Raspberry Pi in **the most extreme environments**

Operation Pico: **make your own board game**

Give kids a **head start in computing**

Raspberry Pi

Official Magazine
#153 | May 2025

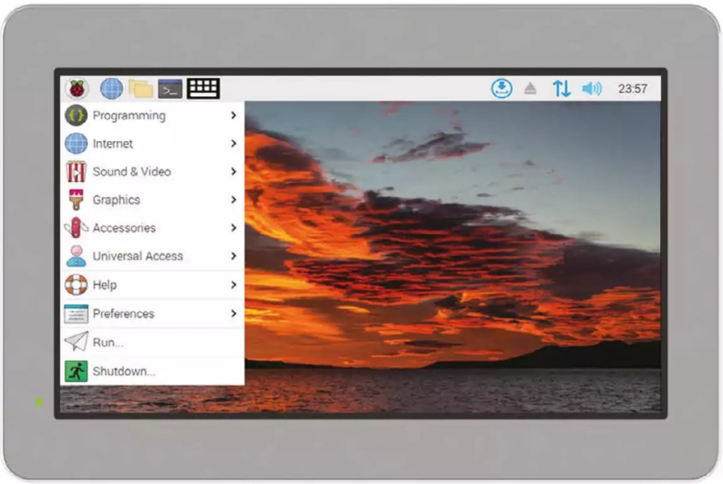
P0WER C0D1NG

HANDS-ON PROJECTS FOR ADVANCED PROGRAMMING

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

Industrial Raspberry Pi

ComfilePi



UL LISTED

Powered by Raspberry Pi

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

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

Visit www.comfiletech.com

COMFILE TECHNOLOGY

Welcome to Raspberry Pi Official Magazine

Editor
Lucy Hattersley
Lucy has just had a coding recommendation approved and won't shut up about it. She is now coaxing AI models to life for upcoming tutorials.

rpimag.co

Learning to code is a modern-age super-power. Armed with code, you can build new things, change the devices you rely on, and understand the world around you.

Even in a world of 'vibe coding' where we augment our powers with AI large language models, it's important to start with a solid base. You need to understand how to deploy code and how computers work if you want to build things that don't fall over.

This month, we tasked Rob Miles with teaching us some Power Coding skills. He'll show us how to beef up our Python with classes, mess around with threads and processes, assign tasks to cores, and explore additional languages like JavaScript and Rust. He works with rotary encoders, NeoPixels, and an old rotary phone to deploy code to the real world.

Meanwhile, Rob Zwetsloot is looking at Extreme Projects. These take Raspberry Pi makes to the absolute edge: from pole to pole, and the edge of space, to the depths of the ocean. These projects deploy Raspberry Pi to the limit.

This month is a powerhouse of an issue. It's packed with projects for you to get stuck into.

Lucy Hattersley – Editor

Contents

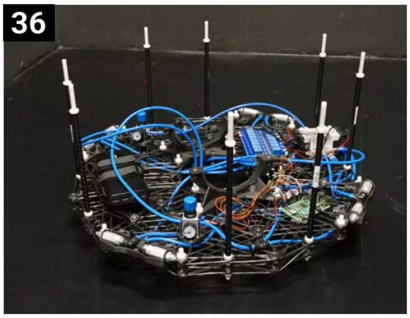
World of Raspberry Pi

Project Showcase

12

28

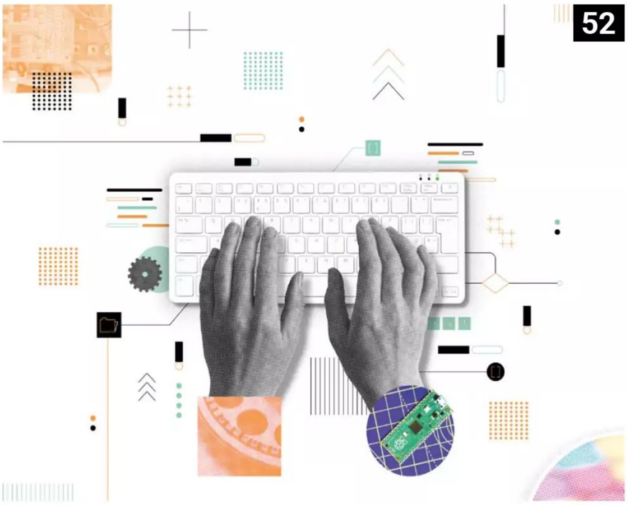
36



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

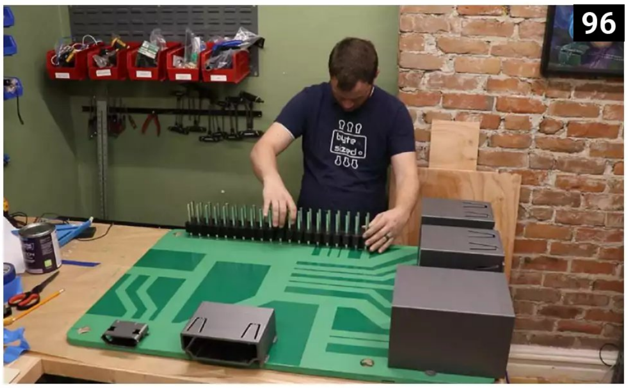
Top Projects

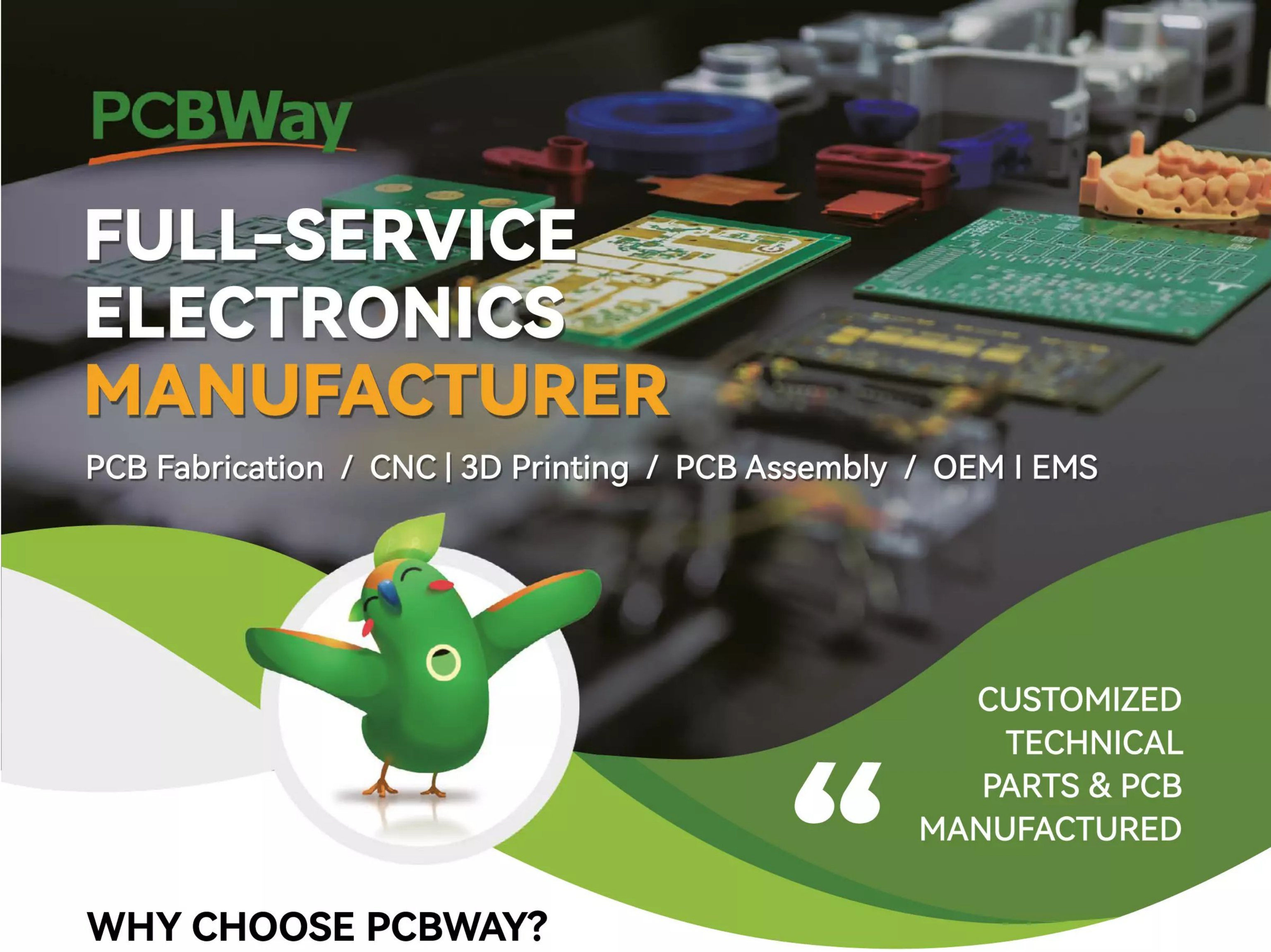
Feature



Tutorials

Feature





104

112

116

Reviews

Raspberry Pi Community

Competition

Win 1 of 5 Pico 2 Starter Kits

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

Raspberry Pi 45W USB-C Power Supply

High-quality power supply now available.
By **Lucy Hattersley**

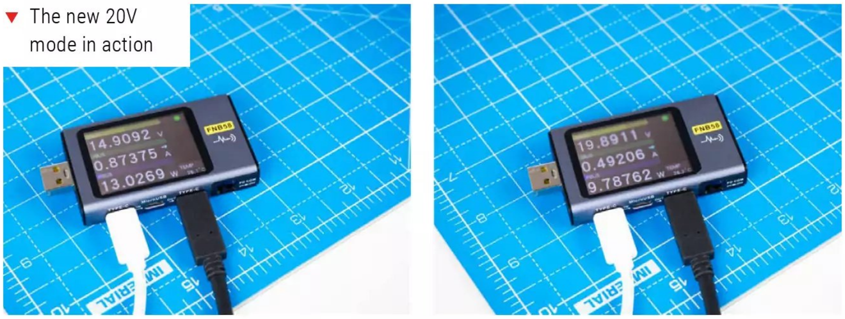
Raspberry Pi introduced its high-quality Raspberry Pi 45W USB-C Power Supply on 7 April, 2025. It is available now for the low price of $15.

"Whether you're running a Raspberry Pi or charging a laptop, the quality of your power supply makes all the difference," says Eben Upton, Raspberry Pi co-founder and CEO.

Efficient regulation

All Raspberry Pi computers need storage and power. While many users repurpose old power supplies and microSD cards, it's beneficial to have high-quality options. "Buying the cheapest SD card or USB wall wart you can find on Amazon is a guaranteed way to have a bad experience," says Eben.

"So over time, we started to regulate the accessories offered by our Approved Resellers. We would test resellers' SD cards, to ensure that they had sufficient random-access performance and were resilient against thousands of unplanned power loss events. Last year, we took this to the next level, launching Raspberry Pi-branded A2-class SD cards and NVMe SSDs, which are now the only storage options promoted alongside our computers."



▼ The new 20V mode in action

You've just found your new favourite power supply

▼ Raspberry Pi 45W with a US two-pin 'Type A' plug

▲ The 45W PSU with a British Type G three-pin plug, which is the best plug

Previously, Raspberry Pi 4 and Raspberry Pi 5 were accompanied by 15.3W and 27W USB-C power supplies, respectively.

"Dominic [Plunkett] worked with our ODM partner KTEC and capacitor vendor Panasonic to validate that the design of these products met our aggressive goals for transient response and product lifetime," says Eben. "In the case of the 27W power supply, we added a custom 5.1V, 5A operating mode to give Raspberry Pi 5 headroom to pass an aggregate 1.6A downstream to power-hungry peripherals."

Power for all

It turns out that building high-quality generic products has appeal beyond your main market. Oli [Wilkin] at Raspberry Pi recounts a story of someone at Micro Center in North Carolina who came up to the Raspberry Pi display just to buy Raspberry Pi's 27W power supply. "In building the best USB-C power supply for our customers, we'd accidentally built the best USB-C power supply for everyone," says Eben.

"But 27W is just a little bit too weedy to rapidly charge my laptop, and so the idea of an upgraded design was born: even better electronics, in (almost) the same form factor, with a longer (1.5m versus 1.2m) cable and a suite of new 45W operating modes, including 20V, 2.25A."

Raspberry Pi users will now have access to a high-quality power supply that delivers the precise power their device needs. You'll still get 5.1V, 3A on a Raspberry Pi 4, or 5.1V, 5A on a Raspberry Pi 5.

The Raspberry Pi 45W power supply is available today from your favourite Approved Reseller (**rpimag.co/45w**).

Raspberry Pi PoE+ Injector

Add power to your network with this new Raspberry Pi product. By **Lucy Hattersley**

Raspberry Pi has announced a brand new product, the PoE+ Injector (Power-over-Ethernet). This enables users an affordable and easy way to add power to a compatible networked Raspberry Pi device for just $25.

Announced by Eben Upton on 24 March, the new device simplifies deploying Raspberry Pi computers in locations that lack convenient power outlets.

"Many of our favourite Raspberry Pi applications, from garden webcams to industrial controllers, involve putting our products in out-of-the-way locations," explains Eben, "where they act as a bridge between the network and the physical world. But the more out-of-the-way the location, the more challenging it is to get power."

The brand new injector, developed with Microchip (**microchip.com**), supports the IEEE 802.3af (PoE, 13W) and IEEE 802.11at (PoE+, 25W) standards, along with mains voltages between 100V and 240V.

◀ PoE+ Injector is "very demure, very mindful," says Eben

Power to devices

This makes it compatible with various devices. It functions as a midspan injector, adding power capabilities to existing non-PoE network switches. Users will need to provide their own standard IEC 'kettle' power lead.

Standing for 'Power over Ethernet', the new product adds PoE support to your existing network, enabling you to both power and connect to your Raspberry Pi with a single Ethernet cable.

While Raspberry Pi has supported PoE since the Raspberry Pi 3B+ back in 2018, Eben acknowledges that the anticipated PoE+ HAT accessory for Raspberry Pi 5 is forthcoming. He describes it as being "in the final stages of development", promising it will be their "smallest, most efficient PD accessory."

The PoE+ Injector is available now through Raspberry Pi Approved Resellers (**rpimag.co/poeplusinjector**).

Adds PoE support to your existing network

◀ PoE+ Injector combining power and networking to Raspberry Pi with a PoE+ HAT via a single Ethernet cable

PoE+ Injector Specifications

- Data rate: 10/100/1000Mbps
- Input voltage: 100 to 240V AC
- Output power: 30W
- Power output on pins: 4/5 (+), 7/8 (–)
- Nominal output voltage: 55V DC
- Dimensions: 159mm (L) × 51.8mm (W) × 33.5mm (H)
- Data connectors: Shielded RJ-45, EIA 568A and 568B
- Power connector: IEC c13 mains power input (not included)
- Operating ambient temperature: –10ºC to +50ºC
- Operating humidity: Maximum 90%, non-condensing
- Storage temperature: –20ºC to +70ºC
- Storage humidity: Maximum 95%, non-condensing
- Operating altitude: –300m to 3000m
- MTBF: 200,000hrs @ 25ºC
- List price: $25

Introducing rpi-image-gen

Build highly customised Raspberry Pi software images. By **Matt Lear**

When it comes to software on Raspberry Pi devices, one size doesn't always fit all. Raspberry Pi OS is ideal for many applications, but we recognise that it doesn't suit every use case or deployment model, particularly in a product that has a specific purpose.

If you are building an embedded system or an industrial controller, you'll need complete control over the software resident on the device, and home users may wish to build their own OS and have it preconfigured exactly the way they want. For developers and organisations that require a custom software image, a flexible and transparent build system is essential; to support these customers, we have created rpi-image-gen, a powerful new tool designed to put you in complete control of your Raspberry Pi images.

Raspberry Pi is pleased to reveal rpi-image-gen, an alternative to pi-gen (**rpimag.co/pigen**), which is the tool we use to create and deploy the Raspberry Pi OS distribution. rpi-image-gen is designed to generate highly customised software images for Raspberry Pi devices, and offers a very granular level of control over file system construction and software image creation.

Why we created rpi-image-gen

There are a number of community-maintained build systems which already exist and which support Raspberry Pi devices. These offer several customisation options and are used by many Raspberry Pi customers, so you may be wondering why we decided to create our own. rpi-image-gen was most definitely not born out of a 'not-invented-here' mindset; there are valid reasons why our customers would benefit from a tool designed, from scratch, to provide the flexibility we know they need to deploy software on their products.

By supporting a build system that has the benefits of Raspberry Pi OS distribution packages, we have one set of sources to maintain, which means that when software gets improved or fixed in one place, it's automatically made available everywhere. Consolidating around centralised package-based delivery of software and updates makes a lot of sense. In addition, being able to

◀ Create custom images for Raspberry Pi computers

▼ A Raspberry Pi thin client setup

help reduce software build time, provide guaranteed ownership of support, and reuse standard methodologies to ensure authenticity of software were all of paramount importance, and among the reasons why we created a new home-grown build tool for Raspberry Pi devices.

Raspberry Pi is pleased to reveal rpi-image-gen, an alternative to pi-gen

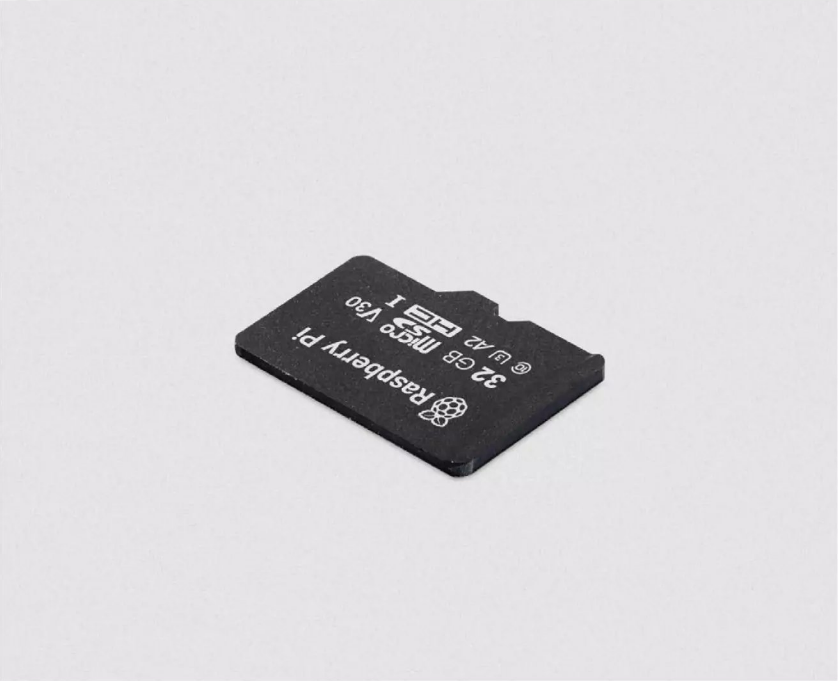
How rpi-image-gen works

Similar to pi-gen, rpi-image-gen leverages the power, reliability, and trust of installing a Debian Linux system for the device. However, unlike pi-gen, rpi-image-gen introduces some new concepts which serve to dictate the build footprint and installation.

A profile is essentially a collection of descriptive layers which group together Debian packages and installation operations. These collections can be selectively picked and customised further, and form the foundation of the software image. The image layout describes how the output software binary image

will be created and laid out on-disk for programming into the device, including the types of file systems, partition table entries, image formats, etc. A config file is a 'top level' text file, written in easy-to-understand .ini syntax, which defines the profile and image layout that rpi-image-gen uses to build the device image.

The config file is typically associated with the underlying device hardware and product, so it can specify applicable attributes accordingly: for example, defining the sizes of individual partition images to match the onboard eMMC size, or using a layout which uses particular mount options for file systems, fine-tuning options exposed by lower levels, or selecting a specific Raspberry Pi device class to target. Likewise, different derivatives of config files can be used to tailor the installation to the product's functional requirements. You could, for example, utilise a Bluetooth audio layer to pull in device support; or use a particular layer to add in a minimal Wayland desktop which runs in a kiosk mode, to install a default set of containers, to seed a default environment for distribution to third-party developers, and so on. There is no limit to the possibilities.



▲ Install it onto an SD card using Raspberry Pi Imager; here's one we made earlier

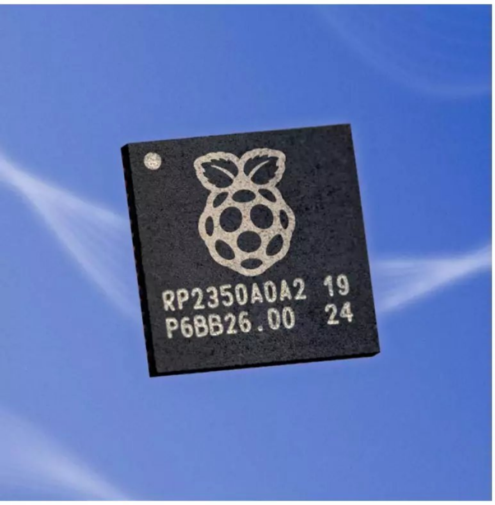
There is a small number of examples in the tree which demonstrate different use cases of rpi-image-gen (**rpimag.co/rpiimagegenexamples**). All of these create bootable disk images and serve to illustrate how one might use rpi-image-gen to create a bespoke image for a particular purpose. The number of examples will grow over time and we welcome suggestions for new ones. You can read more about these examples on the Raspberry Pi blog version of this story: **rpimag.co/rpiimagegenintro**.

Visit the GitHub repository to get started (**rpimag.co/rpiimagegen**). There, you'll find documentation and examples to guide you through creating custom Raspberry Pi images. We encourage you to explore the repository and provide feedback to help us improve the tool further.

Raspberry Pi wins TSMC Trophy

Prestigious award for embedded computing innovation goes to Raspberry Pi.
By **Roger Thornton**

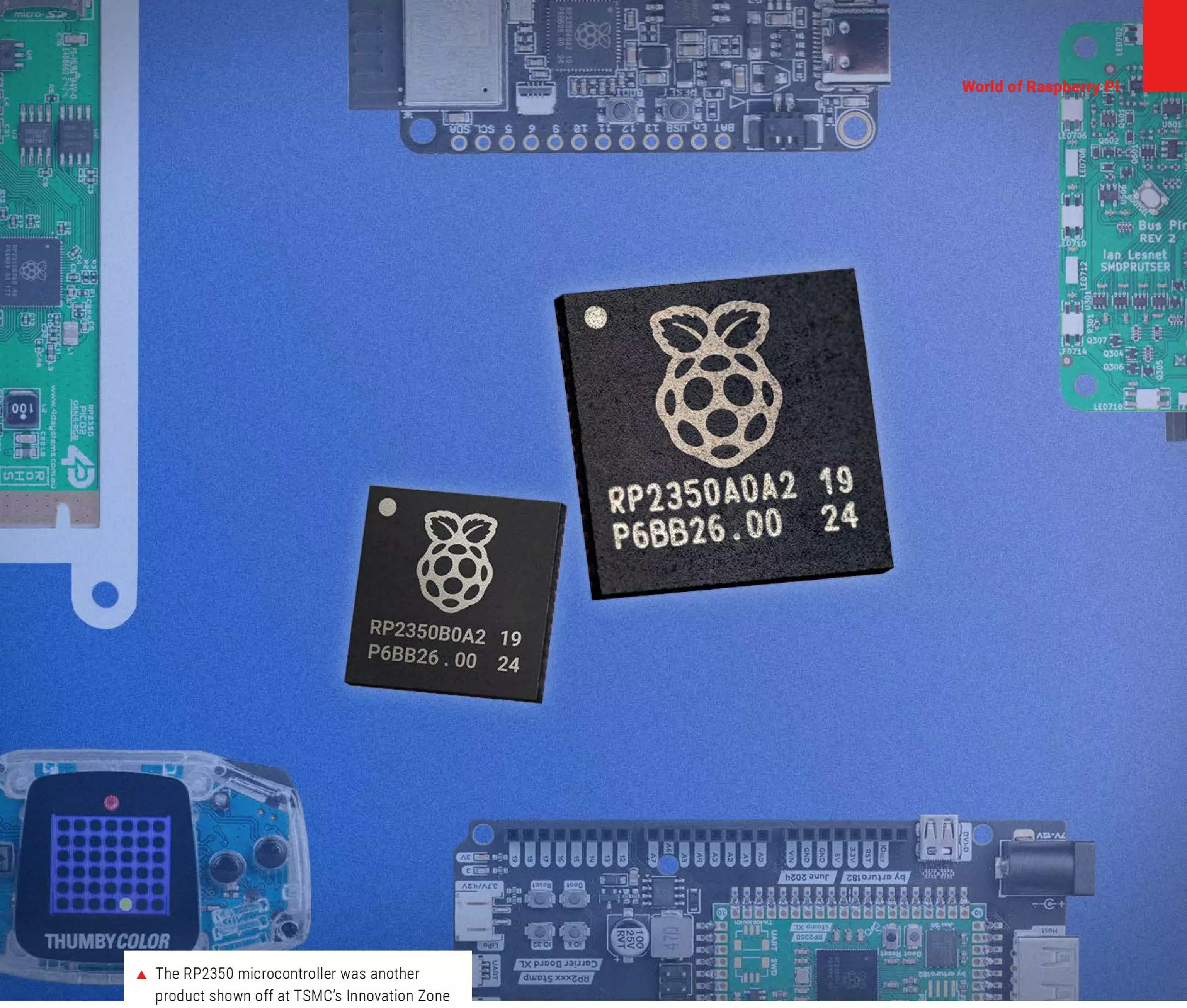
The TSMC Trophy recognises Raspberry Pi's commitment to delivering secure, high-performance, and scalable solutions



We are delighted to share that Raspberry Pi has been awarded the coveted 2024 Europe TSMC Trophy. We are proud to be a leader in scalable embedded secure computing platforms, and this prestigious award recognises our relentless pursuit of cutting-edge technology and our significant contributions to the embedded computing industry, particularly in relation to our RP2350 microcontroller (**rpimag.co/rp2350**) and our much-anticipated Raspberry Pi Compute Module 5 (**rpimag.co/cm5**), both released during 2024.

TSMC (Taiwan Semiconductor Manufacturing Company, **tsmc.com**), a global giant in the field of semiconductor manufacturing, established the trophy to celebrate the groundbreaking achievements of its emerging customers. The award highlights the remarkable innovations developed by startups pushing the boundaries of technology across diverse sectors, including AI, automotive, communication, IoT, and industrial automation.

◀ Raspberry Pi's focus on industrial-grade applications like Compute Module 5 clearly impressed the judges at TSMC



▲ The RP2350 microcontroller was another product shown off at TSMC's Innovation Zone

Revolutionary technology

Raspberry Pi's presence at the TSMC Innovation Zone (**rpimag.co/tsmczone**), a major showcase for revolutionary technologies, provided an opportunity for us to demonstrate our powerful platforms. Amongst 41 innovative companies from across Europe, Raspberry Pi's focus on industrial-grade embedded computing solutions, specifically highlighting RP2350 and Compute Module 5, clearly resonated with the judges.

RP2350, our latest high-performance microcontroller chip, takes embedded applications to the next level. Its dual Arm Cortex-M33 processors deliver a significant boost in processing power and efficiency, together with increased memory, enhanced security features, upgraded interfacing capabilities, and the option to utilise dual Hazard3 RISC-V cores.

Raspberry Pi Compute Module 5, the newest addition to our widely adopted Compute Module line, offers a leap up in processing power and connectivity. With its powerful quad-core Arm Cortex-A76 processor, high-speed interfaces, and enhanced security features, Compute Module 5 is poised to revolutionise industrial embedded systems.

The TSMC Trophy award recognises Raspberry Pi's commitment to delivering secure, high-performance, and scalable solutions that meet the demanding requirements of industrial applications (**rpimag.co/industry**). Our platforms are designed to excel in challenging environments, ensuring reliability, longevity, and robust security for critical industrial systems. The award reinforces our position as a driving force in the industry, enabling businesses to speed time to market and lower the cost of embedded computing by leveraging our technologies.

Raspberry Pi is honoured to receive this recognition from TSMC, the world's largest and most advanced contract chip maker.

Raspberry Pi awarded Green Economy Mark

London Stock Exchange recognises Raspberry Pi's environmental impact. By **Roger Thornton**

We're proud that Raspberry Pi has been awarded the London Stock Exchange's Green Economy Mark – an accolade recognising companies and funds listed on the exchange that generate at least 50% of their revenue from products and services that have a positive environmental impact.

This recognition, achieved at the time of our successful listing in 2024, highlights our dedication to energy-efficient computing and sustainable technology. Our low-power devices help reduce energy consumption in homes, businesses, and industrial and embedded applications, contributing to a lower-carbon future.

What is the Green Economy Mark?

The LSE's Green Economy Mark is a world-first initiative that identifies companies actively driving the transition to a sustainable, low-carbon economy. By spotlighting green leaders, the mark helps investors back businesses that are driving a sustainable future, promoting the growth of the green economy as a whole.

With a combined market capitalisation of £172.8 billion as of 2024, the companies in this exclusive cohort represent some of the most forward-thinking businesses in the world. The value of Green Economy Mark stocks held by institutional investors has surged by 50% over the last five years.

Proving that innovation and sustainability can go hand in hand

Supporting carbon removal

To help mitigate the impacts of climate change, **Raspberry Pi** is supporting the work of **UNDO**, an organisation whose mission is to permanently remove a billion tonnes of CO_2 from the earth's atmosphere.

This token represents an offset of the carbon footprint of manufacturing, shipping, and disposing of one of our boards, and helps support UNDO in providing a healthier planet for future generations.

Find out more at **rptl.io/carbon-credit**

▶ Raspberry Pi's Carbon Credits scheme

Raspberry Pi's green commitment

Earning this mark is more than just an honour; it's a reflection of our long-standing commitment to sustainability. Here are some of the ways we're making a difference:

- **Energy-efficient computing:** our products are designed to maximise performance while using minimal power, thereby helping individuals and businesses to reduce their energy footprint
- **Product longevity:** we prioritise durability and long-term software support, reducing electronic waste
- **Sustainable manufacturing practices:** we work closely with our manufacturing partners to improve efficiency and minimise waste
- **Empowering green innovation:** Raspberry Pi is at the heart of countless projects that support a greener future, from smart energy management to eco-friendly IoT solutions
- **Reducing our carbon footprint:** by designing compact, low-power devices and optimising our supply chain, we're committed to lowering emissions across the life cycle of our products
- **Mitigating our carbon footprint:** in 2025, we have introduced Raspberry Pi Carbon Removal Credits (**rpimag.co/carboncredits**) to enable customers to offset the manufacture, shipping, and disposal of our products

Read the case study on Raspberry Pi in the LSE's 2024 report, 'The Green Economy Mark: five years of green growth', to learn more about our sustainability story here: **rpimag.co/lsegreen** (PDF document).

Sustainability is at the core of what we do, and earning the LSE's Green Economy Mark reinforces our mission to provide powerful, low-cost, and environmentally responsible computing solutions for all. This recognition cements Raspberry Pi's position as a leader in sustainable technology: it not only strengthens our appeal to environmentally conscious investors, but also sets an example for other businesses, proving that innovation and sustainability can go hand in hand.

▲ The LSE Green Economy Mark highlights a company's contribution to the global green economy

SUBSCRIBE TODAY FOR JUST £10

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

SUBSCRIBER BENEFITS

Free Delivery
Get it fast and for free

Exclusive offers
Great gifts, offers, and discounts

Great savings
Save up to 23% compared to stores

SUBSCRIBE FOR £10

Free Pico W

3 issues of Raspberry Pi Official Magazine

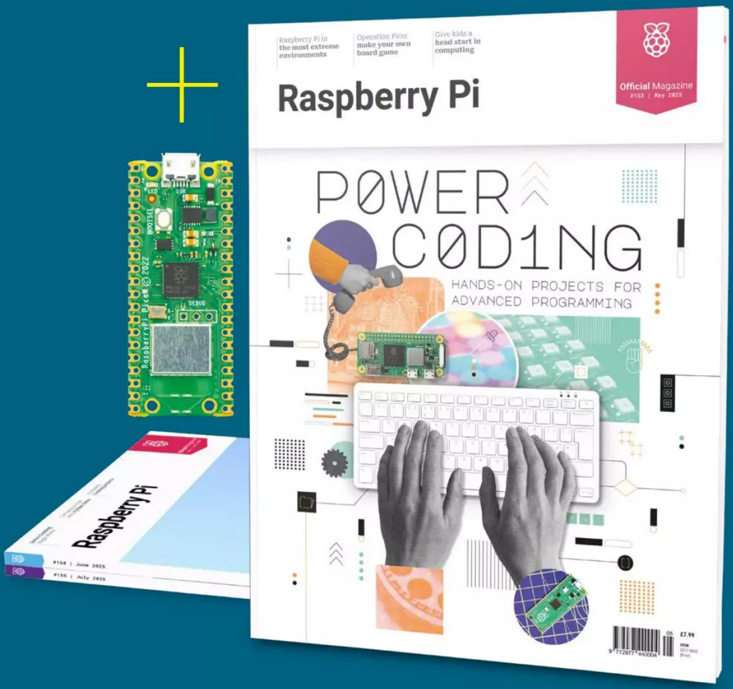
£10 (UK only)

SUBSCRIBE FOR 6 MONTHS

Free Pico 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 free Pico W is included with a six-month subscription in USA, Europe and Rest of World.

Snake game console

A classic game slithers back into action. By **David Crookes**

Maker

Arnov Sharma

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

rpimag.co/snakeconsole

If you owned a Nokia mobile phone in the late 1990s, then you will likely remember the game Snake. It came preloaded on the Nokia 6110 in 1997 and took the world by storm as users – including a host of celebrities – became utterly enthralled by the simple premise and potential of racking up a high score.

For those unaware, Snake was a spin on the 1976 game Blockade, by Gremlin Industries, and the idea was that players would control a limbless reptile around a small area, gobble up food and avoid hitting itself. With each bite, the snake would grow longer, making it more likely to collide with the walls or its own tail. So, it took quite a bit of skill to precisely move around the screen – a task made more difficult given players were controlling the snake using a phone's hard keypad.

> *I completed the entire project in one day*

Arnov Sharma was reminded of this game when he got hold of a matrix panel. Given the blocky, pixelated nature of Snake's rather primitive graphics, he realised the panel could lend itself to a replication of the game, so he began to consider the practicalities of producing a unique game console.

"My objective or goal for developing this project was to create a gaming console that used my 64×32 P3 matrix panel," he says of a Waveshare unit he purchased from PBCWay for $22. "I thought this console would be useful for coding simple games."

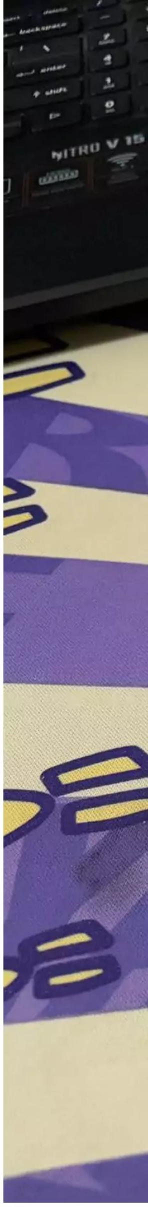
With that in mind, he began to think about how it would be powered and how this large screen could be comfortably held and controlled.

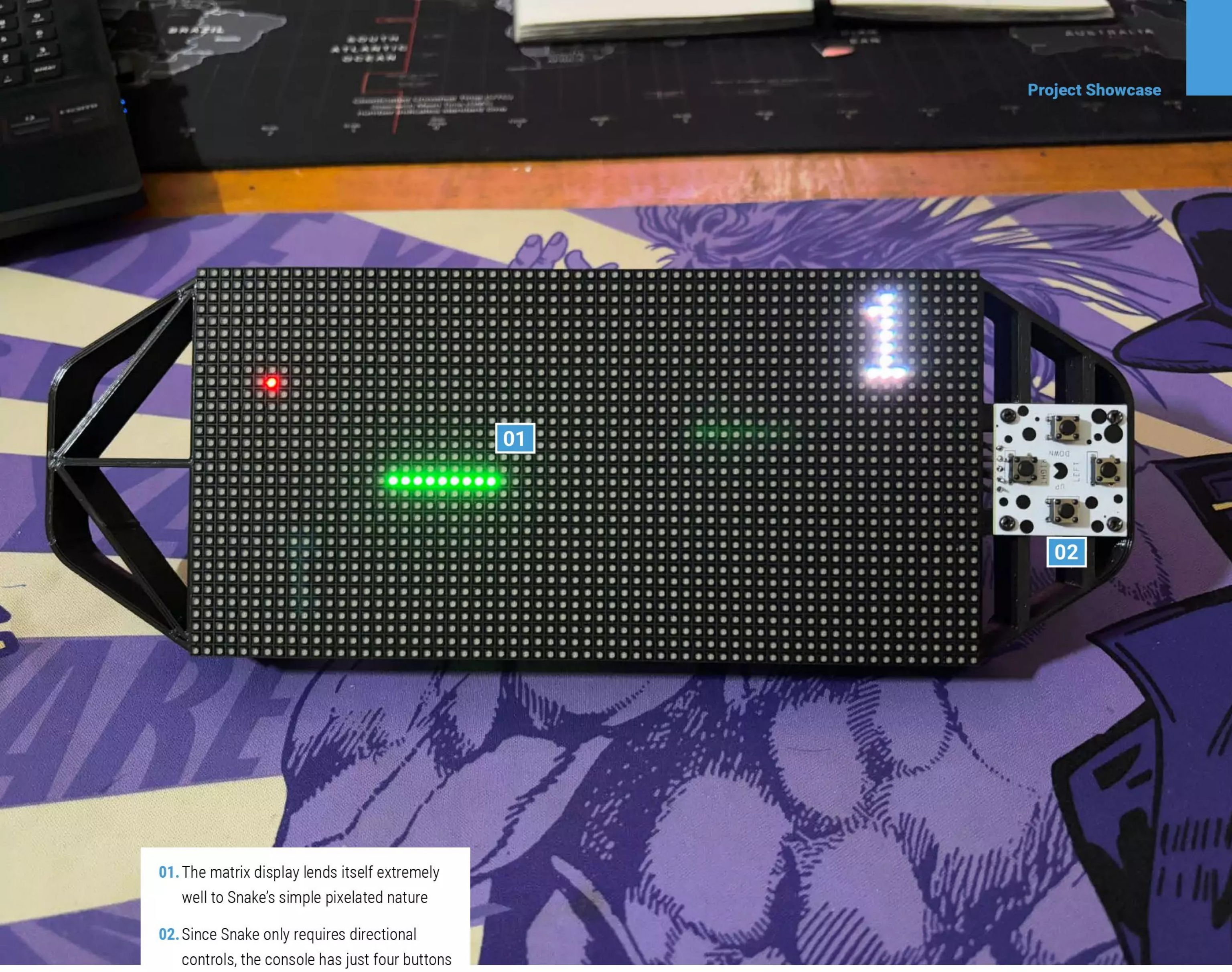
Recreating hiss-tory

Given Arnov was looking to create retro-style games – the likes of which were commonplace back in the 1970s and 1980s – he decided he didn't need the immense processing power offered by a Raspberry Pi 5. At the same time, however, he realised he would still require a powerful device that had a fast CPU, a good amount of RAM, and flash storage, so he decided he could create a streamlined build based around Raspberry Pi Pico 2.

"This project's code was quite large and it involved the use of a matrix library, which requires speed and computing power," he explains. "Raspberry Pi Pico 2 was ideal for this project."

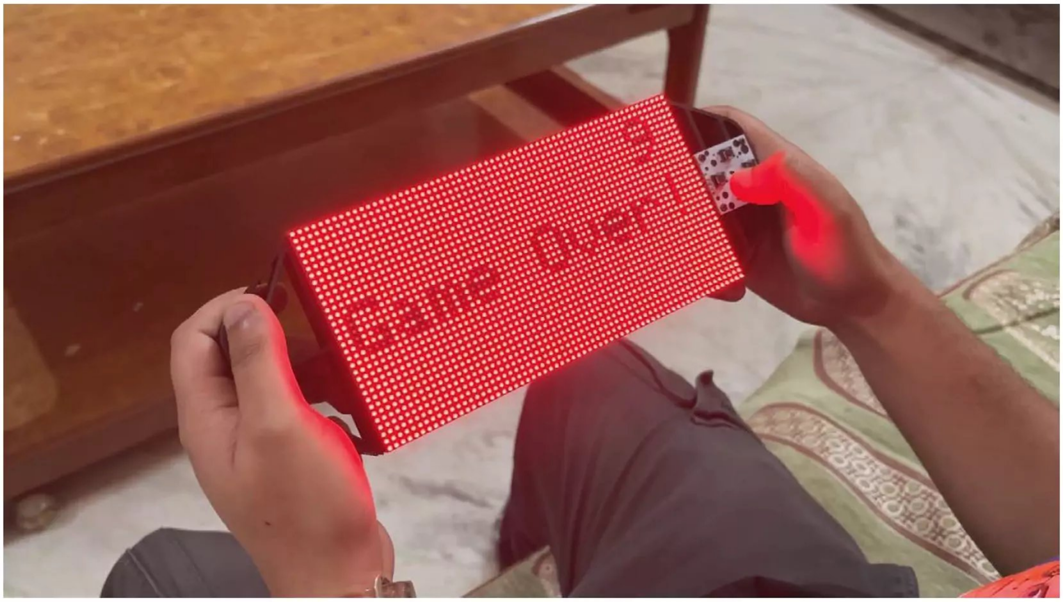
Once those main two components were selected, Arnov then turned his attention to how the game console would be controlled and how it would look. With such a large screen, it made sense to have the controller on one side and something to grip on the other.



