Fun in the sun with **heat-defying summer projects**

Use AI to **identify your Raspberry Pi**

Building a **hydraulic smart lift**

Raspberry Pi

Official Magazine
#154 | June 2025

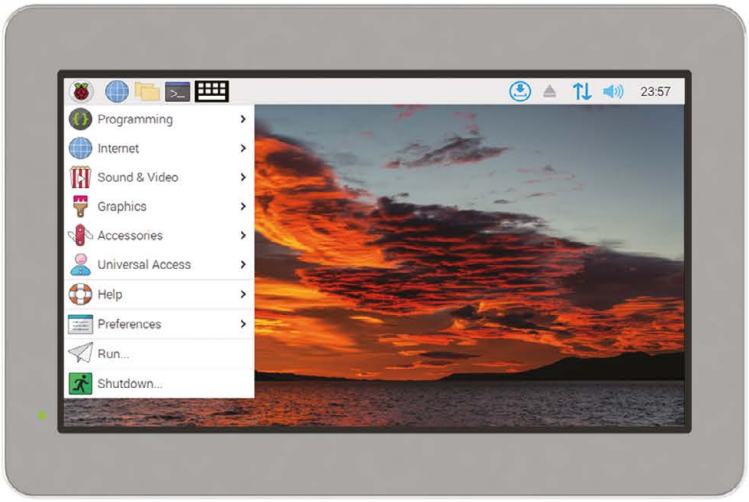
Home Automation Made Easy

Build a **smart house** with Raspberry Pi

06 **£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
This month Lucy has been training AI to recognise Raspberry Pi. You'd think it'd know by now.

rpimag.co

Back in the neon-soaked 1980s, the vision of the future of computing was very different.

We always knew that computing would be central to our lives, but the general idea was that computers would be more like plumbing or gas: integrated into the home as essential utilities.

This is within our reach right now! We can put computers to use and build an incredible responsive home of the future (that is just as smart as how we imagined it would be).

This month we're going to put Raspberry Pi into the heart of your home with our guide to Home Automation: from smart sockets, to wireless light bulbs, and network-connected thermostats and cameras. The great thing about Raspberry Pi is that it can control everything and you're in charge. That means you don't end up giving away all your privacy to a giant technology corporation.

So pick up this month's magazine and take a good look around the house. It's time for us to smarten up our home.

Lucy Hattersley – Editor

Contents

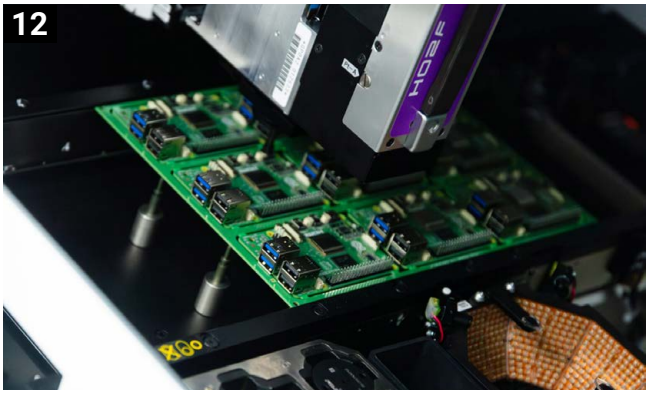
World of Raspberry Pi

010 New pricing for Compute Module 4

012 Raspberry Pi's new soldering process

014 Raspberry Pi OS updated

018 Subscribe to Raspberry Pi Official Magazine



Project Showcase

020 Signal Sentinel

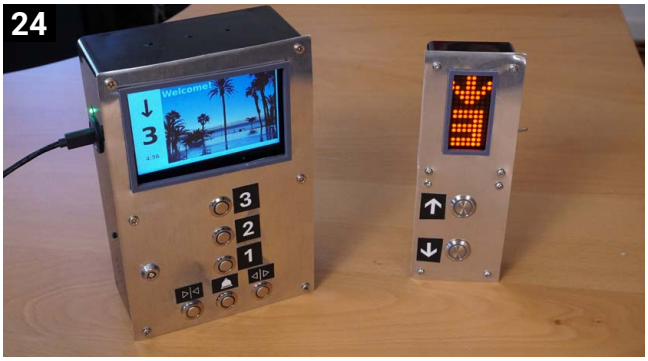
024 Raspberry Pi Elevator

028 Pepper Pi

032 Atari 2600 Digital Frame

036 PiEEG

040 IINTS Insulin Pump



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

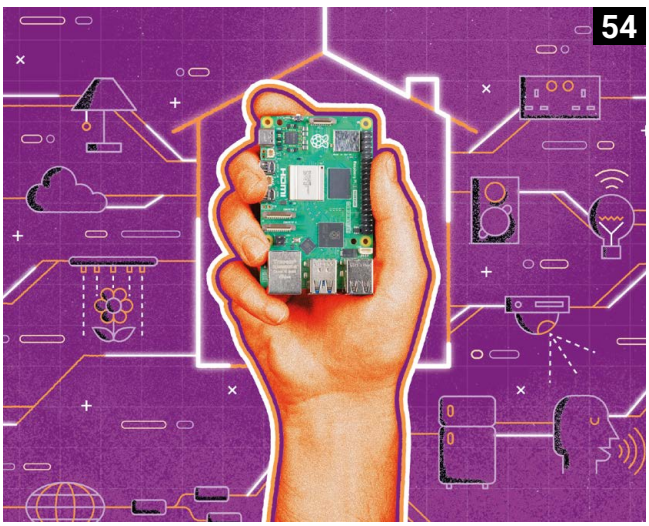
046 Baby North case

048 MIDI Blaster

050 Sony Watchman cyberdeck

052 3D print showcase

Feature



054 **Home Automation Made Easy:** Control your house with a Raspberry Pi

Tutorials

068 Build a Raspberry Pi classifier

076 Unusual tools: cutting mat glue

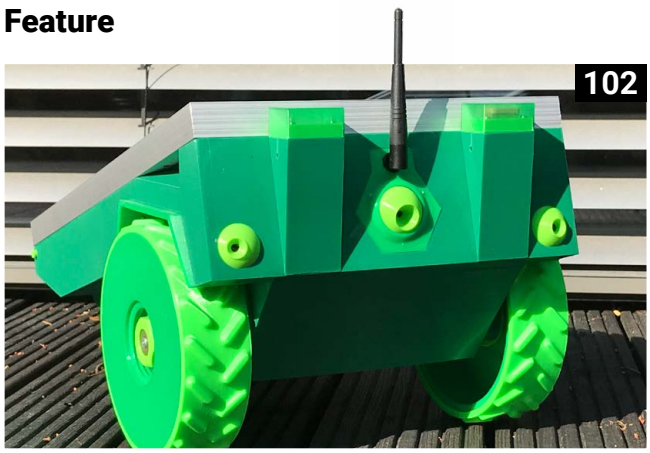
078 Meshtastic LoRA mesh network – part 2

084 MIDI on Raspberry Pi Pico

092 Sewing with LEDs and Pico

098 RISC OS for Raspberry Pi – part 2

Feature



102 **Summer projects:** Get outside with these Raspberry Pi builds

PCBWay

FULL-SERVICE ELECTRONICS MANUFACTURER

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

CUSTOMIZED TECHNICAL PARTS & PCB MANUFACTURED

WHY CHOOSE PCBWAY?

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

Hassle-free ordering

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

On-time shipping

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

Quality assurance

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

Customer support

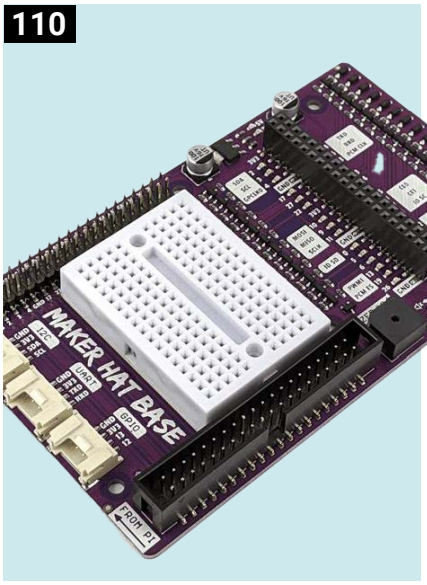
The customer service teams work inshifts to provide 24-hour support. You can always contact a live customer service person.

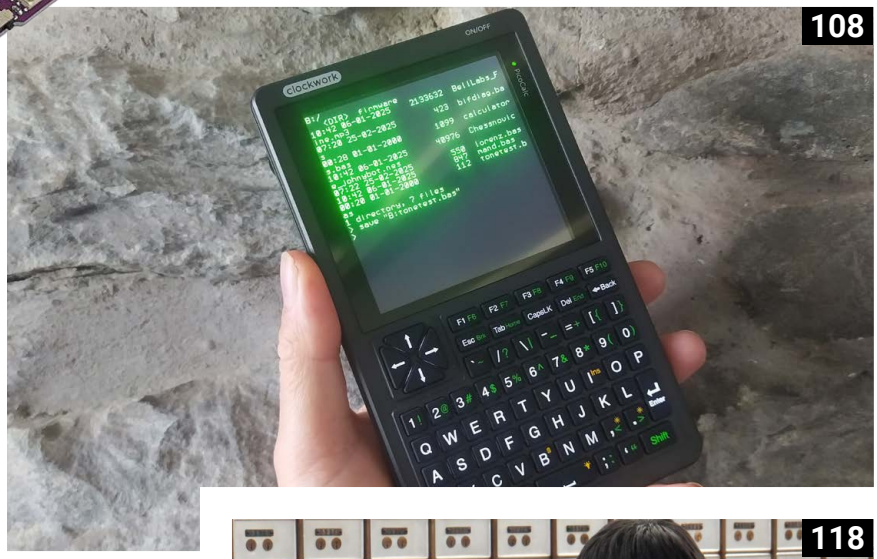
Scan the QR code for more details!

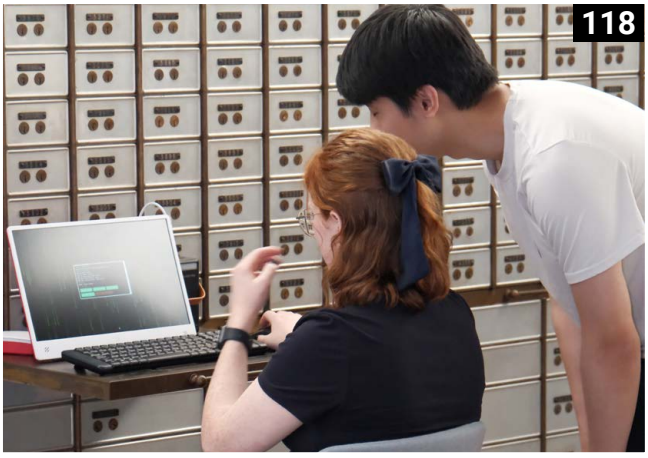
CONTACT US:

www.pcbway.com

service@pcbway.com







Reviews

108 PicoCalc

110 **Only the best:** Electronics prototyping boards

116 **Ten amazing:** tiny Pico projects

Raspberry Pi Community

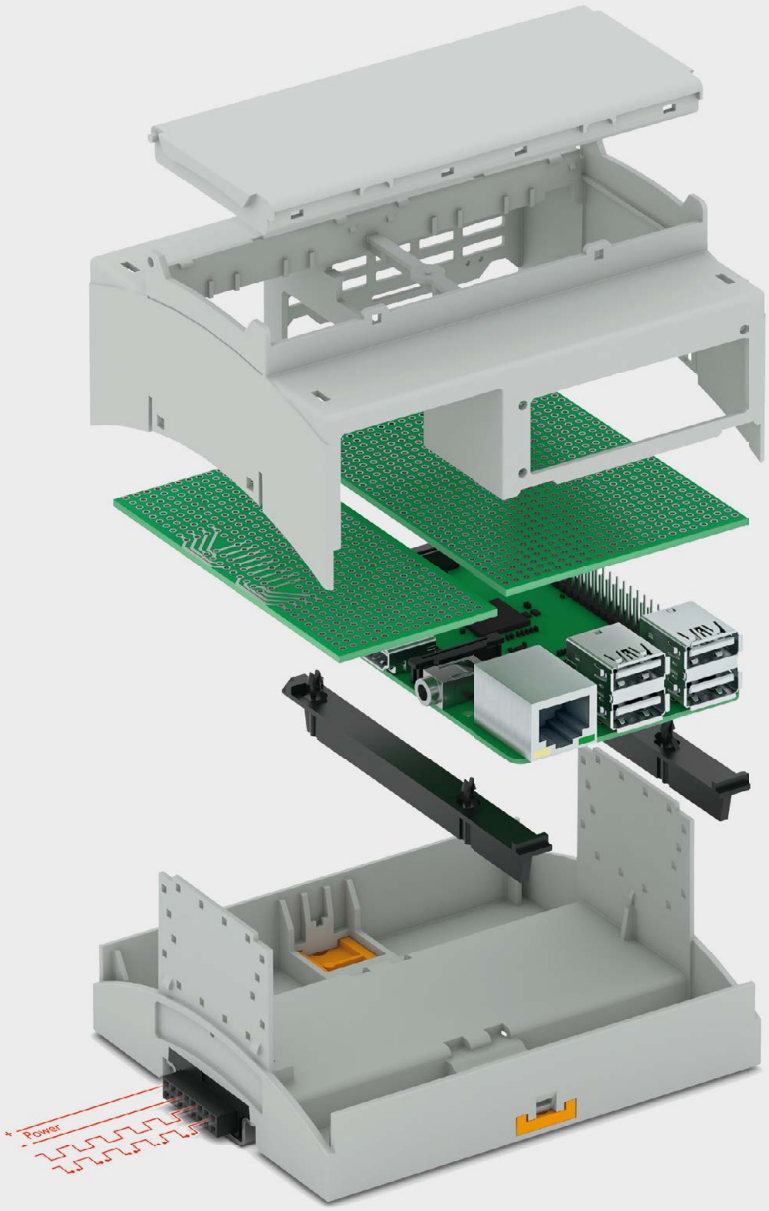
118 This Month in Raspberry Pi

122 Your Letters

124 Community Events Calendar

Competition

Win 1 of 10 KIWI KVM kits



Transform your Raspberry Pi experience

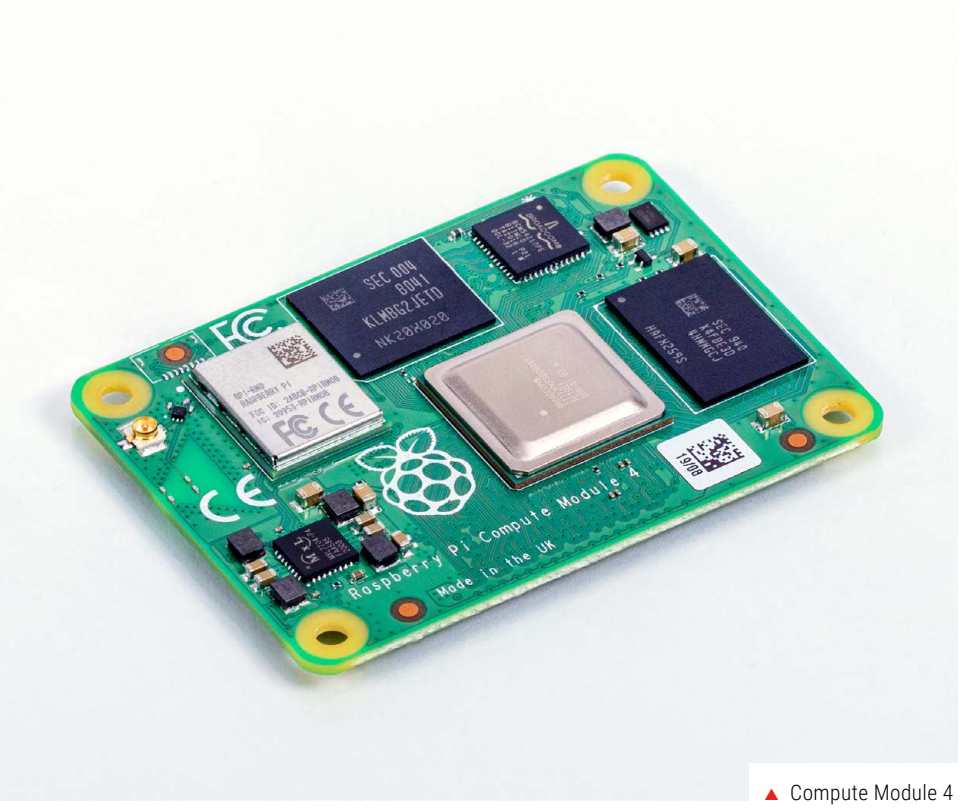
RPI-BC modular housings

Discover the ultimate solution for housing your Raspberry Pi computers with the RPI-BC Modular Housings. Precision-engineered enclosures ensure compatibility with DIN 43880 standards and offer versatile mounting options for seamless integration. Experience superior quality and functionality with RPI-BC enclosures for various installation needs.

For additional information, scan the QR code

New pricing for CM4

Lower prices for 4GB and 8GB Compute Module 4. By **Dave Lee**



▲ Compute Module 4

At Raspberry Pi, our mission is to make computing accessible and affordable for everyone and for businesses at every scale, so we're delighted to announce a reduction in the price of some of the most popular variants of Raspberry Pi Compute Module 4. From now, if you buy a standard operating temperature Compute Module 4 (**rpimag.co/cm4**) from a Raspberry Pi Approved Reseller (**rpimag.co/resellers**), it'll cost you $5 less for a 4GB RAM variant, and $10 less for an 8GB RAM variant.

Broader access to a proven platform

Raspberry Pi Compute Module 4 is the cornerstone of an astonishing variety of applications, from medical equipment (**rpimag.co/biobusiness**) to energy services infrastructure (**rpimag.co/episensor**) and from concrete monitoring (**rpimag.co/concretecompute**) to retro gaming. There is a vast number of

▼ Compute Module 4 provides the power of Raspberry Pi 4 in a compact form factor for embedded applications

embedded use cases that don't require the processing heft of our new (ish) Compute Module 5; by lowering the cost of the higher-memory-density variants of its predecessor, we aim to make these projects more cost-effective, and to unlock new ones that previously weren't viable. We hope the price drop will introduce new possibilities both for businesses and for enthusiasts, helping you bring into existence products and projects we'd never even imagined.

More room to innovate, same trusted performance

Raspberry Pi Compute Module 4 continues to offer the same powerful features that have made it a favourite with designers and developers: with its quad-core Arm Cortex-A72 processor, extensive I/O capabilities, and flexible form factor, it remains the perfect choice for a wide range of applications. We hope that these new lower prices will make it a little bit easier to get your hands on the hardware you need.