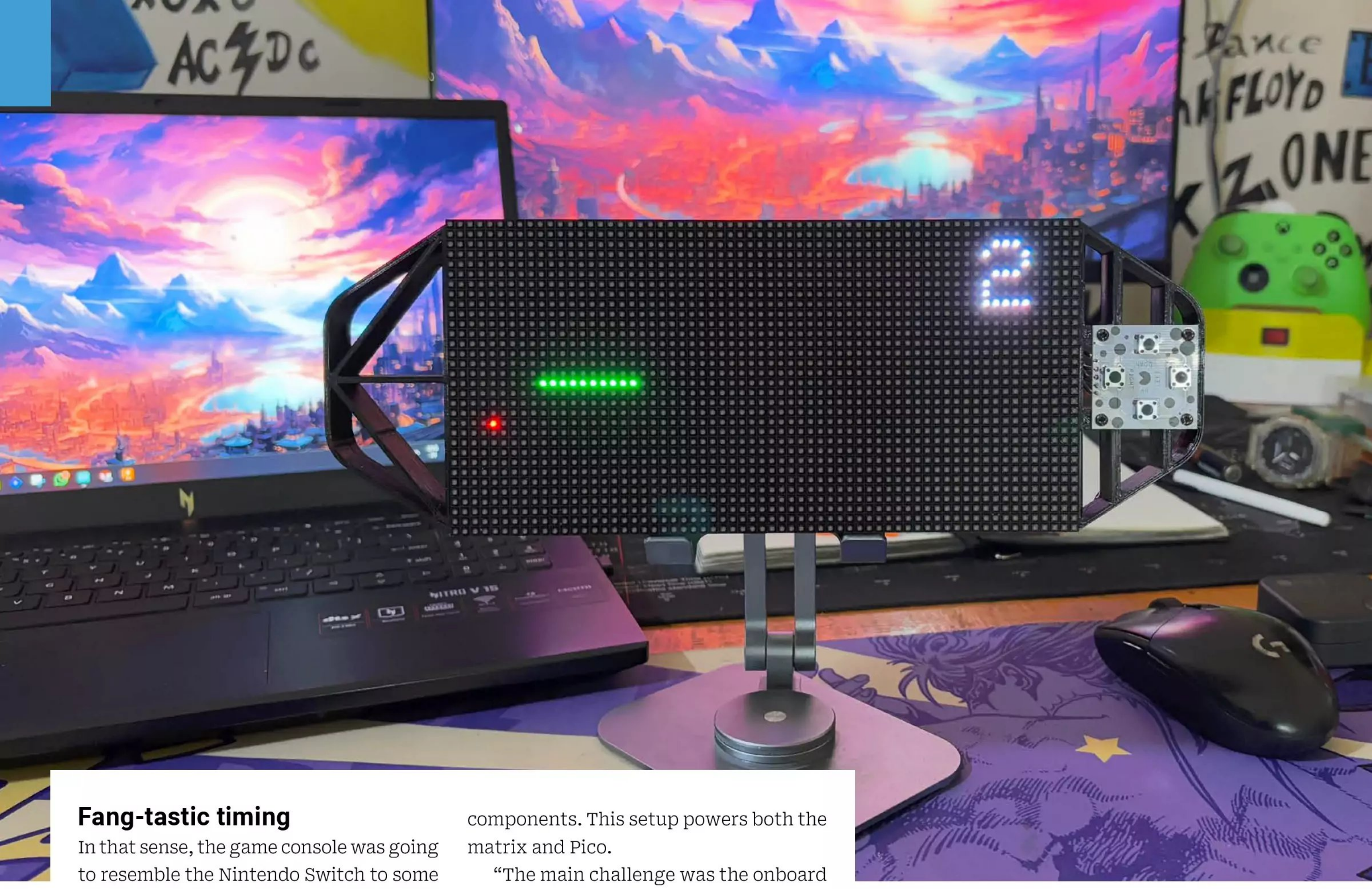


01. The matrix display lends itself extremely well to Snake's simple pixelated nature

02. Since Snake only requires directional controls, the console has just four buttons



▶ The LEDs are used to great effect to clearly show when the player's game has come to end. The resulting score is displayed in the top-right corner



Fang-tastic timing

In that sense, the game console was going to resemble the Nintendo Switch to some extent, although this format also conjures up memories of Sony's PlayStation Portable. In any case, he reckoned that the types of games he would create wouldn't need more than a D-pad (that is, a set of four directional buttons: up, down, left, and right), although it did mean foregoing a fire button. "The most basic game I was able to create in a week was the classic Snake game," says Arnav. This took up most of the project's development time.

"I completed the entire project in one day, but it took me a week to finalise the code," he notes. That is impressive since the project entailed building a 3D model of the console using Autodesk Fusion 360 and working out where the components would go. A custom Pico Driver Board and Button Board were also designed using PCB CAD software. These were produced by PCBWay.

"I created a frame-like part that transforms the matrix into a handheld game console," Arnov explains. "The hardware also features a power circuit that includes a power management IC, a lithium cell, and a few mandatory components. This setup powers both the matrix and Pico.

"The main challenge was the onboard power source; I created a customised circuit that houses the power management IC that powers the matrix and Pico. The challenge was to make the circuit smaller and lighter. I am still thinking about making it smaller, which I may address in a future edition."

Scaling up

Assembling the components was relatively straightforward. "The setup works by connecting Pico 2's GPIOs to the matrix's HUB75 connector and using necessary libraries to control the matrix," Arnov says.

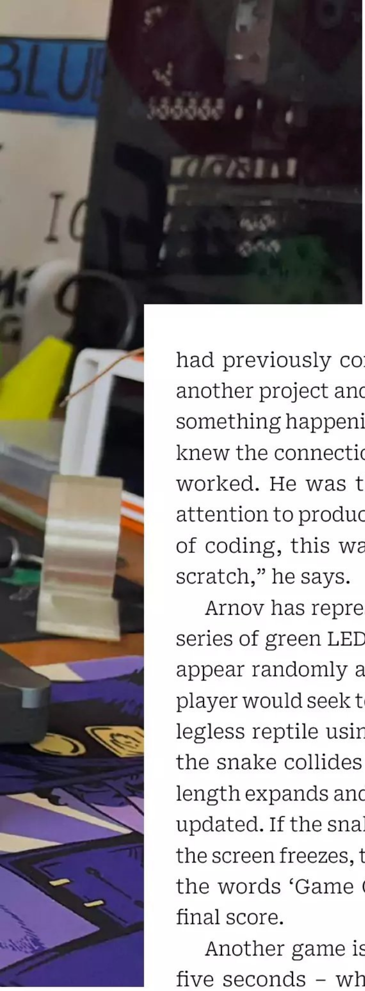
"To power Pico 2 and the matrix, we used a power circuit that uses a lithium ion cell with a nominal voltage of 3.7V. However, Pico 2 and the matrix require a stable 5V, so we used a power management system to boost the cell voltage from 3.7V to 5V. We added a button board with I/O pins that connect to Pico 2's GPIO pins to control the snake's movements."

Arnov tested out the build using a cellular automation called Game of Life, which was created in 1970 by the British mathematician John Horton Conway. He

▲ Although you might have expected a Snake game to be written in Python, this one was developed using a variant of C++

Quick FACTS

- This project is designed around an LED matrix panel
- It allows the classic game Snake to be played
- Only four buttons are currently used
- The code is open source
- The project cost $60 to make

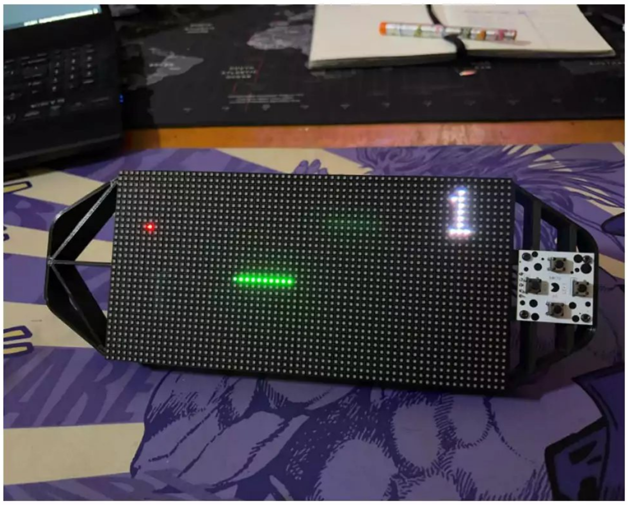


had previously converted the code for another project and it allowed him to see something happening on the screen so he knew the connections and configuration worked. He was then able to turn his attention to producing a game. “In terms of coding, this was put together from scratch,” he says.

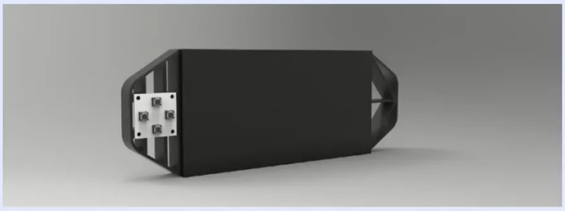
Arnov has represented the snake as a series of green LEDs and he has the food appear randomly as red LEDs which the player would seek to ‘eat’ by directing the legless reptile using the buttons. When the snake collides with the red dot, its length expands and an on-screen score is updated. If the snake collides with itself, the screen freezes, turns red, and displays the words ‘Game Over’ along with the final score.

Another game is ready to play within five seconds – which gives the player enough time to grab a modern smartphone and take a quick photo if the score is impressively high. Given how addictive Snake is, the player is then likely to get stuck in once more, but Arnov is now considering which other classics he may be able to play on his portable console. “I’d need to build them myself or port an existing game, but that’s something for the future,” he says.

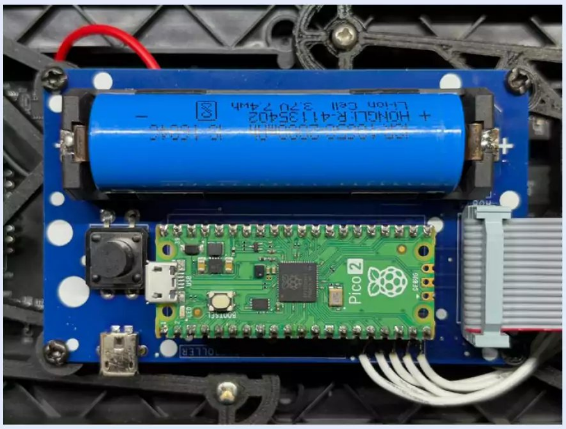
▼ Snake starts off easily enough, but the more food that is consumed, the trickier it becomes to avoid a collision



Building a portable games console



1. Having decided to base a console around a matrix display, Arnov created a housing that has built-in grips on either side of the screen. The buttons are positioned so that they can be reached using one of the user’s thumbs.



2. At the rear of the console is a Raspberry Pi Pico 2 microcontroller connected to a custom-designed PCB and powered by an 18650 lithium battery. This is powerful enough to allow a game of Snake to run quickly and smoothly.

3. A push-button at the rear of the console allows the device to be switched on (and off) and, with sufficient juice via a Micro B cable and standard 5 V smartphone charger, the console will allow for a lengthy gaming session.

Operation Pico

Prolific maker Kevin wanted to recreate the classic board game Operation using Raspberry Pi Pico. **Rosie Hattersley** is buzzed

Maker

Kevin McAleer

Kevin makes robots and shares his builds and expertise via weekly online broadcasts.

kevsrobots.com

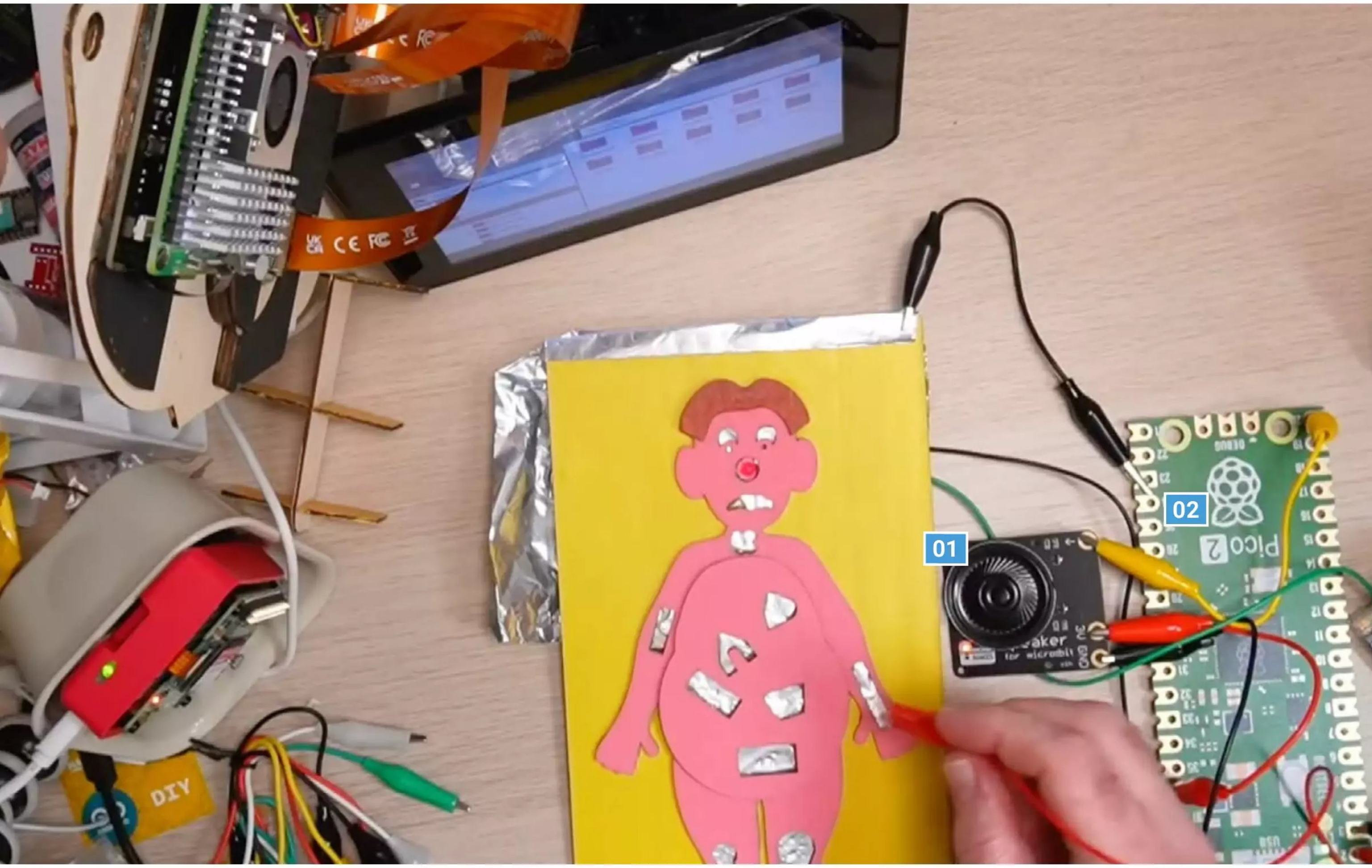
Kevin McAleer is a master at making coding fun, showcasing the creative possibilities via a tranche of simple but clearly explained videos and tutorials that are both memorable and off the wall: rpimag.co/kevsyt. One of the very first projects of Kevin's that we featured involved a Raspberry Pi and Billy Bass, the singing animatronic catch of the day. His latest project is a build-it-yourself version of Operation, a hugely popular game beloved of those of us who grew up in the late 1970s or 1980s. Pico Operation showcases how easy it is to use a Raspberry Pi Pico microcontroller and MicroPython code to recreate the Hasbro board game.

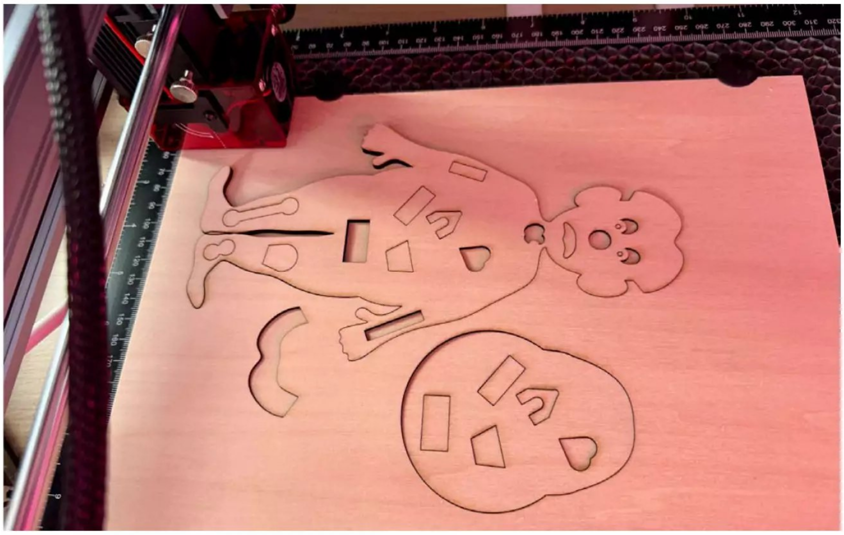
Break it down

Kevin has a knack for making coding accessible and has been showing off his inventive spirit ever since 3D-printing his own Adafruit Mini Mac case with a first-generation Raspberry Pi inside. He credits Raspberry Pi with helping him learn and master Linux and Python, his go-to choice of coding language.

Here, he uses crocodile clips attached to Pimoroni's Pico Jumbo – a "hilariously oversized" version of Raspberry Pi Pico 2 on a matching PCB – connected via GPIO pins to a laser-cut patient from the game Operation (**rpimag.co/operation**). The Jumbo is "perfectly suited for use with crocodile clips and therefore requires zero soldering – you can simply unclip

> Operation Pico has a range of tuneful responses when a shaky hand causes the crocodile clip tweezers to touch the sides of the cavity and trip the buzzer mid-'operation'





01. Pico Operation updates the classic buzzer-wire game with catchy tunes played via a MonkMakes speaker

02. A Pimoroni Pico Jumbo controls everything, although you could just use a standard Pico

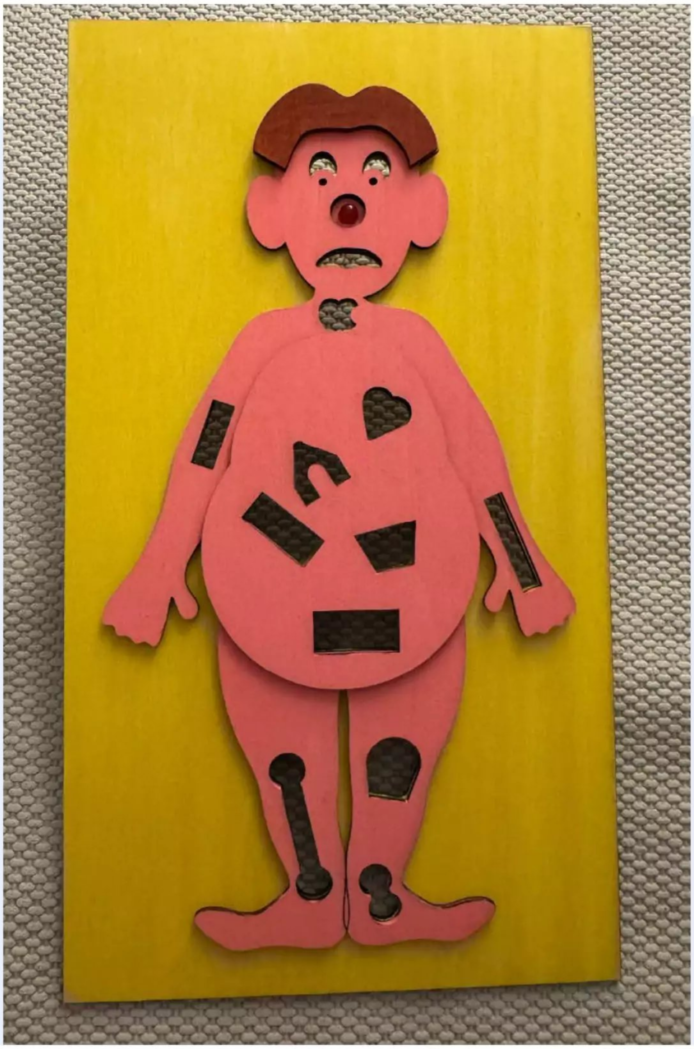
Laser-cut plywood pieces recreate the classic Operation patient everything after playing and put it on the shelf!" He was also impressed with the MonkMakes Speaker which, although labelled as being designed for micro:bit, works perfectly well with Raspberry Pi.

The setup also includes a large 10mm LED nose and a crocodile clip used as the surgeon's removal tool, which "will complete a circuit if it connects with the aluminium foil on the back of the Operation model." The closed circuit is detected in the same way as a button press and causes the speaker to buzz and the patient's nose to light up.

Listen up

Operation Pico has a range of tuneful responses when a shaky hand causes the crocodile clip tweezers to touch the sides of the cavity and trip the buzzer mid-'operation'. "The original Operation game simply had a buzzer that was connected to the battery and made a circuit whenever the tweezers touched the sides of the cavity," says Kevin. "For this project I wanted to include a library for playing music through a simple speaker. I created a music library and included a little tune at the start of the game." He points out that the MicroPython code for the game is only 45 lines long, while adding music extended it to 107 lines, "so it is quite a simple project for people to easily recreate themselves".

It was the first time Kevin had used a MonkMakes mini amplified speaker. He says "it is surprisingly loud and gives the game an extra dimension, and was lots of fun when creating the music player library". Kevin explains the music creation process in his excellent accompanying video: **rpimag.co/operationyt**.



▲ Kevin used Fusion 360 to faithfully recreate the original game's look and feel

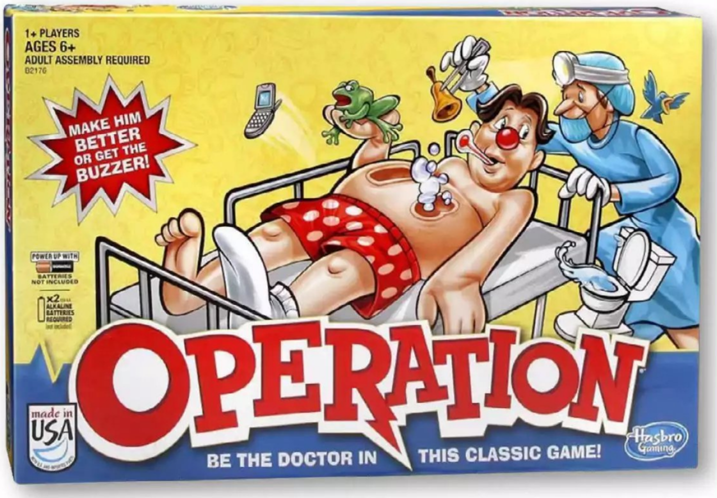
He's got the look

The hardest aspect of the Operation Pico project was getting the iconic patient character design just right. After all, game-playing families from over five decades know exactly what he's supposed to look like. Kevin painstakingly recreated the man in Fusion 360, using circles and splines to mimic the original design. He then glued together multiple layers of plywood to give the character some depth.

As someone commented on the accompanying YouTube video, "Genius! Love how you're combining classic gameplay with modern tech." We couldn't agree more

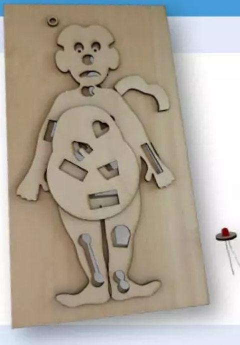
Quick FACTS

- Operation has made Hasbro $40 million to date
- Creator John Spinello was paid $500 for it in 1964
- The original patient is called Cavity Sam
- Kevin has more than 50 active Raspberry Pi projects
- He's keen to hear from anyone currently running more!



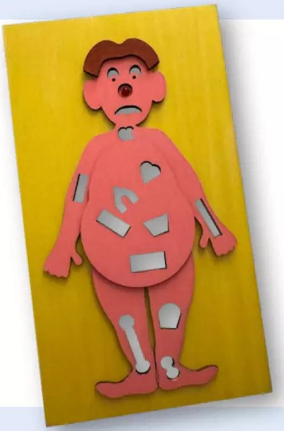
▼ The original game's patient is called Cavity Sam

Body builder



1. Using Kevin's body part templates (**rpimag.co/operationpico**), cut out or laser-cut individual body pieces and spray-paint them. Attach aluminium foil as a base for each body part.

2. Use the GPIO pins on a Raspberry Pi Pico 2 or Pimoroni Jumbo to connect the board electronics, MonkMakes speaker and crocodile clips, along with a power source. Operation Pico code is on Kevin's GitHub page: **rpimag.co/operationgit**.



3. Kevin tracked down a chunky 10 mm LED to serve as the patient's red nose, triggered whenever the player touches the aluminium foil while removing a body part with a crocodile clip.

Raspberry Pi Cluster Briefcase

A clever Raspberry Pi 5 setup needed a stealthy means of travelling internationally. **Rosie Hattersley** gets clued in

Maker

Sean M Tracey

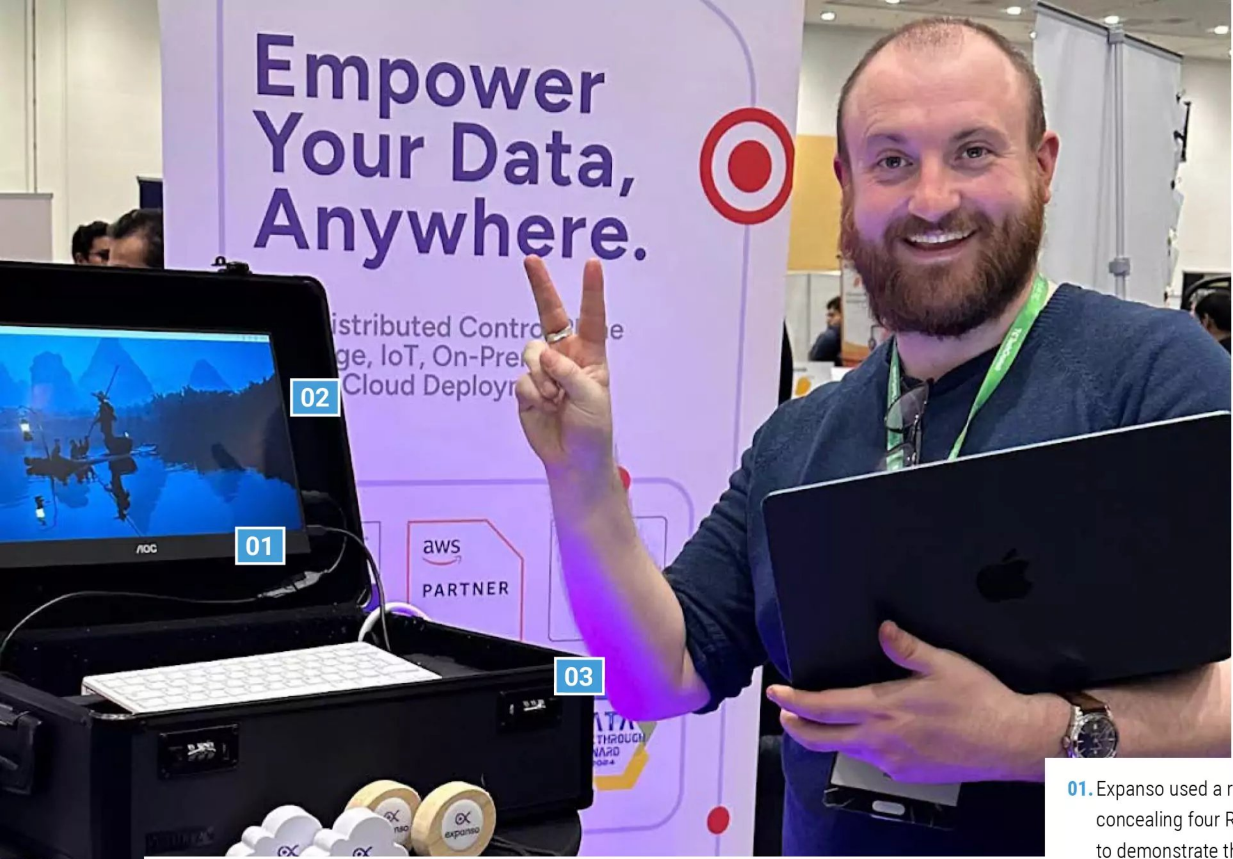
Developer relations bigwig Sean M Tracey has been an ardent Raspberry Pi user since the Model B launched in 2012.

bacalhau.org

Having an inexpensive (but mighty!) Linux-powered computer to hand has always been immensely useful to Sean M Tracey, who fell for Raspberry Pi's abilities almost as soon as it debuted. Already a proficient IoT tinkerer, he quickly incorporated Raspberry Pi into cloud and edge computing projects. These days, he specialises in showcasing how smoothly such products can run. Attracting customers via trade events also involves making a splash, which is how the Raspberry Pi Cluster Briefcase came about…

Smells fishy

Bacalhau is the name of a Portuguese saltfish, but it's also the name of the IoT (Internet of Things) software created by Expanso, where Sean M Tracey works as head of developer relations (**bacalhau.org**). "We wanted to attend TechCrunch Disrupt 2024 with something more interesting than the regular kind of merchandise and posters exhibitors tend to bring with them," he says, but the product the team wanted to showcase is designed to provide access to files and functions from a different location than the exhibition stand. "Bacalhau makes it really easy to set up distributed computing networks, but demoing something that, by design, is usually somewhere else, can be a bit tricky." Turning things on their head, the team decided "to build a distributed compute cluster into a briefcase that we can take with us wherever we go".



01. Expanso used a ruggedised briefcase concealing four Raspberry Pi 5s to demonstrate their distributed computing network

02. A USB-C-powered travel screen Gorilla-glued inside the briefcase lid made for a self-contained setup

03. The secure Bacalhau-powered system features 16 ARM cores and can be run from anywhere

> *Once everything was assembled and fully tested, it took a mere 20 minutes to get all four Raspberry Pi 5s working together inside the briefcase as a Bacalhau cluster*

Presidential approach

The topsy-turvy approach to showing off a means of secure remote access has more than a little in common with the so-called ‘nuclear football’ briefcase that US presidents have been known to carry with them as a mobile command centre for reacting to a nuclear strike. “It’s terrifying that such a thing exists but, world-ending capabilities aside, it’s a cool, impressive piece of technology and perfect for taking a whole bunch of compute capacity with you to various conferences and venues,” notes Sean.

In fact, the resemblance meant the team soon dubbed the project ‘the football’. Raspberry Pi was “a natural fit” given its dainty dimensions, easily able to fit discreetly inside a briefcase. “They are small enough that fitting a couple into a briefcase isn’t too much work, powerful enough to run meaningful workloads on them for our demos (we love running EdgeML models!), and efficient enough that we didn’t need a huge power brick to get the whole thing running,” says Sean. Another serious consideration was how quickly the Expanso team could put together their workable demo. “We built this thing within 36 hours, with a transatlantic flight in the middle!”

Sean dropped by the Raspberry Pi store in Cambridge before setting off to San Francisco. He needed to fit four Raspberry Pi 5s inside the Cluster Briefcase. One would act as the orchestrator node and be used to manage access and distribute jobs. The other three Raspberry Pi 5s were designated as compute nodes where workloads could be executed.

Expanso
Empower Your Data, Anywhere.
The Distributed Control Plane for Edge, IoT, On-Prem, and Cloud Deployments.
aws PARTNER

▶ Expanso’s Lee Baker, David Aronchick, Laura Hohmann, and Sean M Tracey at TechCrunch Disrupt 2024 in San Francisco

Developers, assemble!

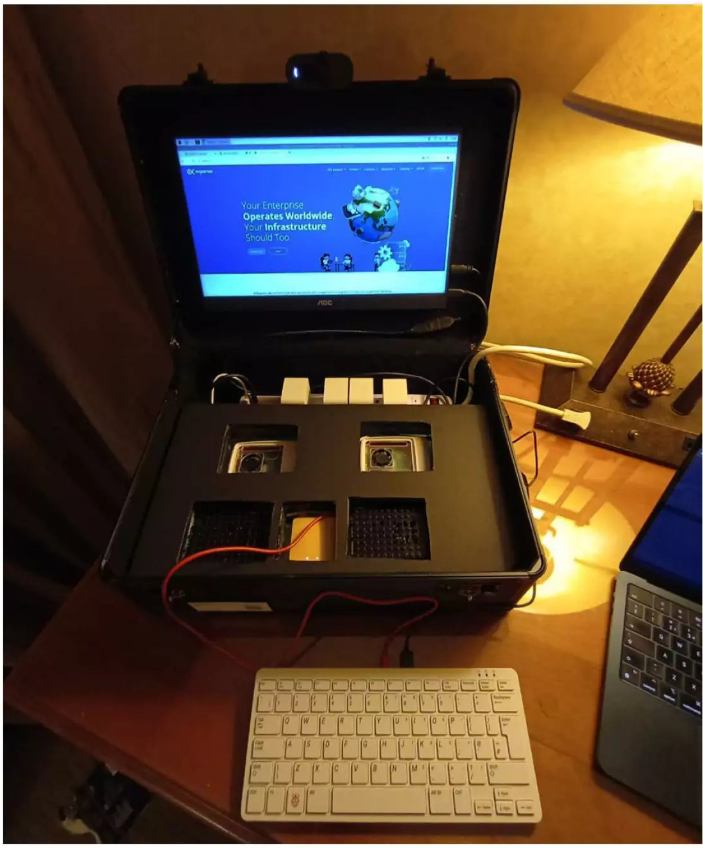
Constructing the top-secret briefcase largely took place in Sean’s hotel room opposite the TechCrunch developer conference where Expanso was due to unveil it. Trundling a trolley full of hardware and wood for the build raised a few eyebrows at reception.

Sean has “learned the hard way that if you’re planning to glue something down – especially a computer – you should check that it works as expected first”, so he set about testing and configuring the Raspberry Pi 5s with Bacalhau before fitting everything into the hardware enclosure and briefcase.

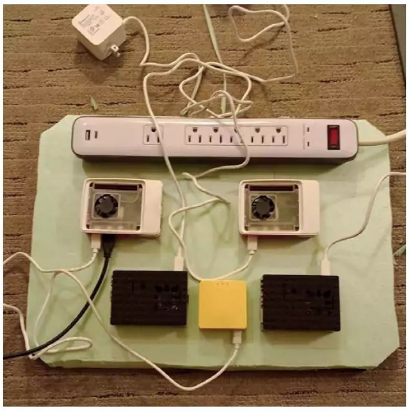
Once everything was assembled and fully tested, it took a mere 20 minutes to get all four Raspberry Pi 5s working together inside the briefcase as a Bacalhau cluster. Being able to set a known hostname for the orchestrator node using Raspberry Pi Imager helped with the ease of setup. Sean also credits the “adaptive and reliable” NATS (**nats.io**) high-performance open-source messaging system Expanso “built right into the heart of Bacalhau”.

Quick FACTS

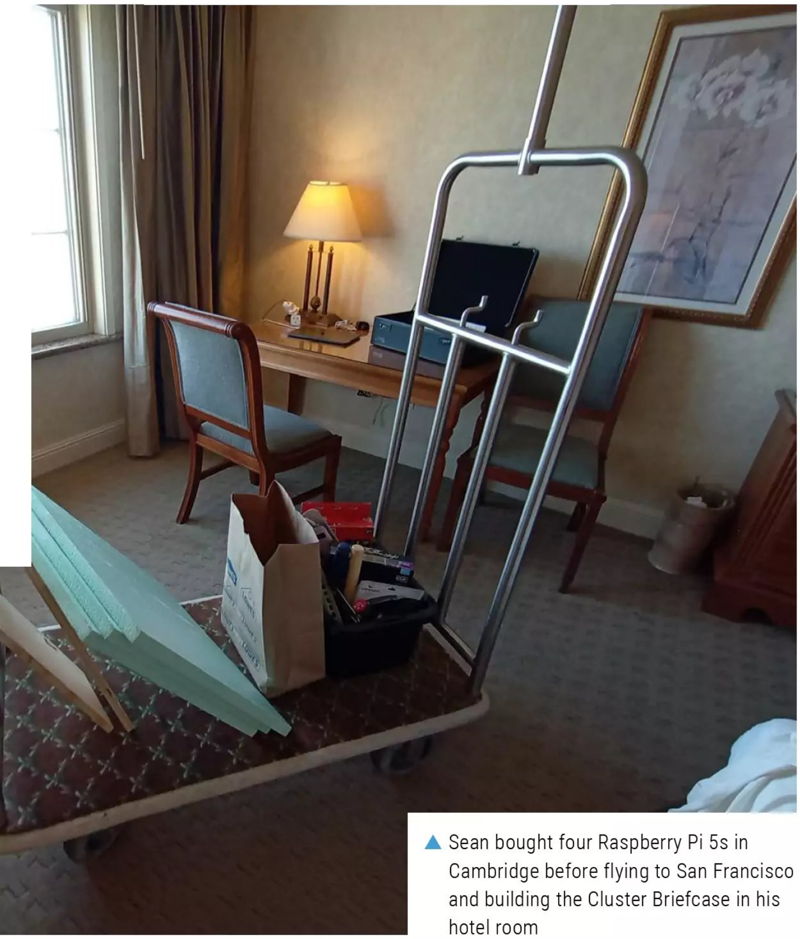
- Sean built almost everything in his hotel room
- Precision-cutting the protective foam mould took nearly four hours
- He spray-painted the (originally bright green) mould to disguise it
- He painted inside bushes rather than stink out his hotel room
- The hidden screen is simply a USB-C powered travel display



▲ Sean had just 18 hours to build and test everything to ensure it was “show-ready”

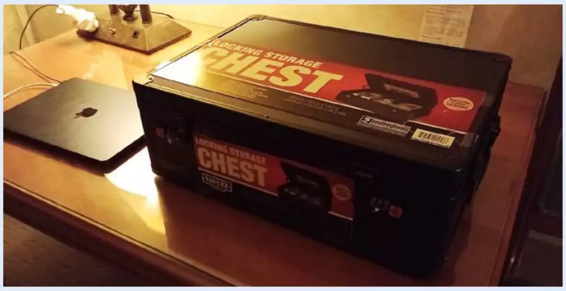


◀ Arranging the Raspberry Pi 5s and power strip before fitting them in the case



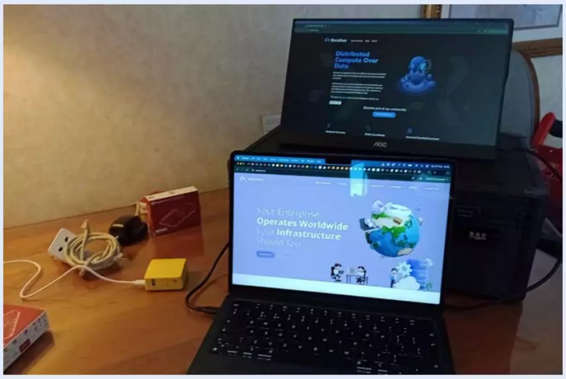
▲ Sean bought four Raspberry Pi 5s in Cambridge before flying to San Francisco and building the Cluster Briefcase in his hotel room

Follow the brief



1. Choose a rugged briefcase large enough to accommodate a travel screen (one powered via USB-C is ideal) along with four Raspberry Pi 5s and a power supply.

2. Shape the padding so everything fits snugly inside. Install the Bacalhau software onto SD cards running Raspberry Pi OS: **rpimag.co/bacalhaugit**.



3. The adaptive and reliable NATS communication protocol is "built right into the heart of Bacalhau" and makes setup on Raspberry Pi straightforward.

Zero-G Laboratory floating platform

Helping explore space by building earthbound robots. **Rob Zwetsloot** investigates the final frontier

Maker

Space Robotics Research Group (SpaceR)

A research group from the University of Luxembourg that specialises in space-bound technology.

spacer.lu

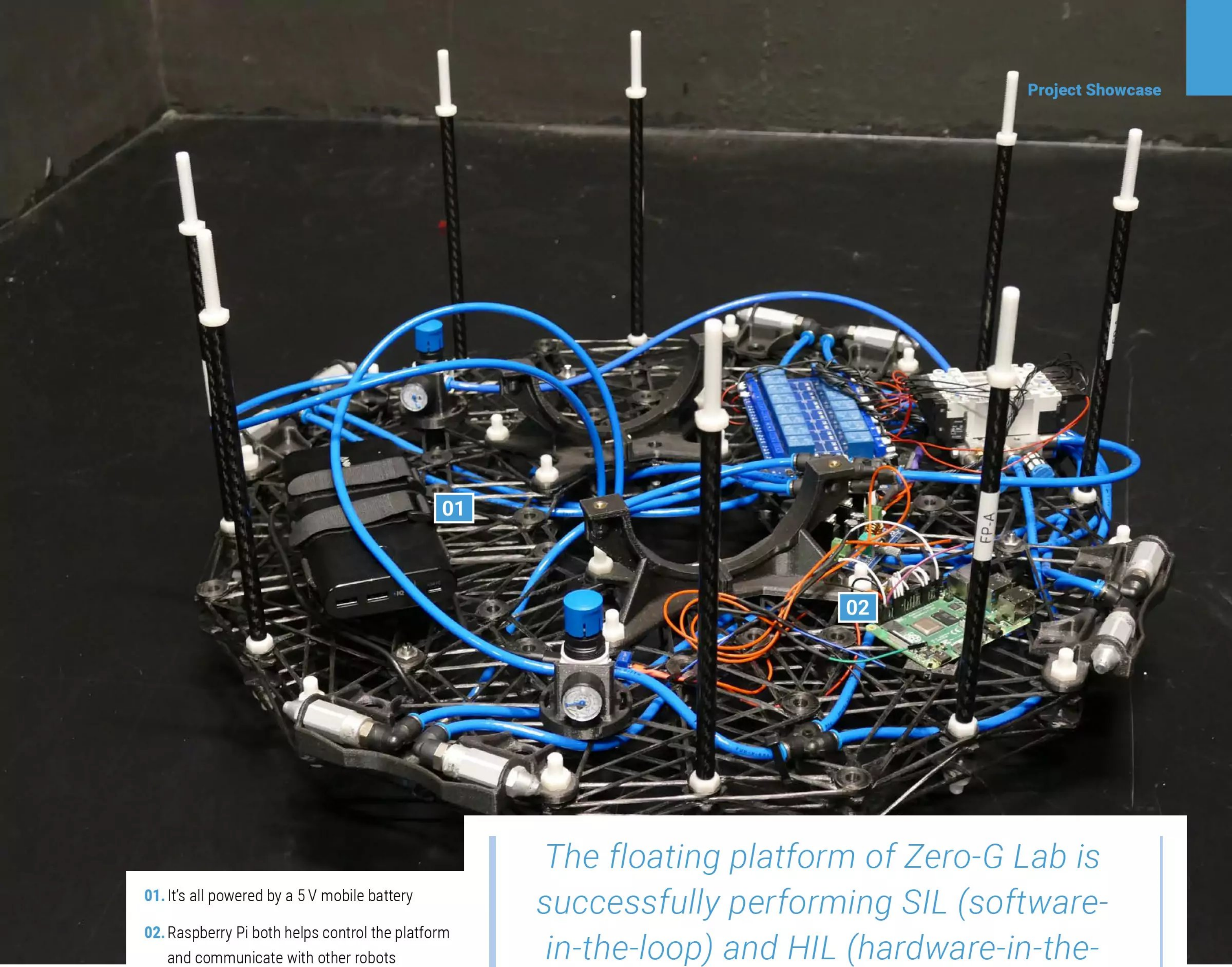
Zero gravity is weird. Most of us can't really conceptualise a weightless environment like you experience on space missions, and there's even a difference between pressurised atmospheres inside a spacecraft and vacuum as well, and even the tiniest variable can affect things. To better understand the various effects, special equipment has to be created.

"The Zero-G Laboratory is specifically designed to emulate scenarios like spacecraft rendezvous, docking, capture, and other interactions between spacecraft," Barış Can Yalçin of the team tells us. "It is equipped with advanced infrastructure, including space-like lighting conditions, a motion capture system, an epoxy floor, mounted robotic rails, and the capability to integrate onboard computers and large mock-ups. These features enable researchers to conduct a wide range of experiments for unique orbital scenarios, allowing for hybrid emulations with robots that integrate hardware and pre-modelled software components. The facility can be operated in real-time and accurately emulate orbital robotics scenarios.

Filling a need

Space is still a huge industry, and labs like this are created to make sure missions are safe and reliable.

"Orbital Servicing, Assembly, and Manufacturing (OSAM), Active Space Debris Removal (ADR), and Asteroid Mining (AM) are becoming increasingly significant in both research and commercial sectors," Barış says. "Earth's orbits are filling up with outdated space assets, while the number of planned missions is set to rise

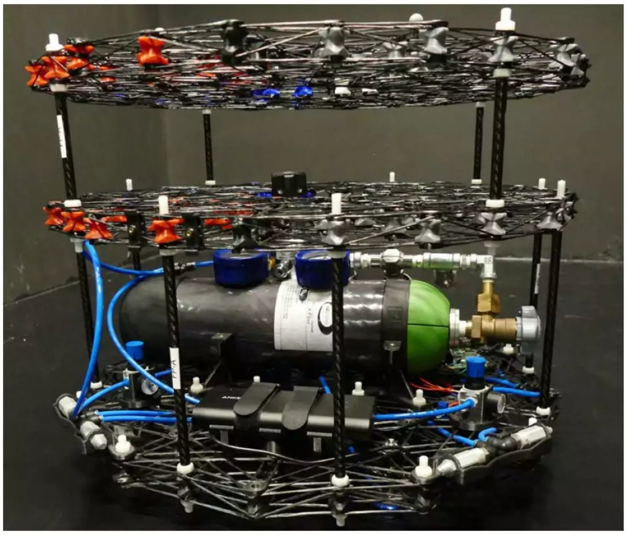


01. It's all powered by a 5 V mobile battery

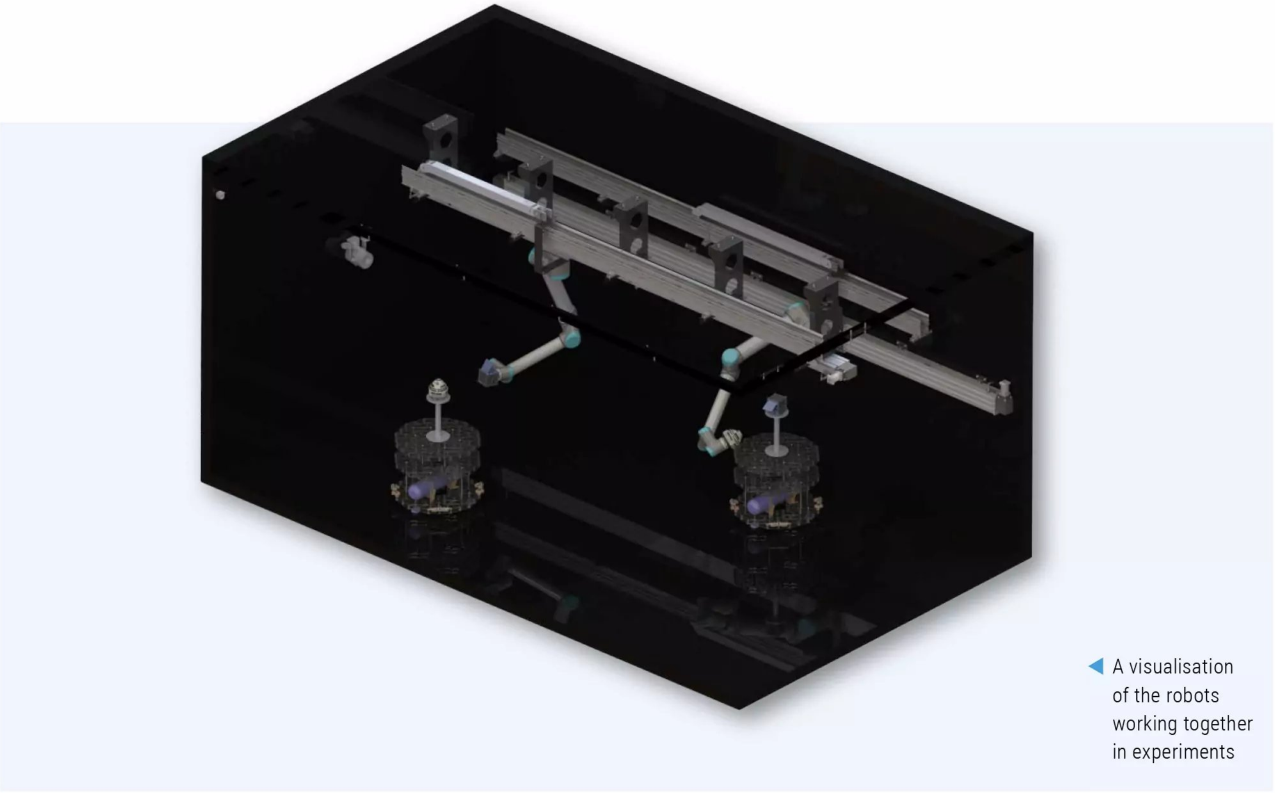
02. Raspberry Pi both helps control the platform and communicate with other robots

sharply in the coming years. Additionally, there are plans to establish multiple space stations and large structures in Earth's orbits over the next decade, which will be partially assembled and/or manufactured in space. These activities require higher levels of autonomy and close interaction. To ensure safe, secure, and reliable in-orbit operations, it is essential to validate and verify Guidance, Navigation, and Control (GNC) algorithms on the ground before launching missions. Consequently, there is a growing need to develop effective experimental setups for testing these algorithms. Therefore, in the academia and industry, floating platforms have been developed and frequently used by many institutions to emulate orbital robotics scenarios."

> *The floating platform of Zero-G Lab is successfully performing SIL (software-in-the-loop) and HIL (hardware-in-the-loop) capabilities*



The fully constructed platform looks like a strange K'Nex set



A visualisation of the robots working together in experiments

Where does Raspberry Pi come in with this scenario? Its small size, low cost, and huge support and compatibility, along with ROS (Robotics Operating System) being available for Raspberry Pi, allow it to perform a great many functions for a project like this. Image processing, data analysis, control algorithms, wireless connectivity, and more make Raspberry Pi ideal for prototyping, then embedding in systems like this.

Some light construction

The system works, to be reductive, like an air hockey table. The floating platforms have an 'air-bearing' that levitates them off the floor a little like a hovercraft, and they have a series of nozzles that grant 3 DoF (degrees of freedom) – they can move along X and Y axes, and rotate around the Z axis. For this to work, the platform needs to be light.

"The floating platform is constructed using additive manufacturing with lightweight carbon-fibre material, which helps extend experiment duration, allowing for the emulation of complex scenarios in the Zero-G Lab," Barış explains. "The lighter the floating platform, the less compressed air it consumes. Moreover, the floating platform features a modular design, allowing easy disassembly of the middle and upper plates, which can also serve to carry equipment for various emulation scenarios. Its string-like topology provides ample space for mounting multiple components. 3D-printed supports can be easily integrated into the string structure, enabling the assembly of additional equipment such as debris removal systems, debris mock-ups, refuelling or docking mock-ups, sensors, and more. Each plate has a 60cm diameter, and the distance between them is adjustable, offering flexibility for accommodating different types of equipment, making the platform highly versatile for various applications."

Lift-off

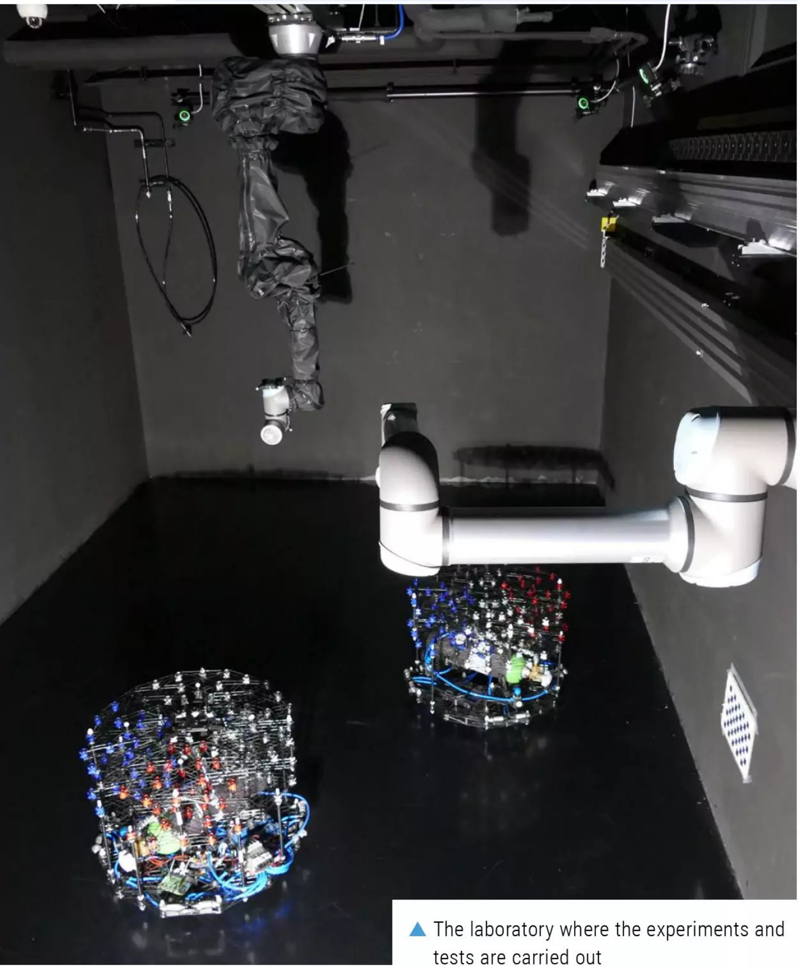
The good news is that the floating platform works as expected.

"From the numerical results gathered during the experiments, we confirmed that the proposed floating platform used with Raspberry Pi is suitable for emulating on-orbit scenarios" Barış reveals. "The floating platform of Zero-G Lab is successfully performing SIL (software-in-the-loop) and HIL (hardware-in-the-loop) capabilities. Several mission-specific proof-of-concept tests such as rendezvous and docking, on-orbit interaction, landing, etc. that leverage the floating platforms in the Zero-G Lab have been realised."

Maybe a Raspberry Pi-powered robot will be tested on here for a future mission? Only time will tell.

Quick FACTS

- Parts of the design are patented
- The platforms communicate with the rest of the lab's robotics
- This communication is handled via ROS
- Solenoid valves manage the pressure, to a max of 10 bar
- A pressure regulator helps control the external compressed air source



▲ The laboratory where the experiments and tests are carried out

You will believe a platform can float

1. "A 3 litre carbon-fibre compact air bottle is mounted on the floating platform as the compressed air source," says Barış. "The selection of carbon-fibre material for the air bottle aims to reduce the overall weight carried by the floating platform."

2. "The criteria for selecting an air-bearing are directly linked to the weight of the floating platform and the load it will support. In this floating platform setup, a 40 mm diameter flat round air-bearing is employed, capable of lifting a maximum of 22.5 kg when actuated at 5 bar. When combined, three air-bearings can achieve a maximum lifting capacity of 67.5 kg."

3. For longer operations, two 30 litre pressure bottles with a maximum capacity of 300 bar can be connected to the platform for a direct source. This extends use from 40 minutes to up to eight hours of operation.

PROJECTS FOR MAKERS & HACKERS
BUILD A FLAT-PACK ROCKET
MAKE ELECTRONIC MUSIC WITH A RASPBERRY PI PICO
BOOK OF MAKING
2025
MAKE A CONNECTED PLANT MONITOR
BUILD SMART HOME LIGHTING
& LOTS MORE
FROM THE MAKERS OF HackSpace MAGAZINE
BOOK OF MAKING 2025
Projects for Makers & Hackers

Tin can plant monitor

HOW I MADE
PRINT YOURSELF A LARGE FORMAT CAMERA
Create super high-quality images using a camera you printed yourself

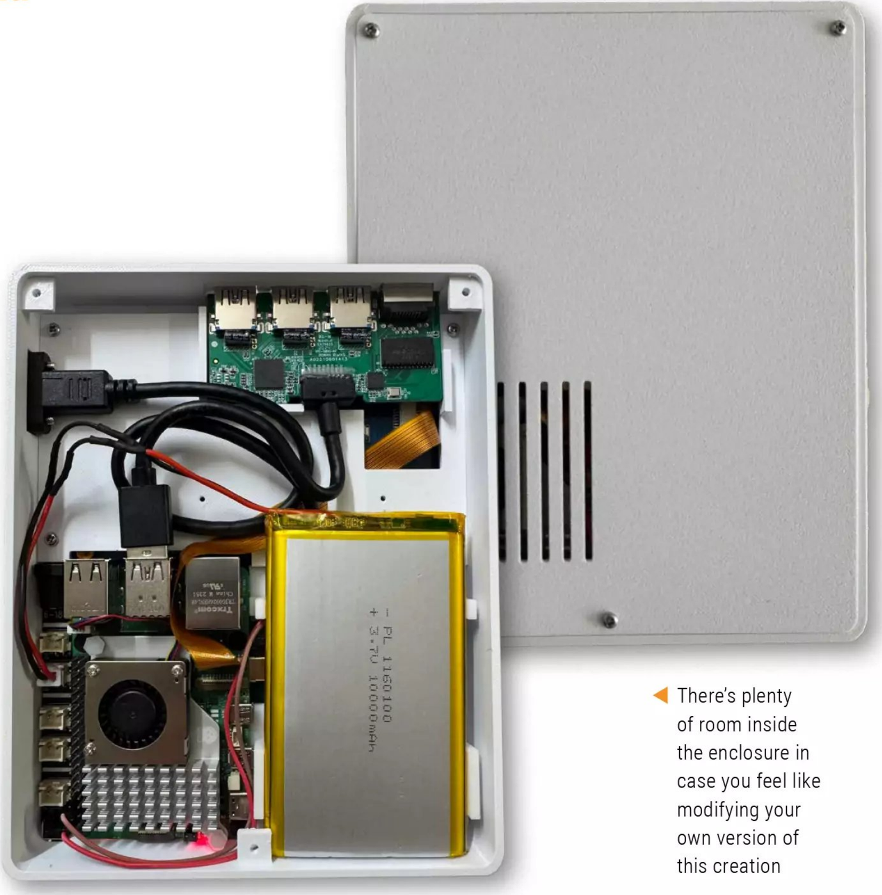
Pi Deck Handheld

By Benjamin Ma

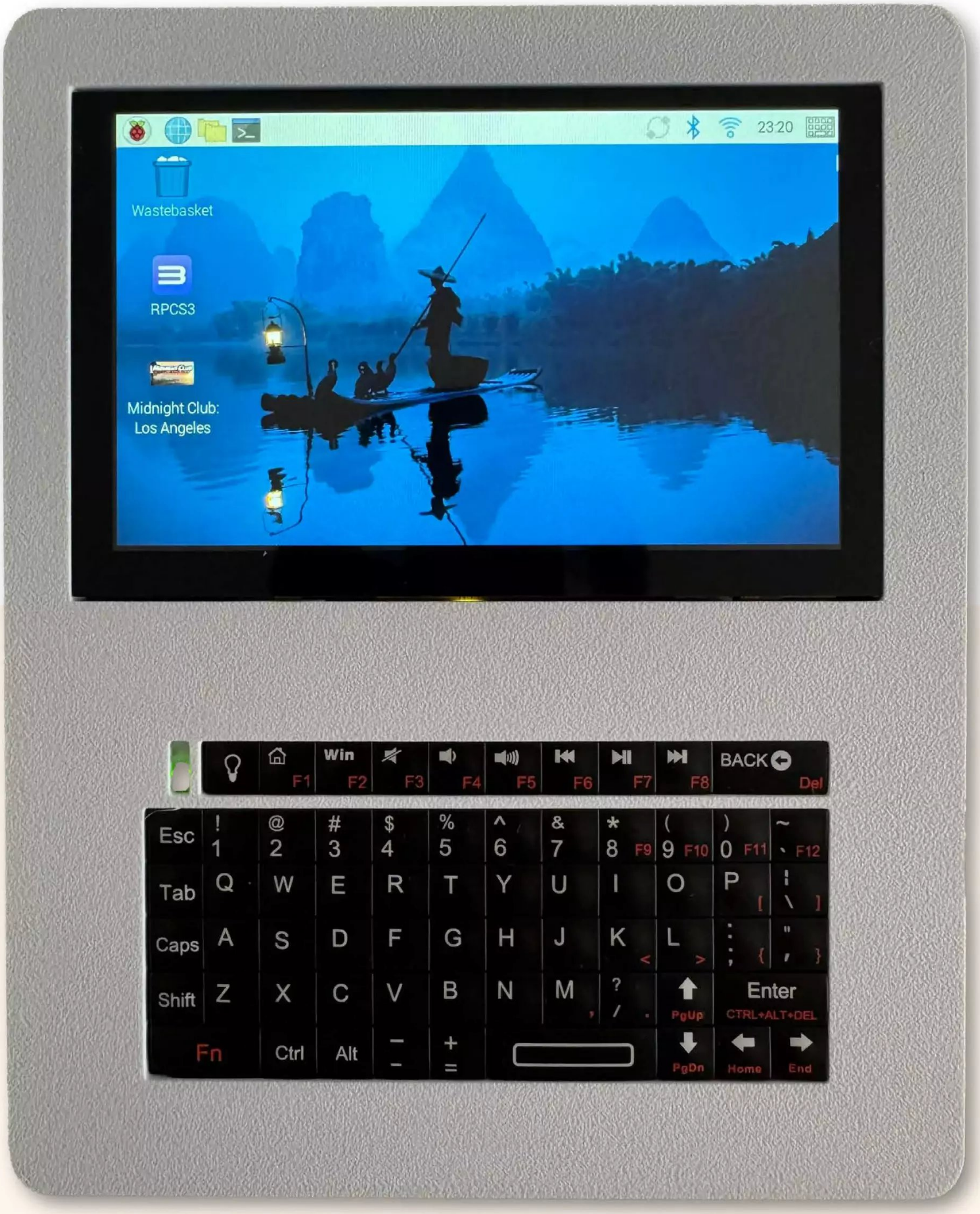
rpimag.co/PiDeckHandheld

We've seen Raspberry Pi computers packaged up into portable, all-in-one devices many times before; search online for 'cyberdeck' and you'll see the variety of uses that Raspberry Pi gets put to. With that in mind, this handheld computer by Benjamin Ma doesn't win any prizes for originality, but in terms of execution it steps up to the plate and hits it for six. Ahem.

Benjamin's gone for an aesthetic somewhere between early Apple and Nintendo Game Boy, with a 3D-printed enclosure held together with screws for easy access. There's a 5-inch touchscreen (so there's no need for a trackpad or mouse input) connected to a Raspberry Pi 5 and a rechargeable Bluetooth mini keyboard – you could connect this to the Raspberry Pi 5 with a wired connection, but it's a fiddly process and so the maker kept the original Bluetooth connection to minimise the amount of wiring in the case.



There's plenty of room inside the enclosure in case you feel like modifying your own version of this creation


23:20
Wastebasket
RPCS3
Midnight Club:
Los Angeles

Split Flap Display

By Morgan Manly

rpimag.co/SplitFlapDisplay

Split flap displays: they're a brilliant bit of retro technology, existing long before LEDs, OLEDs, 7-segment displays, and other modes of showing information. And yet none of those has the same majestic click-clack sound that a split flap display does. Maker Morgan Manley always wanted one of these for his desk, but most split-flap displays are large, loud, and have a load of wires hanging out the back – Morgan wanted something neater. And so he created this: a compact, modular and fully enclosed display.

Each module is approximately 40mm wide and 80mm tall, so the full eight-module device is only 320mm long in total. The modules fit together with only a four-pin connector between each module, and it's powered via a USB-C power adapter.

The build is controlled by an ESP32 board in the leftmost module – all the other modules connect to this via I2C. As the drum within each module rotates, the flaps fall and show the character printed on them – each module displays 37 characters. On a technical note, this is wonderful – the fact that it's modular makes it so easy to think of use cases with fewer modules, and from the evidence of the video Morgan's made it's comfortably quiet enough to sit in our office.

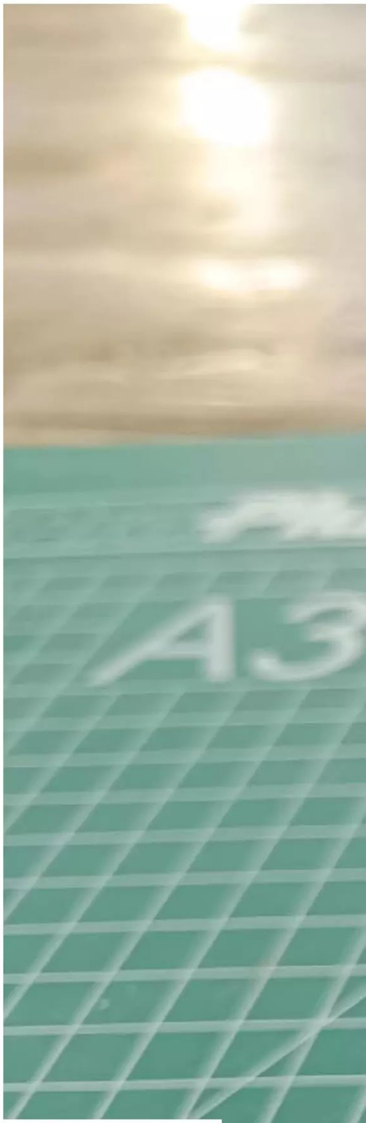
▼ Unlike a screen, a split-flap display doesn't glow blue photons into your eyes – this device will leave your circadian rhythms unruffled!

BANANA

Pi Zero-Based Instant Camera

By Spacerower

rpimag.co/ZeroInstantCamera

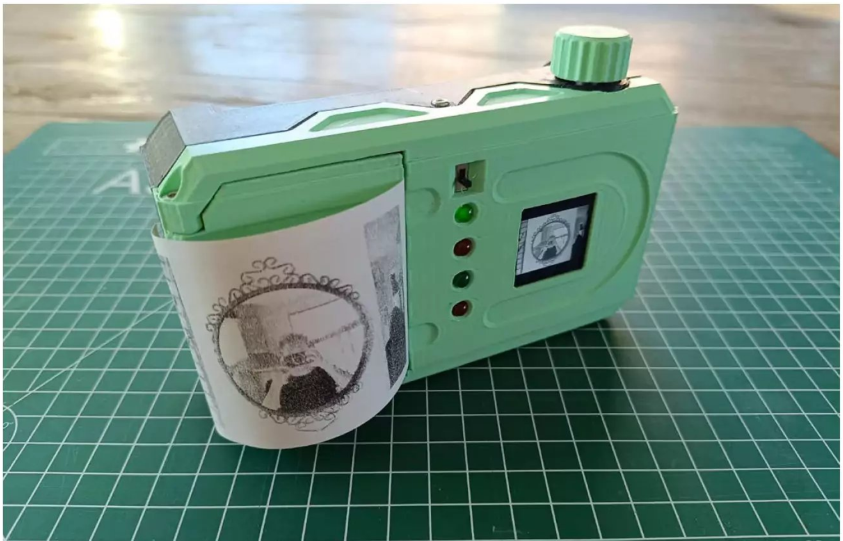


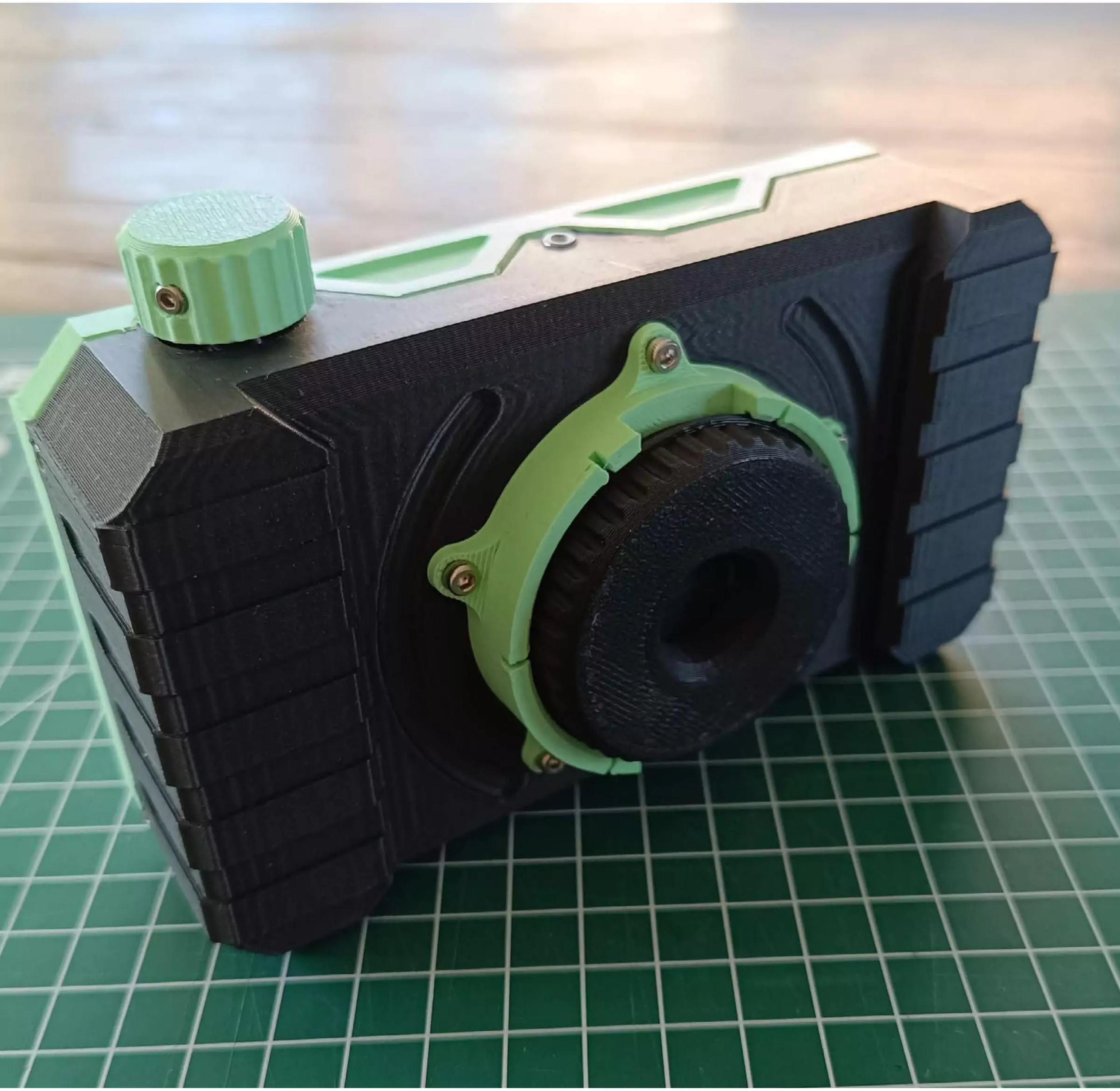
This author has recently acquired a Polaroid camera. They're great fun, they work like magic, and they give you an excuse to stare intently at a blank space until meaning appears (one of our favourite hobbies). The other wonderful thing about Polaroids is that, bereft of the mega-sensitive optical sensors that modern camera phones have, they are really limited in how much light they let in, giving all of our photos a murky, 1970s brown feeling.

Printables user Spacerower has brought some of that instant photo magic to this build, an instant camera that produces low-resolution black and white images using a thermal printer. It's powered by a 1200 mah lithium-ion battery, and uses a Raspberry Pi Zero W and a Raspberry Pi Camera Module.

Other than the fact that this machine is magic in a box, the thing that we're impressed by is the interface. There's no room on the tiny 240×240 screen to display menu options, so the user controls the brightness of the camera using an EC11 rotary encoder. A short press of the green button takes a picture, and a long press prints it out.

▼ Who needs pixels when you've got paper?





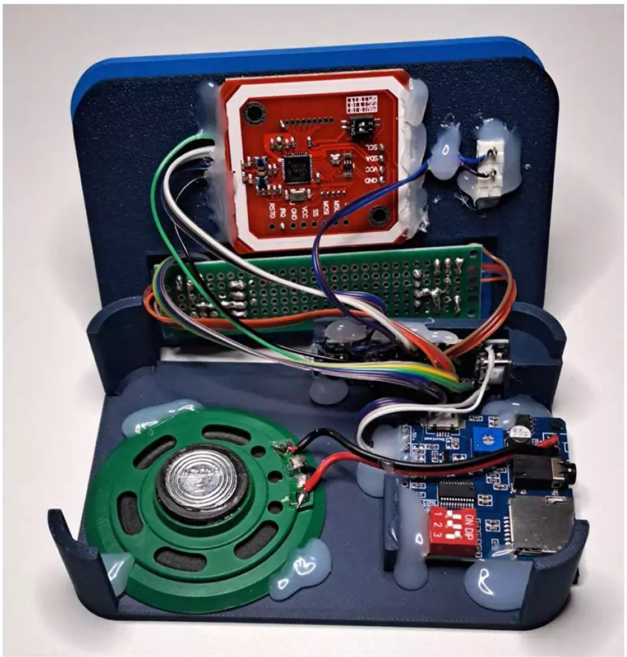
Yaydio

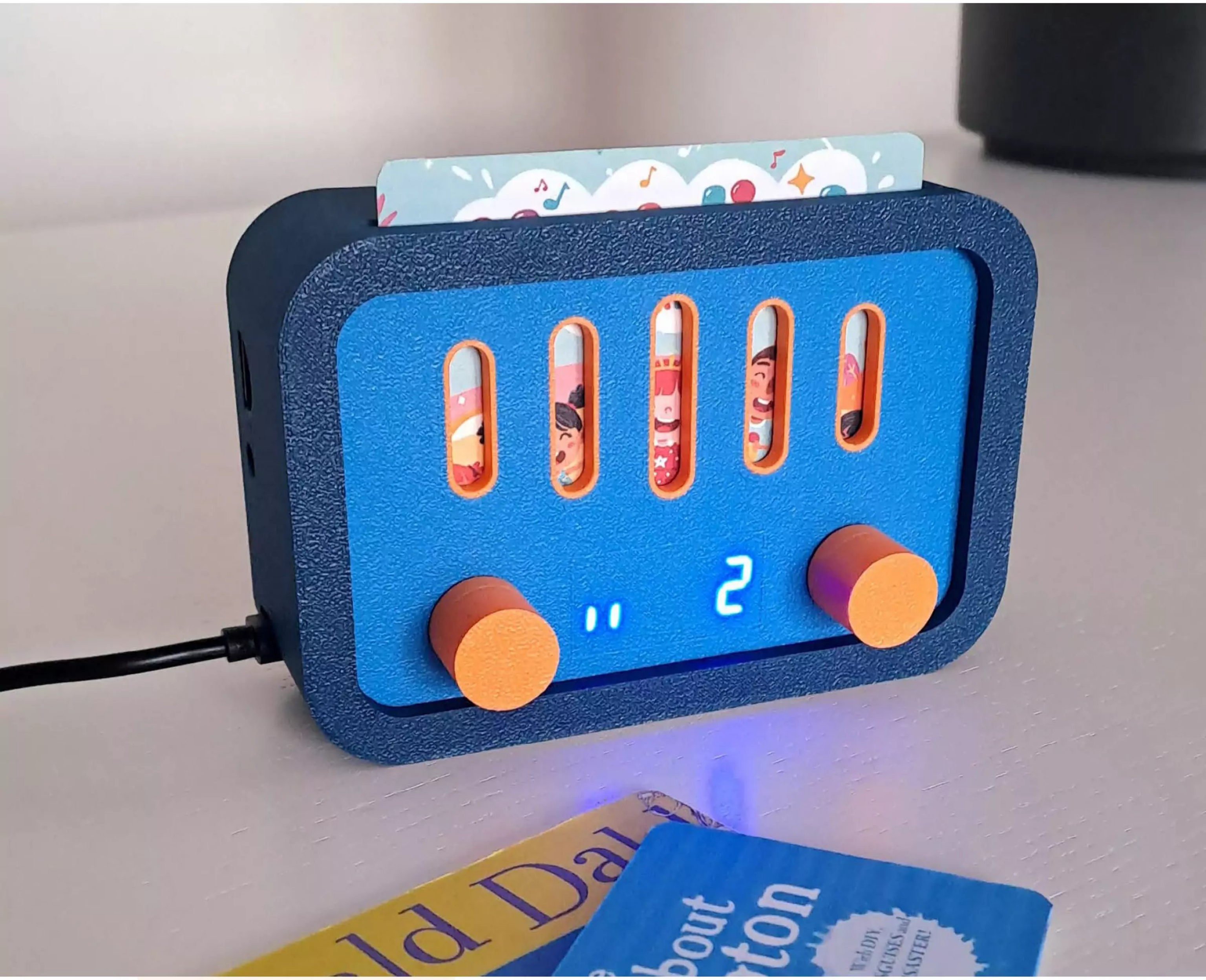
By Jacob Drew

rpimag.co/Yaydio

When this author was a very small child, he took his mother's unopened copy of the Beatles' White Album out of its pristine wrapper and slit it back and forth across the living room carpet like a black vinyl frisbee. It now sits, still unplayed but now scratched to oblivion, a silent reproach to those who would leave valuables within reach of children. Children love to destroy, but they also love to listen to music. How best to introduce them to the classics in a way that they can get involved with, without risking damage to your precious music collection?

Jacob Drew has had a good go at fixing this conundrum with the Yaydio. This beautiful device stores up to 9999 albums (with up to 999 songs per album), and plays them when the right RFID keycard is inserted, which means that a child can spill as much orange juice as they can over the cards and they'll still play. The controls are simple: play/pause, back/forward, and volume up and down. There's a built-in speaker, or you can plug in a 3.5mm jack. And that's about it. It's built around an Arduino Nano, with a four-digit display and a pair of potentiometers, and we think it would be an inspiring challenge to build something similar based on a Raspberry Pi Pico.





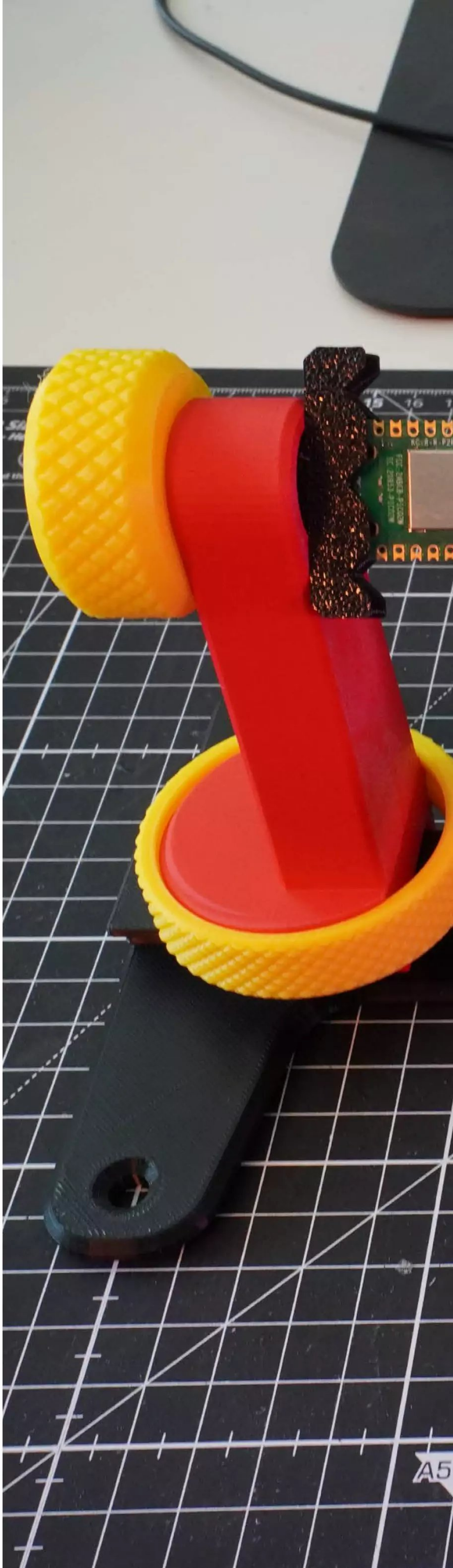
▲ The Yaydio was printed on a Bambu Lab A1 with AMS, using a 0.4 mm nozzle in three shades of PLA: Matte Mandarin Orange, Matte Marine Blue, and Matte Dark Blue

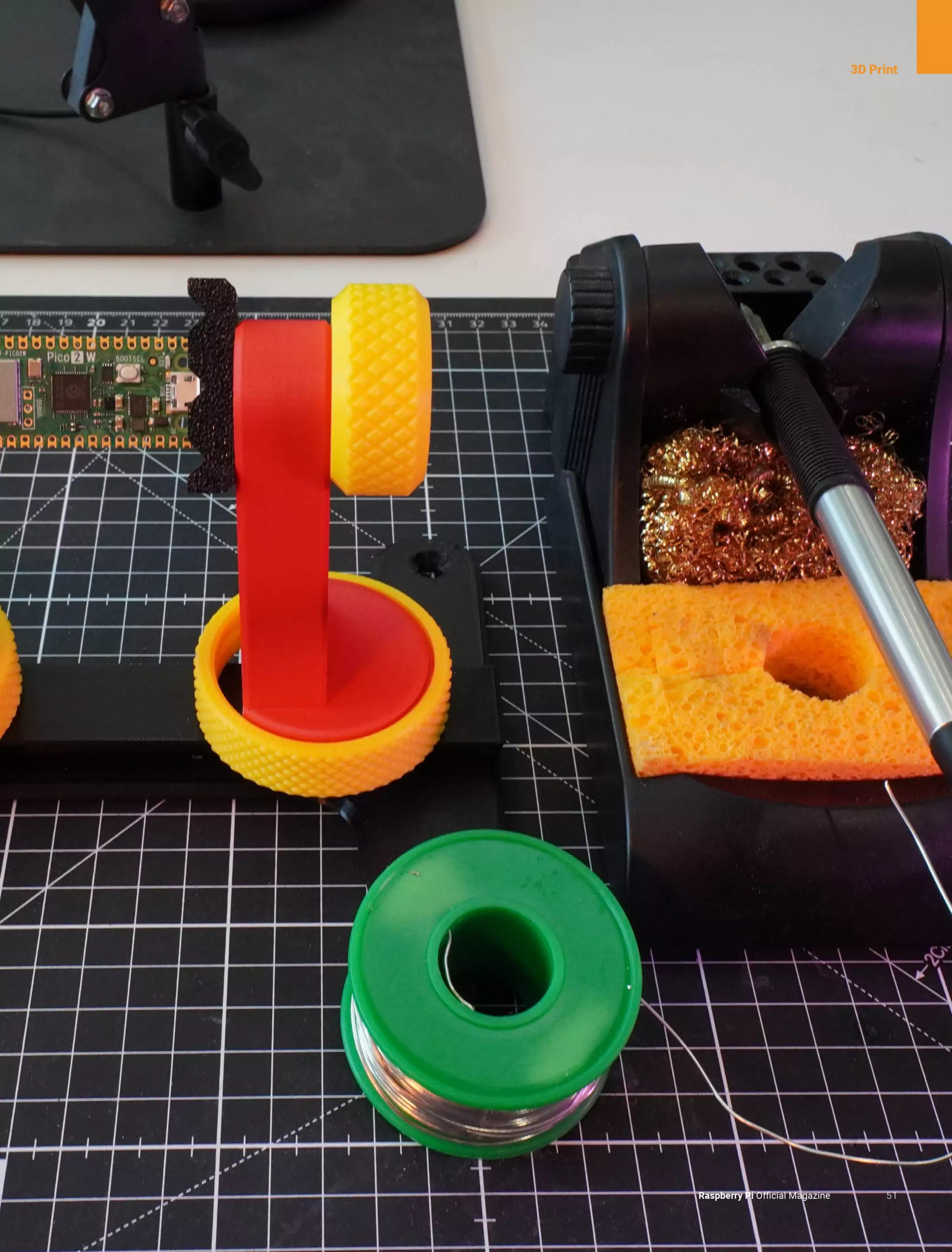
3D print

Keep things steady for soldering with this adjustable PCB holder. By **Toby Roberts**

rpimag.co/PCBHolder

We love a functional print here at Raspberry Pi Mansions. Especially when we have a room full of printers, but inexplicably, not enough clamps to hold the things that we want to solder. One print that has come in handy recently is this perfect little PCB holder (designed by Shawn Quinn), which is the perfect size to hold a Raspberry Pi Pico in place. We printed this clamp in PLA using the Bambu Lab A1 equipped with the AMS (Automatic Material System). Although the AMS supports multicolour printing, each colour was printed separately in this case to minimise material waste from purge towers and excess nozzle 'pooping' – otherwise we'd have wasted loads of filament.

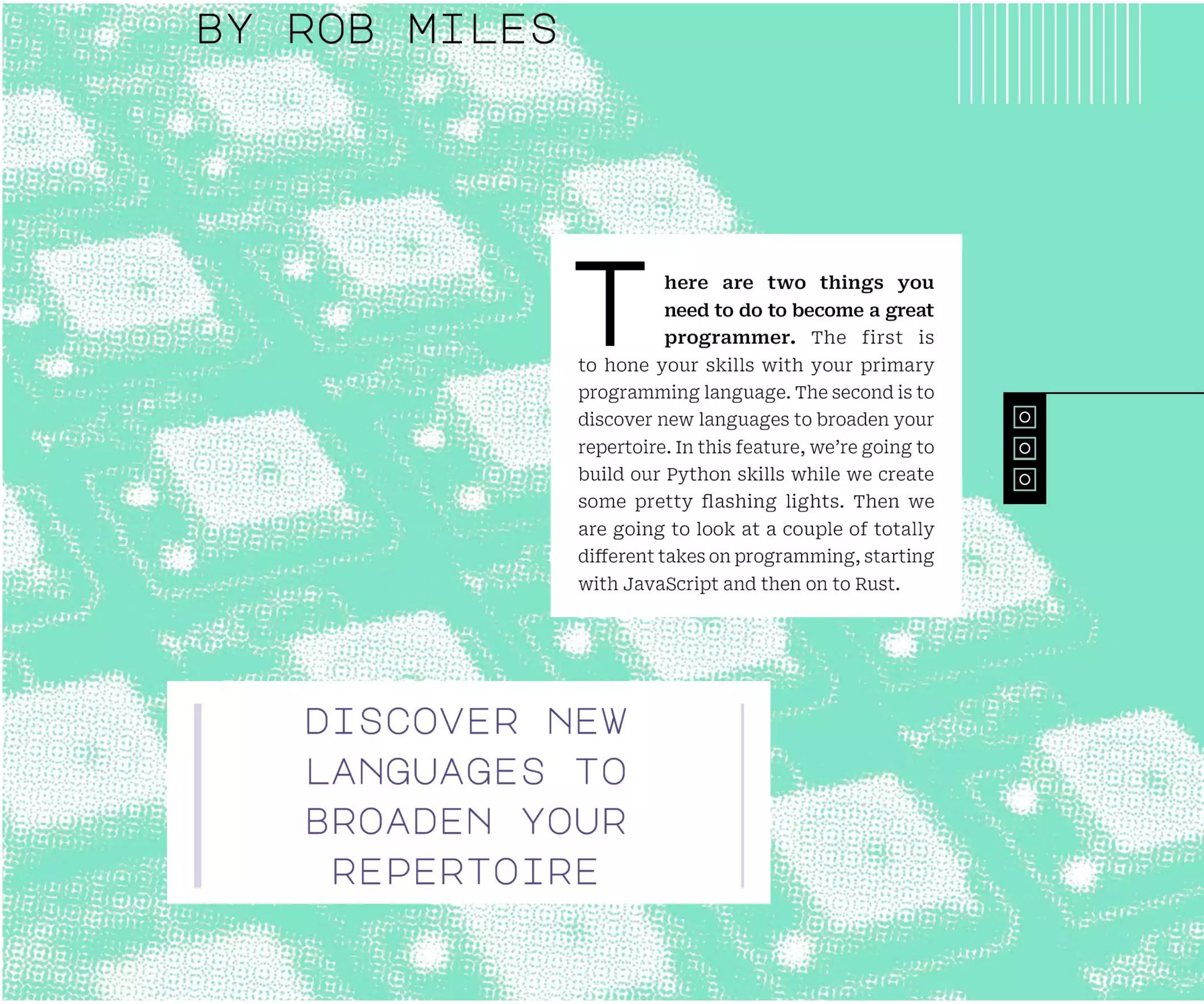




P0WER C0D1NG

TAKE YOUR CODING SKILLS TO A WHOLE OTHER LEVEL.

BY ROB MILES



There are two things you need to do to become a great programmer. The first is to hone your skills with your primary programming language. The second is to discover new languages to broaden your repertoire. In this feature, we're going to build our Python skills while we create some pretty flashing lights. Then we are going to look at a couple of totally different takes on programming, starting with JavaScript and then on to Rust.