CM4 new pricing structure

Part Number	Wireless	RAM	eMMC	Old Price	Price
CM4004000	No	4GB	0GB (Lite)	~~$50~~	$45
CM4004008	No	4GB	8GB	~~$55~~	$50
CM4004016	No	4GB	16GB	~~$60~~	$55
CM4004032	No	4GB	32GB	~~$65~~	$60
CM4008000	No	8GB	0GB (Lite)	~~$75~~	$65
CM4008008	No	8GB	8GB	~~$80~~	$70
CM4008016	No	8GB	16GB	~~$85~~	$75
CM4008032	No	8GB	32GB	~~$90~~	$80
CM4104000	Yes	4GB	0GB (Lite)	~~$55~~	$50
CM4104008	Yes	4GB	8GB	~~$60~~	$55
CM4104016	Yes	4GB	16GB	~~$65~~	$60
CM4104032	Yes	4GB	32GB	~~$70~~	$65
CM4108000	Yes	8GB	0GB (Lite)	~~$80~~	$70
CM4108008	Yes	8GB	8GB	~~$85~~	$75
CM4108016	Yes	8GB	16GB	~~$90~~	$80
CM4108032	Yes	8GB	32GB	~~$95~~	$85

The perfect choice for a wide range of applications

Raspberry Pi's new soldering process

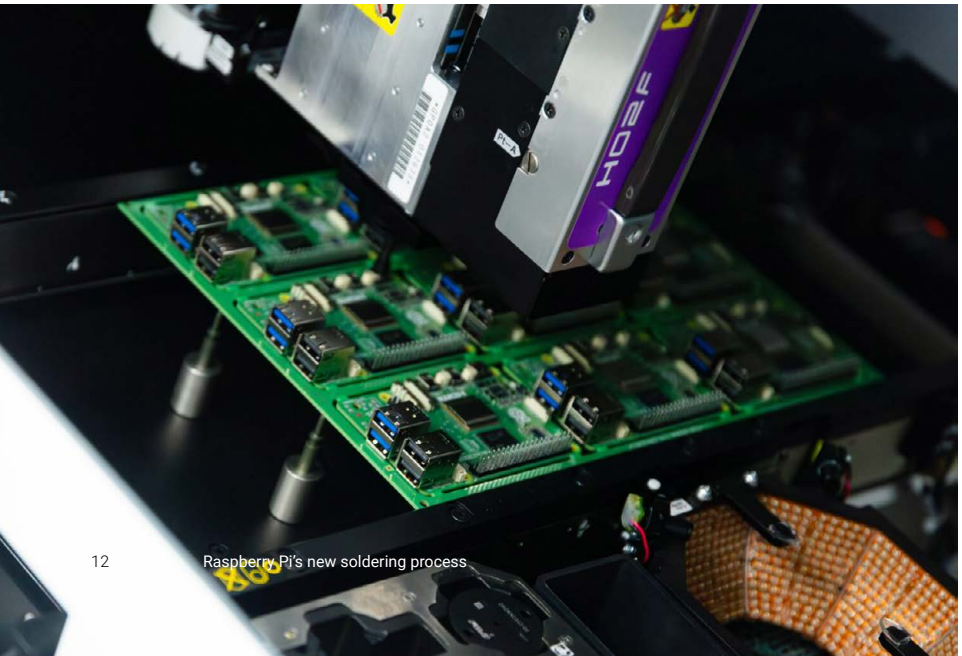
Sustainable solutions with Raspberry Pi: how intrusive reflow soldering boosted our efficiency and cut our carbon footprint.

By **Roger Thornton**

We've reduced product returns by half, cut our manufacturing time by 15%, and eliminated 43 tonnes of CO_2 emissions per year by changing the way we solder connectors onto our computers.

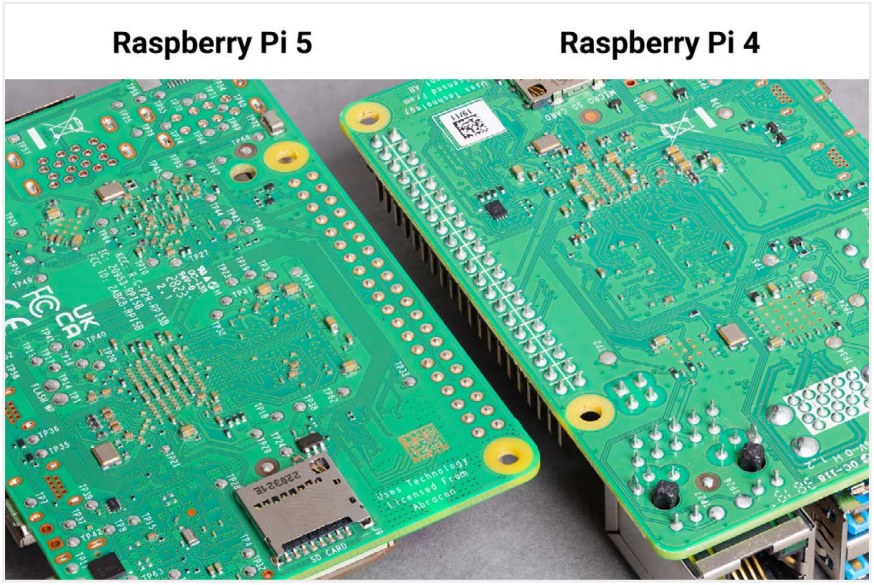
In product design and manufacture, small changes often drive big differences in environmental impact, and at Raspberry Pi we've always made sure we have a thorough understanding of the processes used to manufacture our products so that we can spot ways to improve them. In the run-up to Raspberry Pi 5, we worked with our manufacturing partner Sony (**sonypencoed.co.uk**) to implement a technique called intrusive reflow soldering, an adjustment that has improved product quality, reduced waste, and lowered our environmental impact.

▼ Raspberry Pi 5 boards being created at Sony's factory at Pencoed, Wales



Solving the through-hole bottleneck

Through-hole connectors have long been a sticking point in efficient production. They require robust solder joints made through the printed circuit board itself, meaning they can't simply be handled by the standard processes used with SMT (surface-mount technology, aka pick-and-place) machines. Because of this, we have always endeavoured to minimise the number of through-hole parts, and this type of mounting is typically reserved for connectors. In the early days of Raspberry Pi, these parts were inserted by hand, and later by robotic placement. There then followed a wave soldering



▶ A Raspberry Pi 5 with intrusive reflow soldering and a Raspberry Pi 4 with wave soldering

step – an additional process involving a molten solder bath which the boards pass through. This added time, cost, and complexity to our production line.

Thanks to our work with Sony, we have eliminated all of the through-hole–specific actions from our manufacturing processes. With intrusive reflow, we can now place through-hole connectors using the same pick-and-place machines we already use for surface-mount parts; this means there is no longer any need for bespoke robotics, or for an additional soldering stage. Over a series of trials, we perfected component placement, tweaked the solder paste stencil, refined the PCB layout, adapted the connector design, and adjusted the inspection process. We validated the results against our stringent quality control, successfully achieving all the standards that we set ourselves.

A new manufacturing standard for Raspberry Pi 5 and beyond

This became the production process that we have used on all Raspberry Pi 5 computers, and we're working to roll it out to the manufacture of our earlier models too.

The change delivered a marked increase in product quality, with a massive 50% reduction in product returns. It also increased the speed at which products are manufactured by 15%. Work-in-process (WIP) inventory was eliminated entirely, as there is no longer any break in the production line, all the way from bare boards coming into the factory to finished Raspberry Pi computers being packaged into boxes. And removing a set of machinery – the selective solder bath – from the production line reduced the CO_2 output of our production by 43 tonnes per year.

Smarter manufacturing, smaller footprint

Raspberry Pi's move to intrusive reflow soldering shows how targeted changes in manufacturing can lead to significant improvements in sustainability. By cutting energy use, eliminating wasteful intermediate steps, and improving product quality, we're reducing our environmental impact while making our production more efficient. It's one of many ongoing efforts to manufacture more responsibly and more sustainably.

Targeted changes in manufacturing can lead to significant improvements in sustainability

A new Raspberry Pi OS release

New edition of Raspberry Pi operating system out now. By **Simon Long**

Raspberry Pi has published an improved new version of Raspberry Pi OS (rpimag.co/raspberrypios). This is our recommended (and free) operating system for all Raspberry Pi computers. The new version is now available for download from the Raspberry Pi website, from the Raspberry Pi Imager tool, and you can upgrade directly from inside Raspberry Pi OS.

As many of you already know, Debian Linux works on a two-year release cycle – every odd-numbered year, a new major version is released, and it being 2025, there will be one in the next few months.

This is probably the final release of Raspberry Pi OS based on Debian 'bookworm', before Debian 'trixie' is released in the summer of 2025. The last full release Raspberry Pi made was back in November 2024, and there have been quite a few changes since then, so here's a summary of the most important.