DISCOVER NEW LANGUAGES TO BROADEN YOUR REPERTOIRE

BUILD YOUR: PYTHON SKILLS

"PICK UP PYTHON SKILLS AS YOU PLAY WITH PRETTY LIGHTS AND CONTROLS"

YOU'LL NEED

- Raspberry Pi Pico or Pico W (1 or 2)
- An 8×8 NeoPixel array
- Two rotary encoders

Let's discover some useful features of Python and use them to create coloured lights. We're assuming that you know a bit of Python (enough to flash an LED or turn on some NeoPixels) and want to level up a bit. You can find complete code examples here: **rpimag.co/robmilesgit**. All the samples are in MicroPython and run on any version of Raspberry Pi Pico. You can build your own version of the demonstration hardware using a NeoPixel panel and any model of Pico.

PICO LIGHTBOX

Figure 1 shows the light box. It contains an 8×8 pixel array and a Pico. There are also two rotary encoders connected to Raspberry Pi Pico. Each has an internal switch

Figure 2 shows the circuit diagram for the complete device. You can use a differently sized panel, but you might need a more powerful power supply if you have very large numbers of pixels.

If you just want to control the pixels, you can construct a simple version from just a Pico and a display panel as shown in **Figure 3**. Now that we have the hardware, let's make some software.

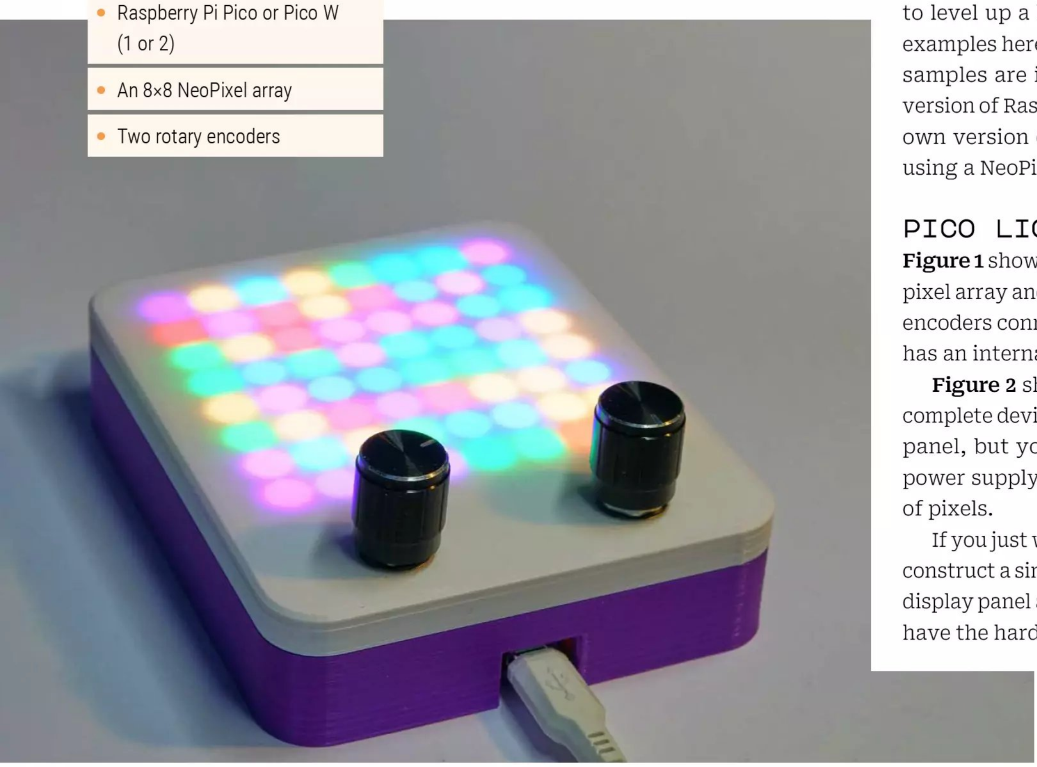


Figure 1: finished project; STL files for 3D-printing the case can be found in the GitHub repo

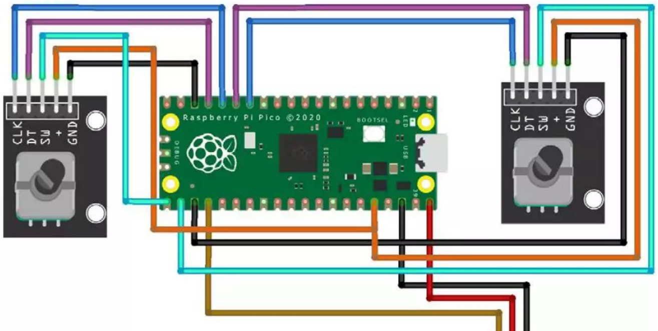


Figure 2: The wiring diagram for the project

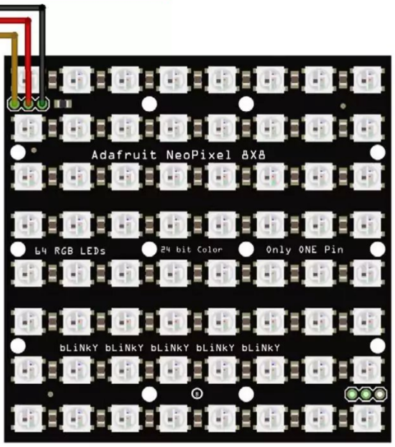
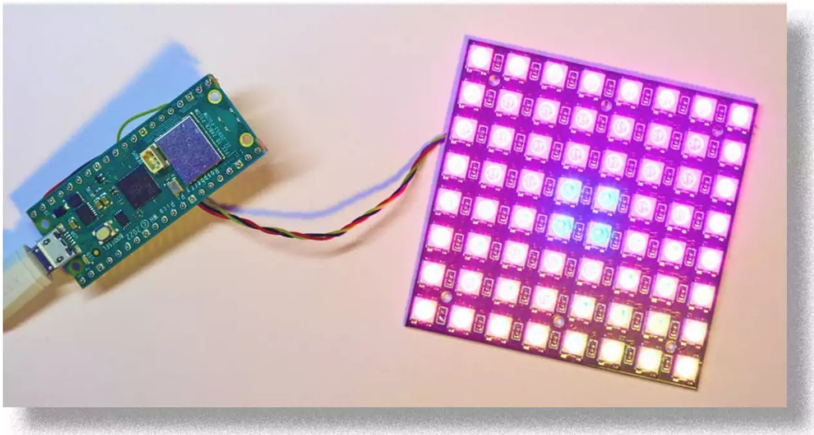


Figure 3: The panel is connected to Pico's VSYS, GND, and GPIO18 pins

QUICK TIP

Use CAPITAL LETTERS to name variables that you want to be used as constant (unchanging) values.



A TOUCH OF CLASS

A Python class is something which holds properties (data variables) and methods(functions). Let's make a class called `Col` to look after pixel colours in our NeoPixels. A colour is represented by the intensity of red, green, and blue in it. Intensity values are integers in the range 0–255. We can put the colour of a pixel in a Python tuple containing three values:

```
class Col:
    RED = (255, 0, 0)
    YELLOW = (255, 150, 0)
    GREEN = (0, 255, 0)
    CYAN = (0, 255, 255)
    BLUE = (0, 0, 255)
```

A tuple is like a list, but its values are immutable, which means the values cannot be changed once created. The code above creates a class called `Col` which defines some colour properties. Now let's use these to draw some pixels.

QUICK ON THE DRAW

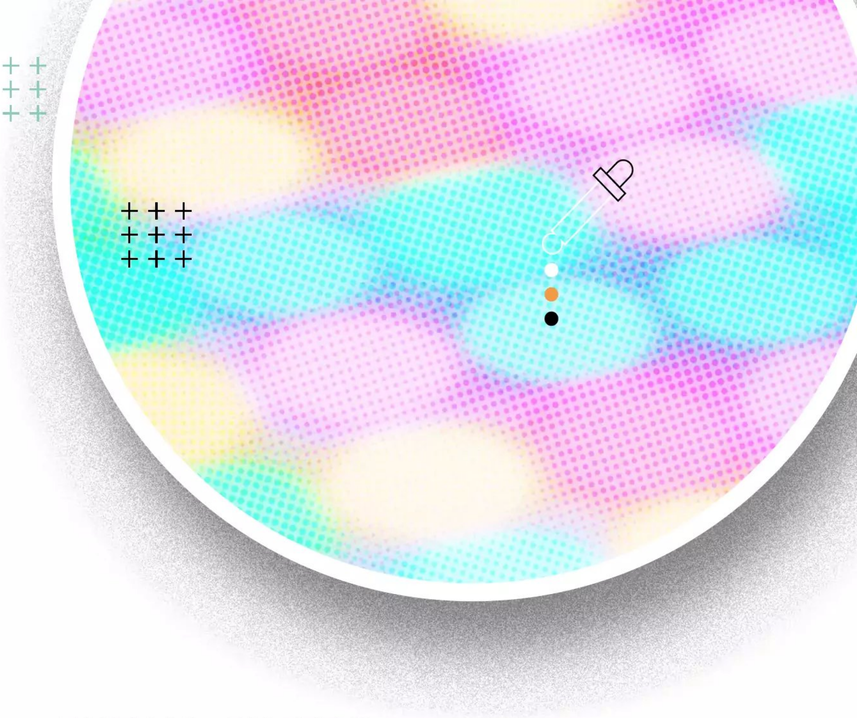
```
import machine
import neopixel
```

The two statements above import the `machine` library which we will use to interact with hardware pins. The `neopixel` library is also imported so that our program can control NeoPixels.

```
PIXEL_PIN = 18
NUM_PIXELS = 64
```

These two statements define the pixel pin and the number of pixels in the display. In our hardware, the pixels are connected to GPIO18, so the pixel pin is set to 18. If your pixels are connected differently, you must change this value.

```
np = neopixel.NeoPixel(machine.Pin(PIXEL_PIN),
NUM_PIXELS)
```



This statement creates a `NeoPixel` object called `np` to control the pixels. Now we can use this to set a pixel a particular colour:

```
np[0]=Col.RED
np.write()
```

The code above sets the pixel at the start of the pixels (in location 0) to red and then writes new colours to the NeoPixels. The pixel will turn red. We can use the values in `Col` anywhere in our program. Now let's put `Col` to work for us.

> A PYTHON CLASS CAN CONTAIN METHODS

METHODS IN OUR MADNESS

A class can contain methods. We can make `Col` into a container for colour manipulation code, starting with a method that lets us create dimmed versions of our colours:

```
@staticmethod
def dim(col):
    return (int(col[0]/20),
            int(col[1]/20),
            int(col[2]/20))
```

The method is marked as `@staticmethod` because it is part of the class, not an instance. The `dim` method takes in a colour and returns a dimmed version of it. The dimming is achieved by dividing the intensity of each component (red, green, and blue) by 20.

```
np[10]=Col.dim(Col.GREEN)
```

The code above would set pixel number 10 to a dim green. We could add other functions to `Col` to mix colours or calculate intermediate ones.

TASTEFUL PALETTES

We can also add 'palettes' to the `Col` class so that we can bring together a set of colours that work well together.

```
 RAINBOW = (RED, ORANGE, YELLOW, GREEN, CYAN,
BLUE, INDIGO, VIOLET)
 FIRE = (RED, ORANGE, YELLOW)
 COOL = (CYAN, BLUE, MAGENTA)
 MONO = (BLACK, GREY, WHITE)
```

These are defined inside the `Col` class. We will use them later when we start making flickering lights.

RANDOM COLOURS

And finally, we can add a method to `Col` that will pick a random colour from a palette:

```
 @staticmethod
 def random_from_palette(palette):
     """Pick a random colour from the given
palette."""
     if not palette:
         return Col.BLACK  # fallback default
     return palette[random.randint(0,
len(palette) - 1)]
```

The method is provided with a palette to choose from. It uses the `randint` method from the `random` library to pick a random colour from that palette. If no palette is provided, the method will return black.

CLASSY CONTAINERS

Whenever you have a bunch of values and some little behaviours that you want to bind together, putting them in a class like this is a great idea. Now that we have our colours, let's make some flickering lights.

FLICKER LOADS OF LIGHTS WITH ASYNCIO

Flickering lights are always fun to make and watch. Let's start by making a single NeoPixel display a colour sequence.

```
def flicker_pixel(index, cols, delay):
    while True:
        for col in cols:
            np[index] = col
            np.write()
            time.sleep(delay)
```

The function `flicker_pixel` above is given the number of the pixel to flicker (`index`), a list of colour values (`cols`), and the time interval between each flicker (`delay`). It then repeatedly cycles through the colours, calling `time.sleep` after each update to pause the display. We would call the function like this:

```
flicker_pixel(0, Col.RAINBOW, 0.5)
```

The statement calls `flicker_pixel` to animate pixel 0 through the rainbow palette with half-second gaps between each colour. The `flicker_pixel` function will never return from this call because it contains a `while` loop which never ends. However, the function will spend most of its time sleeping between colour changes, which seems like a waste.

THE KITCHEN ASYNC

The Python asyncio framework lets a program perform other actions while a function is 'asleep'. Let's see how we can use this to create multiple flickering pixels that flicker at different rates. The first thing we need to do is include asyncio in our program:

```
import uasyncio as asyncio
```

Now we can rewrite the `flicker_pixel` function to make it 'asynchronous'.

```
async def flicker_pixel_async(index, cols,
delay):
    while True:
        for col in cols:
            np[index] = col
            np.write()
            await asyncio.sleep(delay)
```

There are only a couple of changes to the code. The first is that the function is now marked with the keyword `async`. The second is that the function now calls `asyncio.sleep` to perform the delay. Let's look at what these changes mean.

COMPUTER WORLDS

Normally, when you call a Python function, you get back whatever result the function returns. However, when you call a function marked as `async`, you get back a reference to a 'coroutine'. This is a tiny little 'computer world' object containing the function and what it needs to run. Programs can work with coroutines as they would with any other variables.

```
flicker_coroutine = flicker_pixel_async(0,
Col.RAINBOW, 0.5)
```

The statement above creates a variable `flicker_coroutine` that refers to a coroutine holding a call to `flicker_pixel_async`. Later, we'll see how to start a coroutine running, but now let's take a look at the other change in `flicker_pixel_async`: the use of `asyncio.sleep`.

ASYNCHRONOUS SLEEPING

The `time.sleep` function we used in the original flickering light causes a program to pause for a specified time. The `asyncio.sleep` function tells the asyncio framework that the coroutine can go to sleep for a while. This means that asyncio can switch to another coroutine that does need to run. At some point the other coroutine will sleep, at which point execution will transfer somewhere else. After the sleep is completed, the coroutine will be eligible to run again. The asyncio framework is a bit like a juggler, swapping between coroutines as they run and sleep. We can now bring this coroutine to life by running it.

LIVING CODE

```
asyncio.run(flicker_coroutine)
```

The statement above would start the NeoPixel flickering. The `run` method in the asyncio framework takes a coroutine and runs it. This doesn't look very useful so far, but we can make several calls of `run` to start multiple asynchronous functions.

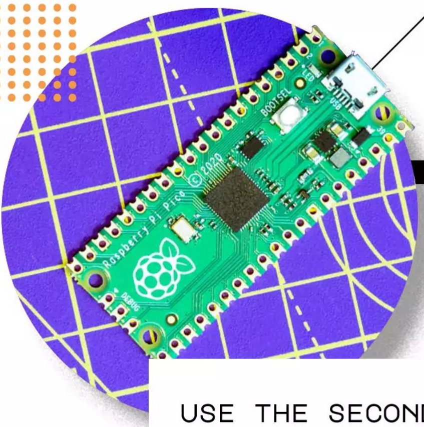
```
 Led0_coroutine = flicker_pixel_async(0,
Col.RAINBOW, 0.5)
 Led1_coroutine  = flicker_pixel_async(1,
Col.COOL, 0.7)
 asyncio.run (asyncio.gather(Led0_coroutine,
Led1_coroutine))
```

The code above creates two coroutines for the first two LEDs. It then runs them using the `asyncio.gather` function. The latter accepts multiple coroutines as parameters and starts them running. It is called 'gather' because if the functions returned results (which the pixel ones do not), it would gather the results and return them when all the functions had finished. What we see is two LEDs flashing at different speeds using different palettes.

We are just scratching the surface of asyncio here. It is a very powerful mechanism. In the code examples for this article, you can find a program that flickers 64 LEDs. There is also an example that uses asynchronous processes to control pixel drawing and the flicker speed using a rotary encoder.

Things to remember about using asyncio:

- At no time are two processes running at the same time – this mechanism just allows a function to 'hand back' the thread of execution when it no longer needs it.
- If one asynchronous function gets 'stuck' and never sleeps, it will stop everything else from running.
- The best way for asynchronous processes to share data is to use a global dictionary which holds named variables.



USE THE SECOND PROCESSOR CORE IN THE PICO

We've seen we can use asyncio to allow a processor to support multiple coroutines, but these don't run in parallel. Instead, asyncio is constantly switching between coroutines. If you want to do things in parallel, you need more than one processor. Let's have a look at how this works and use the second processor in Raspberry Pi Pico to continuously generate a nice display for our pixels.

CORE VALUES

Raspberry Pi Pico has two processor cores. Programs normally run on core 0, but in MicroPython you can start a function running on core 1. The function that accesses core 1 is in the `_thread` library. A 'thread' is the name given to a process that executes concurrently with others. Note that a thread is not the same as a coroutine. Threads run at the same time as other threads. Coroutines cooperate to share a single processor.

```
import _thread
```

The statement above imports the `_thread` library. Now we need a function to run on core 1:

```
def core1_task():
    while True:
        print("Hello from core 1")
        time.sleep(0.6)
```

The function above repeatedly prints 'Hello from core 1' with a gap of 0.6 seconds between each print. Now we need to get this running as a thread on core 1:

```
_thread.start_new_thread(core1_task, ())
```

The `start_new_thread` function is given the function to run, followed by the parameters for the function call. In this case, the parameters are empty because the `core1_task` function does not receive any parameters. At this point in the program, the function is running in a thread on core 1 and messages start to be printed. So now, let's start another loop running on core 0:

```
while True:
    print("Hello from core 2")
    time.sleep(0.5)
```

This prints a hello message every 0.5 seconds. The output is interesting:

```
Hello from core 2
Hello from core 1
Hello from core 2
Hello from core 1Hello from core 2
```

The messages run together because sometimes the two threads try to print a message at the same time. This illustrates a problem when using threads. They can end up fighting over resources.

THREAD LOCKDOWN

We can stop our threads fighting by using a 'lock' which controls access to the `print` function:

```
print_lock = _thread.allocate_lock()
```

The statement above creates a lock called `print_lock`. Only one thread can have the lock at any time.

```
def core1_task():
    while True:
        with print_lock:
            print("Hello from core 1")
        time.sleep(0.6)
```

The updated version of `core1_task` uses the Python `with` construction to get the lock before printing. When execution leaves the code indented under the `with` line, the lock is released. We can add the same mechanism to code on core 0:

```
while True:
    with print_lock:
        print("Hello from core 2")
    time.sleep(0.5)
```

A thread will wait for the lock to become available. If the lock never becomes available, the program will be stuck forever. A system can contain multiple locks to protect different resources.

PRETTY NOISE

The final example (in the repo) uses core 1 to generate a pretty Perlin noise display which is rendered on the display by a process running on core 0. It works because one thread is producing data and another is consuming it. The lock is only used when data is being transferred between the processes. However, multiprocessing can be the source of horrendous bugs involving the timing of processor interactions. Things to remember about using the second core:

- There is a mechanism which allows a thread to do something different if a lock is in use – take a look at the `lock.acquire(False)` method.
- If you don't use the `with` construction to get your locks, you need to be very careful about exceptions being thrown which might prevent the release of a lock and stop your program.
- Both threads share the same variables in your program.
- Do not allow both threads to use hardware such as I2C or serial connections at the same time as these are not 'thread safe'.
- When you start a thread on core 1, you will need to pass in all the item values it needs to work. Look at the Perlin noise example for how to do this.
- When writing code that uses the second core, you might find it hard to interrupt a running program on the second core.

A TASTE OF JAVASCRIPT

A programmer who can only use one language is like a pilot who can only fly one kind of plane. Not as useful as they might be. So, with that in mind, let's look at one alternative to Python: JavaScript.

JAVA HISTORY

The JavaScript language was first developed as a way of making web pages more interactive. JavaScript code can be embedded in web pages to run inside the browser on a user's machine. However, the Node.js platform lets you use JavaScript as a general-purpose language, even in the embedded space. There are libraries allowing JavaScript programs to interact directly with hardware. JavaScript is very like Python in that you can get something going in JavaScript quite quickly. However, JavaScript is also like Python in that code can be quirky and hard to understand unless you know what is going on. Let's look at JavaScript through a Python lens.

INSTALL JAVASCRIPT

Unlike Python, JavaScript is not installed in Raspberry Pi OS by default. Enter the following to install it:

```
sudo apt update
sudo apt install -y nodejs
```

Test that it's working with a simple Hello, World file:

```
nano hello.js
```

Enter the following line of code:

```
console.log("hello, world!")
```

Exit and save the file (**CTRL**+**O**, **CTRL**+**X**), then run with:

```
node hello.js
```

LAID-BACK JAVASCRIPT VARIABLES

Let's start with some stupid code:

```
x = 5
y = "hello"
z = x + y
```

Python does not like this code. When it runs you get an error:

```
 TypeError: unsupported types for __add__:
'int', 'str'
```

We can decode this cryptic message as Python saying: "You've asked me to add an integer to a string, and I don't know how to do that". Python keeps track of variable types. It looks at what is being put into a variable and works out the type from that. It knows that `x` contains a number, and `y` contains a string, and it won't let you add them together. If you really want to add a number to a string in Python, you must explicitly convert the number to a string first:

```
z = str(x)+y
```

JAVASCRIPT AND PYTHON ARE GREAT LANGUAGES

However, if you run the above statements in JavaScript, you don't get an error. And if you have a look at the contents of the variable `z`, you find that it contains the string "5hello". You could interpret this as meaning that JavaScript is cleverer that Python, in that it knows how to add a number to a string. You could also interpret this as meaning that JavaScript is more dangerous than Python, because you might have made a mistake trying to add a number to a string, and you'd like to know when you make mistakes like these.

Welcome to the world of programming language comparisons. And just like there are pilots that claim their favourite plane is better than anyone else's, there are also lots of people claiming their favourite language is 'best'. The author doesn't have any strong opinions on this. He enjoys a good discussion as much as anyone else, but he thinks that the best programming language is the one that you get paid the most to write. With that in mind, JavaScript and Python are great languages. Let's dig a little deeper into JavaScript:

JAVASCRIPT CODING

JavaScript and Python programs look a bit different. JavaScript programs use curly brackets (braces) to denote the start and end of blocks.

```
function add(a, b) {
    let result = a + b;
    return result;
  }
```

The JavaScript code above creates an `add` function which accepts two numbers and returns their sum. You can call it like this:

```
x = add(3, 4);
```

This would set the value of `x` to 7. However, the following statement would compile and run in JavaScript without an error:

```
x = add();
```

The value of `x` would be set to NaN which means 'not a number'. Items missing from a function call are treated as values set to the 'undefined' value in JavaScript. Adding two undefined values makes a result which is not a number.

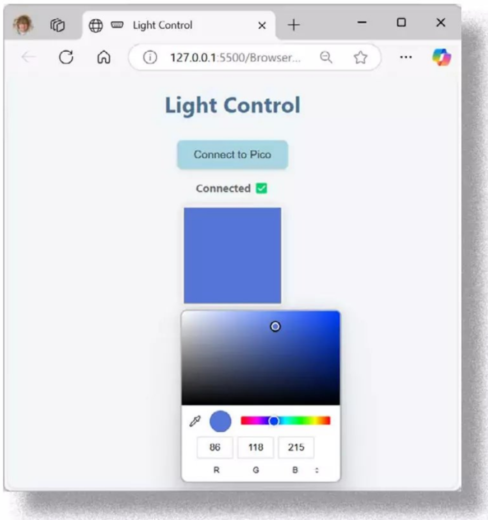
JAVASCRIPT: THE SHOW MUST GO ON

JavaScript has a 'the show must go on' philosophy. Actions such as dividing one by zero or using a variable before you've put a value into it do not cause runtime errors. Instead, JavaScript has values built into the language to mean 'infinity', 'undefined', or 'not a number'. JavaScript programs can (and should) explicitly test for these values when they run to make sure that all is well. This changes the way that you write code when using JavaScript.

When learning a new programming language, you need to be careful that you don't take assumptions from your previous language into the new one. And a lot of learning a new programming language is understanding the underlying philosophy behind the language design. Python is relaxed about some things; for example, JavaScript insists you declare variables before using them in blocks whereas

▶ **Figure 5:** The dial signals are also processed using JavaScript events

▼ **Figure 4:** This page could be hosted on the web so anyone can use it to configure a device



R 86 G 118 B 215

Python doesn't. However, JavaScript does everything it can to make sure a program will run, even though the output might not be what you wanted. So, where can you run JavaScript?

BROWSER JAVASCRIPT

JavaScript running in a browser works with the HTML (HyperText Markup Language) that describes the web page framework within which the program runs. This can be useful for embedded development because browsers based on Google Chrome (including Microsoft Edge) contain a serial port library that makes it possible to connect embedded devices to a serial USB port on a computer and then send commands to them.

Figure 4 shows a Light Control page that can be used to set the colour of a NeoPixel light. JavaScript code in the web page sends colour configuration messages over a serial connection to the light. The light contains a Python program that sets the colour and stores the colour information in the light. You can find these programs in the repository for this article.

NODE.JS FOR THE WIN

If you want to run JavaScript programs on a computer, you use the Node.js framework. Node.js provides a runtime environment for JavaScript programs. A Pico doesn't have enough power to run Node.js, but it will run quite happily on even the lowest-power Raspberry Pi computer. The Node Package Manager (NPM) works alongside Node.js to provide access to a wealth of prewritten libraries, such as ones for hosting web pages, interacting with databases, and hardware interfacing including support for GPIO, I2C, SPI, and serial communication.

Figure 5 shows the 'red telephone'. This contains a Raspberry Pi Zero which is used to detect the handset being lifted and can simulate a telephone call, using text-to-speech to deliver messages when the phone is answered. The phone also hosts a web page which can be used to send messages and trigger the phone to ring.

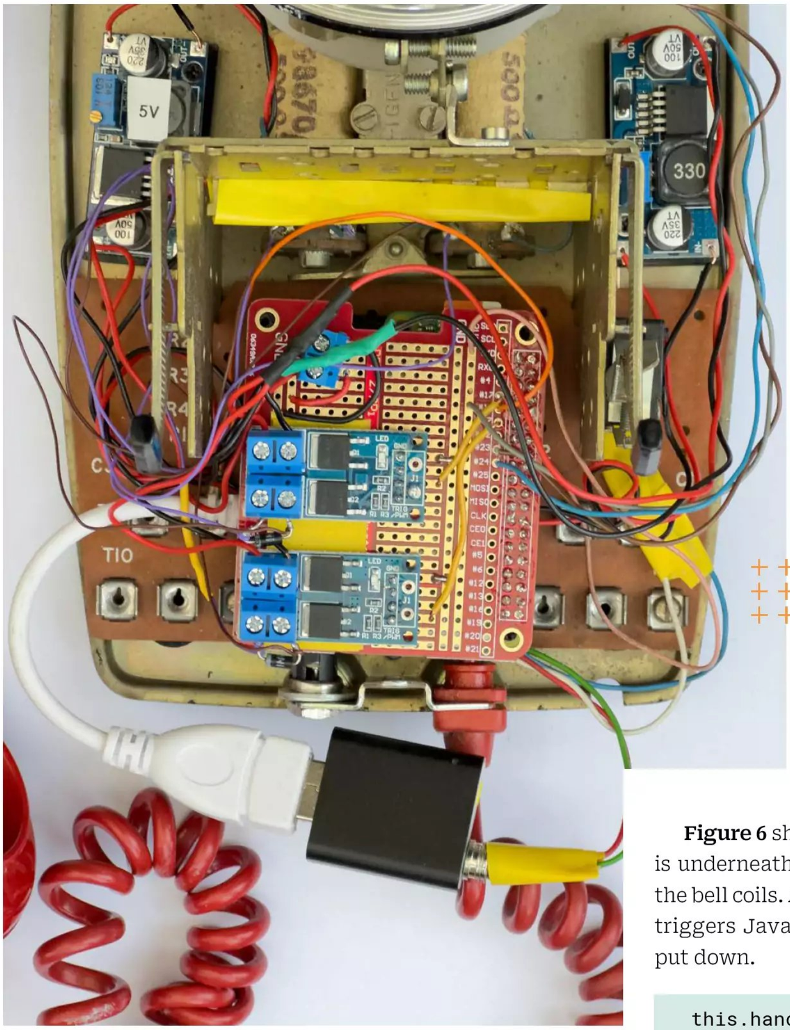


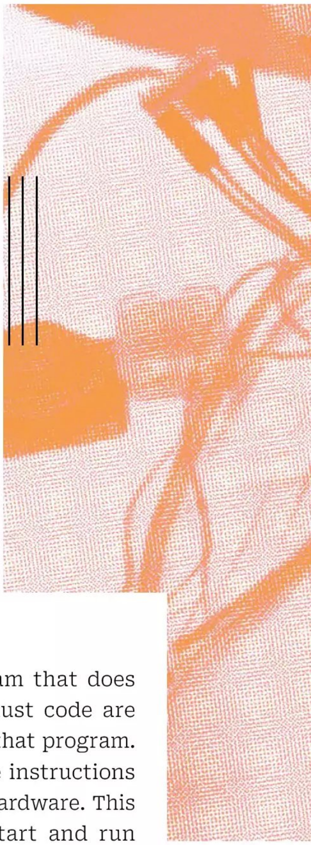
Figure 6: A USB audio interface is connected to the handset speaker

Figure 6 shows the inside of the phone. A Raspberry Pi Zero is underneath the expansion board which holds to drivers for the bell coils. A microswitch connected to the handset switches triggers JavaScript code when the handset is picked up and put down.

```
this.handsetGPIO = new InGPIO(18, (value) => {
    if(value == 1){
        owner.handsetPickedUp();
    }
    else {
        owner.handsetPutDown();
    }
});
```

The code sample above shows how JavaScript code can be attached to input pins. The `handsetPickedup` method is called if the input changes state and the input is high. Otherwise, the `handsetPutDown` method is called. You can find all the JavaScript code for the phone on GitHub here: **rpimag.co/rpidialphone**.

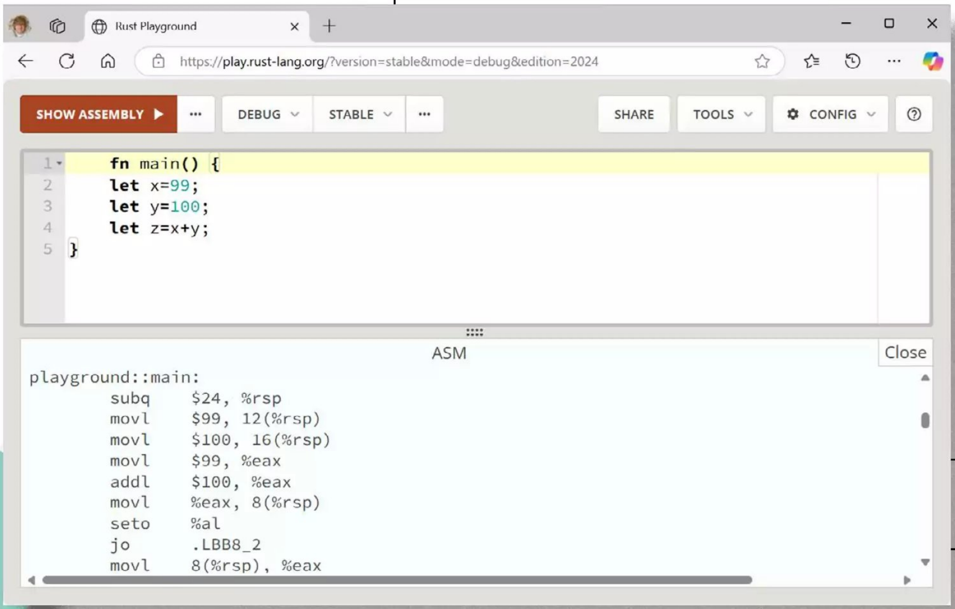
GETTING RUSTY



We can end this exploration of programming with a look at the Rust programming language. Rust is probably not the first programming language you should learn. Maybe it shouldn't be your second either. But at some point, you are going to wonder if there is more to programming and how you can make your programs properly reliable, self-contained, and easy to manage. These are issues you can't really consider when your main problem is getting your code to work, but once you start making useful solutions, they are things you need to look out for. Rust was created with all these issues in mind. Python wants to be easy to learn and use; JavaScript wants every program to do something. Rust wants every program to be solid, safe, predictable, and built to last. Let's look at how it does this.

RUNNING RUST

Figure 7 shows a Rust program that does some arithmetic. Below the Rust code are the assembler instructions for that program. When the program runs, these instructions are performed directly by the hardware. This means that Rust programs start and run quickly. Statements in JavaScript and Python programs are decoded as they run, which slows them down slightly. There are Rust compilers for most platforms, including Pico and Pico W (but not Pico 2 devices at the time of writing).



Rust Playground
https://play.rust-lang.org/?version=stable&mode=debug&edition=2024
SHOW ASSEMBLY
DEBUG
STABLE
SHARE
TOOLS
CONFIG
ASM
Close

```
fn main() {
    let x=99;
    let y=100;
    let z=x+y;
}
```

```
playground::main:
        subq    $24, %rsp
        movl    $99, 12(%rsp)
        movl    $100, 16(%rsp)
        movl    $99, %eax
        addl    $100, %eax
        movl    %eax, 8(%rsp)
        seto    %al
        jo      .LBB8_2
        movl    8(%rsp), %eax
```

Figure 7: You can have a go with Rust at **play.rust-lang.org**

LEARNING RUST IS NOT AN EASY ROAD TO TRAVEL

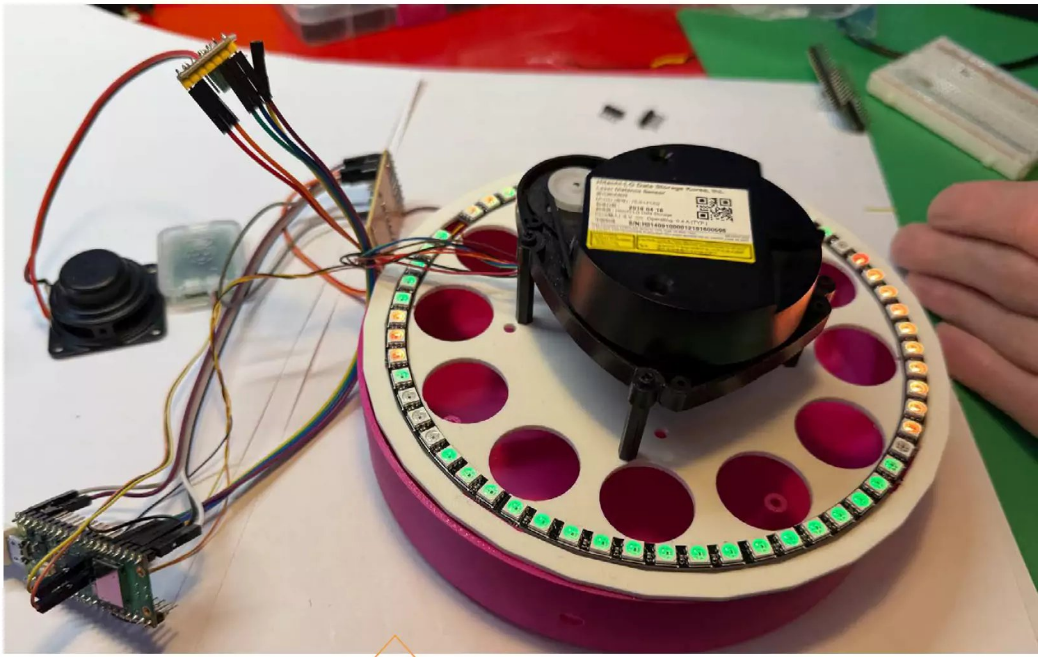


Figure 8: You can find the project at **rpimag.co/rustylidar**

STATIC ON THE LINE

Rust programs are ‘statically linked’, which means that everything needed to run is determined when the program is built. Data must also be statically allocated and managed, so that there is no chance of a Rust program running out of memory or pausing while a garbage collector runs to free up space. Rust also encompasses a build system that allows you to specify and manage the versions of the components in a solution. If you are creating critical systems for widespread deployment, this is very important. It stops code from failing because of changes to libraries that they depend on.

FUN WITH RUST

Figure 8 shows a rotary lidar sensor on top of a ring of NeoPixels. The pixels light up red when objects are detected near to the sensor. The next phase of the project is to add sound output so that we can make a ‘sneak-up’ game with the device in the middle of the room. The Pico processes the high-speed serial stream from the lidar sensor and then drives the NeoPixels. All the software is written in Rust, which is quite up to the task of dealing with the incoming data. The author is enjoying learning the language: it has been designed to instil a professional approach to those using it. Learning Rust is not an easy road to travel, but it is well worth the effort.

A Raspberry Pi for younger children

Kids want to emulate their parents, and parents want their kids to learn useful stuff

Maker

Dr Andrew Lewis

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

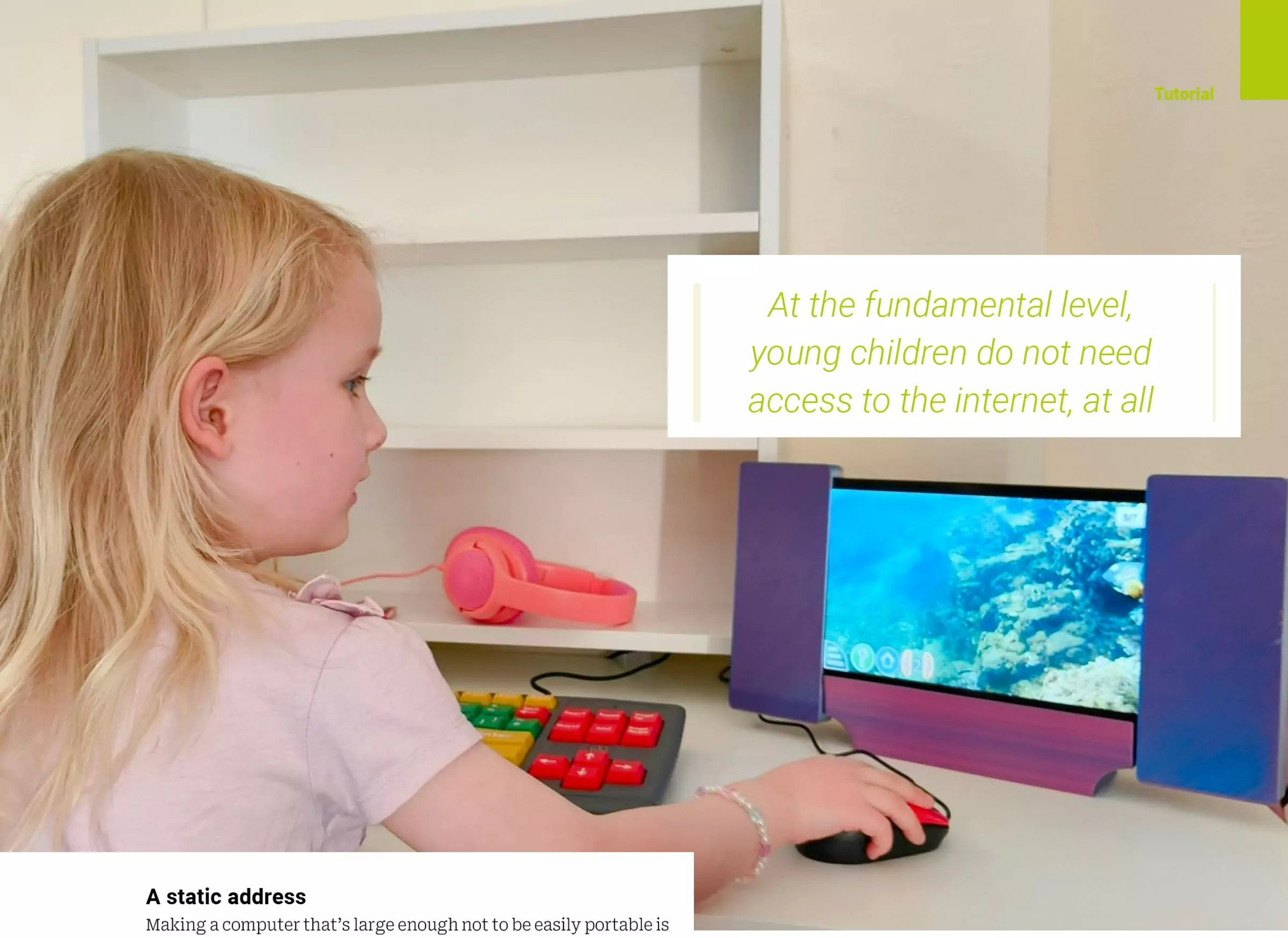
lewiswork.etsy.com

In the modern world, desktop computers are increasingly rare. Laptop computers, mobile phones, and tablet computers with touchscreens are ubiquitous, while the traditional mouse and keyboard are less easily discovered. Learning to type and use a computer is a valuable skill for kids, but you can't just sit a young child in front of a laptop or desktop computer and let them play unsupervised. Aside from the obvious security issues, a desktop computer is not designed with the needs of small children in mind. This article shows some of the issues and potential solutions that you will encounter if you decide to make a real computer for a young human.

In the context of this article, when we talk about 'young children' or 'small children', it means mostly children around the age of three to six. Unlike in a Bethesda game, children in the real world are all different, and what's appropriate for one child might not be appropriate for another. It's up to parents to decide what is or isn't safe or appropriate for their own children.

A small computer like Raspberry Pi 500 is ideal for older children to learn with, but for young children (around four years old), there are some other considerations that make rolling your own machine from a Raspberry Pi a more attractive solution. So what exactly are the issues you should consider when thinking about a computer younger children, and what sort of machine are we looking to build?

Firstly, little hands are very good at pulling cables that they're not supposed to, and delicate plugs and sockets like USB cables are likely to get pulled out. Securing these plugs using cable ties and sticky pads will go a long way towards keeping the machine running. The same goes for SD cards, which small fingers are very adept at removing if they are accessible. The most sensible way to deal with this is to hide everything inside a box of some description. Whether that's a pre-made project case, or a custom 3D-printed enclosure that supports the monitor is up to you.