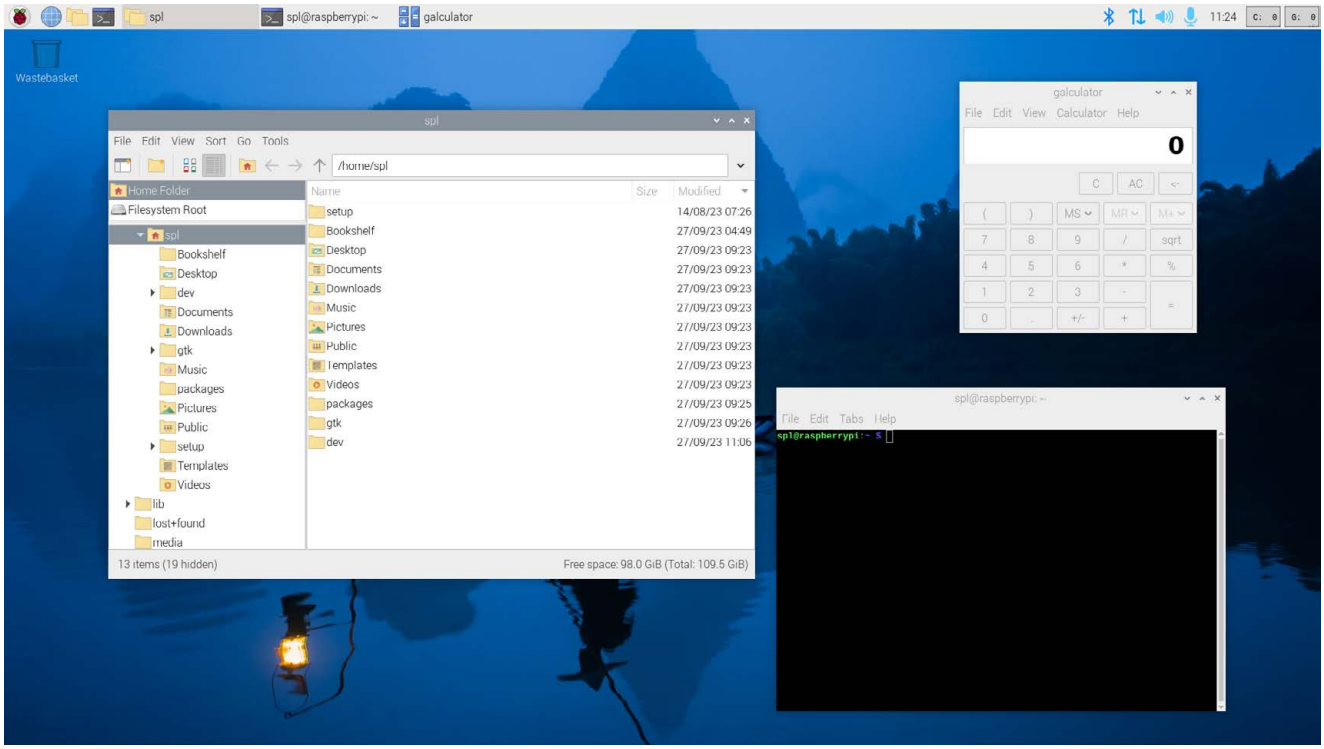
Screen locking

We have installed a modified version of the swaylock screen-locking application that helps you preserve privacy when you're away from your Raspberry Pi. Anyone who has used swaylock will be familiar with its somewhat minimal interface – when you lock the screen, you just get a completely white screen with no indication of what has happened or what you need to do. We felt this was a bit unhelpful, so we've added a custom front end which gives a bit more feedback as to what is happening and what you need to do to unlock it again!

You can now lock the screen by pressing **CTRL+ALT+L**, or by choosing 'Shutdown...' from the main menu and selecting Lock Screen in the dialog. You'll then see the lock screen, with a password entry box.

Type in your password, hit **ENTER**, and the desktop should return.

There have been dozens of the usual small tweaks to fix bugs, add new translations, and just generally tidy things up



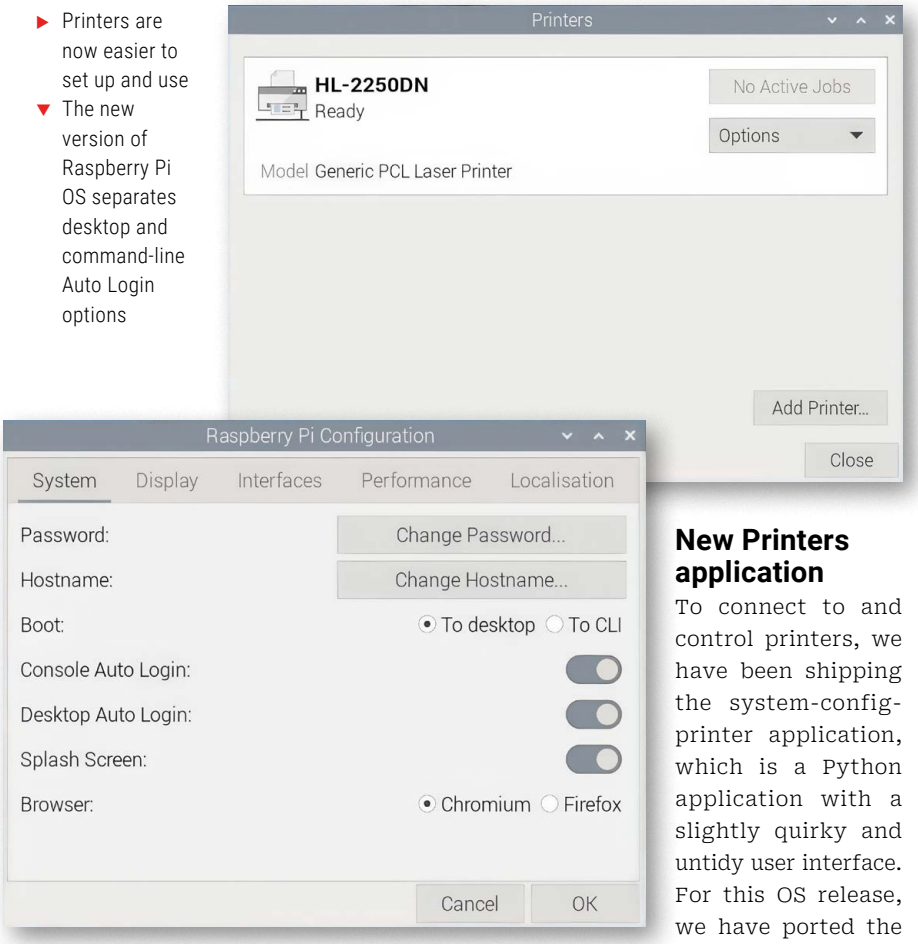
▲ The Raspberry Pi OS desktop interface

Auto login options

In Linux desktops, it is usually possible to access a command-line console (known as a TTY) by pressing **CTRL+ALT** and one of the function keys from **F1** to **F7**. We have always set up Raspberry Pi Desktop so that if you boot to the desktop and enable auto login, then the TTY on **CTRL+ALT+F1** is also automatically logged in. If you use the screen lock, this gives a potential security hole, as the TTY switches are not disabled when the screen is locked.

What this means is that if you lock the screen, you should need to enter a password to be able to access the Raspberry Pi desktop again. But if a TTY is also logged in, someone can just hit **CTRL+ALT+F1**, switch to the logged-in TTY, and gain access to the computer.

In order to prevent this, we have now separated console and desktop auto login options. On a new image, both console and desktop are automatically logged in, but if you want to prevent someone using this to get around the screen lock, we recommend turning off console auto login. There are now controls for this both in Raspberry Pi Configuration and in raspi-config.



- Printers are now easier to set up and use
- The new version of Raspberry Pi OS separates desktop and command-line Auto Login options

New Printers application

To connect to and control printers, we have been shipping the system-config-printer application, which is a Python application with a slightly quirky and untidy user interface. For this OS release, we have ported the printer control plug-in from the GNOME desktop control centre into a standalone Printers application (along with fixing a few of GNOME's more puzzling user interface decisions...). The new application can be found in the Preferences section of the main menu and should hopefully make managing printers a bit more intuitive.

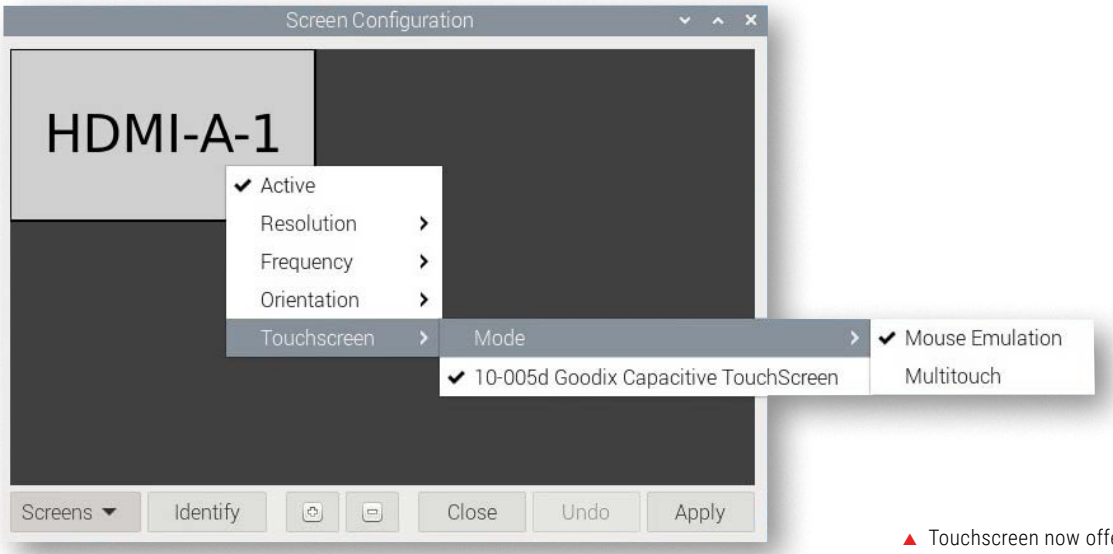
Better touchscreen handling

Touchscreen handling in Wayland is relatively new and sometimes doesn't do everything you might hope. We hit a problem when we first moved to Wayland in that some touch features, like the ability to double-click, were simply not available, and we had to find a workaround.

What we did was to enable mouse emulation by default, whereby touchscreens just pretend to be mice – when you tap the touchscreen, it generates a mouse click instead of a touch, and if you tap it twice, it generates a double-click. The problem with this was that it meant that touch-specific features, like swiping the screen to scroll, were disabled, and some people noticed their absence.

For this release, we are making it easy for touchscreen users to choose whether they want mouse emulation behaviour, or native touchscreen behaviour. There is a new menu under the 'Touchscreen' section of the context-sensitive menu in Screen Configuration.

The main disadvantage of no longer using mouse emulation is that it isn't possible to double-click by tapping the screen twice, and this makes navigation in the File Manager rather difficult. There are a couple of workarounds specific to the File Manager: you can enable 'Open files with single click' in the File Manager preferences, or use a tap-and-hold to open the context-sensitive menu and then choose 'Open'.



▲ Touchscreen now offers mouse emulation behaviour, or native touchscreen behaviour

Hopefully, at some point Wayland touchscreen support will be mature enough that it is no longer necessary to offer this option, but in the meantime, this lets users choose their preferred behaviour.

Other changes

This release is running version 0.8.1 of the labwc Wayland window manager – this is a couple of releases behind the very latest version, but has had a lot of testing and is very stable. We'll be updating this to a newer version in the near future. We are also now running on version 6.12 of the Linux kernel for this release.

The Squeekboard virtual keyboard for use with touchscreens has been modified to allow users with multiple monitor configurations to choose the screen on which it is shown – the relevant option is on the Display tab of Raspberry Pi Configuration.

Unfortunately, due to changes made by the authors of the Chromium web browser, it is no longer possible to pre-install the uBlock Origin ad blocker. As a result of this, from this release onwards, we are pre-installing the slightly less full-featured uBlock Origin Lite.

A lot of work has gone into optimising the startup of the wf-panel-pi application used to create the taskbar in Wayland, and this has resulted in a noticeable improvement in the time taken for the desktop to start after your Raspberry Pi is booted.

In another performance optimisation, we have stopped using the zenity tool to create prompts and dialogs from the command line, and have written a more efficient tool of our own, called zenoty – this saves installing some packages which were slowing down startup.

There have also been a lot of changes under the hood aimed at making maintenance of the desktop more straightforward and easer to manage going forward into trixie, but they shouldn't (hopefully) be noticed by most users.

And of course there have been dozens of the usual small tweaks to fix bugs, add new translations, and just generally tidy things up.

How do I get it?

As is usual, you can do most of the update automatically via apt. Just open a terminal and type:

```
sudo apt update
sudo apt full-upgrade
```

While the upgrade is in progress, you may get prompts asking you to confirm changes to configuration files; just answer **Y** for yes to these.

If you want to write a fresh image to an SD card or other media, visit our software page (**raspberrypi.com/software**), where you can download Raspberry Pi Imager – the most straightforward way for most users to prepare a new SD card – or browse download options to install manually.

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.

Signal Sentinel

Harnessing Raspberry Pi 5's power to detect and deflect radio jammers made for a great research project, hears **Rosie Hattersley**

Maker

Josh Perryman

Cybersecurity student Josh scratched a long-standing itch to experiment with Raspberry Pi for his final-year project.

rpimag.co/signalsentinel

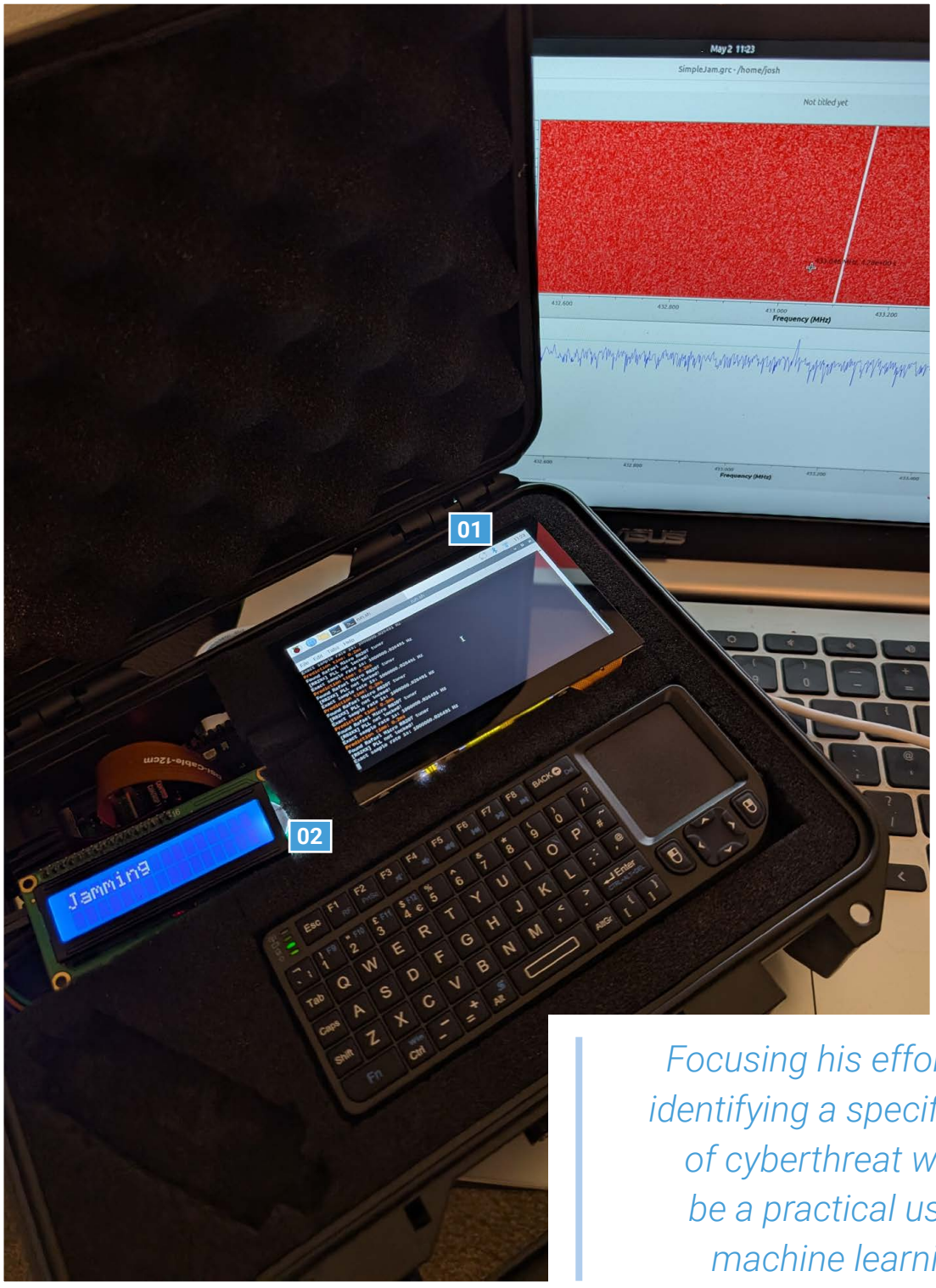
Cybersecurity gets plenty of media coverage, not least because most of us have become so dependent on electronic communication and internet connectivity that disruptions can cause major havoc. In splat-a-rat fashion, one type of threat gets beaten back, only for another to rear its head.

Solent University final-year student Josh Perryman reasoned that focusing his efforts on identifying a specific type of cyberthreat would be a practical use of machine learning, and also make a great excuse to finally try Raspberry Pi's capabilities in this field for himself. His Signal Sentinel can detect jamming attacks on wirelessly connected devices including IoT hardware and even cars – an increasingly prevalent target for thieves (**rpimag.co/bbccartheft**). Josh envisages his Signal Sentinel being integrated with other systems which mitigate or prevent the attack.

Demonstration model

Having trained and worked for a decade as a chef, Josh says he "didn't have much of a technical background, just some basic Arduino robotics projects" and a Raspberry Pi weather station build as part of his course. However, he was interested in embedded systems and IoT, had always wanted to build a project using Raspberry Pi and his final-year project gave him the perfect excuse.

The plan was to get Raspberry Pi to classify and passively detect radio frequency jamming signals. Josh would build a proof-of-concept model to demonstrate how effectively and efficiently machine learning could be used for the purpose. He selected Raspberry Pi 5 for its processing power and with an eye on planned future projects involving the AI+ HAT and Camera Module NoIR. Keen to keep build costs down, he began with an unbranded LCD that seemed to have similar capabilities to Waveshare's.



Warning!

Detection only

The use of RF (radio frequency) jammers in the UK is strictly prohibited under the Wireless Telegraphy Act 2006.

rpimag.co/ofcomradio

01. Signal Sentinel passively detects radio frequency jamming attempts

02. An ML-trained 4GB Raspberry Pi 5 analyses real-time traffic provided by an RF-SDR dongle

Focusing his efforts on identifying a specific type of cyberthreat would be a practical use of machine learning

▲ Some signal jammers are sold as kits hidden in flight cases; Signal Sentinel is designed as a portable setup to detect them

However, Josh soon found out that this was a false economy: "the unbranded screen was quite cheap and caused some electrical issues, causing Raspberry Pi to turn off when [it] was connected. Changing this for a better-quality screen fixed this issue.

Perfect fit

The Signal Sentinel was easy to build, with little call for custom parts (he bought most items from The Pi Hut, where he is a regular customer) and a flight case that could house all the components. Next, Josh mapped out areas where the RTL-SDR dongle could fit through when in use. This was critical since it would provide live radio transmission data. He also carved a channel into the case's foam lining for the power cable, plus sections to snugly house the Waveshare screen, keyboard, and Raspberry Pi 5.

As a confident Python coder with experience of using MQTT, Josh's main technical challenge for the RF jammer was collecting a diverse enough database. It needed to include "a good variety of jamming signals and 'safe' signals such as background noise, key fobs, RF devices, and radio stations". Collecting sufficient data accounted for the large majority of the four-month project. The hardware build took only a few hours.

Josh used Ubuntu VM on his laptop for data collection, prototyping, and testing. Once he was satisfied that his machine learning models were effective, he created a simple terminal interface for the Signal Sentinel in Raspberry Pi OS and attached a miniature keyboard. The main menu monitors whichever frequency is selected by the user, while a smaller LCD indicates whether the airwaves are currently 'safe' or 'jamming', based on the model's prediction.