> At the fundamental level, young children do not need access to the internet, at all

▲ It seems strange, but in a world where we are immersed in technology, there are still some young people who struggle to use a desktop computer simply because they've never been exposed to a physical keyboard and mouse. You can help break that trend by introducing the next generation to skills that they might build a future career on

A static address

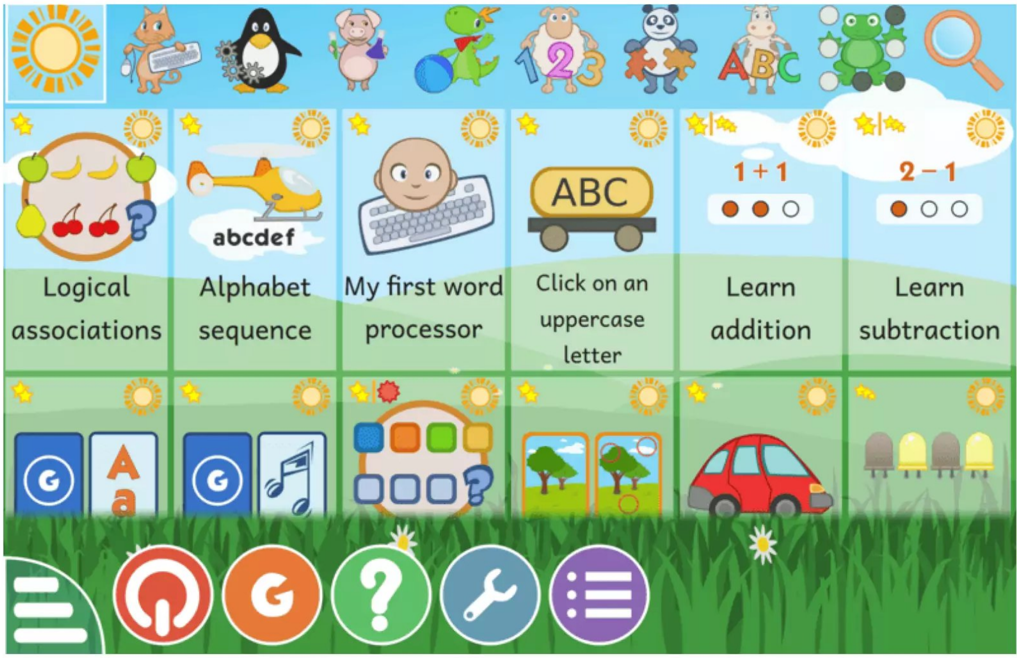
Making a computer that's large enough not to be easily portable is a great way to prevent unwanted access and enforce a schedule. With a small tablet or smartphone, a child can walk away from you into a different room or position the screen so that it's difficult for you to see what they're doing. A desktop computer is exactly that – the computer sits on a desktop. It is a fixed location where your child can sit down and use the computer. If you ever took typing classes, you'll know that posture and position are important when using a keyboard.

One of the biggest advantages to using a Raspberry Pi 5 over a Raspberry Pi 500 for young children is that you are not tied to using the stock keyboard. Young children are more familiar working with lower-case letters, and usually have less fine motor control than an adult. The small keys and upper-case lettering can be confusing to some children. Additionally, it's very common for children to be long-sighted. It makes sense if you think about it, because a child has smaller eyes than those of an adult.

The eye simply isn't developed enough and the lens focusses behind the retina instead of on it. That means that while they might be able to see a monitor or projector screen in a classroom, closer objects like keyboards or books are more difficult to focus on. A dedicated children's keyboard with large font and larger keys takes care of this issue, and can be switched for a regular keyboard as the child grows.

QUICK TIP

It's tempting to think of giving children headphones to use with their computer. Children's headphones must be power limited to less than 75 decibels so that their maximum volume isn't high enough to cause permanent damage to a child's sensitive ears. Check this before you buy headphones for obvious reasons, and be aware that the term 'kid safe' or 'child friendly' doesn't mean that they're power-limited.



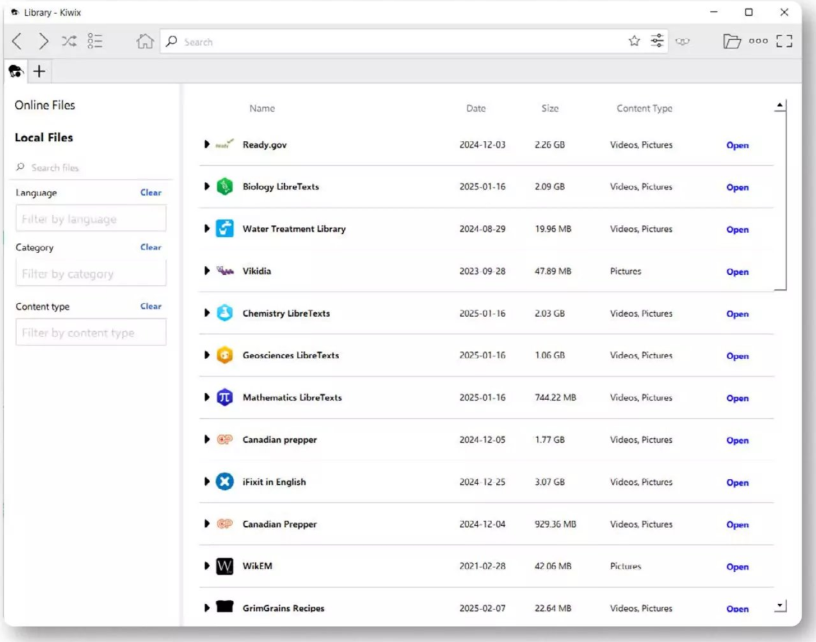
▶ GCompris is an excellent educational package for children. However, the current Raspberry Pi installer has a missing dependency file that breaks some of the games. Make sure that you install qt5-image-formats-plugins separately, or some of the games will be hopelessly broken and will have missing graphics

A standard computer mouse is far too big for a small child to use comfortably, so it doesn't take much consideration to solve that problem by using a smaller mouse instead. However, learning to use a mouse properly takes time, and until they master the skill, children are likely to get frustrated every time they use the computer. For this reason, it helps to have a computer monitor with a touchscreen matrix. If they are struggling to do something on the computer using the mouse, they can just touch the screen and then carry on using the mouse when they are more confident. For this reason, you don't want to give the child a giant monitor unless you want them to get a sore neck and arms. Although young people these days no longer sit in front of a glass vacuum tube with an electron gun firing at their faces, a large monitor can still be overwhelming and uncomfortable for a child to use.

The idea that 'less is more' is carried through from the monitor to the speakers. Ideally, you'll have a monitor with a small set of built-in speakers. If not, a very cheap set of Bluetooth speakers will be fine. When it comes to children, you really don't want them to be louder than necessary.

With all of these considerations in mind, you should be able to assemble a suitable Raspberry Pi for a child to use. The last practical consideration is to hide every plug, socket, and cable as much as possible in a way that is still attractive to the child. 3D-printed covers for the monitor sockets and stands to make a portable monitor look like a desktop monitor go a long way to achieving this. Some children may have never seen a real-life desktop computer, but they've probably seen them on TV or in books.

They'll expect their computer to look like the representations they've seen, and might be a bit confused or disappointed if it doesn't. A great way to engage them with the computer is to let them decorate the outside with stickers and choose their own desktop background. If you're designing a custom 3D-printed case, it's worth allowing plenty of flat surfaces for them to get creative and decorate.



Name	Date	Size	Content Type	
Ready.gov	2024-12-03	2.26 GB	Videos, Pictures	Open
Biology LibreTexts	2025-01-16	2.09 GB	Videos, Pictures	Open
Water Treatment Library	2024-08-29	19.96 MB	Videos, Pictures	Open
Vikidia	2023-09-28	47.89 MB	Pictures	Open
Chemistry LibreTexts	2025-01-16	2.03 GB	Videos, Pictures	Open
Geosciences LibreTexts	2025-01-16	1.06 GB	Videos, Pictures	Open
Mathematics LibreTexts	2025-01-16	744.22 MB	Videos, Pictures	Open
Canadian prepper	2024-12-05	1.77 GB	Videos, Pictures	Open
iFixit in English	2024-12-25	3.07 GB	Videos, Pictures	Open
Canadian Prepper	2024-12-04	929.36 MB	Videos, Pictures	Open
WikEM	2021-02-28	42.06 MB	Pictures	Open
GrimGrains Recipes	2025-02-07	22.64 MB	Videos, Pictures	Open

◀ The Kiwix library has a huge amount of offline information that is of interest to adults and children alike, accessible from the Kiwix app or a browser extension. You can also create a Raspberry Pi-powered Kiwix Hotspot and use it to serve 20 devices at a time

Control the internet

You don't want to unleash your child onto a completely open computer, so let's deal with some of the big software and networking issues. At the fundamental level, young children do not need access to the internet, at all. Preventing access to the internet can be done in several ways; the easiest is to provide no access to networking. No Wi-Fi passwords, no cable connections. This is a 100% certain way to prevent children from accessing something that they shouldn't. A 2m air-gap leaks no data. There are of course other methods of limiting access to the internet, such as changing to a child-safe DNS like the OpenDNS Family Shield, or using a Pi-hole or Squid proxy in your home network. The sad reality is that filtering technology is not a perfect replacement for parental supervision, and you'll never be able to shield a child completely from the shadier side of the web without a Faraday cage. However, if you are set on providing some internet access to young children, then a combination of these technologies and close supervision, education, and logging are about as effective as you will ever get. If your four-year-old is secretly an elite member of a secret hacking community, your results may vary.

The safer 'air-gap' approach doesn't necessarily mean starving your children of knowledge or access to everything the web can offer. Some people forget that the internet is relatively new, and for all of the streaming content it provides, there are generally offline methods for presenting the same information. For example, the Kiwix project provides a way to view websites offline. There's a whole list of pre-packaged sites to install, including Wikipedia, Vikidia (a children's encyclopaedia), children's story books, and even curated collections aimed specifically at children. If you can't find what you need in the Kiwix library, there are tools to package your own sites. There's also nothing to prevent you from augmenting the static information on their computer with DVD, CD, MP3, or streaming content provided from your own tablet or mobile phone. In fact, the distinction that 'this is your computer for working, this is a tablet/TV for watching videos' can be a useful one for younger children and stop them from becoming fixated on a specific device.

The ever popular Tuxpaint is an excellent application for young people to experiment with for multiple reasons. It's a fun drawing app that helps teach fine motor and mouse control, but it also has a huge selection of buttons and menus to navigate through. Once your child starts clicking through the dozens of options and exploring the painting application, any fear of exploring other apps on the computer will diminish. In fact, Raspberry Pi itself is part of this idea. Let your child loose on the operating system. The worst they will do is break it, and with Raspberry Pi and a backup SD card, it's only a few minutes work to get things working again. You could create a new user and lock down the desktop to only the items you want your child to access, but because a Raspberry Pi can be reinstalled very quickly, it might be less effort to keep a backup SD card ready and let them break things.

▼ A colourful keyboard designed for a child to use can be educational even when it isn't plugged in. The colourful keys on this keyboard help to differentiate between vowels and consonants, numbers, and control keys. The larger keys and text are easier to read and easier to type without making a mistake

Create a Sense HAT interactive pixel pet

Create a pint-sized pocket monster living in the digital world of a Raspberry Pi Sense HAT

Maker
Lucy Hattersley
Lucy is editor of Raspberry Pi Official Magazine and has a real cat that doesn't like being prodded one bit.

rpimag.co

Using sensors and output devices is a great way to make your computer programs more interactive. The Raspberry Pi Sense HAT contains a whole set of sensors that can detect movement, and in this activity we'll use them to take a digital pet for a walk.

You'll need to design your pet avatar before you program any actions. There are examples of some famous characters you can make on an excellent sprite sheet created by Johan Vinet, which can be found at **rpimag.co/spritesheet**. You can use this to draw your space pet avatar, as seen in **Figure 1**.

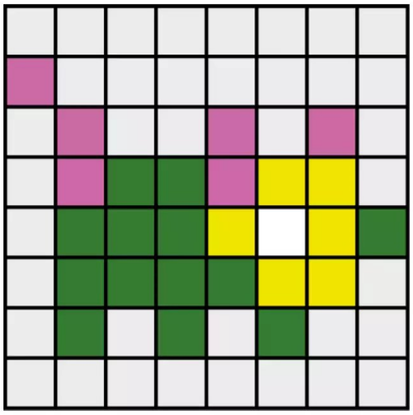
Draw your picture out on squared paper using coloured pencils. You'll need two pet designs, the second preferably being very similar to the first so that you can animate your pet. In **Figure 2**, you can see that our two images are almost identical to each other – only the feet are in a different place.

Later, when you code your animation, this will create the illusion that the pet is walking.

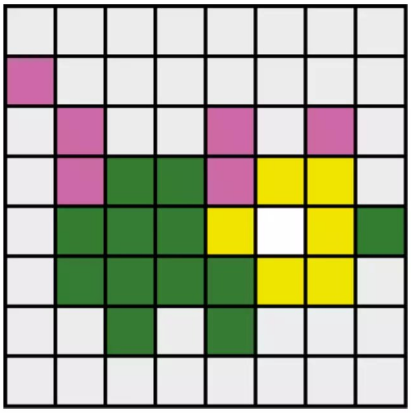
Labelling each pixel

Think of a letter from the alphabet to represent each colour in your pixel pet image: for example, 'w' for white or 'r' for red. If using squared paper for your design, you can write the letters on top, as shown in **Figure 3**. Note that 'e' stands for empty.

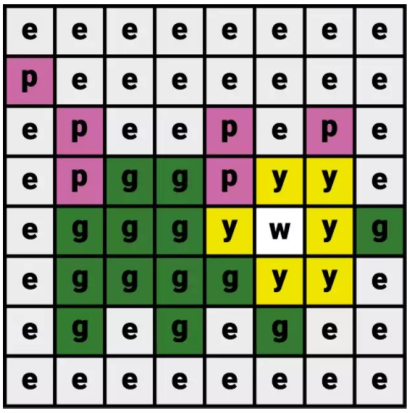
> *Nothing should happen until you shake your Raspberry Pi. Then the pet animation will run*



▲ **Figure 1:** The first animation frame



▲ **Figure 2:** The second animation frame



e	e	e	e	e	e	e	e
p	e	e	e	e	e	e	e
e	p	e	e	p	e	p	e
e	p	g	g	p	y	y	e
e	g	g	g	y	w	y	g
e	g	g	g	g	y	y	e
e	g	e	g	e	g	e	e
e	e	e	e	e	e	e	e

▲ **Figure 3:** The pixel pet with colour labels

pixel_pet.py

> Language: **Python**

DOWNLOAD THE FULL CODE:

rpimag.co/pixelpetpy

```python
from sense_hat import SenseHat
import time

sense = SenseHat()

p = (204, 0, 204) # Pink
g = (0, 102, 102) # Dark Green
w = (200, 200, 200) # White
y = (204, 204, 0) # Yellow
e = (0, 0, 0) # Empty

pet1 = [
 e, e, e, e, e, e, e, e,
 p, e, e, e, e, e, e, e,
 e, p, e, e, p, e, p, e,
 e, p, g, g, p, w, w, e,
 e, g, g, g, w, y, w, y,
 e, g, g, g, g, w, w, e,
 e, g, e, g, e, g, e, e,
 e, e, e, e, e, e, e, e
 ]
pet2 = [
 e, e, e, e, e, e, e, e,
 p, e, e, e, e, e, e, e,
 e, p, e, e, p, e, p, e,
 e, p, g, g, p, w, w, e,
 e, g, g, g, w, y, w, y,
 e, g, g, g, g, w, w, e,
 e, e, g, e, g, e, e, e,
 e, e, e, e, e, e, e, e
 ]

def walking():
    for i in range(10):
        sense.set_pixels(pet1)
        time.sleep(0.5)
        sense.set_pixels(pet2)
        time.sleep(0.5)

sense.clear(0, 0, 0)
while True:
    x, y, z = sense.get_accelerometer_raw().values()
    if x >= 2 or y >= 2 or z >= 2:
        break
walking()
```

You'll notice that we have eight rows and eight columns of letters, each separated by a comma, to make up the LED matrix on the Sense HAT. Repeat this step for your second pet design so that you end up with two grids of letters.

Code your pet

Now that you have your designs represented as letters in a grid, you can start to code them in Python. 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, empty file as **space-pet.py**.

First, you need to import all the modules and libraries required for the project into your code and create a `SenseHat` object:

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

Note that capital letters, full stops, and commas are very important in Python. Your code might not work if you don't include these. Next, create a variable for each colour label in your pet design, like this:

```
p = (204, 0, 204) # Pink
g = (0, 102, 102) # Dark Green
w = (200, 200, 200) # White
y = (204, 204, 0) # Yellow
e = (0, 0, 0) # Empty
```

The numbers inside the brackets are the RGB (red, green, and blue) values. Mixtures of these colours make different shades. The higher the number, the more of that colour it will contain. For example, (255, 0, 0) would make a solid red, whereas (0, 255, 0) would create a vivid green. You can change these numbers in your code to get the colours that you want.

Next, use a list to store your pixel pet design, like this:

```
pet1 = [
 e, e, e, e, e, e, e, e,
 p, e, e, e, e, e, e, e,
 e, p, e, e, p, e, p, e,
 e, p, g, g, p, w, w, e,
 e, g, g, g, w, y, w, y,
 e, g, g, g, g, w, w, e,
 e, g, e, g, e, g, e, e,
 e, e, e, e, e, e, e, e
 ]
```

Here you have created a variable called `pet1` and stored a list of labels for each colour by using `[` at the start and `]` at the end. Repeat for the second pixel pet design, using a different variable name like `pet2`. Your code should start to look something like the drawing you created earlier.

If you ran your code now, nothing would happen, because so far you have only told the program to store information. To make something happen, you will need to write a command to call on that data and display your colours in the correct order on the Sense HAT's LED matrix. Type this command underneath your lists:

```
sense.set_pixels(pet1)
```

Save your code by clicking Save, then run your program to test it works by clicking Run or pressing **F5**. Note what happens. Why did only one of your pet designs appear? It's because you only called `pet1` in your command.

Add a delay using the `time.sleep` function, then call the second picture using the same command as before, like this:

```
time.sleep(0.5)
sense.set_pixels(pet2)
```

Save and run your code to see your pet 'move'.

Animate your pet

So far, your pixel pet only changes once. To animate it fully, you will need to switch repeatedly between the two pictures with a time delay. You could write the commands out over and over again, but it makes more sense to put them in a loop.

Move to the end of your program and locate the `sense.set_pixels(pet1)` line and the two lines that follow. Change them to look like this:

```
for i in range(10):
    sense.set_pixels(pet1)
    time.sleep(0.5)
    sense.set_pixels(pet2)
    time.sleep(0.5)
```

Don't forget to add the extra `time.sleep(0.5)` on the last line, and remember to indent the lines after `for i in range(10):` to place them inside the `for` loop. This `for` loop with the `range` function will repeat the indented code ten times and then stop.

Save and run your code to watch the animation. You will notice that after the animation has finished, you are left with the same image still displayed on the LED matrix. There is a

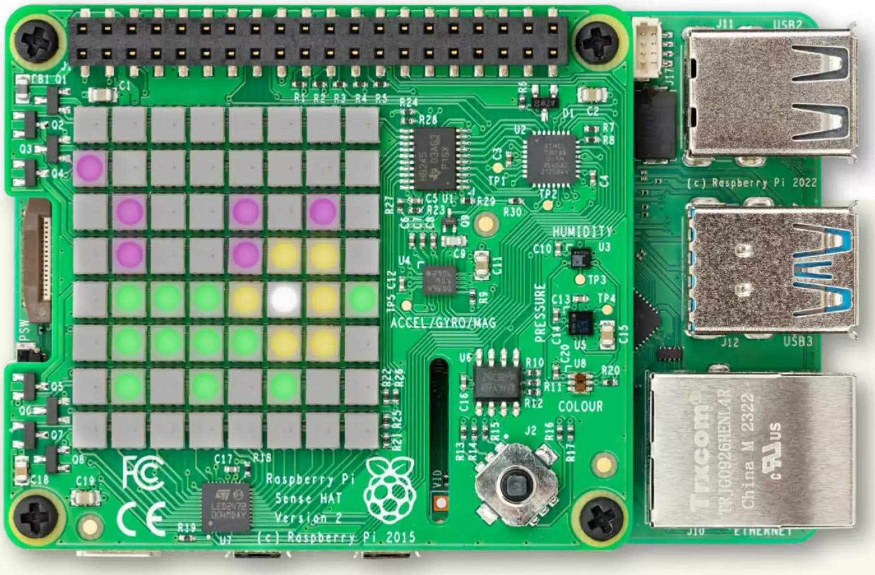


Figure 4: The final code running on a Sense HAT

useful function that you can use to clear the LEDs; add this line above your new loop to clear the LEDs when you first run your program:

```
sense.clear(0, 0, 0)
```

Create a walking function

A function is a piece of code that you can use over and over. As the goal is to trigger the walking animation later on, it makes sense for us to put the animation code into a function that can be called when an action has been sensed by the hardware.

To put your code into a function, you simply need to add the line `def walking():` above your `for` loop and indent the lines beneath, like this:

```
def walking():
    for i in range(10):
        sense.set_pixels(pet1)
        time.sleep(0.5)
        sense.set_pixels(pet2)
        time.sleep(0.5)
```

The use of `def` here means that you are defining a function you have named `walking`. Now you need to call the function. So, at the bottom of your code, type:

```
walking()
```

Shake for more

It's time to use the Sense HAT's movement sensors, particularly its accelerometer, to trigger the walking function and make the project more interactive. Between `sense.clear(0, 0, 0)` and the function call to `walking()`, type:

```
 sense.set_pixels(pet1)
 while True:
     x, y, z = sense.get_accelerometer_raw().
values()
     if x >= 2 or y >= 2 or z >= 2:
         break
```

The first line shows a (stationary) space pet, and the second creates an infinite loop. The next line obtains current movement readings from the Sense HAT on its x, y, and z axes. As your Raspberry Pi is presumably sitting still on a desk, those readings will have a very low value. The loop continues forever until the Sense HAT moves enough for the statement `x >= 2 or y >= 2 or z >= 2` to evaluate to `True`. When that happens, the `break` statement stops the loop. You can help the Sense HAT take an accelerometer reading above 2 by shaking it! Your final code is in the **pixel_pet.py** listing (previous page). Save your code and then run it. Nothing should happen until you shake your Raspberry Pi. Then the pet animation will run, and your pet will start to walk (**Figure 4**).

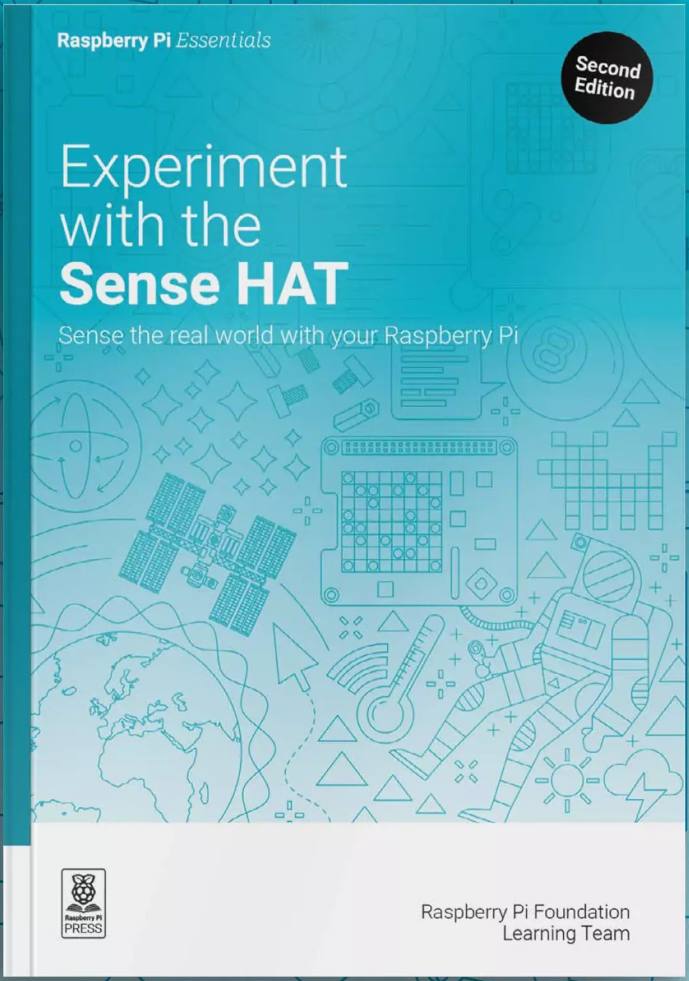
Your pet walks for a few seconds, as the `for` loop executes ten times, before the program stops. If you find your pet doesn't move at all, you can try changing the `if` statement to require less movement:

```
     if x >= 1 or y >= 1 or z >= 1:
```

You could even add `x`, `y`, and `z` together to only take action when the sum of the raw values is over a certain amount:

```
     if x + y + z >= 2:
```

Can you think of a way to modify the program so that the pet walks for a bit every time you shake it, rather than just once?



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

Part 01

RISC OS for Raspberry Pi

In the first of a series of guides to using RISC OS on a Raspberry Pi, we take a look at how to install it and use it to go online

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

ijosborne.bsky.social

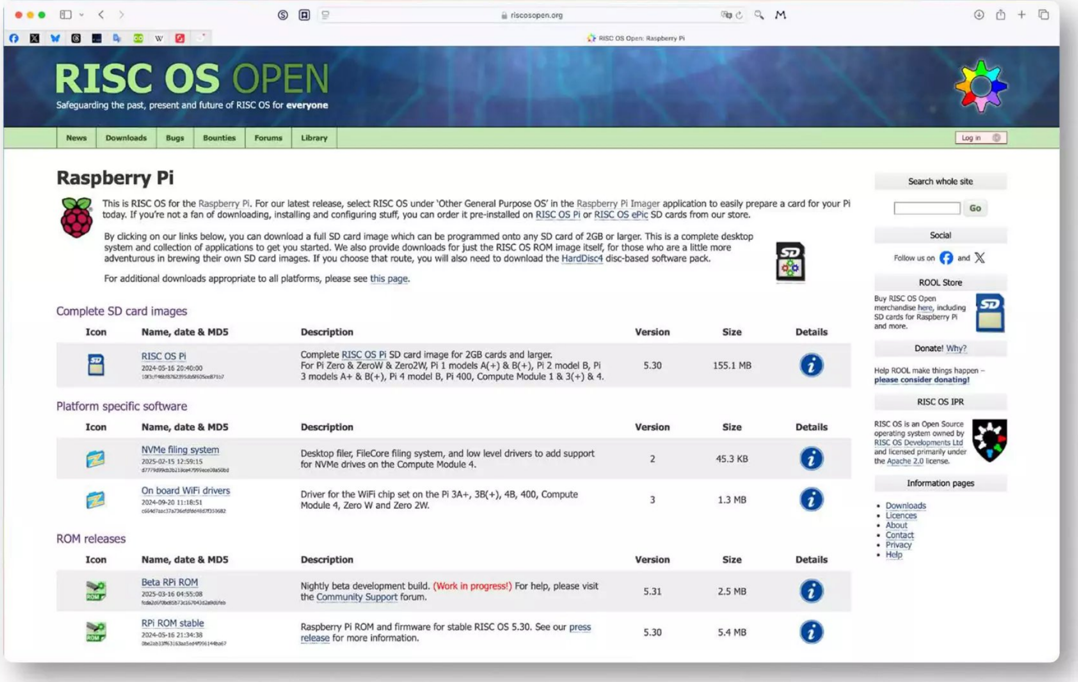
Raspberry Pi OS isn't the only operating system you can use on your Raspberry Pi. Retro fans can also use RISC OS, which is older than Raspberry Pi itself. First released on 25 September 1987, it was developed by Acorn Computers to run on its Archimedes machines, which used ARM (Advanced RISC Machine) processors. The operating system survived the demise of Acorn in 1999, and exists today as an open-source project.

Reduced instruction set, increased speed

RISC OS is named after the Reduced Instruction Set Computer (RISC) architecture on which it runs. RISC processors use a simplified instruction set, allowing them to focus on speed and efficiency by using a smaller, more optimised set of instructions. This means you can get a very good performance even from relatively low-powered processors. As the Raspberry Pi uses ARM CPUs, it's ideal for RISC OS.

To run RISC OS on your Raspberry Pi, you need to download and install it on a microSD card with a capacity of at least 2GB. You probably have one at the back of a drawer somewhere, gathering dust because we rarely use cards that small any more.

Use the Raspberry Pi Imager application, available for Macs, Windows and Linux from **rpimag.co/software**. It is installed by default in Raspberry Pi OS (64-bit).



riscosopen.org

RISC OS Open: Raspberry Pi

RISC OS OPEN

Safeguarding the past, present and future of RISC OS for **everyone**

Raspberry Pi

This is RISC OS for the Raspberry Pi. For our latest release, select RISC OS under 'Other General Purpose OS' in the Raspberry Pi Imager application to easily prepare a card for your Pi today. If you're not a fan of downloading, installing and configuring stuff, you can order it pre-installed on RISC OS Pi or RISC OS ePic SD cards from our store.

By clicking on our links below, you can download a full SD card image which can be programmed onto any SD card of 2GB or larger. This is a complete desktop system and collection of applications to get you started. We also provide downloads for just the RISC OS ROM image itself, for those who are a little more adventurous in brewing their own SD card images. If you choose that route, you will also need to download the HardDisc4 disc-based software pack.

For additional downloads appropriate to all platforms, please see this page.

Complete SD card images

Icon	Name, date & MD5	Description	Version	Size	Details
SD	RISC OS Pi 2024-05-16 20:40:00 10f3cff46bf8762395db5f605ce871b7	Complete RISC OS Pi SD card image for 2GB cards and larger. For Pi Zero & ZeroW & Zero2W, Pi 1 models A(+) & B(+), Pi 2 model B, Pi 3 models A+ & B(+), Pi 4 model B, Pi 400, Compute Module 1 & 3(+) & 4.	5.30	155.1 MB	i

Platform specific software

Icon	Name, date & MD5	Description	Version	Size	Details
Z	NVMe filing system 2025-02-15 12:59:15 d7779d99cb3b218ce47999ece08a50bd	Desktop filer, FileCore filing system, and low level drivers to add support for NVMe drives on the Compute Module 4.	2	45.3 KB	i
Z	On board WiFi drivers 2024-09-20 11:18:51 c664d7aac37a736efdfdd48d7f350682	Driver for the WiFi chip set on the Pi 3A+, 3B(+), 4B, 400, Compute Module 4, Zero W and Zero 2W.	3	1.3 MB	i

ROM releases

Icon	Name, date & MD5	Description	Version	Size	Details
ROM	Beta RPi ROM 2025-03-16 04:55:08 fcda2d6f0bd85b73c167043d2a9d6feb	Nightly beta development build. (Work in progress!) For help, please visit the Community Support forum.	5.31	2.5 MB	i
ROM	RPi ROM stable 2024-05-16 21:34:38 0be2ab33ff63163aa5ed4f956144ba67	Raspberry Pi ROM and firmware for stable RISC OS 5.30. See our press release for more information.	5.30	5.4 MB	i

Search whole site

Go

Social

Follow us on and X

ROOL Store

Buy RISC OS Open merchandise here, including SD cards for Raspberry Pi and more.

Donate! Why?

Help ROOL make things happen – **please consider donating!**

RISC OS IPR

RISC OS is an Open Source operating system owned by RISC OS Developments Ltd and licensed primarily under the Apache 2.0 license.

Information pages

- Downloads
- Licences
- About
- Contact
- Privacy
- Help

> *RISC OS is named after the Reduced Instruction Set Computer (RISC) architecture on which it runs*

▲ You can also download the RISC OS operating system for Raspberry Pi from riscosopen.org

On the cards…

Preparing the microSD card is simplicity itself. Skip the next two paragraphs if you don't need instructions, but if you're a beginner, this is how you do it. Launch Raspberry Pi Imager on your computer of choice. Click the first button, under Raspberry Pi Device, and select 'No filtering'. Click the second button, under Operating System, and choose Other general-purpose OS > RISC OS Pi > RISC OS at the bottom of the list. The latest version should be 5.30 or higher.

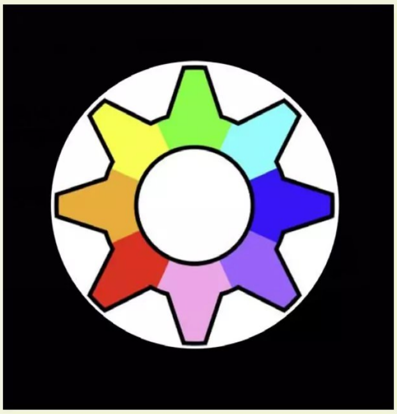
If you haven't done so already, plug the microSD card you wish to use for RISC OS into your computer. Now click the button under Storage, and choose the microSD card from the list of connected storage drives offered.

Click Next. When you get the warning about everything currently on the card being erased, click 'Yes' to continue, and the OS will be written to the card. It may take a while, so be patient.

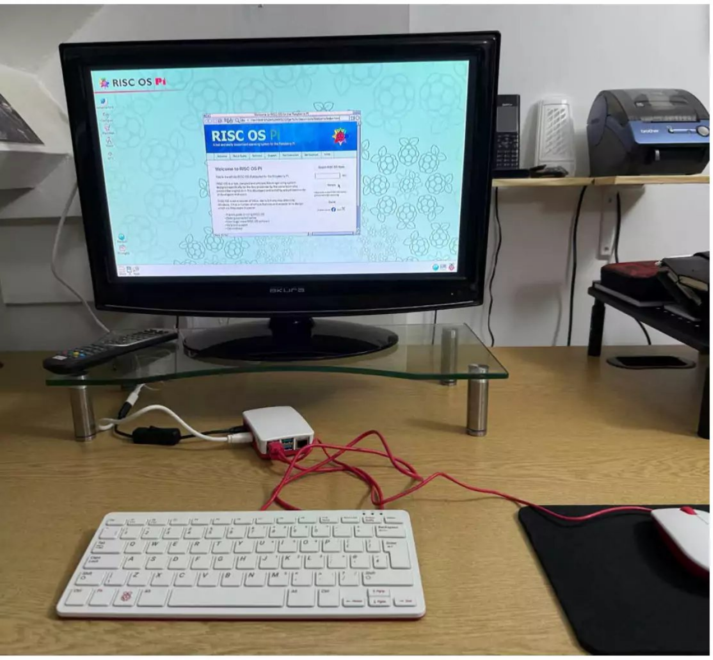
Now you have your RISC OS microSD card ready, insert the card into your Raspberry Pi and power it up. It should boot straight into RISC OS. At the time of writing, the current release of RISC OS Pi doesn't indicate support for the Raspberry Pi 5 (see the 'No RISC OS for Raspberry Pi 5… yet!' box, overleaf).

Why RISC OS?

Why would you want to use RISC OS on your Raspberry Pi? According to RISC OS Open: "RISC OS is not a flavour of Linux. It's not a type of Unix. It's certainly got nothing to do with Windows! RISC OS is its own thing – a very specialised ARM-based operating system. So if you've not used it before, you will find it doesn't behave quite the same way as anything else."



So there you have it. Check out RISC OS at **riscosopen.org**.



▲ RISC OS, up and running on a Raspberry Pi 4

Going online

Now we have RISC OS up and running, it's time to get it online. Here is our first challenge: RISC OS was designed for a mouse with three buttons. Fortunately, Raspberry Pi's official mouse has a clickable scroll wheel that acts as the middle button.

First of all, middle-click on the Task Manager toolbar icon – that is, the Raspberry Pi icon in the bottom-right corner of the desktop. In the menu that appears, click on the Configure… option. Now click Network, then Internet. You're now in the Internet Configuration window.

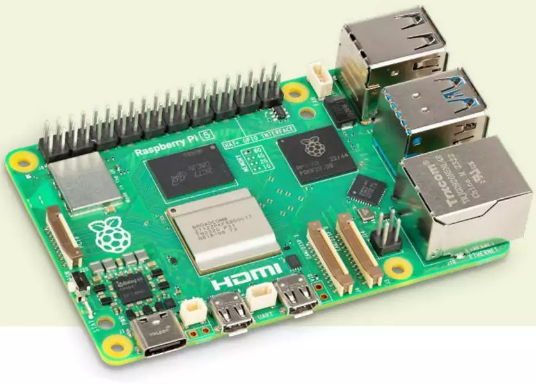
For Ethernet connections, make sure Enable TCP/IP Protocol Suite is ticked and then click Interfaces. Add a tick next to Broadcom GENET or Ethernet over USB, depending on which is present, and then click on its Configure button. The IP Address setting is usually 'via DHCP', unless your setup is different. Confirm your choice with the Set button, then close the windows, click Save when invited to do so, and select Restart Now to restart your Raspberry Pi and make your changes active.

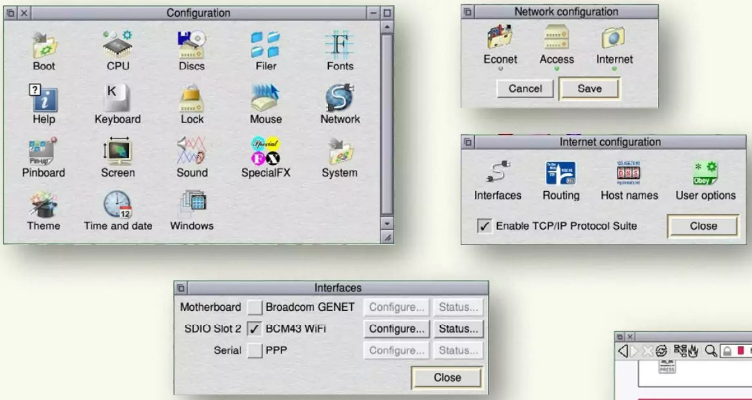
If you're planning to use Wi-Fi for your Internet connection, again, make sure Enable TCP/IP Protocol Suite is ticked in the Internet Configuration window, and then click the Interfaces icon. Check the BCM43 WiFi box (if this option is greyed out, Wi-Fi isn't available on your Raspberry Pi and you will need to add an Ethernet cable).

Untick all the other interface options, so only Wi-Fi is ticked. Click the Configure… button, and select 'via DHCP' (unless your Internet Router uses a different configuration, however DHCP is most common). The IP Address and Netmask fields are filled in automatically. Click Set in the current window, Close in the Interfaces window, and Close in the Internet Configuration window and Save in the Network configuration window. Finally click Reset Now to restart your machine with these networking options configured.

No RISC OS for Raspberry Pi 5… yet!

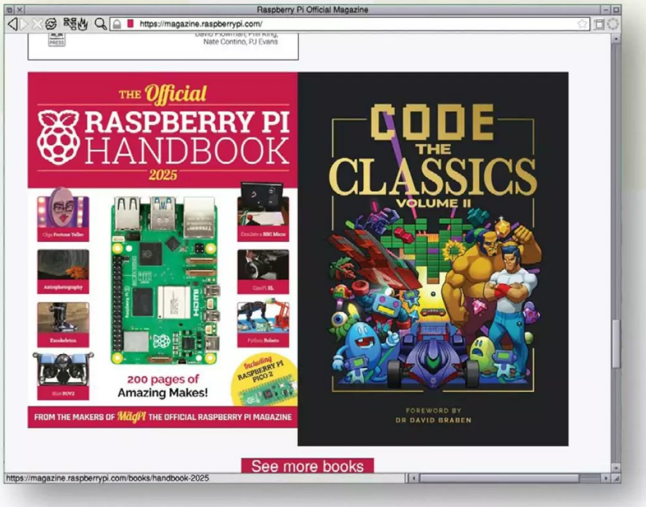
The current build of RISC OS is not yet available for the Raspberry Pi 5 range of computers, such as Raspberry Pi 5 (shown here), Raspberry Pi 500, or the Compute Module 5. This is because these machines are primarily 64-bit and lack the 32-bit operating mode that RISC OS needs. There are plans to bring RISC OS to the Raspberry Pi 5 range, but it won't come any time soon. See **rpimag.co/riscaarcheol** for more information.

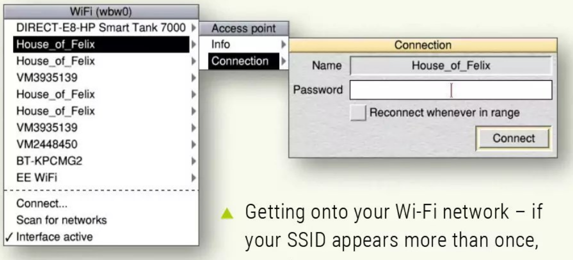




Configuring RISC OS for a Wi-Fi internet connection

Up and running with RISC OS and its native browser, NetSurf


Raspberry Pi Official Magazine
https://magazine.raspberrypi.com/
THE Official RASPBERRY PI HANDBOOK 2025
200 pages of Amazing Makes!
FROM THE MAKERS OF MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
CODE THE CLASSICS VOLUME II
FOREWORD BY DR DAVID BRABEN
See more books
https://magazine.raspberrypi.com/books/handbook-2025



Getting onto your Wi-Fi network – if your SSID appears more than once, use the strongest one

Wireless wonders

In the toolbar at the foot of the screen, middle-click the new 'WiFi' icon. From the list that appears, choose your network. If the network name (SSID) crops up more than once, it's probably because you have more than one access point set up, maybe through a repeater. If this is the case, drag your pointer to the right-pointing triangle next to your network name, then do the same for the Info option in the menu that appears, and make a note of the signal strength. Do the same for the other instances of your network name, and then connect to the one with the greatest strength.

To connect, drag the pointer next to the right-pointing arrow next to your network name, then in the next menu do the same for Connection. In the Connection window that appears, type your network password. Make sure 'Reconnect whenever in range' is checked so you don't have to do this again next time you use your RISC OS Raspberry Pi, and click the Connect button. Take another look at the WiFi icon in the toolbar. The black mast will have turned green (or at least partially green), indicating that you have a wireless Internet connection.

And that's it. You should now be online, and able to surf the net using RISC OS's bundled browser, NetSurf.

Part 03

Build a home recording studio with Raspberry Pi 500

How to install and set up the software you'll need for high-quality audio production

Maker

K.G. Orphanides

K.G. holds that tuning to C# and using arpeggiators are both cheating at music, and does them with great enthusiasm.

twoot.space/@owlbear

Raspberry Pi OS, like most other Linux distributions, comes with a wide range of audio production software of somewhat variable quality. We'll guide you through your options for audio. We're going to use JACK2 for low-latency audio alongside pipewire for system audio, with a pipewire-pulse compatibility layer to route between JACK and standard desktop applications.

First, open a terminal and install the audio software:

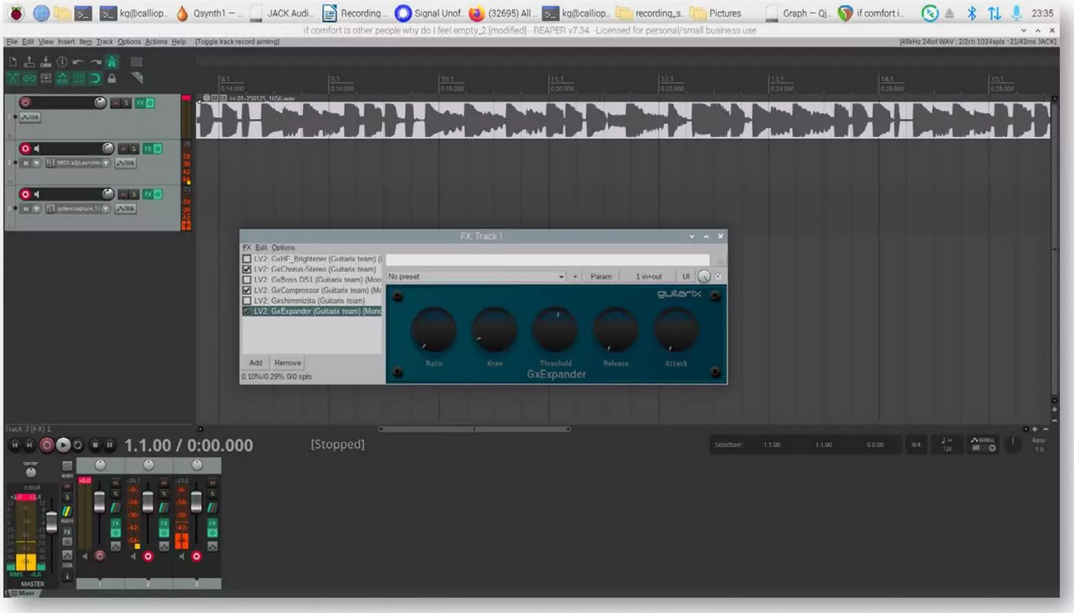
```
 $ sudo apt install jackd2 qjackctl
pipewire-pulse qsynth guitarix
```

With that done, enter:

```
 $ qjackctl
```

Click 'setup', and in the Settings tab, make sure the driver is set to ALSA and the MIDI driver is set to raw. Click Advanced and make sure your audio interface is set as the output device and, if you have a standalone USB MIDI keyboard, that it's set as the input device. Finally, click the Options tab and put the following into the 'Execute script after startup' section:

```
 pacmd set-default-sink jack_out && pacmd
set-default-source jack_in
```


FX: Track 1
FX Edit Options
LV2: GxHF_Brightener (Guitarix team)
LV2: GxChorus-Stereo (Guitarix team)
LV2: GxBoss DS1 (Guitarix team)
LV2: GxCompressor (Guitarix team)
LV2: Gxshimmizita (Guitarix team)
LV2: GxExpander (Guitarix team)
Add
Remove
No preset
Param
1 in+out
UI
guitarix
Ratio
Knee
Threshold
Release
Attack
GxExpander
1.1.00 / 0:00.000
[Stopped]
MASTER

We're fans of Reaper's excellent FX interface for PCM audio and capable MIDI input options

Setting up your audio interface

Most USB audio interfaces are class-compliant, so will work on Linux, but you might encounter issues with devices that rely on proprietary software to configure them or update their firmware.