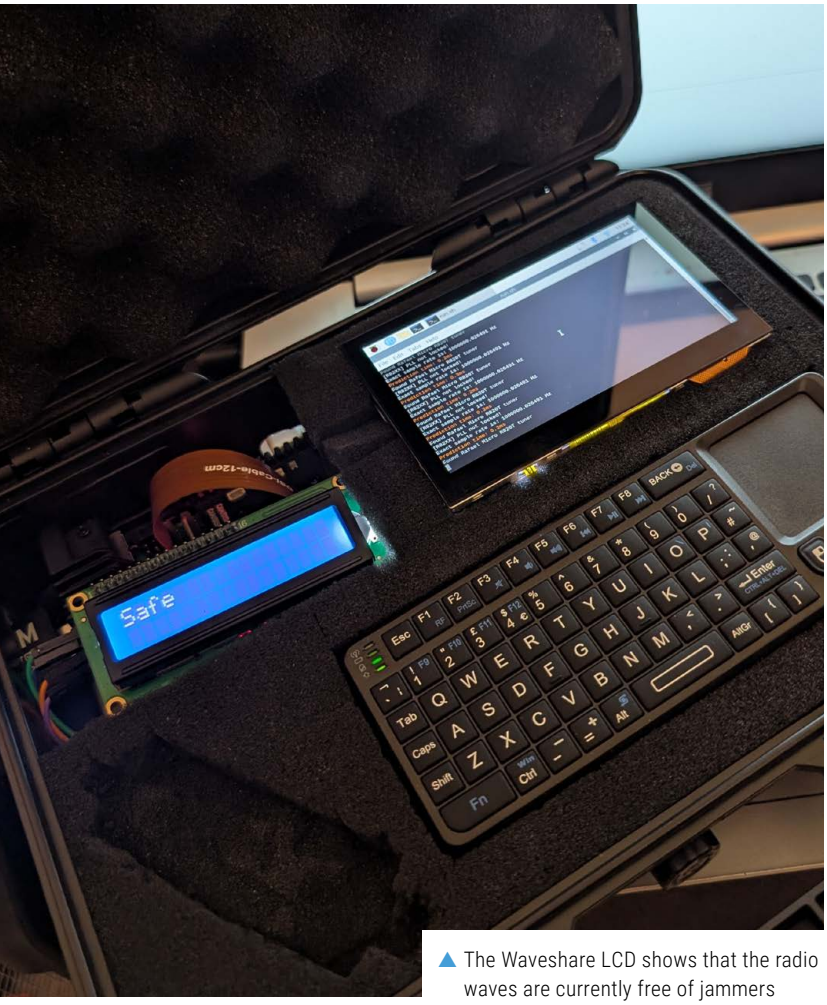
Building experience

Although the SDR dongle worked well, Josh suggests someone with a keen interest in radio wanting to replicate his build might prefer the better bandwidth and range offered by a HackRF One or Pluto SDR, though these are larger and not necessarily ideal for fitting inside a small flight case. However, the Signal Sentinel is best used in a lab environment where jamming signals can be contained to prevent accidental interference. Having submitted his final dissertation, Josh intends to refine his model "to detect jamming attacks which aren't just noise and signal blocking, but more sophisticated attacks which would otherwise go unnoticed in their current stage".

Quick FACTS

- Josh wanted to show small embedded systems were useful for ML
- He was keen to build a portable, use anywhere, device
- The lab environment also contained accidental signal interference
- In the lab, he could create signal jams to prove his device worked
- The Signal Sentinel cost just £118 excluding Raspberry Pi 5

▶ Remote unlocking has left cars vulnerable to theft using RF jammers, the Perfect Jammers website warns



▲ The Waveshare LCD shows that the radio waves are currently free of jammers

Signal seeker

1. You will need an RF-SDR dongle, Raspberry Pi display, I2C LCD, Raspberry Pi, mini keyboard, and a power supply. Carve spaces for each in the protective foam of a flight case and cut a hole in its side for a power cable.

2. Cut a hole to accommodate the SDR-dongle so it can be used from inside the case. Install drivers for the RF-SDR dongle from the libstlsdr library. Attach the I2C LCD to Raspberry Pi's GPIO pins.

```
--------------------------
Choose how long to capture for: (seconds)
Time in Seconds: 5
Is this signal Safe or Jamming?: Jamming
Capturing on frequency 433050000.0 MHz for 5 seconds
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Frequency: 433050000.0
Signal to Noise ratio: 0.0003033729735761881
Magnitude: 0.7356888055801392
Phase: -45020.05859375
Entropy: 4.654413822285635
Power Spectral Density: -7.339263916015625
Amplitude: 0.36507144570350647
RMS: 0.1651252806186676
dBm: -17.64
Data saved to Training_data.csv
```

3. Download and install Josh's dataset and code from his GitHub page: **rpimag.co/signalsentinel**.

Raspberry Pi Elevator

Here is an uplifting replication. **David Crookes** gets the lowdown

Maker
Alan Boris
Alan is an edge AI enablement lead at **balena.io** who loves Raspberry Pi, home automation, and building projects.

rpimag.co/elevatorvid

Warning
Mains Electricity
Electricity is dangerous. Be careful when working with electrical projects that involve mains electricity and relay devices to control high-power loads.
rpimag.co/electricalsafety

Life can be full of ups and downs and, for maker Alan Boris, that can be something to celebrate. After all, he's a self-proclaimed lover of elevators, having grown to appreciate just how complex these vertical transportation devices actually are. And, when he rode a "very cool" example recently, his tech ambitions rose to another level.

"When you start to think about the design, elevators have a lot of inputs and outputs – buttons, sensors, motors, LEDs and floor indicators," he notes. "They may be simple to use, but they hide a lot of complexity behind the scenes, so I created one using Raspberry Pi devices because it seemed like a great programming challenge."

Now, it has to be pointed out that Alan didn't crack on with the task of building a real-life, working elevator for his home. That would be a disruptive endeavour costing many thousands, which is why he decided to produce a much smaller replica instead. To do this, he made use of a Raspberry Pi 4 Model B computer and some Raspberry Pi Pico 2W microcontrollers. He also created an impressive hydraulic system. "And it was all a lot less expensive than the real thing," he tells us.

Uplifting concept

The project also scratched an itch. "I was looking to do a project where the Raspberry Pi single-board computer and the Raspberry Pi Pico microcontroller communicated with and worked together," he explains. To that end, he decided he would create a car operating panel wirelessly connected to some hall stations, so he began to create about half a dozen diagrams on paper to guide him. He wanted it to be possible to call the elevator by pressing either an up or down button on a station – the idea being there would be one on each of three floors. He would then have the interior car panel display, with buttons to take the elevator to one of the three floors, react in some way.

> *People often don't notice most elevators ding once for up and twice for down*

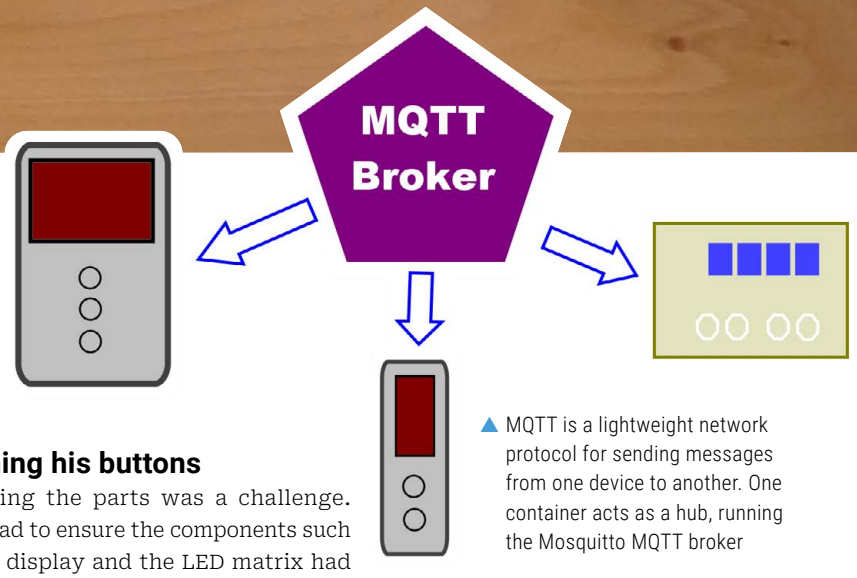
"I wanted the [interior car] panel to be similar to an interesting elevator I was in a few months ago, so I knew it would have an LCD display and buttons that had built-in LEDs," he says. "The elevator also had spoken messages such as 'watch your

01. The hallway station includes a Pico which communicates with the elevator panel

02. The car panel with Raspberry Pi includes a 5-inch display that shows the current state of the lift

step', so I knew I wanted to add a DAC and amplifier board to the Raspberry Pi 4 Model B computer to support that. In addition to the messages, the elevator's panel beeped as each floor was passed and there were dings for the up and down direction notification. People often don't notice that most elevators ding once for up and twice for down when they arrive on a floor, and I wanted all of that in my build."

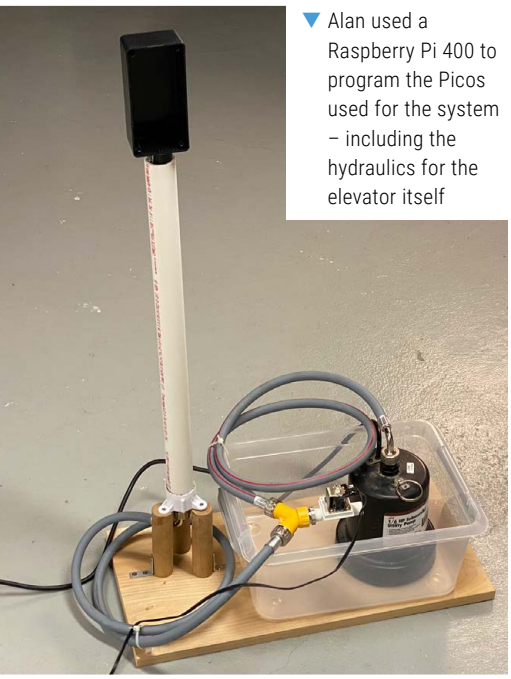
The hall station panel only needed two buttons for up and down, but Alan needed a way to show which floor the elevator car was on and the direction it was travelling. "I searched for a suitable LED matrix display," he says, "and the buttons were inspired by a different elevator system, so you could say my setup was beginning to be an amalgam of elevator controls I've admired over the years."



▲ MQTT is a lightweight network protocol for sending messages from one device to another. One container acts as a hub, running the Mosquitto MQTT broker

Pushing his buttons

Selecting the parts was a challenge. Alan had to ensure the components such as the display and the LED matrix had software libraries that he could easily use and there was a lot of improvisation as he realised the panel would be full of parts that needed to clear each other. "Substituting a part or moving it slightly could affect other part placements," he reveals. "For instance, the push-buttons are very deep, almost the full depth of the case, so very little if anything could



▼ Alan used a Raspberry Pi 400 to program the Picos used for the system – including the hydraulics for the elevator itself

be placed behind them."It meant that finding cases the correct size was tricky. "I initially considered 3D-printing the entire case, but that would have taken too much time. I could not find a long rectangular project case in stock for the floor call button assembly, so I ended up bolting two shorter ones together."

To make the panels look more like the ones on real elevators, he decided to use sheet metal and stick the button labels on (he printed them white-on-black with a laser printer and decided he'd stick them on using double-sided tape). Alan had never worked with sheet metal before, but found it was in stock at a local hardware store. He picked up 0.019-inch (0.5 mm) thick sheets that could be cut with simple shears.

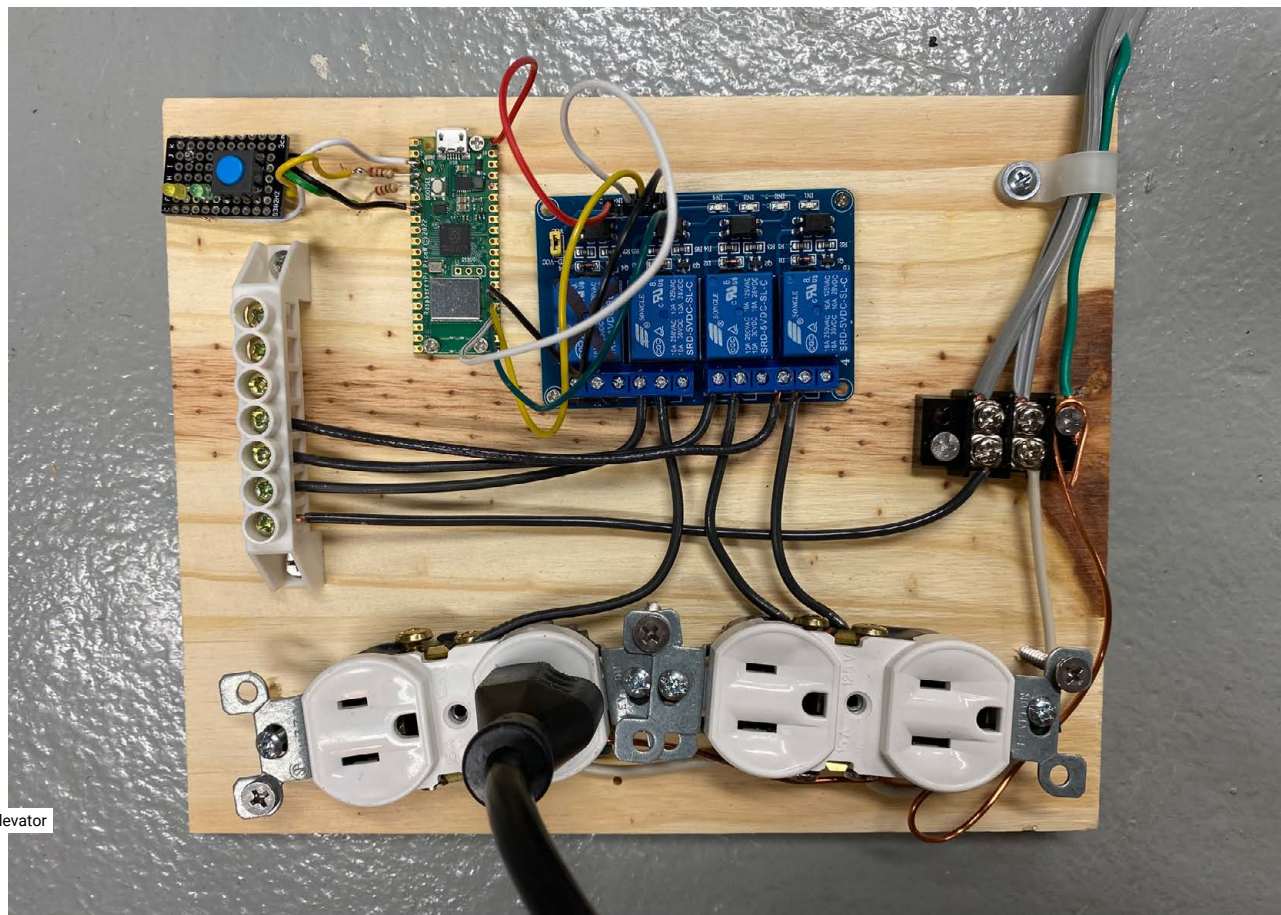
"My interior rectangular cutouts for the display were terrible – very uneven with shards of metal sticking out. I filed them down and hid my poor cuts with a simple 3D-printed frame that snapped on to the cut out sections."

On the rise

Since Alan had once seen a hydraulic elevator being repaired, he decided this was the system he'd use. "I created a mock-up showing how they work, but this required me to design a board that could allow the Raspberry Pi Pico 2W's GPIO to switch high voltages," he explains. "But it allowed me to work with a relay board, which was a lot of fun, and I like how the tiny Pico can control huge pumps and solenoids."

With no experience of hydraulics or plumbing, this was a learning curve. "I had to educate myself on the various couplings and fittings to fit the PVC pipes and hoses," Alan adds. Everything was then brought together using a program written in Python. "It basically waits for button presses via a Raspberry Pi 4 GPIO pins and takes appropriate action."

Coding the project was another challenge. "It was soon full of 'if-then-else' spaghetti code," he recalls. He then



▶ It's very much a concept model purely for fun, but it demonstrates the complexities involved in coding and operating an elevator

stumbled upon the elevator algorithm. “It basically states to keep moving in one direction and fulfil all floor requests, then start moving in the opposite direction to serve any remaining requests. I crudely implemented this algorithm using a Python list which the program loops through over and over again.

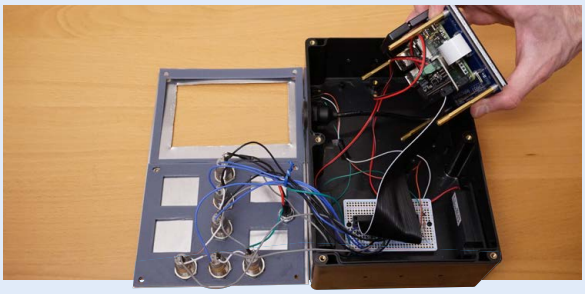
“When there is no activity, the program keeps repeating the current step. Everything happens in a large loop in 4.5-second intervals. Only the ‘door open’ and ‘door close’ buttons can interrupt a loop. Again, the programming gets complex because you can’t always open the door (like when the elevator is moving or between floors), so the code has to keep track of all of that.”

Luckily, it all worked and Alan looks at the project with a great deal of satisfaction and even deeper admiration for anyone who had coded for an elevator. “This simulation was for just one three-floor elevator,” he says. “I can’t imagine the coding required for a bank of elevators in a skyscraper that have to act cooperatively and efficiently.”

Quick FACTS

- This project replicates a hydraulic elevator
- It uses a Raspberry Pi computer and a Pico
- They communicate via MQTT over wireless LAN
- It took Alan 3.5 months to complete
- The project cost about $365

Give yourself a lift



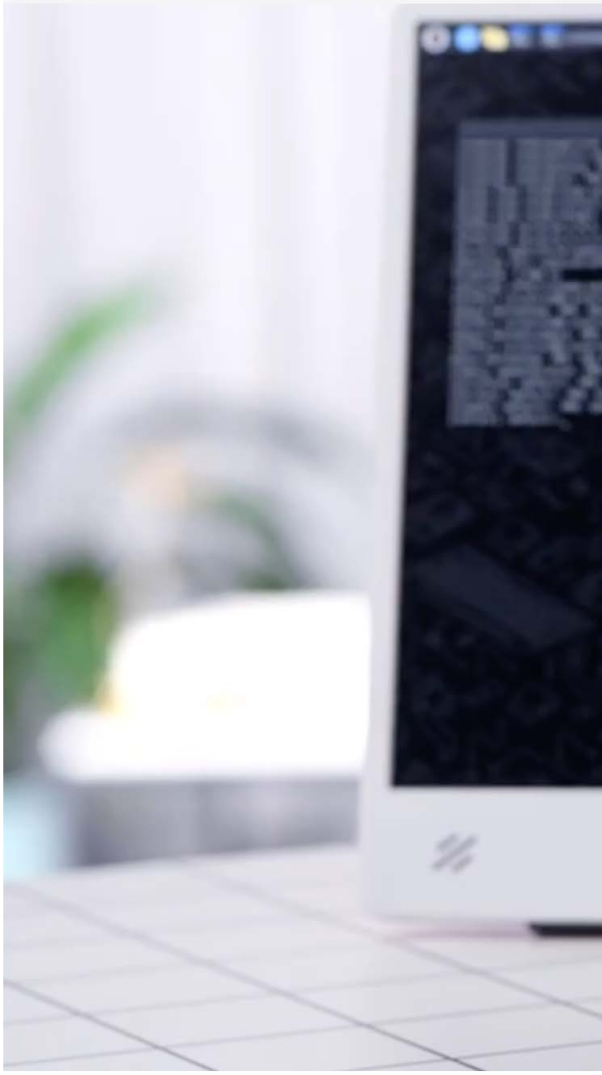
1. Alan says the elevator car panel is the brains of the system. It contains a Raspberry Pi 4, the floor buttons, LCD display, and audio output. It runs on balenaOS and it communicates with the floor call buttons and a high-voltage board using MQTT over wireless LAN.