We chose a Focusrite 2i2 for our studio setup because it has full Linux support thanks to cooperation between Focusrite and the Linux kernel development community, and the hard work of driver developer Geoffrey Bennett in particular.

Ensure that you have a compatible version of the kernel – Focusrite audio interface support is built into Linux kernel 6.8 and above. At the time of writing, the default Raspberry Pi kernel was 6.6.74+rpt-rpi-2712; this doesn't include the module for the 2i2, but fortunately the testing kernel (6.12) does.

You first step should be to check your kernel version. By the time this article is in your hands, the default kernel may have been updated to 6.12. Let's find out. Open a terminal and type:

```
$ uname -a
```

You'll have to find digital effects built for aarch64 architecture or compile them yourself

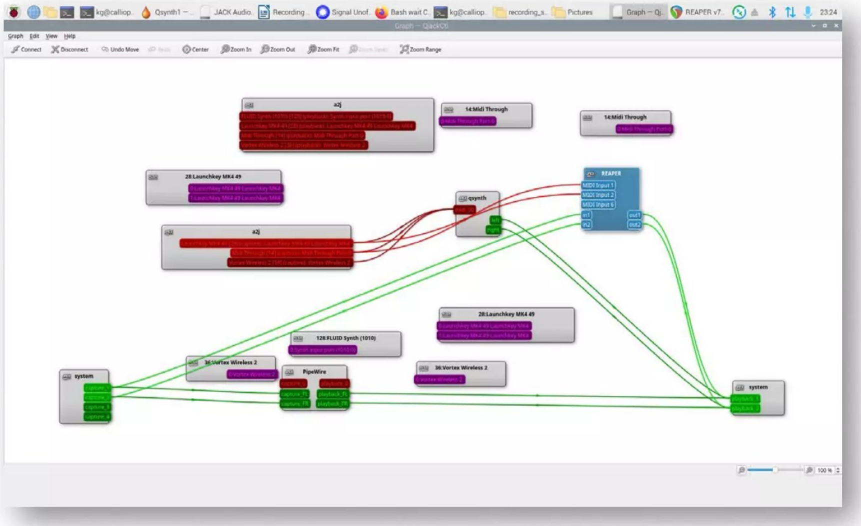
audio_start.sh

> Language: **Bash**

```bash
#!/bin/bash

#Run all processes required for JACK driver, assign MIDI connections, control interface and SF2 synth patch loader

jackd -d alsa&
a2jmidid -e&
qjackctl&
qsynth&
;
```



▲ QjackCtl's connection graph looks messy but makes connecting MIDI inputs particularly easy

If it reports a kernel version below 6.8, update to the experimental new kernel:

```
$ sudo rpi-update
$ reboot
```

Install the Focusrite control interface

```
$ git clone https://github.com/geoffreybennett/
alsa-scarlett-gui.git
$ cd /home/kg/Software/alsa-scarlett-gui-0.4.0
$ cd alsa-scarlett-gui-0.4.0/
$ sudo apt -y install git make gcc libgtk-4-dev
libasound2-dev libssl-dev
$ cd src
$ make -j4
```

Test to make sure that it works:

```
$ ./alsa-scarlett-gui
```

Then install it system-wide:

```
$ sudo make install
```

The Focusrite Scarlett GUI should be automatically added to the Other section of the main menu. You can use it to update your Scarlett's firmware, connect and activate various inputs, contain gain, and more.

Microphone selection

We use a Blue Yeti Pro condenser mic connected via XLR, rather than via its integrated USB audio interface. While you can use multiple audio interfaces, it's easiest to manage latency and audio routing with a single device. If you use a dynamic microphone, like our Shure SM-58, you'll want a pre-amp, such as Triton Audio's convenient in-line FetHead (**rpimag.co/fethead**). This connects to the cable between your mic and audio interface and give you a boost of 27dB, which means that you won't have to turn the gain up so high on your USB audio device. Cheaper in-line pre-amps with a little less boost are also available, as are many more expensive ones. This is a useful because many USB audio interfaces hit their signal-to-noise threshold at a little about 50% gain, which can lead to a staticky hiss on your recordings.

Your MIDI keyboard

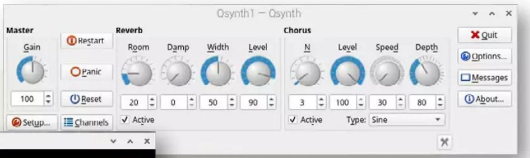
We use various MIDI instruments, including a Novation LaunchKey 49 keyboard and an Alesis Vortex Wireless 2 keytar. Connect USB keyboards to Raspberry Pi directly or via a powered USB hub – unpowered hubs don't provide sufficient power.

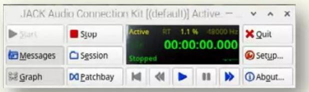
Drivers can be an issue under Linux, but recent Novation devices can be managed and updated in a Chromium-based browser via a web app at **rpimag.co/novation**. This is still a relatively unusual feature, but other brands with web app firmware updaters include PirateMIDI, Morningstar, and PandaMIDI. Our Alesis doesn't require any firmware management, nor do older MIDI keyboards connected via USB MIDI adapters.

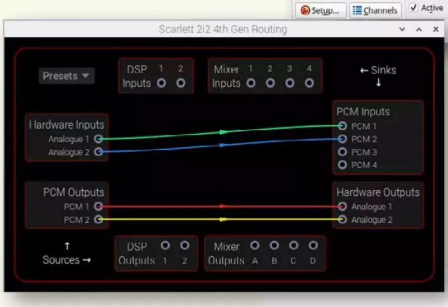
Once your keyboard is set up, the easiest way to test it is via Qsynth, a GUI frontend for Fluidsynth. This ships with an accurate, high-quality General MIDI voice set, but you can use it to load any SoundFont (SF2) file, such as the opl3-soundfont you'll find in the repositories.

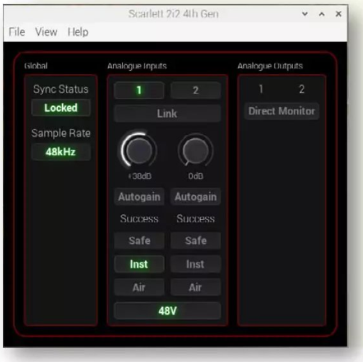
Instant FX rack

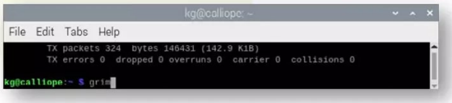
Guitarix lets you stack digital guitar pedals and other effects to give you the equivalent of hundreds of pounds worth of stomp-boxes and tools like a guitar tuner. These effects can also be used in most DAWs, with some being particularly well suited for integration with one or the other, such as a range of Ardour-specific LV2 effects. You can use these, as well as LADSPA effects, with Guitarix, while the DAWs also support VST3 and CLAP plug-ins, among others.











```
        TX packets 324  bytes 146431 (142.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

kg@calliope:~ $ grim
```

There's an incredibly polished Linux control GUI for Focusrite Scarlett hardware. Note the virtual cables connecting hardware inputs to PCM audio sinks

When downloading digital effects, you'll have to find files built for aarch64 architecture or compile them yourself. Fortunately, there are plenty in the repositories. Some useful effects will be installed along with Guitarix to get you started.

Setting up a DAW

A digital audio workstation (DAW) is a complete environment for recording and producing music using either MIDI or live analogue instruments connected via your audio interface. Reaper, Ardour, FruityLoops-like LLMS, and lightweight favourite Qtractor are capable tools, as is PCM-audio focused Audacity and its forks.

We're pushing the limits of what Raspberry Pi 500 can do when it comes to some of these applications. Most PCM recorders, with the notable exception of Qtractor, struggled to keep their visual waveform display in time with the audio recording, although the resulting audio capture was perfect. Some processor-intensive synth voices and effects also affected performance of the DAW.

Most of these tools can be installed directly from the command line:

```
$ sudo apt install lmms audacity ardour qtractor
```

Reaper is commercial software, but is very reasonably priced at $60 (£46) and is highly customisable, with a clear interface, loads of features, and a generous free trial.

To install Reaper, go to **rpimag.co/reaperfm** and download the latest Linux aarch64 version of the software and extract it. Open a terminal and `cd` to the directory where you extracted it:

```
$ cd reaper_linux_aarch64/
$ ./install-reaper.sh
```

Type **I** to install it, then **1** to put it in **/opt**, **Y** to add a desktop integration, **Y** to symlink paths, and **Y** to proceed. Provide your sudo password when prompted. Reaper will be added to the Sound & Video section of the Raspberry Pi OS main menu.

Making connections

Once you've started JACK's control interface, the Focusrite Scarlett GUI, and your music software of choice, you'll quite probably find that you're not actually hearing sound where you expect to. Both the Scarlett GUI and JACK's graphic interface require you to make connections between devices to make sure they're talking to each other properly, as shown in our screenshots.

Getting aarch64 effects and synth plug-ins

For some good indexes of audio plug-ins, visit **rpimag.co/zynthian**, **rpimag.co/lv2plugins**, and **rpimag.co/kxplugins**. KXstudio also provides a repository of software and plug-ins at **rpimag.co/kxrepos**.

Numerous plug-ins – including many of those listed at the above sites – are available from the Raspberry Pi OS repositories. Use the 'Add/remove software' GUI interface to search for VST, LADSPA, LV2, CALF, and more plug-ins.

Or, for a selection to get you started, open a terminal and type:

```
$ sudo apt install calf-plugins amb-plugins
ardour-lv2-plugins invada-studio-plugins-
ladspa lsp-plugins
```

Electronics enclosures

Projects can live a long life on a breadboard, or as a collection of point-to-point soldering on a workbench – but there comes a time where we might need to move our creation into something more robust

Maker

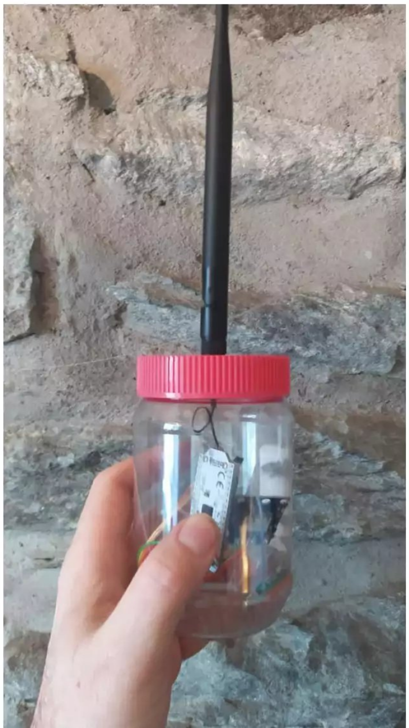
Jo Hinchliffe (AKA Concretedog)

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

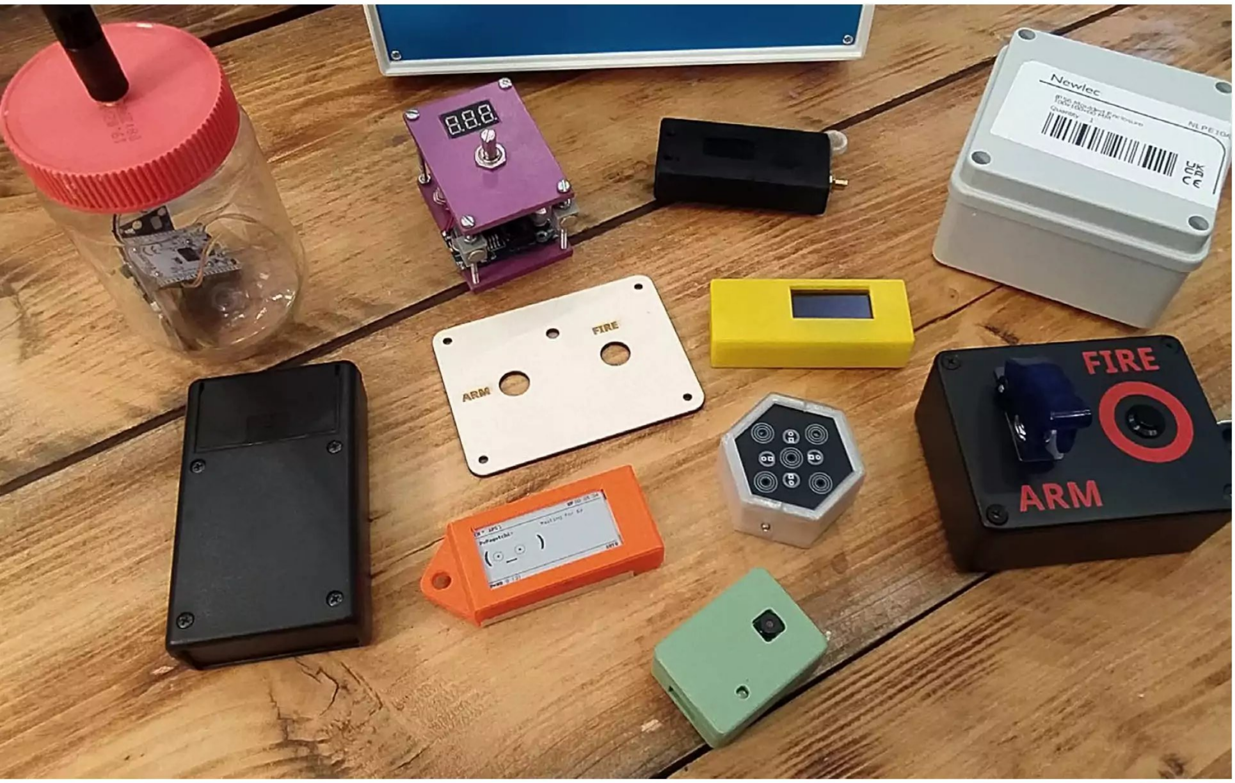
concretedog.blogspot.com

When it comes to enclosures, it's fair to say that there is a never-ending variety of approaches out there in the world of electronics projects (Figure 1). From improvised recycled enclosures, to ordering thousands of injection moulded cases, there's lots of interesting options in between. Let's take a look at some approaches.

A great starting point for when you simply want to take your project or experiment out into the world is an improvised enclosure. As an example in **Figure 2**, we wanted to place a device outside for a while in a static location as an experiment. It didn't need to be a permanent solution, so we stuffed the device and battery inside a plastic peanut butter jar. Using a drill, we made a hole for the antenna connector. We weren't aiming to deploy this project outdoors for extended periods and could choose to avoid wet weather, so we didn't really need to waterproof the antenna connector. However, we were confident that if we misjudged the weather, it would survive a short period in some light rain. Another popular item for a cheap and somewhat weatherproof enclosure is a small Tupperware (or similar style) food storage tub; these are readily available in most supermarkets.



▼ **Figure 2:** A plastic jar destined for the recycling box can make an excellent temporary enclosure and is a good reuse for plastic waste



Alert!

Power Tools

Be mindful of problems caused by power tools. Be careful of strain from vibration, noise hazards and (if necessary) wear protective clothing and goggles. Do your research first!

rpimag.co/powertools

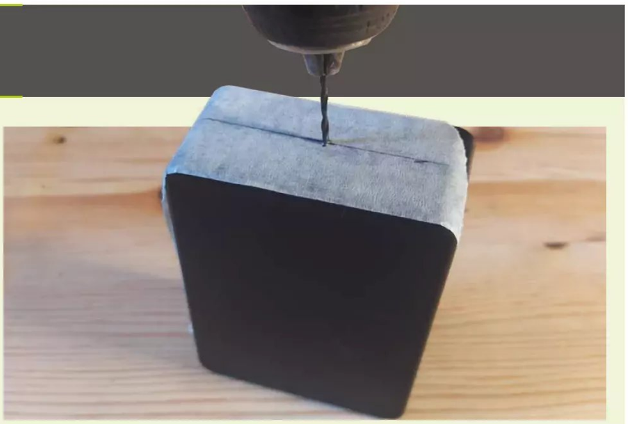
Figure 1: A collection of different enclosure options for a variety of electronics projects

Once you start to consider items as enclosures, you will start to see ideas everywhere. Indeed, there are some classic enclosures made from packaging, none more so than the Altoids tin. This enclosure took a simple mint confectionery tin and spawned a huge movement of building projects in that form factor; in fact, there are numerous kits and products that are designed to be built into this classic tin.

Once you start to consider items as enclosures, you will start to see ideas everywhere

Drilling holes

When you get into using either commercially available cases or found items to home your projects, you will often end up drilling to create holes for panel components. A good starter tip to preserve the good appearance of your enclosure is to use masking tape on it to mark out your drill positions. Apart from not marking your enclosure directly, the masking tape provides a soft surface for your first drill or centre punch to press into, thereby reducing the risk of it bouncing across the panel surface and causing marks and scuffs. It's a good idea to abide by the old adage here when drilling: it's definitely a process in which to take your time and measure twice and drill once!



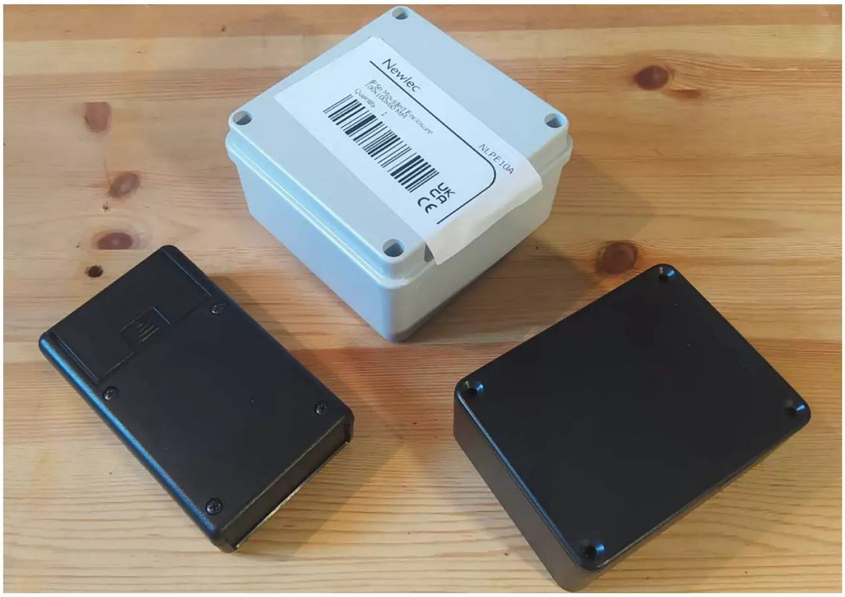


Figure 3: A collection of small commercially available plastic project boxes

Off the shelf

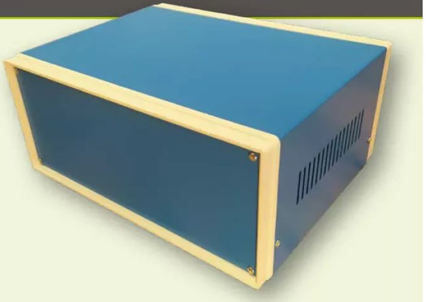
Moving to commercially available enclosures, we can find an amazing amount of choice available online. Probably the most common option is the simple ABS plastic box (**Figure 3**).Whilst perhaps not the most visually exciting, there is a wide array of sizes, shapes, colours, and features for ABS enclosures, across a large number of companies making them. Many come with screw-on lids and you can of course drill, cut, and grind shapes for components through the faces of enclosure. Some will have mounting points inside, allowing for PCBs to be slid into position, and some will have screw mounts for specific-size PCBs to be robustly mounted. Whilst not the full level of engineering rigour, it's also common to use double-sided tape or even hot glue to mount electronics into ABS boxes and this can be a very successful approach. Some ABS boxes will feature drill-to-size cable grommets. These are rubberised stepped small panels cast into the faces of the enclosure into which you drill a slightly undersize hole for the cable you wish to send through the surface. You can then push the cable through the hole to create a reasonably waterproof ingress/egress point. It's fair to say that these types of grommets are OK, but not as robust a solution as buying grommets that are specified for the size of cable and you fit entirely into a pre-drilled hole.

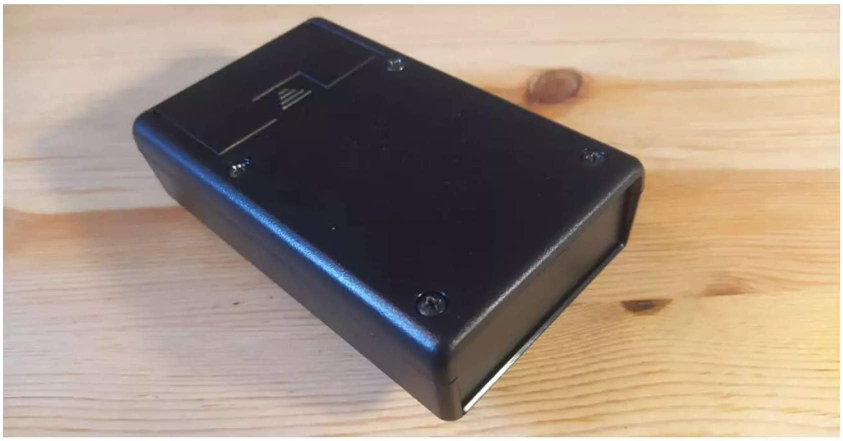
Another consideration with ABS plastic enclosures is that you'll find there is a range of fastening types, from simple self-tapping screws, and bolts with captive nuts, to custom twist-to-close plastic fittings. Also, you can find enclosures with rubber seals/o-rings to create a more weatherproof seal between the enclosure and its lid. Often, at the cheaper end of cases you won't have any particular weather rating, but if you are buying from an electronics retailer you'll often see enclosures with Ingress Protection or 'IP' ratings. These give an accurate measure of two different types of potential ingress: solid objects (dust and other particulates) and water. IP ratings are supplied as the letters 'IP' followed by two digits. The first digit refers to the solid object ingress and the second the water ingress rating. So, for example, an enclosure with an IP rating of IP22 would offer less protection than one rated IP99. You can find lists of charts explaining the IP rating numbers online; a good example can be found on the IEC website (**iec.ch/ip-ratings**). A common higher-end level of protection for an electronics enclosure is IP65 and, looking at the IEC site, we can see that the '6' means that the enclosure is 'dust-tight' and that the '5' means that it should endure jets of water being sprayed on it – often considered more than enough protection for an electronics project destined for the garden!

A really nice off-the-shelf solution for electronics enclosures that is pretty popular is the Hammond-style enclosure. These come in various shapes and sizes, including aluminium extruded versions as well as all manner of plastic injection-moulded.

Benchtop enclosure

Whilst we've mainly considered smaller portable electronics projects in this article, it's worth noting that there are interesting commercial options for benchtop device enclosures. This aluminium panelled case costs less than £15 online and definitely has that lab desk feel about it. It would be perfect for test equipment or perhaps for a DIY variable power supply.



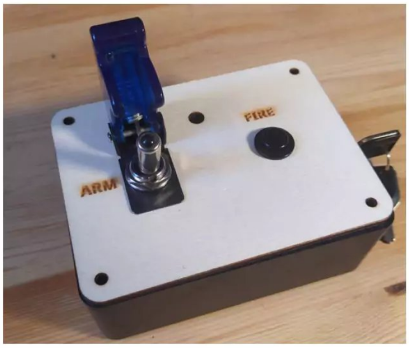


▲ **Figure 4:** A small Hammond 1593QBK enclosure which features a small panel insert, a battery compartment, and internal PCB slots

Hammond cases use removable sections which can be modified to create custom panels. The panels used can be aluminium, fibreglass or plastic, but often people opt to replace them with a custom panel, either hand-cut from material, or indeed by creating a custom PCB panel. In **Figure 4**, the small Hammond enclosure has a heap of features to mount projects internally and a snap-fit battery compartment; it's perfect for many pocket-sized electronics devices.

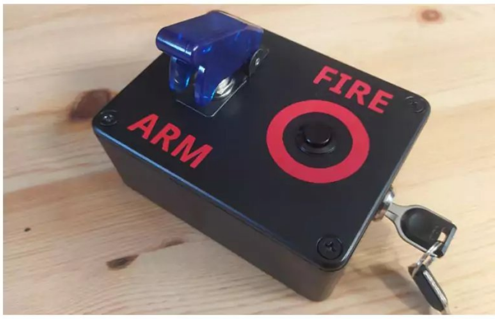
Bespoke

If you have access to some makerspace type equipment, you may have lots of options for creating enclosures and front panel designs for projects. One of the most highly regarded tools for this work is the laser cutter. Of course, you can use a laser cutter to create all manner of boxes and enclosures, but perhaps even more attractive is that you can use a laser to create accurate panels for through-mount components without drilling and you can laser-engrave into the panel to label your project. Another option with this is to mix and match approaches. Using a ruler and a pair of digital callipers, it was pretty straightforward to design a front panel for a model rocket launch controller project in the wonderful free and open-source Inkscape. The panel is designed to fit on top of our simple 100×75mm ABS enclosure (**Figure 5**). Whilst we then have a spare ABS lid, there's no denying that this is a neat approach for custom panels.



▶ **Figure 5:** Using a laser cutter to create a front panel for an existing plastic enclosure

▼ **Figure 6:** Using a vinyl cutter can be an attractive way to make durable decals for an enclosure panel

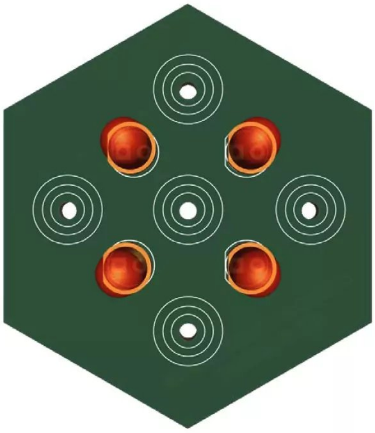


Again, using the 100×75mm ABS enclosure as an example of using makerspace type tools, this time in **Figure 6**, we've used the original enclosure panel, carefully using our tape and drilling techniques to create the holes in the right places in the box and panel.

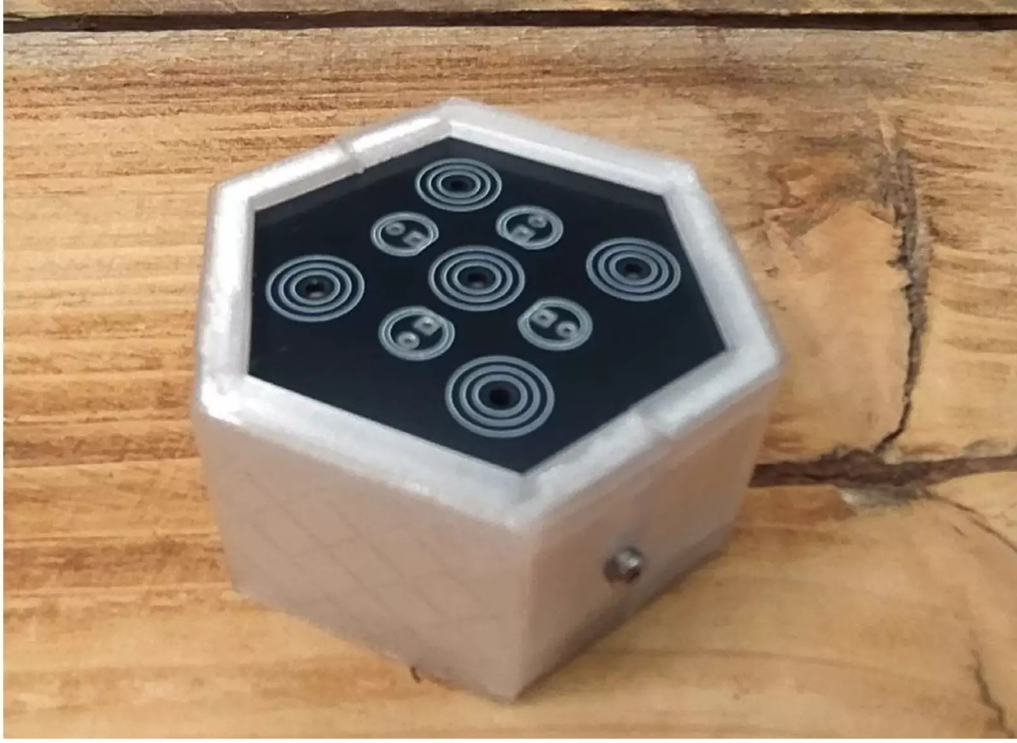
Setting out some labels for vinyl cutting is again pretty straightforward in Inkscape, although we added a little complexity by creating a circle decal to make the fire button extra-visible. Once cut on the vinyl cutter, we simply weeded away the waste vinyl and then used some transfer tape – this allows you to lift off the entire design off the backing paper, to accurately place the sticky vinyl.

Of course, it would be remiss to write a piece on electronics enclosures and DIY and not mention 3D printing. This has been a game-changer in terms of being able, with some CAD knowledge, to create accurate custom enclosures for projects. However, although we definitely think people should develop their CAD skills, it's also worth checking out what designs exist for projects and enclosures on sites such as Printables, GrabCAD, Thingiverse, and more. Many established electronics projects will have pre-designed enclosures, saving you some time and effort.

A nice option if you have, or are looking to develop, some CAD skills for both mechanical design and for electronic PCB design is to perhaps emulate a Hammond-style case. Using KiCad, it's pretty easy to create panels for project enclosures and you can use the silkscreen tools to add text or graphical items to label parts of your design. We opted, without a specific project in mind, to try and create a PCB panel that could be used for creating small panels in future projects. We set



▼ **Figure 7:** Creating a small PCB in KiCad that can act as a front and rear panel in DIY enclosures



▲ **Figure 9:** A completed DIY enclosure with PCB front and rear panels; it would be simple to extend the case length in FreeCAD for larger projects

up a design and, for visual interest, made the design hexagonal; in fact, it's the same dimensions as hexbin-compatible stickers at 50.8mm along its longest axis. Into the hexagonal PCB we added some through-hole pads for 5mm LEDs, as an LED is a common addition to a panel (**Figure 7**). We then added five small uncoated holes through the board, with the idea that these could be drilled out to larger sizes to accommodate other devices like switches, buttons, potentiometers, and more. Rounding out the design, we stuck a logo and email address on the reverse of the panel, meaning that these also have a dual function as a cheap PCB give-away business card.

Made to measure

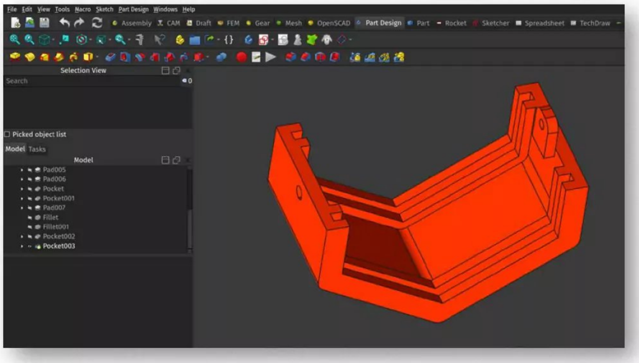
Moving to the enclosure, we fired up the excellent FreeCAD. Our approach here was to model an entire hexagonal tube part, modelling in a groove at each end that would accommodate the PCB edges (**Figure 8**). Once this basic geometry was created, we then used a large rectangle drawn over exactly half of the design and used a pocket operation to cut away half the hexagon. As a simple but effective closure, we've added a hole on one side and on the other a small internal rectangle piece with a hole in it that will align with the outer wall hole. The nice thing about this approach is that both sides of the enclosure are identical. 3D printing is also pretty straightforward as if you place the half hexagon on a flat surface on the build plate of the 3D printer slicer, the rest of the object's walls will sit at printable angles with no supports needed.

Having printed a pair of enclosure halves, once assembled the whole unit is pretty rigid. With a front and a rear PCB panel combined with the two M2 self-tapping bolts, there is no skew or twisting in the design. Whilst not water- or weatherproof, it would certainly make a decent indoor enclosure that's simple to extend and reprint for different lengths. Dare we say it, we also think it looks pretty cool (**Figure 9**).

Hopefully, you've found some hints and tips in this article interesting and if nothing else perhaps it's inspired you to box up some of those long-term breadboard projects we all have!

QUICK TIP

If you are looking for a primer on KiCad, that takes you all the way to having full PCB assembly skills, you might want to check out our *Design An RP2040 Board With KiCad* book: **rpimag.co/kicad2040**.



▼ **Figure 8:** Designing a hexagonal enclosure in FreeCAD

Pico-controlled cable release

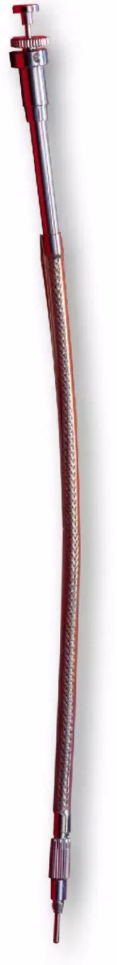
Make an old camera shutter computer controlled using a Raspberry Pi Pico and a slightly beefy servo

Maker

Rob Miles

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

robmiles.com

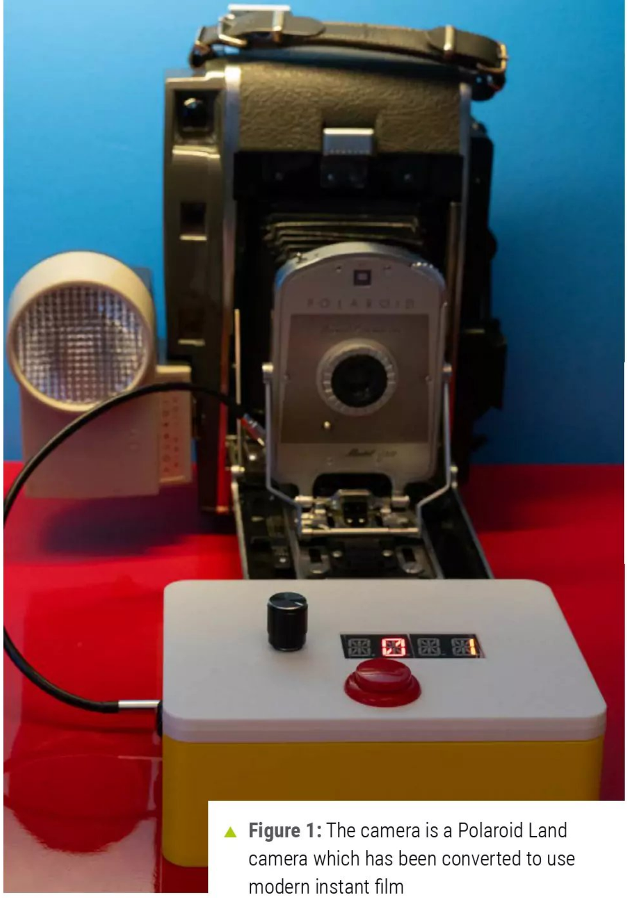


▶ **Figure 2:** This cable release has been half-pressed to show how it works

YOU'LL NEED

- Raspberry Pi Pico or Pico W (version 1 or 2)
- MG995 micro digital servo
- 0.54-inch 14-segment red and orange LED display (HT16K33)
- CJMCU-219 bidirectional DC current sensor (INA219)
- Large push-button (preferably a nice colour)
- Connecting wire (search for '30 AWG wire wrap') which needs a wire wrap tool (search for 'wire wrap tool'). You can use Dupont cables if you would prefer
- A box. There is a 3D printable design, or you can put the release in any box you fancy
- Screws. You'll need some screws sized M2 24 mm in length to fix things to the case (search for 'laptop screws')
- Cyanoacrylate (super) glue to stick the 3D-printed gear to a servo horn

In this article, we'll build a remote release for use with older cameras which have a cable release fitting. This will allow accurate long exposures and provide a self-timer you can use to get yourself into the picture. This is also a good way to explore microcontroller-triggered mechanical actuation. We'll also discover how you can use current monitoring to detect when your device is in the process of destroying itself, which is always something that is good to know. You can find all the software and build instructions here: **rpimag.co/picocamerareleasegh**.



▲ **Figure 1:** The camera is a Polaroid Land camera which has been converted to use modern instant film

Figure 1 shows the remote release connected to a camera. Pressing the red button at the front will trigger the camera shutter for a set time. You turn the knob to select the exposure time. If you hold the knob down and turn it, you select the self-timer delay in seconds. The display shows the delay in red and the exposure in orange. In **Figure 1** the exposure time is set to 1 second and the self-timer delay is 0. The box contains a Pico-controlled servo which presses the cable release you can see coming out of the left-hand side of the box. Let's start with a look at what a cable release does.

Figure 2 shows a camera cable release. If you press the plunger at the top, a metal pin is pushed out of the fitting at the bottom. The fitting is screwed into the camera and the pin triggers the shutter. Using a cable release is a good way to reduce blurring in a photograph, especially if you are using a long exposure. Triggering an exposure with a button on the camera can cause camera shake, but a cable release isolates the movement, keeping the camera steady. Cable releases of this design this have been in use for over 100 years.

Figure 3 shows the first version of the shutter actuator. It is based on a design you can find here: **rpimag.co/servoactuator**. There are two versions of the design for different-sized servos. The author started with a micro servo that he had to hand. When the servo turns, the rack is moved in and out. The end of the rack presses the button on the cable release. This worked very well right up to the point where the author tried to use it to drive his camera. At that point, the servo made some very unhappy noises and nothing moved. The camera button was too hard for the servo to press.

> *It turns out that the servo is strong enough to damage itself*

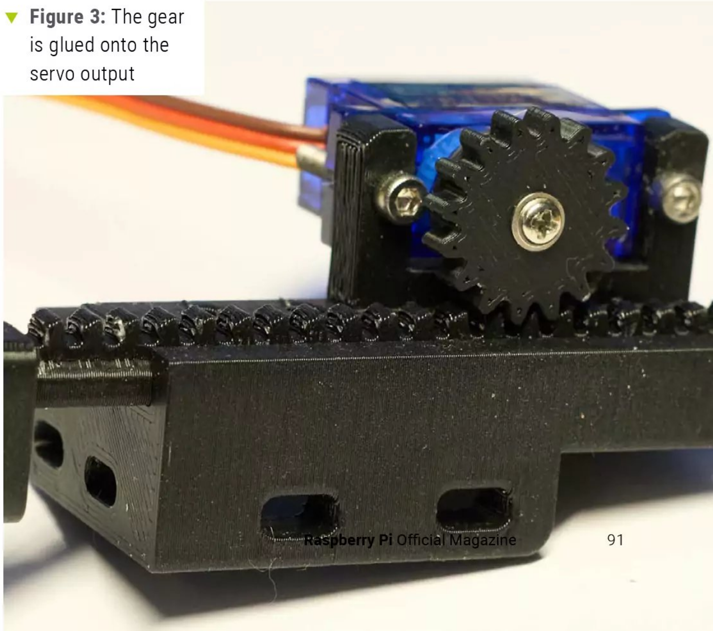
QUICK TIP

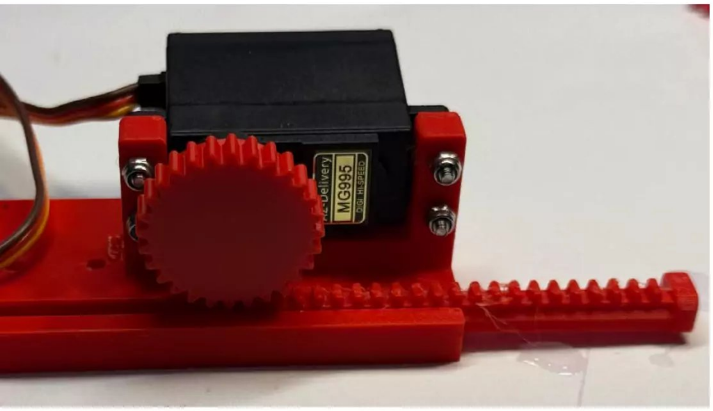
Some modern cable releases have a flexible outer sheath which stretches when the plunger is pressed, and the pin tries to push the mechanism in the camera. These might not work with older cameras which may have shutters that are hard to press. If you have an old camera, it is worth seeking out an older camera release (preferably with a metal sheath) to use with it.

Shutter cocking blues

When a film camera takes a picture, tiny blades in the shutter move to open a hole to let light onto the film. At the end of the exposure, the blades are closed. In older cameras, the blades are moved by a spring-loaded mechanism. Before the shutter can be fired, a spring must first be wound up, or 'cocked'. Usually, this is performed before taking the photo; sometimes, cocking the shutter takes place as part of the action of winding the film onto the next frame. Unfortunately, the Polaroid camera used in this project has a mechanism that tensions the shutter as the shutter button is being pressed. This makes the shutter simple to make and less prone to wear, but it also means that you must press the button quite hard take a picture. Unfortunately, a micro-servo doesn't provide enough torque. It was time to level up the servo.

▼ **Figure 3:** The gear is glued onto the servo output





▲ The servos can be controlled by the same software, but the large servo needs a higher voltage power supply

Figure 4 shows the complete circuit diagram for the remote release. It is powered by five AA batteries that provide 7.5 volts for the servo. The 7.5V is reduced to 5V by a DC-DC voltage converter to power the Raspberry Pi Pico. A four-character text display and a rotary encoder provide the user interface. The encoder contains a push switch that is triggered when the user pushes in the control. The CJMCU-219 sensor reads the current flowing through the servo. The hardware is all controlled by a CircuitPython program.

Classy photography

A class called `CableReleaseController` manages the operation of the device. The code below constructs an instance of this class (called `cable`) with the default hardware configuration. If you use different ports for your build, you can change this to match your hardware.

```
cable = CableReleaseController(
  i2c_sda=board.GP0, i2c_scl=board.GP1,
  encoder_a=board.GP8,encoder_b=board.GP9,
  encoder_button=board.GP22,
  time_button=board.GP16,
  servo_pin=board.GP4 )
```

The `cable` object provides a `run` method which is called to run the cable release.

```
cable.run()
```

The `run` method never returns. It calls the `update` method in the `cable` object 100 times a second:

```
def run(self):
    while True:
        self.update()
        time.sleep(0.01)
```

The `update_shutter` method checks to see if the button has been pressed to trigger the shutter. If the button has been pressed, the method pauses for the length of the self-timer delay and then opens the shutter for the length of the exposure. The `update_times` method checks to see if the rotary encoder has been turned; if so, the method updates the self-timer or the exposure value as appropriate.