2. The display is driven by a web browser container in kiosk mode. A tiny web server runs in a separate container to serve the page. The main Python code establishes a web socket with the server to update the static web page. It uses the Pygame library to play audio.

3. The other panels have Picos inside and run on CircuitPython, which also has a compatible MQTT library. The hydraulics use a Pico W to power relays that allow sufficient water in and out of the system to move the elevator car up and down.

Pepper Pi

Raspberry Pi fans with multiple models will relate to the rationale behind the origins of the transparent computer, says **Rosie Hattersley**



Makers

Martin Spendiff and Vanessa Bradley

Serial inventors Martin and Vanessa enjoy videoing their Raspberry Pi builds and have inspired many makers to create their own fun projects.

veeb.ch

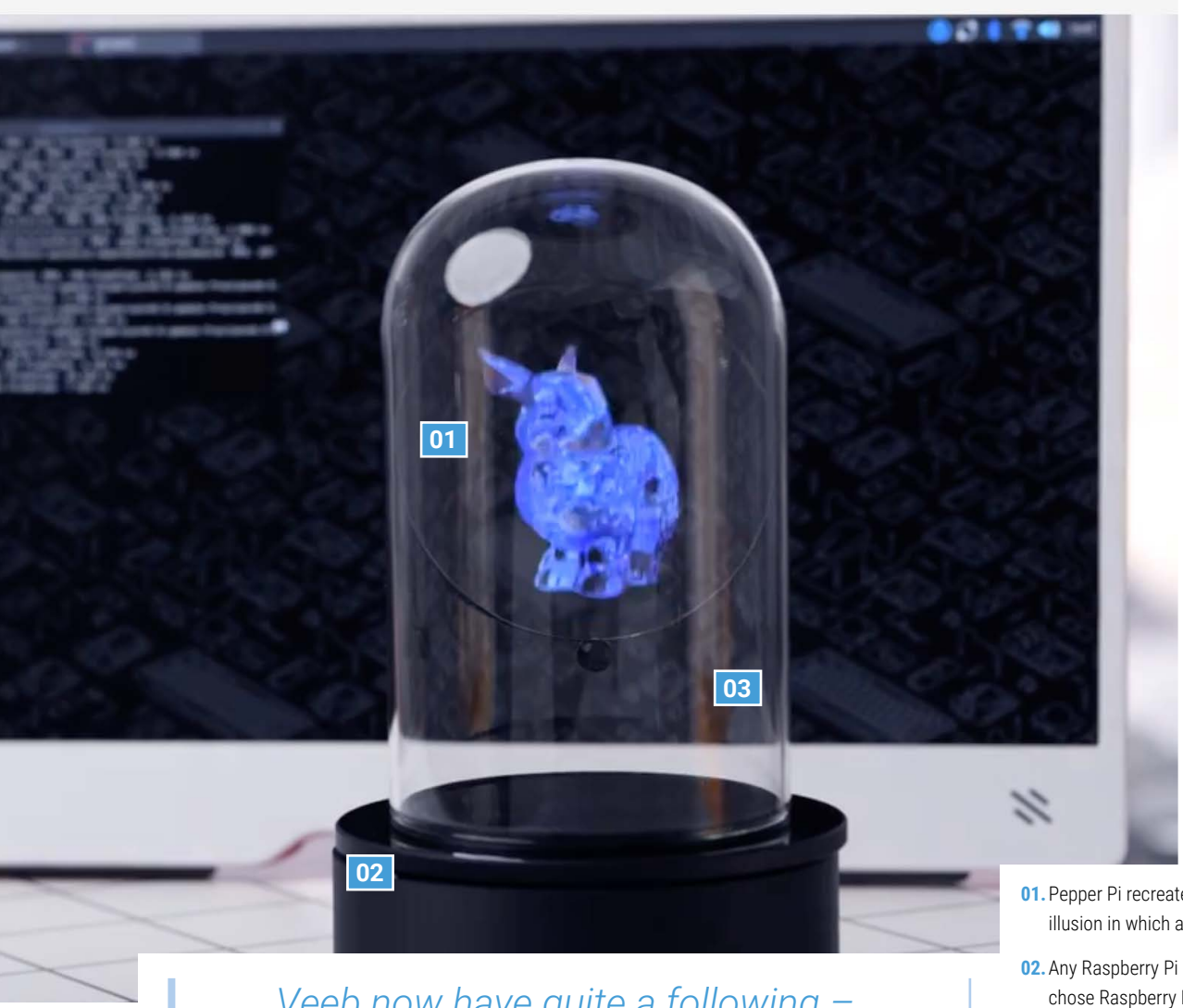
Documenting quirky makes comes naturally to creative couple Martin and Vanessa. Collectively known as Veeb, they've been making madcap designs accompanied by fun videos of each build for the past five years. Their latest project, Pepper Pi, makes use of a headless Raspberry Pi 5 they had available to recreate a Victorian illusion known as Pepper's Ghost in which an object appears to float in a glass dome. The project also has its roots in early 1900s stock tickers. Impressively, Pepper Pi functions as a perfectly usable transparent PC, too.

Seeing is believing

Originally conceived during lockdown, Veeb's videos soon caught on (and meant they didn't need to do quite so much worthy documentation!). Imitation is flattering, and when Martin and Vanessa realised other people enjoyed some of the videos they'd made combining old tech with new sufficiently to create their own versions, they embarked on an ongoing series. "We started to hear about people reproducing what we'd done. That felt good and was enough to make us keen to keep sharing projects whenever we thought we'd done something vaguely interesting." Veeb now have quite a following – 11,500 according to their YouTube follower clock: **rpimag.co/ytsubscounter**.

The day-to-day business involves custom builds and "reimagines vintage design for contemporary living". Martin tells us. "The Pepper Pi project was prompted by the fact that there are headless Raspberry Pi computers dotted about our office taking care of various tasks."

Although "ever so useful", a scattering of hardware components doesn't produce quite the contemporary finish to their aspirational design studio that, say, scatter cushions or artfully placed house-plants might. Instead, the aim was to spruce things up and to make a selection of "Raspberry Pi computers into interesting things to look at".



01. Pepper Pi recreates a Victorian mirror illusion in which an image appears to float

02. Any Raspberry Pi can be used, but Veeb chose Raspberry Pi 5 with 16GB RAM so it would also serve as a PC

03. Mirror angles and a privacy shield plus some **mpv.io** footage create the compelling visual trickery

Veeb now have quite a following – 11,500 according to their YouTube follower clock

▼ 1920s ticker tape machines and Victorian illusions inspired this build, as Veeb's video explains

Old hands

Martin and Vanessa have many Raspberry Pi builds under their belts. Here, they used Raspberry Pi 5 to run the statistical package R (**r-project.org**), "so we wanted to use something meaty for number-crunching". However, the illusory effect with the screen could be done using "more modest hardware" – they have since created one using Pico. A previous project involving an old Super 8 video viewer (**rpimag.co/boostbox**) showed them the potential of lower-resolution screens and how to make the most of them. They still use the Boostbox cyberdeck daily. As with the Pepper Pi project, it combined "a redundant bit of technology and something box-fresh".

Their take on the 150-year-old Pepper's Ghost illusion involves a pair of mirrors bouncing a hidden image to make it appear within a transparent dome they bought online. They had seen versions hidden in arcade cabinets, but worked out the trickery for themselves. It took a while before they hit on the idea of using a privacy screen to shield the image so it was not on obvious display. In Vanessa and Martin's version, the source image is hidden behind an acrylic disc which they commissioned "a nice man in Germany (**formulor.de**)" to laser-cut.

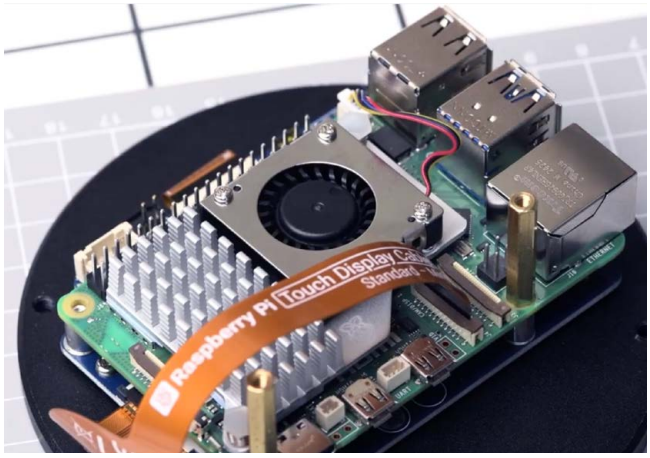
Originally, they set the viewing angle for the mirror to 45 degrees, but soon realised this didn't work well for viewing the illusion when the Pepper Pi was sat on a desk and in use as a PC. Since the angle of the reflector is pretty important, Veeb created a shape calculator to assist anyone keen to recreate the project: **rpimag.co/pepperpi**.

With the trigonometry taken care of, (and an optional screen attached), they ended up with "a functional computer that has a nice steampunk vibe". You can even run it as an ornament that can act as a server while intermittently playing that hologram message that Princess