Servo specifics

The Pico produces a sequence of control pulses to set the angle of rotation of the servo output. The `servo` class is used to generate these pulses.

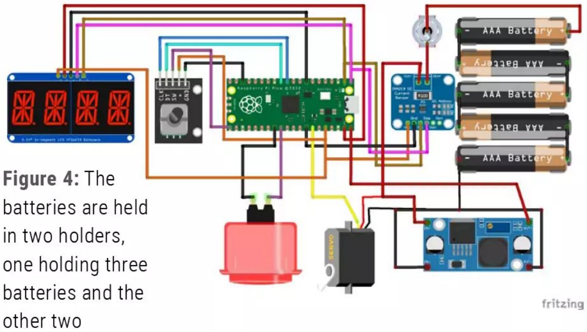
```
from adafruit_motor import servo
```

The statement above imports the `servo` class from the library. The `servo` object is created when the `CableReleaseController` instance is created:

```
pwm_servo = pwmio.PWMOut(servo_pin,
    duty_cycle=2 ** 15,
    frequency=50)
self.servo = servo.Servo(
    pwm_servo,
    min_pulse=500,
    max_pulse=2200
)
```

QUICK TIP

You can set the `min_pulse` and `max_pulse` values to tune the code for your servo.



▲ **Figure 4:** The batteries are held in two holders, one holding three batteries and the other two

The statements above are in the CableReleaseController constructor. They create a servo object and add it to the class as the member self.servo. The servo is then used to control the camera triggering. A program sets the position of the servo by setting the angle property to a value in the range 0 to 180 degrees.

```
cable.servo.angle=70
```

The above statement would set the servo to 70 degrees.

The software moves the servo between a minimum servo angle where the cable release plunger is released to a maximum servo angle where the plunger is pressed in far enough to trigger the shutter.

```
MAX_SERVO_ANGLE=107
MIN_SERVO_ANGLE=20
```

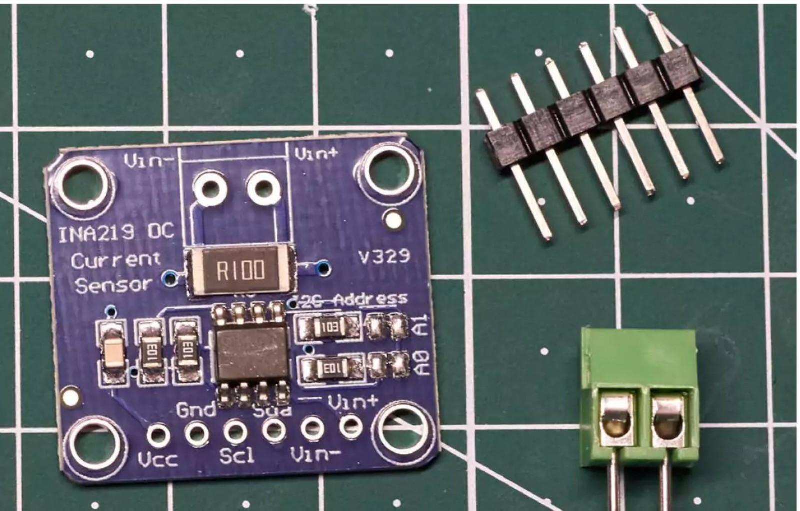
The two values above give the maximum and minimum values for the servo in the cable release controller. These may be different for your release and the camera that you are using.

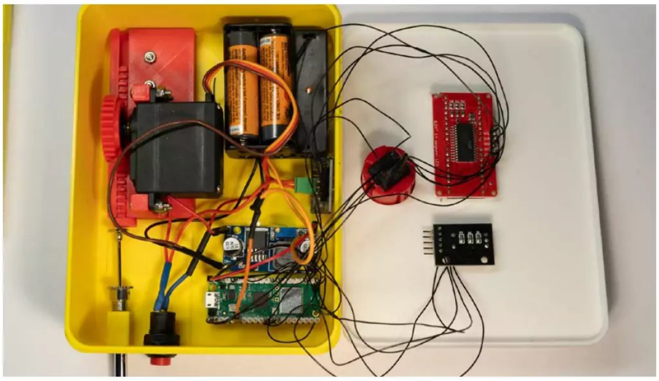
```
def servo_home(self):
    self.servo.angle=self.MIN_SERVO_ANGLE

def servo_trigger(self):
    self.servo.angle=self.MAX_SERVO_ANGLE
```

The servo_home and servo_trigger methods are called to move the cable release in and out. The values of MIN_SERVO_ANGLE and MAX_SERVO_ANGLE must be configured for a particular cable release and camera, otherwise the cable release might push the trigger too far and damage something. The author discovered by accident that the servo mechanism is quite capable of breaking itself when the servo tried to pull the actuator too far back and snapped the end off.

Figure 5: The R100 component of the current sensor is the shunt resistor, which is placed in series with the circuit to be monitored





The current sensor is stuck on the side of the box on the right

Disable self-destruct

It turns out that the servo is strong enough to damage itself if sent to a position outside the range of movement available. A current-monitoring device is used to determine the range of movement automatically and avoid creating a device which will destroy itself the first time it is switched on.

Figure 5 shows a CJMCU-219 current sensor which is fitted with an INA219 device. It measures the current flowing in a circuit by placing a small resistor called a ‘shunt resistor’ in series with the circuit being monitored. When current moves through the shunt resistor, it causes a voltage to appear across it. As the current increases, so does this voltage.

The INA219 device reads this voltage, works out the current that the voltage represents, and delivers information to Pico over an I2C connection. The harder the servo is working (and the more potential for damage), the higher the current. By monitoring the current value, a device can make sure that it doesn’t try to do something that would break the hardware.

You must be a little bit careful when using the current value in this way. When the servo is moving from one position to another, it can consume a lot of current for a very short time. This might make software think that something was wrong when it wasn’t. The current should be checked once the servo can be considered to have arrived at the requested position.

QUICK TIP

During assembly, the servo angle should be set to 180 before the actuator rack is inserted into the gear. Reducing the angle of the servo should then pull the rack into the actuator.

QUICK TIP

When 3D-printing the gear and the actuator, you should use a smaller layer height (perhaps 0.08 mm) and print with a larger infill value (perhaps 60 or so). This will make them stronger.

```
from adafruit_ina219 import INA219
```

The `adafruit_ina219` library contains the `INA219` class which lets CircuitPython programs talk to the INA219 sensor.

```
self.i2c = busio.I2C(i2c_scl, i2c_sda)
self.ina = INA219(self.i2c)
```

The statement above creates an I2C connection and then uses this to create an instance of the `INA219` class. The instance is referred to by the variable `self.ina`. This instance exposes a set of properties given the current and voltage present in the system:

```
def show_current(self):
    bus_voltage = self.ina.bus_voltage
    current = self.ina.current
    power = self.ina.power
    shunt_voltage = self.ina.shunt_voltage

    # Print readings
    print(f"Bus Voltage: {bus_voltage:.2f} V")
    print(f"Current: {current:.2f} mA")
    print(f"Power: {power:.2f} mW")
    print(f"Shunt Voltage: {shunt_voltage:.2f}
mV")
```

The method `show_current` above reads data from the sensor and then prints it to the console. This would allow manual monitoring, but it would be nice to have the cable release automatically determine the start and end points.

```
def limit_test(self,start,delta,max_current):
    result = start
    while True:
        self.servo.angle = result
        time.sleep(1.5)
        current=self.ina.current
        print(f"   Current:{current}
Angle:{result}")
        if self.ina.current >= max_current:
            result = result - delta
            self.servo.angle = start
            print(f"Result:{result}")
            return result
        result = result + delta
```

The `limit_test` function receives a `start` angle, a `delta` amount the angle should change for each step and `max_current`, the maximum current that should be used. The function puts the servo in the start position and then moves it in delta-sized steps until the current flowing to the servo exceeds the max value. The function then returns the angle before that point was reached.

```
def set_limits(self):
    self.servo.angle = self.CENTRE_SERVO_ANGLE
    # do the inner limit first
    self.MIN_SERVO_ANGLE = self.limit_test(
        self.CENTRE_SERVO_ANGLE,-1,self.MAX_
CURRENT)
    self.MAX_SERVO_ANGLE = self.limit_test(
        self.CENTRE_SERVO_ANGLE,1,self.MAX_
CURRENT)
```

The `set_limits` function calls `limit_test` to set the servo limits. Note that for the inner limit, the delta value is negative so that the actuator is moved back from the cable release. The `set_limits` function can be called each time that the cable release is started, or you can use it once to obtain the maximum and minimum values and set them in the code.

Happy snaps

The release works well and there is considerable scope for development. The Pico could host a web page to allow the camera to be triggered remotely. The release could be programmed to take multiple pictures when the button is pressed, rather than just one. The current monitoring could be performed during movement, rather than just when the release is set up.

The cable release mechanism can also be used in any situation where you want a Pico-powered device to move something mechanical. You could even use it to create a mechanical ‘thumb wrestler’ which measures the strength of the opponent by tracking the current value, although the author hasn’t done that yet.

2nd Edition
Updated for Raspberry Pi Pico W
Get started with
MicroPython on Raspberry Pi Pico
The Official Raspberry Pi Pico Guide
PLAYER 1 WINS!
Raspberry Pi PRESS
Gareth Halfacree,
Ben Everard

Extreme Raspberry Pi

Take Raspberry Pi to its very limits with these incredible projects.

By the pretty pedestrian **Rob Zwetsloot**

Raspberry Pi is a very powerful, very small, very customisable device and we have seen it be used for so many things over the years because of this. Whether folks are slipping them into 3D-printed classic console cases or simply hiding them away as home file servers, we've covered many of the cool things the community have done with Raspberry Pi.

Which begs the question – how can you push the limits with a Raspberry Pi? Well, with 13 years of projects and Raspberry Pi models, you'll be surprised just how far/fast/high/deep Raspberry Pi has gone. We're taking Raspberry Pi to the extreme.

Have a more extreme use of Raspberry Pi? An updated record? Email us at magazine@raspberrypi.com

Longest-running Raspberry Pi

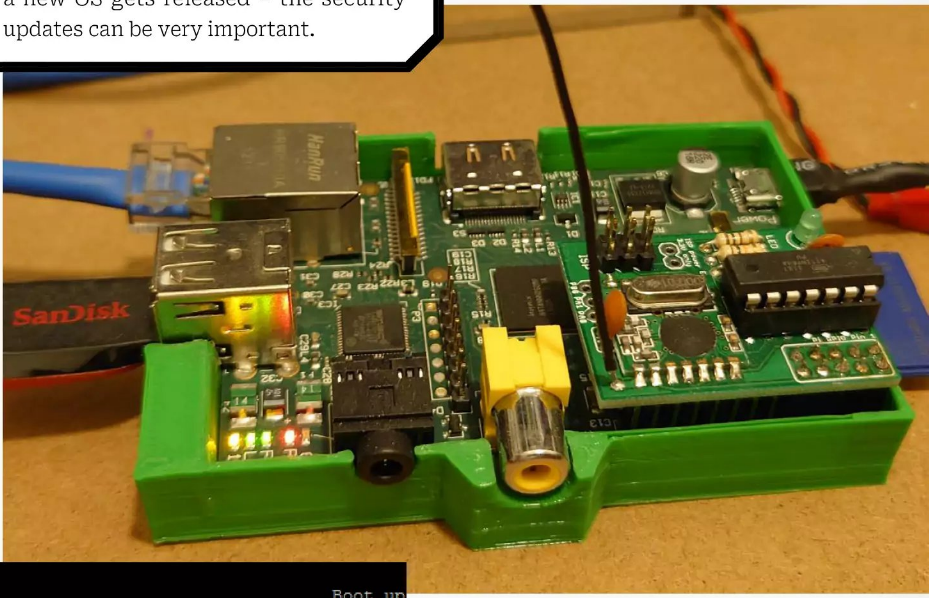
2331 days of uptime for the Model B that could

rpimag.co/uptime

“This Raspberry Pi has been running for ten years,” says Reddit user KerazyPete, although not without a few reboots. At the time of posting this in late 2023, his Raspberry Pi 1 Model B (revision 0002 with 256MB of RAM no less, one of the first production versions) had been ‘up’ for 2331 consecutive days – since July 2017.

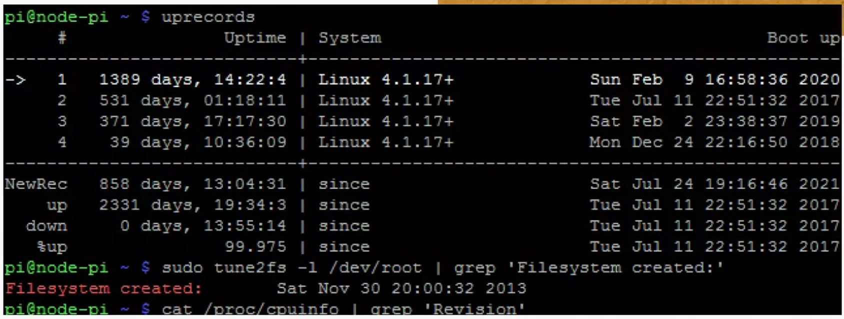
It’s the longest uptime we’ve seen for a Raspberry Pi – in fact, it's pretty impressive with occasional power cuts, power surges, and accidental unplugs. Our file servers and media centres have undergone hardware upgrades as well.

You can easily check how long your Raspberry Pi has been on yourself by opening a terminal and typing `uprecords` – maybe you’ll be surprised. Although we do recommend upgrading the operating system on your Raspberry Pi whenever a new OS gets released – the security updates can be very important.



▶ The system in question. The original Raspberry Pi 1 design is quite nostalgic

▼ The `uprecords` command shows how long the system has been running



```
pi@node-pi ~ $ uprecords
     #               Uptime | System                                     Boot up
----------------------------+---------------------------------------------------
->   1   1389 days, 14:22:4 | Linux 4.1.17+             Sun Feb  9 16:58:36 2020
     2   531 days, 01:18:11 | Linux 4.1.17+             Tue Jul 11 22:51:32 2017
     3   371 days, 17:17:30 | Linux 4.1.17+             Sat Feb  2 23:38:37 2019
     4    39 days, 10:36:09 | Linux 4.1.17+             Mon Dec 24 22:16:50 2018
----------------------------+---------------------------------------------------
NewRec   858 days, 13:04:31 | since                     Sat Jul 24 19:16:46 2021
    up   2331 days, 19:34:3 | since                     Tue Jul 11 22:51:32 2017
  down     0 days, 13:55:14 | since                     Tue Jul 11 22:51:32 2017
   %up               99.975 | since                     Tue Jul 11 22:51:32 2017
pi@node-pi ~ $ sudo tune2fs -l /dev/root | grep 'Filesystem created:'
Filesystem created:       Sat Nov 30 20:00:32 2013
pi@node-pi ~ $ cat /proc/cpuinfo | grep 'Revision'
```

Largest Raspberry Pi

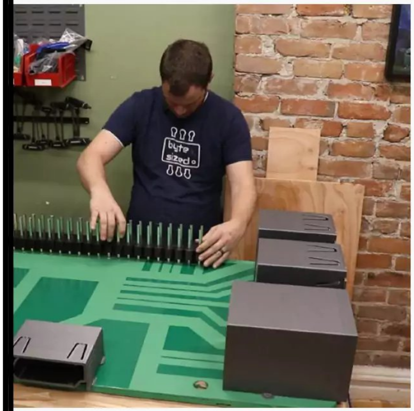
Jumbo working Raspberry Pi 3

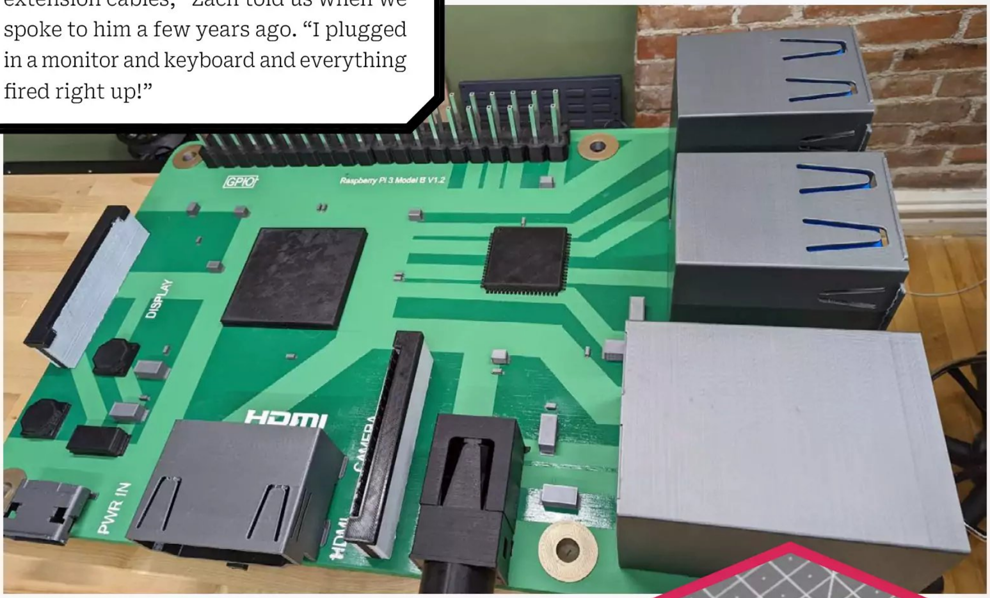
rpimag.co/largest

There are a few huge, working Raspberry Pi computers. A 10× Raspberry Pi 3 was shown off at Maker Faire Bay Area in 2016, and Raspberry Pi's own Toby Roberts built a 6× Raspberry Pi 4 for a little exhibition in the shopping centre where the official Raspberry Pi Store resides. This 12× working Raspberry Pi takes the crown, though, designed and 3D-printed by Zach Hipps.

The PCB is made from plywood, the GPIO pins are made from aluminium tubing, and over 5kg of PLA filament was used for the other large-scale components. "I connected my Raspberry Pi to all the large-scale connectors with extension cables," Zach told us when we spoke to him a few years ago. "I plugged in a monitor and keyboard and everything fired right up!"

▼ Raspberry Pi 3; human being for scale

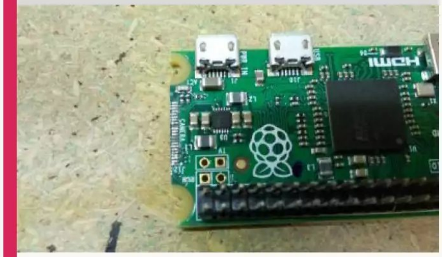




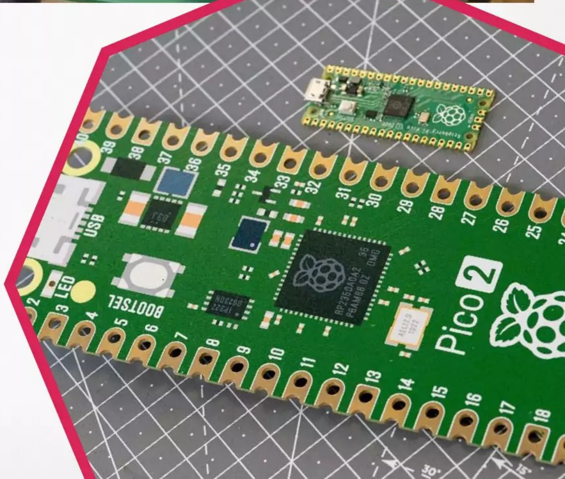
▲ Most Raspberry Pi devices are a little less conspicuous than this

Smallest Raspberry Pi

While some tiny RP2040/RP2350 boards are technically the smallest Raspberry Pi products, the smallest standard Raspberry Pi computer is Raspberry Pi Zero. However, you can make it smaller still – 5mm smaller as it goes. Raspberry Pi Zero v1.3 has a camera connector which can be removed from the board and, with no circuits in that section of the board, you can trim it. Check out the forum post about it: **rpimag.co/smallest**.



▶ The Pico Jumbo is an adapter for Raspberry Pi Pico 2 that makes it 3.5× larger: **rpimag.co/picojumbo**



Pi in the Sky

During the dawn of Raspberry Pi, and for many years afterwards, high-altitude balloonist Dave Akerman would regularly send balloons about 40km into the mesophere (which is above the stratosphere, Queen fans) with a Raspberry Pi attached to take photos and other telemetry. In 2016 he broke the world record for the highest live image sent down from an amateur balloon, at 41,837 metres. That probably makes it the highest a Raspberry Pi has gone without the use of a rocket.

▼ A successful mission photo

▶ CubeSats are very unassuming, but this one did go into space with a Raspberry Pi aboard

Highest Raspberry Pi

GASPACS

428km over the Earth

rpimag.co/highest

Is it cheating to go to space so your Raspberry Pi can claim the altitude record for the computer? Not when there are several Raspberry Pi computers in space – the current highest are the Astro Pi boxes up on the International Space Station. However, GASPACS beat their record by a mere six kilometres on its 117-day mission in 2022.

The GASPACS (Get Away Special Passive Attitude Control Satellite) was a 1U CubeSat built by students at Utah State University to test aerobraking with an inflatable ‘AeroBoom’. Since the Earth is not a perfect sphere, the CubeSat’s orbital altitude was 416km (258 miles) at its perigee, and 428km (267 miles) at its apogee.

◀ The first pair of Astro Pi boxes went up for Tim Peake's mission

Fastest Raspberry Pi
Astro Pi

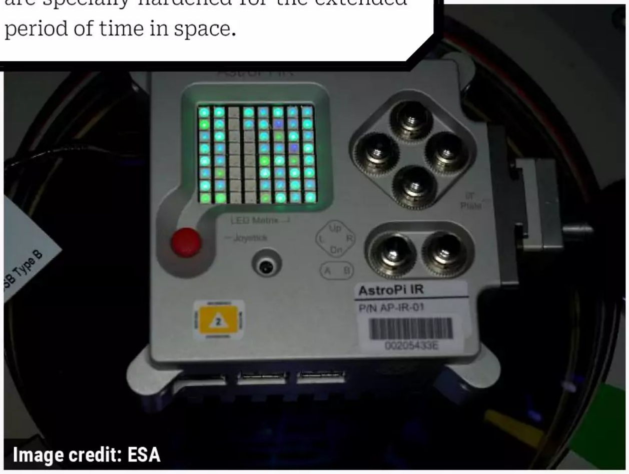
17,100mph is easier with no friction

astro-pi.org

While Astro Pi on the ISS doesn't hold the altitude record for a Raspberry Pi, due to the peculiarities of orbital physics it is the fastest. Very basically, having a lower orbit means you need to go faster so that you don't fall to Earth.

The ISS is clocked at 17,100mph (27,520km/h) – that's 4.77 miles per second (7.67km/s), and 22.5 times the speed of sound at ground level. Those numbers are difficult to conceptualise, but it orbits the Earth in just shy of 93 minutes, meaning it orbits 15.5 times a day. Pretty quick!

Astro Pi in orbit has various sensors thanks to a Sense HAT, including motion sensors, which are used by schoolchildren to run experiments via code. Unlike GASPACS, these Raspberry Pi computers are specially hardened for the extended period of time in space.



▶ A humble Raspberry Pi living on the ISS

Deepest Raspberry Pi

Maka Niu

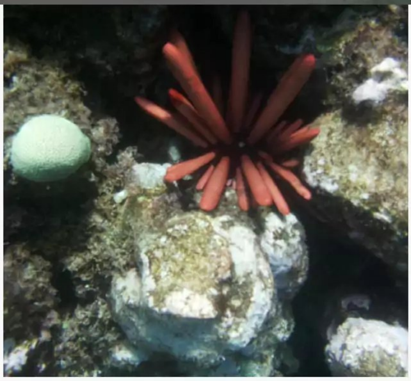
1500m underwater – and possibly able to do 6000m – is no mean feat

rpimag.co/deepest

Going deep underwater is hard. The deeper you go, the more the weight of the water above you becomes a huge issue, and human-made devices need to withstand the pressure in the depths. It's why your watch may only be rated to 100m underwater – stuff will then start to break.

A descent of 1500 metres – nearly a mile underwater – is very, very far. At this depth in salt water, pressure is 148 atmospheres, or 148 times normal air pressure at sea level.

Maka Niu is a very special system then, able to safely contain a Raspberry Pi Zero and Camera Module V2 in a low-cost device – opening up citizen science to more folks, and helping to explore the largely unknown deep sea.



▲ It may just be a tube, but it's a very sturdy one

◀ They're called sea urchins because hedgehogs used to be called urchins

Raspberry Pi submersible at 300m depth

While unable to make it to depths of 1500m, this pilotable Raspberry Pi craft can go to 300m, which is still very impressive. The price tag of $4250 (£3288) is a little eye-watering, but it's at least a factor of ten cheaper than other commercial ROVs of its type. Find out more at **bluerobotics.com**.

Highest clock speed

Reaching a limit of 3.6GHz

rpimag.co/fastesthz

'Fastest Raspberry Pi' can really mean two things, which is why we've awkwardly titled this 'highest clock speed'. Every new numbered model of Raspberry Pi is slightly faster than the last, but the question for some people is: how fast?

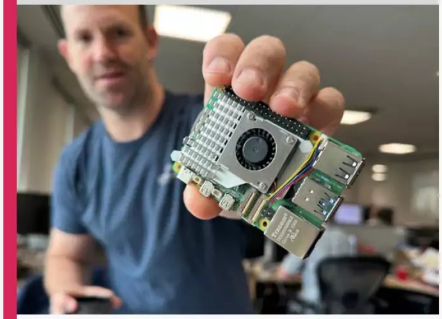
Overclocking computer hardware – the process of telling it to run faster than it was designed to, i.e. increase its clock speed – is an age-old tradition amongst tech nerds. The caveat is, this tends to make the hardware hotter and can damage it in the long-run. In normal overclock scenarios, this is solved with liquid cooling or other cooling solutions, but if you want to take a chip to its absolute limits, you need something really cold: liquid nitrogen.

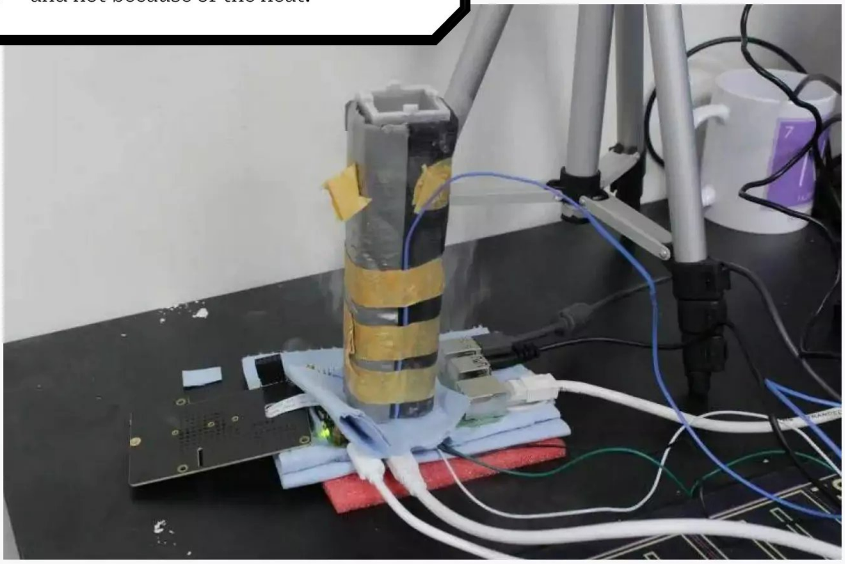
With a specialised tube over Raspberry Pi 5, Pieter-Jan Plaisier poured liquid nitrogen directly onto the chip as it ran at 3.6GHz. Can it go higher? Supposedly not – trying at 3.7 GHz the system crashed, and not because of the heat.

Warning!
Nitrogen
Be careful when handling liquid nitrogen. Wear protective clothing and eyewear and follow all instructions. Work in a well-ventilated area.
rpimag.co/nitrosafety

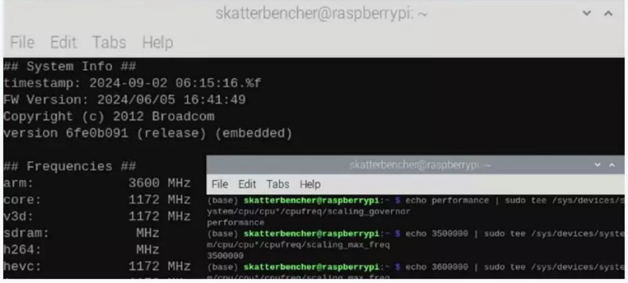
Hottest Raspberry Pi?

What is the hottest environment a Raspberry Pi operates in? Unfortunately, we couldn't find an answer – some have operated in deserts that regularly clock 40°C (104°F). The real trick here is that with good enough cooling, you can have a Raspberry Pi operate in very high temperature environments, and the BCM2712 chip that runs Raspberry Pi will operate at temperatures as high as 80–85°C (176–185°F) before throttling is enabled.





▲ It works but it's not very practical – it's just a test after all



```
## System Info ##
timestamp: 2024-09-02 06:15:16.%f
FW Version: 2024/06/05 16:41:49
Copyright (c) 2012 Broadcom
version 6fe0b091 (release) (embedded)

## Frequencies ##
arm:           3600 MHz
core:          1172 MHz
v3d:           1172 MHz
sdram:          MHz
h264:           MHz
hevc:          1172 MHz
```

```
(base) skatterbencher@raspberrypi:~ $ echo performance | sudo tee /sys/devices/s
ystem/cpu/cpu*/cpufreq/scaling_governor
performance
(base) skatterbencher@raspberrypi:~ $ echo 3500000 | sudo tee /sys/devices/syste
m/cpu/cpu*/cpufreq/scaling_max_freq
3500000
(base) skatterbencher@raspberrypi:~ $ echo 3600000 | sudo tee /sys/devices/syste
m/cpu/cpu*/cpufreq/scaling_max_freq
```

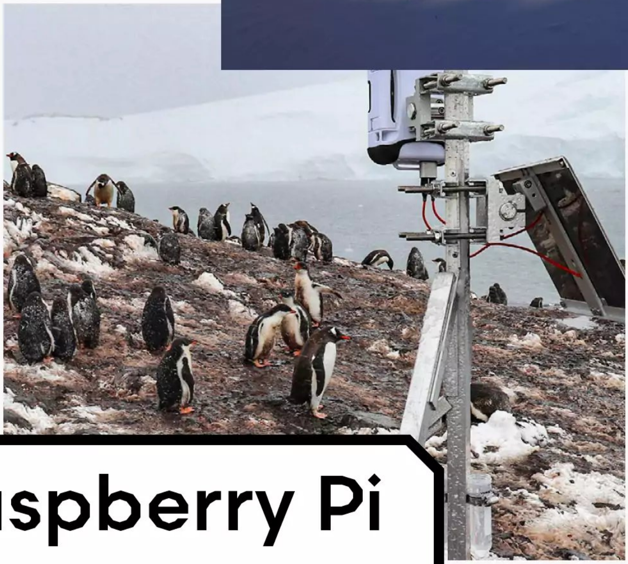
◀ Proof from the terminal of the amazing feat

Raspberry Pi operating temperatures

Recently, Compute Module 4 got an extended temperature variant, giving it an even lower operating temperature of -40°C (-40°F). Many parts of the world, not just Antarctica, can get to below -20°C (-4°F), which is the standard lowest operating range of other Raspberry Pi products. 85°C/185°F is the highest operating temperature of Raspberry Pi.

▶ The photos were used for time lapses

▼ The solar-powered station taking photos of some feathered friends



Coldest Raspberry Pi

Arribada Penguin Monitoring

It gets to -60°C in the Antarctic winters

rpimag.co/coldest

Sending any equipment to Antarctica is tricky – it gets very, very cold there. When your tech accidentally stays there for three years, and remains working, that is quite the feat. The Arribada Penguin Monitoring project (accidentally) managed that, being unable to pick up the camera at the end of 2019 and then, well, the world shut down in 2020 during the start of the Covid-19 pandemic.

While the temperature regularly gets below -30°C (-22°F), it can reach -60°C (-76°F) in the winter. Antarctica is very large though and depending on the placement of the camera, the temperature will be different. Having said that, after three years in the frigid wasteland, the Arribada camera returned safely with 32,764 photos for the researchers to go through.

ONLY THE **BEST**

Audio accessories

Make your Raspberry Pi sound even better with an audio add-on

By **Phil King**

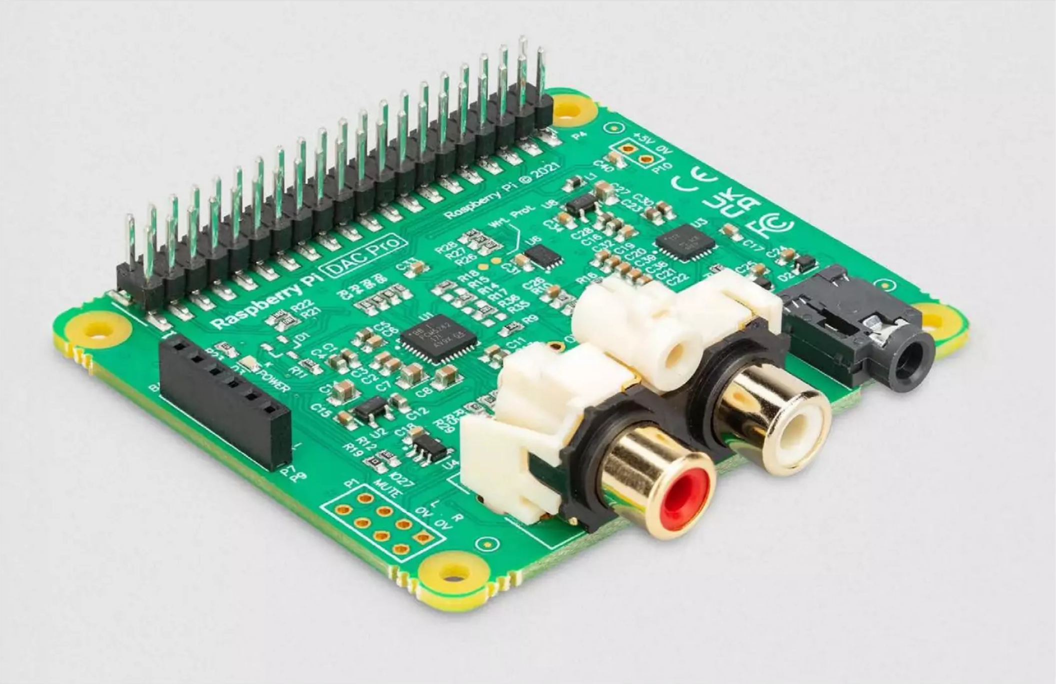
While some Raspberry Pi computer models, such as Raspberry Pi 3 and 4, feature an audio jack (3.5mm, shared with composite video), others – such as Raspberry Pi 5 and Zero – can only output sound via HDMI by default. And a Raspberry Pi Pico microcontroller is completely silent by itself.

So, if you want to output high-quality audio for a custom hi-fi system or home theatre PC setup – a very popular use for Raspberry Pi – you'll need some kind of audio HAT or add-on featuring a DAC (digital to analogue converter) to turn the digital data into sound. Fortunately, there's a wide array of them to choose from. We've rounded up some of the best and most interesting audio boards around. One even combines a DAC with an NVMe interface for SSD storage.

One thing to note is that some HATs include an on-board amplifier and are designed to be used with headphones or directly wired speakers. Others don't and their line-level audio output needs to go into an external amplifier (such as on a hi-fi) linked to speakers.

Raspberry Pi DAC Pro

Raspberry Pi | £24 / $26 | raspberrypi.com



Verdict

Superb sound quality when you hook it up to a hi-fi.

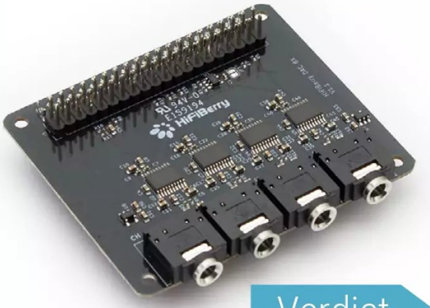
◀ Output outstanding audio with this hi-fi HAT

This is one of four official Raspberry Pi digital audio boards – the others are the DAC+, DigiAmp (with integrated amplifier), and smaller Codec Zero. It boasts the highest audio quality of the range thanks to its Texas Instruments PCM5242 chip, which outputs high-definition 24-bit audio at 192kHz. That's far better than standard 16-bit CD quality, although you'll need to be an audiophile to notice the difference. What you will appreciate is the crystal-clear sound with a rich, warm feel.

Just mount the board onto the 40-pin GPIO header of any Raspberry Pi computer and it's ready to use – no drivers required, nor external power. You can then use standard audio cables to hook up its twin RCA (aka phono) outputs to a hi-fi amplifier and speakers. Alternatively, there's a 3.5mm jack stereo output connected to the board's built-in headphone amp. As a bonus, there's a mini header for use with an optional XLR board for a differential/balanced output – as used by some high-end hi-fi systems.

HiFiBerry DAC8x

HiFiBerry | £60 / $65 | hifiberry.com



▲ If you need more channels/outputs, the DAC8x fits the bill

Verdict

Ideal for multi-speaker home theatre setups.

HiFiBerry is one of the best-known makers of audio boards for Raspberry Pi. The range includes the DAC 2, delivering 24-bit 192kHz sound and minimal noise and distortion due to a custom current-to-voltage output stage featuring high-end filtering.

The DAC8x is a unique beast, however, While most Raspberry Pi DAC HATs feature a standard two-channel stereo output, this one boasts eight channels, making it ideal for multi-speaker audio and home theatre setups. The possibilities are exciting – you could even use it for a DIY synthesizer project using something like SuperCollider.

Those eight channels are output via four stereo 3.5mm jacks. They are fed by four dedicated 192kHz/24-bit high-quality TI Burr-Brown DACs with a typical signal-to-noise ratio of 112dB. Two ultra-low-noise linear voltage regulators are used to optimise audio performance.

Note that the DAC8x is only compatible with Raspberry Pi 5 and you'll need to alter a line in the **config.txt** file to get it working in Raspberry Pi OS.

Audio DAC SHIM

Pimoroni | £13 / $14 | pimoroni.com

Verdict

Good value and leaves the GPIO header fully accessible.

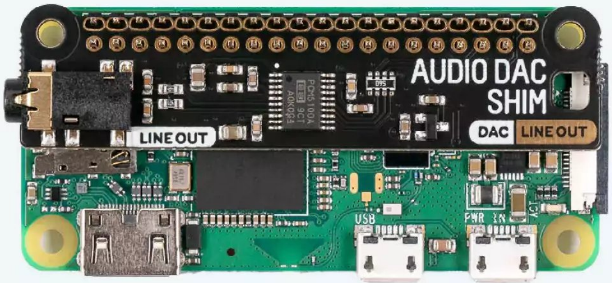
▼ The smallest DAC add-on around

We've covered Pimoroni's range of Pirate Audio boards before – Zero-size HATs featuring an on-board mini LCD and a choice of features and outputs.

The firm has now launched a couple of even smaller audio DAC products in the form of super-slimline SHIMs – short for 'Shove Hardware In Middle'! – that fit over Raspberry Pi's GPIO pins and stay put thanks to a friction-fit connector. You can even sandwich a SHIM between a Raspberry Pi and a HAT.

The Audio DAC SHIM (Line-Out) features a stereo line-level output via a 3.5mm jack, so you can hook it up to a hi-fi amp and speakers. There's also an Audio Amp SHIM with a built-in 3W mono amp and push-fit connectors for a speaker – suitable for a mini DIY audio project requiring sound.

Both SHIMs deliver 24-bit/192kHz audio over I2S – Inter-IC Sound, a serial interface protocol – and can be used with Pimoroni's Pirate Audio software, which configures I2S and also installs the Mopidy audio player.



NVDAC

52Pi | £29 / $30 | 52pi.com

A two-in-one solution for audio and storage

If you really like your HATs to be multifunctional, this one may well be of interest. Designed for use with Raspberry Pi 5, it combines a DAC with an NVMe interface to add SSD (solid-state drive) for storage. Thinking about it, that makes a lot of sense for a NAS or home theatre setup, for which you want extra storage and high-quality sound.

The DAC part is supplied by a HiFiBerry DAC+ Standard, delivering high-quality 24-bit/192kHz line-level audio via two RCA (phono) ports that you can connect to a hi-fi amp and speakers – there's no headphone jack. The board utilises the I2S protocol to connect directly to Raspberry Pi 5's CPU, preserving audio signal integrity.

For the NVMe functionality, there's a standard M.2 connector with enough room on board for a 2230 or 2242 SSD. A 40mm FFC ribbon cable (supplied) is used to connect the board to Raspberry Pi's PCIe slot.

Verdict

A unique HAT with dual functionality.

Pico Audio Pack

Pimoroni | £15 / $16 | pimoroni.com

The Raspberry Pi Pico range of microcontroller boards is ideal for smaller electronics projects which don't require the full processing power of a Raspberry Pi single-board computer. If you need to output any audio, however, there's no obvious way to do it from a Pico, which lacks an audio output.

You could always fiddle around with pulse-width modulation (PWM) to output audio signals from the GPIO pins by switching them on and off very quickly, possibly making use of Pico's Programmable IO blocks. That's fine for basic audio effects, but don't expect stellar sound quality.

Alternatively, just grab a Pico Audio Pack. It adds a PCM5100A DAC to output high-quality (up to 32-bit, 384kHz) stereo audio through a 3.5mm line-out connector – or amplified stereo from a 3.5mm headphone jack, thanks to its built-in mini amplifier.

Verdict

A neat way to get sound out of your Pico.

You even get a choice of audio outputs

Pirate Audio Dual Mic

Pimoroni | £26 / $29 | pimoroni.com

Verdict

A handy way to record high-quality sound clips.

◀ Record stereo sound clips with this mini HAT

Outputting sound is all very well, but what if you need audio input? Sure, you could always plug in a USB mic or sound card connected to a microphone. Alternatively, the Pirate Audio Dual Mic features two mini mics on either side of the mini HAT to capture sounds in stereo.

The mics feature a SiSonic acoustic sensor, a serial ADC (analogue-to-digital converter), and an interface to convert the signal into the industry-standard 24-bit I2S format. They record sound with crystal-clear quality, with volume levels shown on the LCD, although the mics aren't that far apart so the stereo effect is limited. It could prove very useful for any project that requires audio input, such as a portable recorder or voice assistant.

To make the board work, you will need to download the related software, which includes a couple of Python code examples. Playback of recorded sounds is via HDMI by default.

40 MM SPEAKER

The Pi Hut | £4 / $4 | thepihut.com

▶ You'll need a mini amp to power this mini speaker

If you just want some sound output for a portable project, such as a hand-held games console, a full-size audio HAT would likely be overkill – in which case you can just use a small amp board (such as Pimoroni's Audio Amp SHIM) teamed up with a mini speaker, like this one from The Pi Hut. With 4 Ω impedance and up to 3 W of power, it's loud and clear enough for most mini projects.

KIWI KVM

A digital KVM for connecting a Raspberry Pi to another PC – no networking required. **Rob Zwetsloot** gets connected

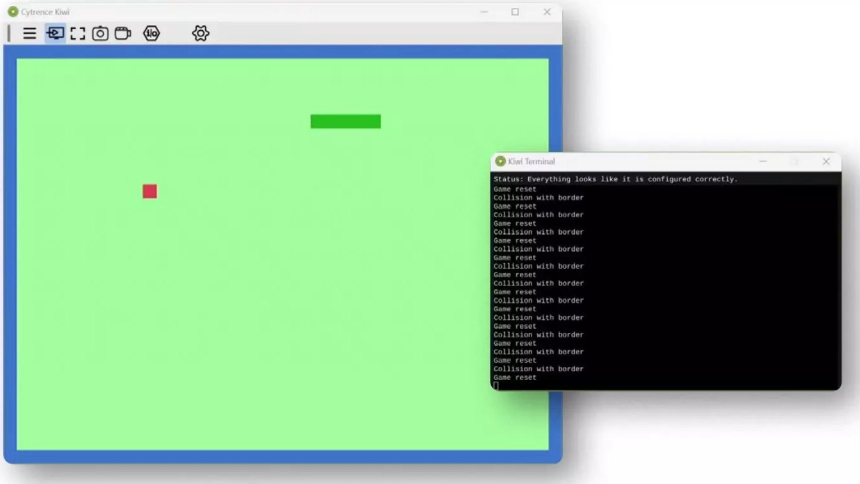
Cytrence **rpimag.co/kiwikvm** From £70 / $89

SPECS

I/O: USB-C (host connection), USB-C (input connection), HDMI, 6 × GPIO (PRO version)

DIMENSIONS: 46 × 46 × 15 mm

CONNECTIVITY: 1080p video, human input devices, virtual serial connection, UART, ATX



```
Status: Everything looks like it is configured correctly.
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
Collision with border
Game reset
```

▶ Debugging a Snake game with KIWI KVM

Screen capture tools are very common these days; however, usually you cannot interact with the screen on the other end. KIWI, on the other hand, hooks your mouse and keyboard right up to it.

The concept is fairly simple – download the software (**rpimag.co/kiwisoftware**), do a quick install, and then plug your Raspberry Pi into the device using an HDMI cable and an (included) USB-A to USB-C cable to connect keyboard and mouse. We've used several screen capture tools in the past for various reasons – whether it's live streaming, video making, or screenshot taking – and while a lot of them are in theory plug-and-play like this, that's not always the case. Surprisingly, we had no such issue with KIWI, despite it also providing input.

The device is ultra-compact

Plug 'n' play

There is a little bit of noticeable lag when working over KIWI. The mouse cursor feels a little bit like it's being dragged through mud, although keyboard inputs feel more responsive. It's faster than Raspberry Pi Connect at least, but we wouldn't want to be doing any twitch FPS gaming using it.

It also provides several very useful functions for this kind of screen capture – sending some basic key commands, allowing you to paste from your host computer and even do a screen recording. The interface for this is all very customisable, even allowing for a random 'mouse jiggle' to keep a screen alive.

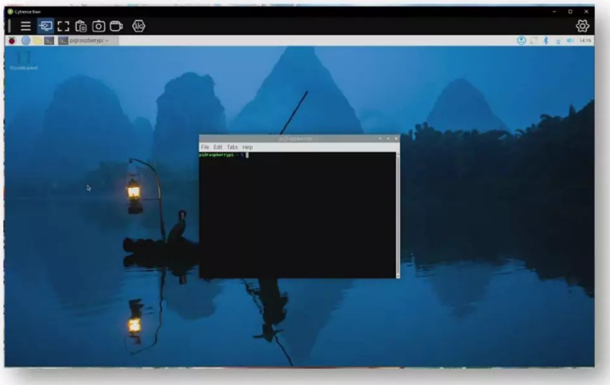
Extra functions

A fairly unique ability in its PRO version is the ability to (physically) uncover some extra GPIO ports on the device and control them via the interface, which also have UART. They're marketed at IT professionals, allowing for debugging and control of ATX power, and there's even an extra function of turning the input cable into a virtual serial cable. It's quite impressive.

> *Due to the speed of getting it working, no need for any local networking, and its fairly small footprint, it's a great alternative to Connect*

With the myriad ways you can connect to a Raspberry Pi from another device, we did wonder if this would be superfluous. However, due to the speed of getting it working, no need for any local networking, and its fairly small footprint, it's a great alternative to Connect and other network-based remote connection tools. These network tools also only work once a system has fully booted up, whereas KIWI's physical connection allows you to see the boot process, which means you can troubleshoot any misbehaving Raspberry Pi without plugging it into another monitor.

If you use Raspberry Pi a lot, this is definitely worth considering adding to your arsenal of add-ons.



```
pi@raspberrypi:~ $
```

Controlling the desktop of Raspberry pi

Verdict

A surprisingly functional and full-featured digital KVM and screen capture device, and much smaller than even a Raspberry Pi.

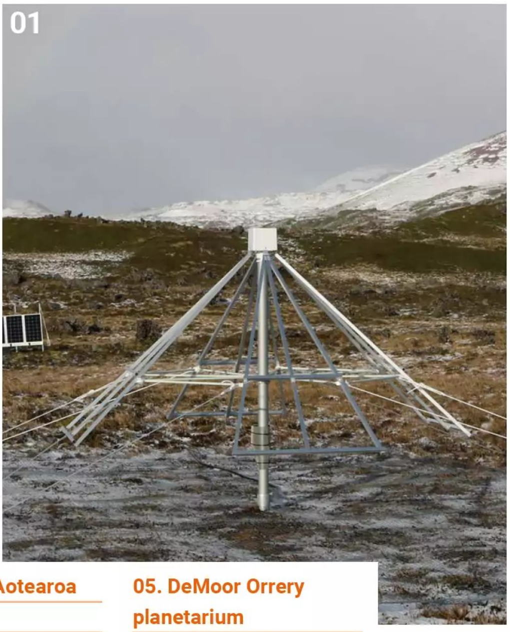
9/10

10 amazing:

Astronomy projects

Look to the stars with a Raspberry Pi

Writing about projects in space for our Extreme Projects feature this month got us thinking about other ways Raspberry Pi can be used with the vast cosmos. It's one of our favourite uses and we always love seeing what folks are doing when it comes to space observation and Raspberry Pi.



01. ALBATROS radio astronomy
Listening for signals
rpimag.co/albatros
An array deployed in the 'sub-Antarctic' and then the Arctic and is being used to map out the universe – and find out how old it is.

02. Mini Observatory
Table-top miniature
rpimag.co/miniobs
This 3D-printed observatory replica uses a Raspberry Pi 4, RP2040, and 50mm SLR lens to capture beautiful photos with a beautiful project.

03. Hubble Pi
Earth-bound telescope
rpimag.co/hubblepi
Not the one up in space, but a telescope modified to use a Raspberry Pi HQ Camera for wonderful, and customisable, astrophotography.

04. Fireballs Aotearoa
Great balls
rpimag.co/aotearoa
Make sure no meteor goes undetected around the world (and specifically New Zealand here) with this fantastic citizen science project.

05. DeMoor Orrery planetarium
Track the planets
rpimag.co/demoor
This project inspired by an 1781 orrery is a beautiful piece of art you can hang on your ceiling. It also includes astrological labels.

06. Raspberry Pi Pico solar system
Cheaper orrery
rpimag.co/picosolar
This tiny clock shows the current orientation of the solar system and, with a bit of editing, can show you how the planets aligned in the past or will do in the future.

07. Astroberry
Track the stars
astroberry.io
A system for Raspberry Pi that can hook up to computer-controlled telescopes and other similar devices for tracking celestial objects. Perfect for long-exposure photos of far-away targets.

08. Raspberry Pi Sunspot & Solar Eclipse Camera
Total coverage
rpimag.co/eclipsecam
Remember, never look directly at an eclipse, have a Raspberry Pi do it for you. Then you can convert it, like creator CapeGeek, into a solar camera for stunning shots of our very own star.

09. PiFinder
Pre-made tracking
pifinder.io
$485 / £375
Built around a Raspberry Pi, this product is already fully set up to help control your telescope (or other observation equipment) to track celestial objects.

10. Cheap All Sky Camera
Time-lapse on a budget
rpimag.co/allskycam
Using a plant pot and an acrylic dome, you too can create an all-weather camera on a tight budget that can capture meteor showers and other cosmic events.

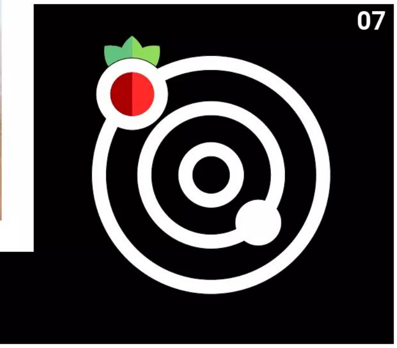
02

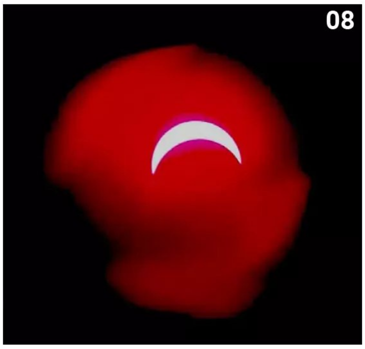
03

04

05

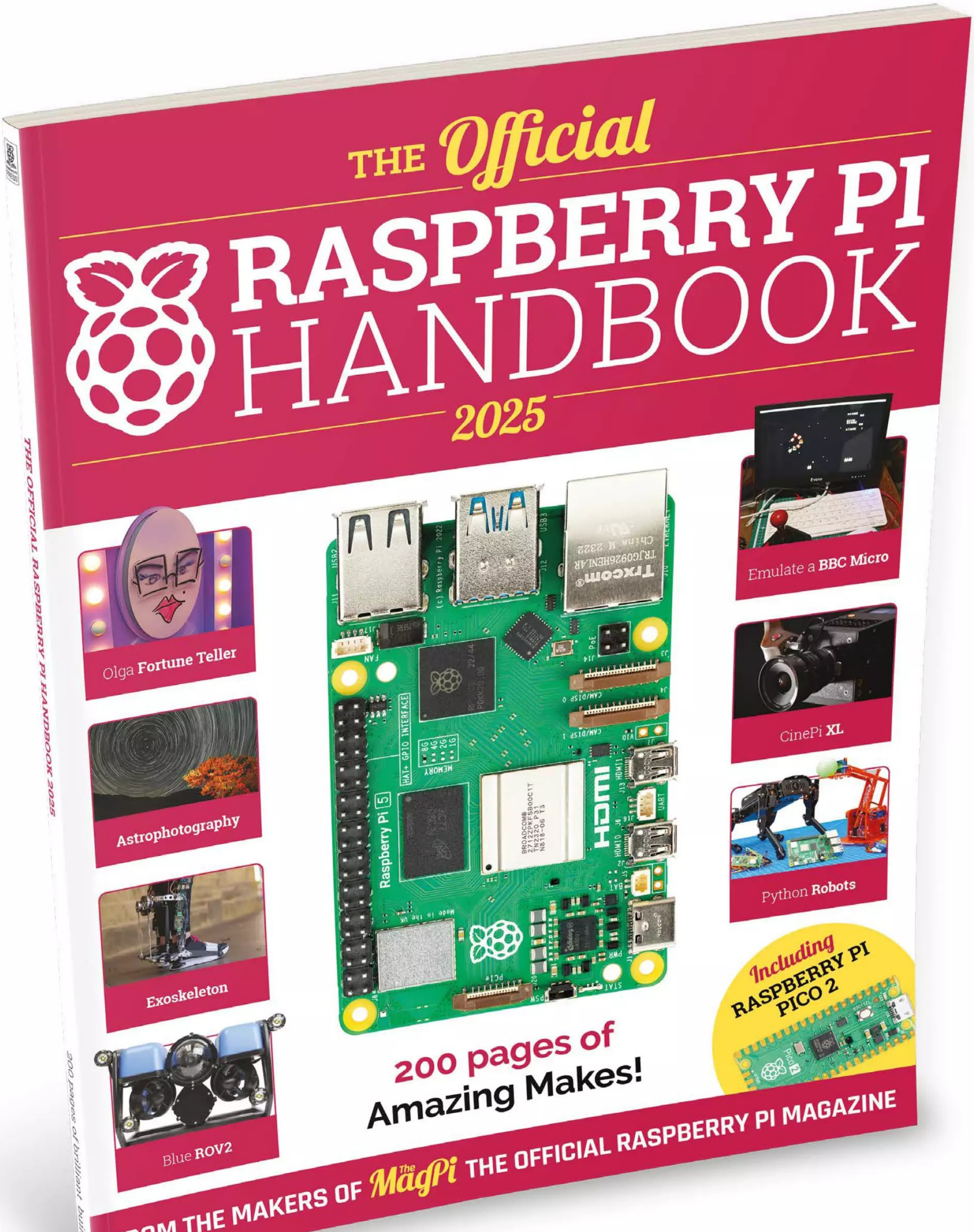
06


07


08

09

10


THE Official
RASPBERRY PI
HANDBOOK
2025
Olga Fortune Teller
Astrophotography
Exoskeleton
Blue ROV2
Emulate a BBC Micro
CinePi XL
Python Robots
Including RASPBERRY PI PICO 2
200 pages of Amazing Makes!
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI HANDBOOK 2025
200 pages of brilliant builds

Robert Wiltshire

Organiser of the Tech Jam Cornwall

Name Robert Wiltshire
Occupation Youth programme coordinator
Community role Jam organiser
URL techcornwall.co.uk

One of the most regular events in our calendar is Tech Jam Cornwall, put on by Tech Cornwall, an organisation promoting and building a vibrant tech community in Cornwall and the Isles of Scilly. Robert Wiltshire is the current organiser of this long-running event.

"I suppose like a lot of the older generations my making started in childhood with Lego, Meccano and Airfix," Robert tells us. "I still enjoy scale modelling. When we had young children, costumes for the local carnival got a little out of hand. One-third scale-model Proton Packs for Ghostbusters and a Slimer puppet, a Batmobile on an old-fashioned soap cart (which was also a land yacht sailed at the local aerodrome), and full-on Minions with a Gru head made from witheys."

Robert had done some coding in the 1990s, and when his sons came home from school one day and talked about using Scratch, it reignited his interest in coding and resulted in him starting up a Code Club in their school.

▼ Pico and Raspberry Pi 500 – things have come a long way

When did you first learn about Raspberry Pi?

A year after the Code Club started, Raspberry Pi was launched. I had heard about it before and was following the development with interest, as the intentions were closely linked with my ideals about the real need to encourage children to code.

I missed out on the first release, but received my first Raspberry from the second production run. It wasn't long before I'd linked my wife's rowing machine into a game within Scratch and then a radio-controlled car both using the original Raspberry Pi 1 Model B.

▼ Parents and children alike attend the Tech Jams

How did Cornwall Tech Jam start?

The Cornwall Tech Jam was created by a number of local business leaders. They realised that there needed to be more regular inspiration to local school children around digital and coding careers and skills. They started to do school visits and provided a software development work experience as well, the Mission to Mars. The free-to-attend Cornwall Tech Jam started in 2015 and has run every month, except in August and during the lockdowns. As part of the Cornwall Festival of Tech, the Jams were taken on tour with a different location every day for a week. Reaching new areas and more children than before.

At around the same time as the Tech Jam started, I wanted to get more children to code than the more limited numbers we could cope with at the local school. I put a message out and Tech Cornwall suggested I come along to their event. And from there it's almost history. I kept attending and learning from the professionals who were helping to run the Cornwall Tech Jam. Then one day they put out a call for a new Education Outreach Officer to take the Tech Cornwall call for more children to learn to code to schools all around the county. And to run their work experience. I applied and was lucky enough to get the job.

> I wanted to get more children to code than the more limited numbers we could cope with at the local school

▼ Even older students enjoy learning about coding

What have been some of your favourite moments of Tech Jam?

A mother came along with her son and made it perfectly clear that she didn't know how to code and wasn't at all interested in learning herself. Her son was given the task to write the code to make a micro:bit remote control for our Martian Rover. When he finally got the rover to move under his control, he was asked why it only went forwards and didn't turn – despite him trying to make it do so with the micro:bit in his hand. His mother instantly started hopping around saying, "I know, I know!" When asked why she thought it wasn't working, she quite rightly pointed out that his code hadn't included the instructions to do that yet. She was hooked!

[Another time,] a young girl attended our Tech Jam and really took an interest in coding in Python on a Pico for one of the new rovers with Mecanum wheels. She spent the entire session developing the route the rover would take. When it was time to leave, her older brothers came in and she was very delighted to show them what she had been doing and how she did it. They were very impressed by her work and a little disappointed that they hadn't decided to come along themselves.

Maker Monday

Amazing projects direct from social media!

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

Follow along to #MakerMonday each week over on our various social media platforms!

01. Hmm, we need to ask the engineers if they put any safeguards in to prevent spider robot rebellion

02. Aki is 100% correct: it looks very cute (and it's practical!)

03. Ooh, a lovely cyberdeck-esque idea that Mike Horne is working on

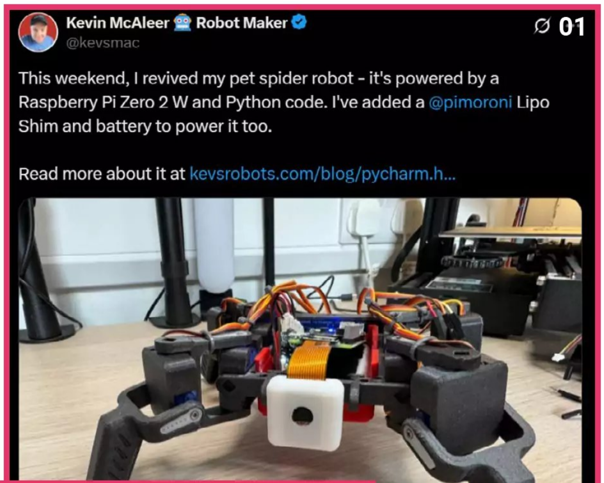
04. We never knew we wanted this, but Nick is completely right: we yearn for it

05. Use your brain to navigate your tabs. Pray you don't have 100 open

06. A lot of batteries for one robot – we hope the obstacle avoidance goes smoothly

07. More great code work from the Penguin Tutor himself

08. Is this to a model train what a deconstructed Michelin star meal is?

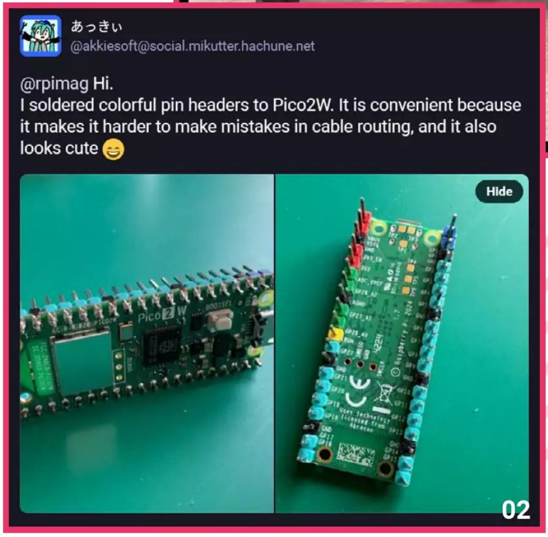


Kevin McAleer 🤖 Robot Maker
@kevsmac

01

This weekend, I revived my pet spider robot - it's powered by a Raspberry Pi Zero 2 W and Python code. I've added a @pimoroni Lipo Shim and battery to power it too.

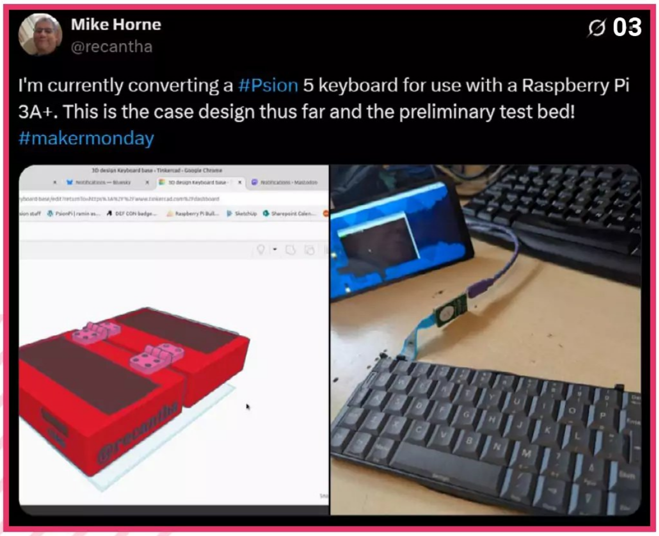
Read more about it at kevsrobots.com/blog/pycharm.h...



あっきい
@akkiesoft@social.mikutter.hachune.net

@rpimag Hi.
I soldered colorful pin headers to Pico2W. It is convenient because it makes it harder to make mistakes in cable routing, and it also looks cute 😄

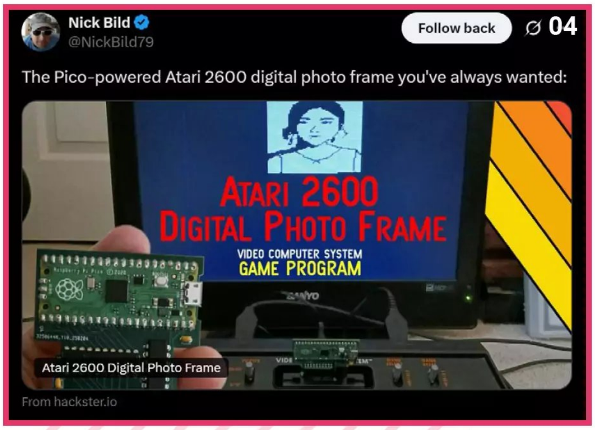
02



Mike Horne
@recantha

03

I'm currently converting a #Psion 5 keyboard for use with a Raspberry Pi 3A+. This is the case design thus far and the preliminary test bed! #makermonday



04

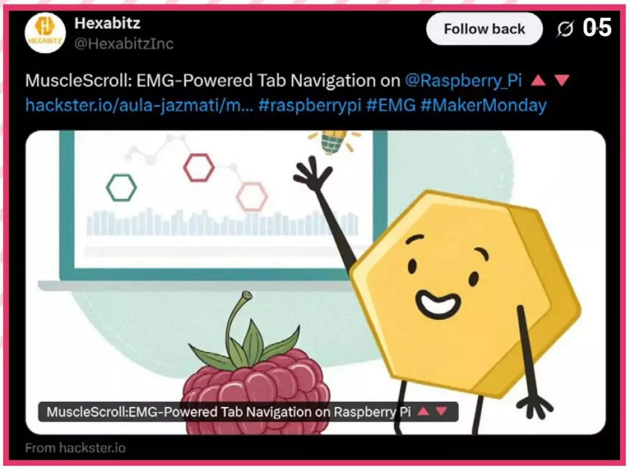
Nick Bild
@NickBild79

Follow back

The Pico-powered Atari 2600 digital photo frame you've always wanted:

Atari 2600 Digital Photo Frame

From hackster.io



05

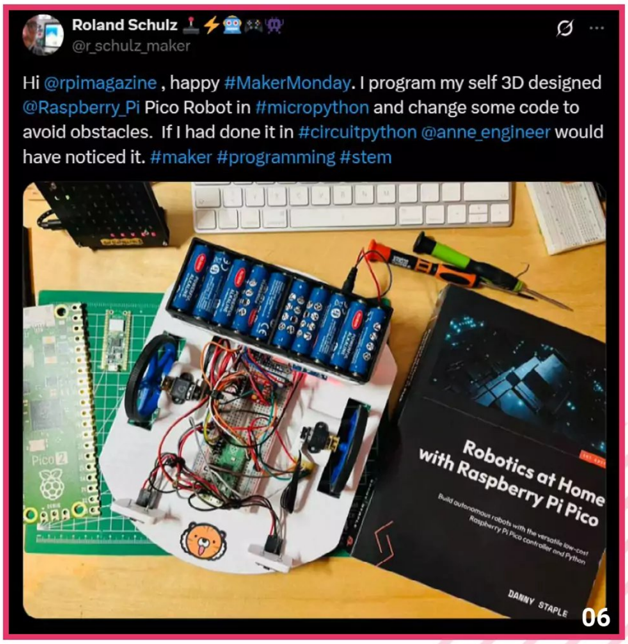
Hexabitz
@HexabitzInc

Follow back

MuscleScroll: EMG-Powered Tab Navigation on @Raspberry_Pi ▲▼
hackster.io/aula-jazmati/m... #raspberrypi #EMG #MakerMonday

MuscleScroll:EMG-Powered Tab Navigation on Raspberry Pi ▲▼

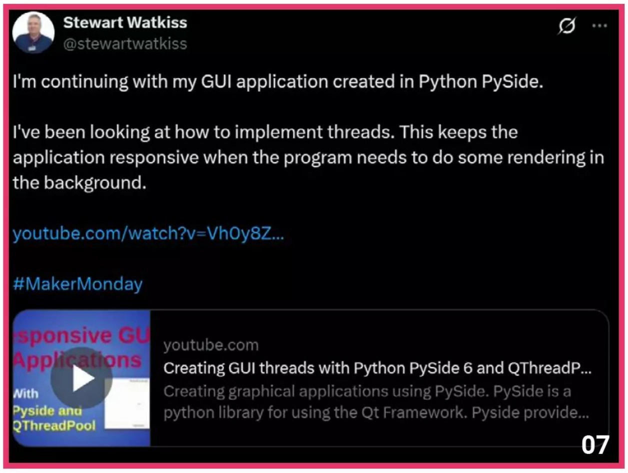
From hackster.io



06

Roland Schulz
@r_schulz_maker

Hi @rpimagazine , happy #MakerMonday. I program my self 3D designed @Raspberry_Pi Pico Robot in #micropython and change some code to avoid obstacles. If I had done it in #circuitpython @anne_engineer would have noticed it. #maker #programming #stem



07

Stewart Watkiss
@stewartwatkiss

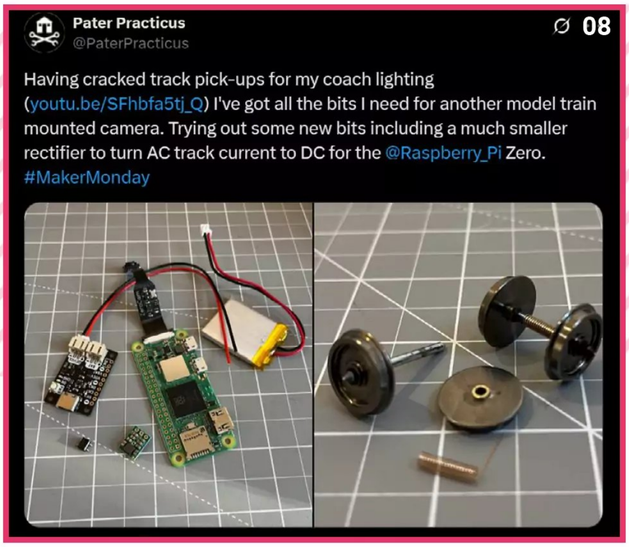
I'm continuing with my GUI application created in Python PySide.

I've been looking at how to implement threads. This keeps the application responsive when the program needs to do some rendering in the background.

youtube.com/watch?v=Vh0y8Z...

#MakerMonday

youtube.com
Creating GUI threads with Python PySide 6 and QThreadP...
Creating graphical applications using PySide. PySide is a python library for using the Qt Framework. Pyside provide...



08

Pater Practicus
@PaterPracticus

Having cracked track pick-ups for my coach lighting (youtu.be/SFhbfa5tj_Q) I've got all the bits I need for another model train mounted camera. Trying out some new bits including a much smaller rectifier to turn AC track current to DC for the @Raspberry_Pi Zero. #MakerMonday

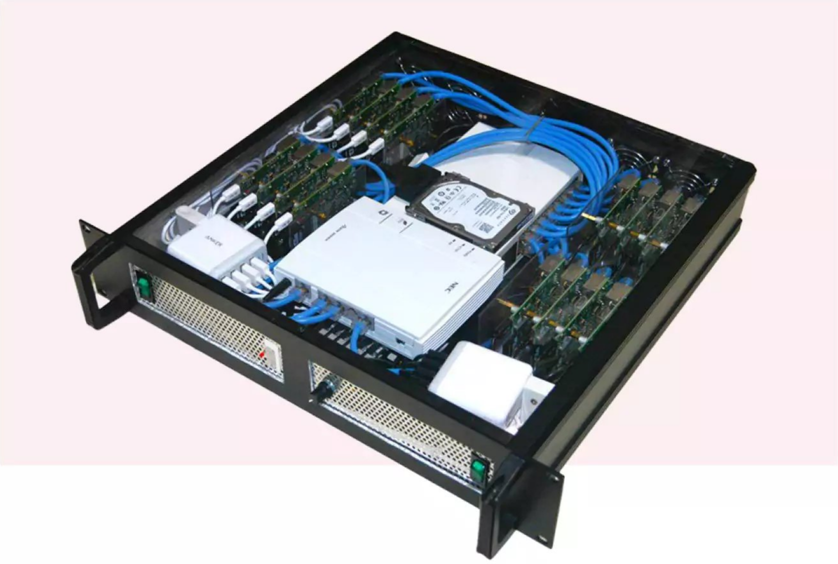
Lens – a Raspberry Pi 1 cluster

Reusing spare, decade-plus old computers for parallel processing

An email from a Japanese Raspberry Pi user caught our attention. D.M Entry Point has been creating cluster computers using old Raspberry Pi computers, including this one built with 16 Raspberry Pi 1 boards.

They're planning to build one using every numbered model of Raspberry Pi through to 5, with a Raspberry Pi 2 version already completed. We're really interested to see the performance jump between models, although we'd hazard a guess that the jump from the single-core BCM2835 on Raspberry Pi 1 and the quad-core BCM2836 on Raspberry Pi 2 might be the highest.

We are now talking to D.M. about a showcase of his project to explain how it works in a future issue.



Raspberry Piで
スーパーコンピュータ
を作ってみた！

D.M 著
D.M Entry Point 編

▲ D.M has written an ebook about his experiences building the cluster

◀ The 16 Raspberry Pi 1 cluster in all its glory

Best of the rest!

Other great things we saw this month

RP2040 macro keyboard

Using RP2040 for little macro keyboards is a fun thing that we've done ourselves – they're great as a sort of media control suite for streaming. With the added dials, we think it would be even better.

▶ rpimag.co/rpmacro153

E-Ink Weather Dashboard

While perhaps a simple project, it's also very cool, and looks great. There's a great sense of satisfaction from making something like this well, especially when it becomes a useful display piece.

▶ rpimag.co/weathdash153

Crowdfund this

Raspberry Pi projects you can support this month

Recalbox RGB DUAL 2

Recalbox is an add-on we've covered before – it allows you to hook a Raspberry Pi up to a CRT TV so that you can emulate classic console games the way they were meant to be seen – with beautiful scanlines. This new version is specific to Raspberry Pi 5, allowing even more retro and classic games to be played on an old telly

▶ kck.st/4iNbP96

AudioPlug

This cool little audio dongle for Raspberry Pi includes a huge number of audio features beyond adding a 3.5mm jack back to the board – it also has an onboard microphone, Bluetooth audio functions, a JST stereo speaker connection, and more.

▶ kck.st/4iO3L86

Your Letters

Bat Frequency

I bought my first Raspberry Pi way back when they needed keyboards attached. Whilst I was impressed at the time, my interest was mostly in microcontrollers.

The arrival of Raspberry Pi Pico and Pico W has caused me to return to the community. I even bought a Raspberry Pi 500 as a Christmas present to myself and that is now my main go-to computer.

I saw The MagPi [now Raspberry Pi Official Magazine] advertised and was attracted by the possibility of being both inspired and educated. It's a great publication… so thank you.

I have been rapidly learning how to do things with my growing collection of Picos and Pico Ws , including building an internet weather display and planetary magnetic field (Kp) display.

I have been working on monitoring sound signals and the recent article on 'Pipistrelle bat detector ' in the *Raspberry Pi Handbook 2025* caught my eye. I tried the link to try to learn about the open-source software used, but was hit by a '404'. Inspiration turned to frustration.

I looked at the back catalogue of Raspberry Pi Official Magazine tutorials for something on detecting and counting pulses/measuring frequency etc. on the Pico, but could not see anything there.

So, my request: lots more tutorials with explanations, circuits, plans and, crucially, programs please.

My experience is that there is a plethora of stuff out there for the hobbyist for the Arduino, but not so much for Pico.

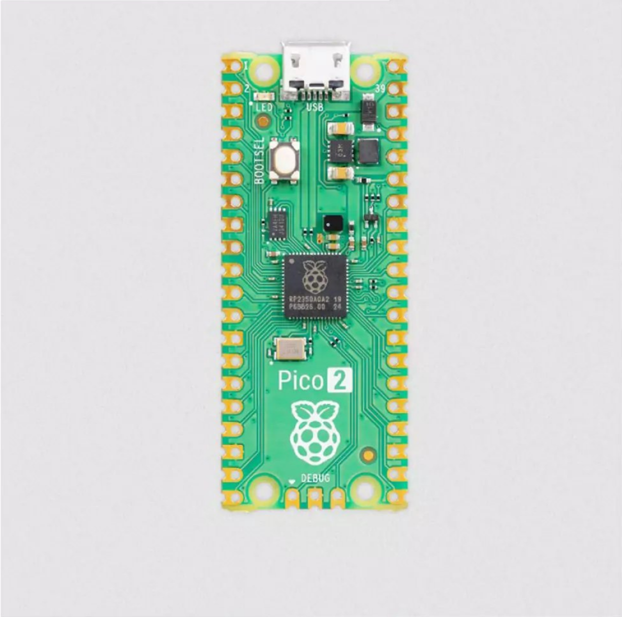
Keep up the good work.

Peter via email

We're glad to hear that you have had a lot of fun with Pico! The Pipistrelle bat detector fortunately still exists and you can check it out at **pippyg.com** – we'll update shortlinks where we can.

As for the pulse measuring – that sounds like a job for PWM, and there is a Python module called PWMCounter (**rpimag.co/pwmcounter**) that does the trick. We're always trying to get more hands-on circuits and programming in the mag, so keep an eye out for more in the future!

▼ PWM and PIO on Raspberry Pi Pico are powerful tools



Au Français

Do you ever plan to make a French version of the magazine, even if it's just a PDF?

Peter via email

In a previous lifetime, the magazine was licensed and translated into French. However, due to low demand, they stopped releasing it. As the magazine is licensed under Creative Commons, folks are free to translate it under the terms of the license.

elektor
NOUVEAU : LE MAGAZINE OFFICIEL DU RASPBERRY PI
Le magazine officiel du RASPBERRY PI
MagPi
01-2018 mars – avril
www.magpi.fr
Le magazine officiel du Raspberry Pi
INTELLIGENCE ARTIFICIELLE
Reconnaissance d'images, réseaux neuronaux…
AIY Vision Kit de Google, la caméra intelligente
KaleidoCam kaléidoscope sur RPi
Renaissance du simulateur de conduite Tomy Turnin' Turbo
Surveillance d'un jardin hydroponique (caméra et détecteur d'humidité)
Et bien plus encore :
› PI-TOP : TRANSFORMER RPI EN VÉRITABLE ORDINATEUR DE COIN DE TABLE
› RETROGAMING AVEC LAKKA : RETROUVEZ VOS JEUX PERDUS !
› SURVEILLANCE DE LA QUALITÉ DE L'AIR INTÉRIEUR
› JEU D'ÉCHECS : RÉSOLVEZ LE PROBLÈME DES HUIT REINES !
Raspberry Pi pour les débutants
· Ma première fois avec Raspberry Pi
· Où trouver de l'aide
LE SEUL MAGAZINE ÉCRIT PAR ET POUR LA COMMUNAUTÉ RPI

▲ It's fun seeing your work translated into other languages

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. Hobby-X

Thursday 1 May to Sunday 4 May
Kyalami Grand Prix Circuit, Johannesburg, South Africa
▶ **rpimag.co/hobbyx**

Raspberry Pi Approved Reseller PiShop is hosting an exhibit at Hobby-X, a trade and consumer show that showcases a variety of hobbies and crafts. At Hobby-X, visitors can explore the latest products and supplies related to their favourite hobbies and crafts, learn new techniques, and connect with other hobbyists and experts in their field.

02. Experience Raspberry Pi @ Purple Space

Sunday 4 May
Purple Space Library, Udupi, India
▶ **rpimag.co/erpps153**

Raspberry Pi enthusiasts, assemble for an exciting day of innovation and creativity! This event is the perfect opportunity to explore how Raspberry Pi inspires learners and makers to design, build, and bring ideas to life.

03. Southend Raspberry Jam Maker Meetup

Thursday 15 May
The Board Game Hut, Southend-on-Sea, UK
▶ **rpimag.co/srjmm153**

Southend Raspberry Jam Maker Meetup is a monthly event for those who are interested in building hardware and software projects using Raspberry Pi hardware and want to join a friendly group of enthusiasts and makers.

They welcome beginners to professionals. If you have any ideas for projects, talks or demos, they're especially keen to hear from you. Over 18s only.

04. Pi Force v2

Thursday 15 May
Lincoln Corners Pakistan, Karachi, Pakistan
▶ **rpimag.co/piforcev2**

Go and join the Pi Force for a day of tinkering, hacking, and learning – they will be having multiple hands-on sessions for coding and Raspberry Pi. Bring a friend, but remember there are limited seats.

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

05. Automate 2025

- **Monday 12 May to Thursday 15 May**
- **Huntington Place, Detroit, MI, USA**
- **rpimag.co/automate2025**

Raspberry Pi will have a stand at Automate 2025, Detroit, from 12–15 May. There you will be able to meet us and experience a wide array of our products, including Raspberry Pi Pico 2, our AI product range, RP2350-based solutions, and our latest industrial device: Compute Module 5.

You'll be able to see how companies around the world use Raspberry Pi to support their industrial applications and discover how Raspberry Pi can help you with your own solutions.

Official Raspberry Pi Event

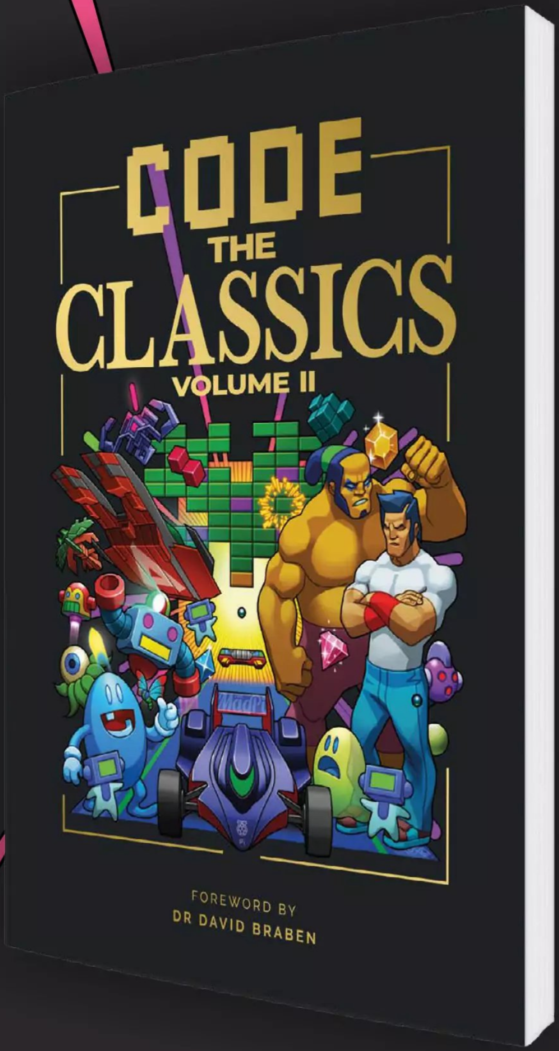
Win 1 of 5 Raspberry Pi Pico 2 Starter Kits

This incredibly sci-fi looking briefcase is actually a Pico playground, using an RP2350 at its core to let you experiment and code with 17 different sensors over 21 coding lessons. Also, it looks really, really cool.

Head here to enter:
rpimag.co/win
Learn more:
rpimag.co/pico2starter

Terms & Conditions

Competition opens on **23 April 2025** and closes on **29 May 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.



CODE
THE
CLASSICS
VOLUME II
FOREWORD BY
DR DAVID BRABEN

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 #154

Next Month

HOME AUTOMATION

Deploy Raspberry Pi around the house

On sale 29 May

PLUS!

The shiniest summer projects

A Compute Module 4 printer

Build an AI Raspberry Pi detector

Editorial

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Phil King

Advertising
Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

Design

Head of Design
Jack Willis

Designers
Sara Parodi, Natalie Turner

Illustrator
Sam Alder

Brand Manager
Brian O Halloran

Contributors

David Crookes, Rosemary Hattersley,
Jo Hinchcliffe, Phil King, Matt Lear,
Andrew Lewis, Rob Lewis, Rob Miles,
KG Orphanides, Ian Osborne, Roger Thornton

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)

Timely upgrades

Recreating a project in the age of Raspberry Pi Pico means grand ideas of silly additions

Long-time readers may be aware that, despite my age, I am a fan of the Power Rangers. Well, more specifically, Super Sentai these days, but I'm not going to sit you down and chart out the history of tokusatsu TV shows, so let's just stick with Power Rangers. As a kid I was a big fan of the Megazord (giant robots, to this day, remain cool), but these days I find the morphers more interesting: the devices the heroes use to transform into their superhero alter ego.

With roots in kabuki theatre, the whole rehearsed dance-like sequence and roll-call introduction that accompanies the transformation pleases a performance and production side of my brain, along with the interaction with the toy they're trying to sell to kids (and, apparently, 38-year-old magazine editors). These days such toys produce a cacophony of music and sound effects and vocalised yells, and while I think it's a bit much, it's something Pico can handle pretty easily. See, I managed to tie it into the theme of the mag eventually.

> *The ease at which I realised I could add a bit of Pico-power to such a creation proves just how far we've come*

It's 3D-printin' time

I recently (five days ago at time of writing) got a new smartwatch, nine years after my last one. I haven't used that old watch in a few years now, but when it was part of my daily fashion, I did design, 3D-print, laser-cut, paint, and detail up a case for it that when clipped on would make my watch look like a morpher. I felt cool wearing it. People asked me about it with great interest. I may have tried to work out a cool transformation pose, but let's not dwell on that.

This new watch has a round face, though, compared to the previous square one. So, if I want to recreate it, I need to design from scratch. And maybe I could make the fake buttons into real buttons and have it activate some LEDs and feasibly even a bit of sound. Some folks on the forums have done some testing with button-cell batteries, and a Pico will run off one – although not for long. More research is needed.

Clap your hands!

The ease at which I realised I could add a bit of Pico-power to such a creation and had already roughly mapped out the circuits and code in my head really proves just how far we've come in the last few years and even decades. As a kid, I watched *Robot Wars* and thought about how cool it would be to enter. When it came back in the late 2010s, I realised I could make a robot for it. Now I'm just casually thinking about dropping a Pico into a custom watch case because I can. Also, Toby, if you're reading this, I think this time I'll be using a resin printer, so be prepared for a million questions.

Rob Zwetsloot – Author

Rob is a Features Editor on Raspberry Pi Official Magazine and owns far too many henshin items.

rpimag.co

PiKVM

Remote control redefined

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

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

PiKVM V4 Plus

The most feature-rich edition

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

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

Available at the main Raspberry Pi resellers

core ELECTRONICS
PiShop.ca
RPishop.cz
Welectron.
elektorstore
PIMORONI
Pi-Shop.ch
The PiHut
PiShop.us
CloudFree
KUBII
Cytron Marketplace
ROBU.IN Your ideas, Our Parts
NETTOP COMPUTERS & ELECTRONICS
netsket mall
DIREKTRONIK
RaspberryPi.dk

HiPi.io

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

List of official resellers by country: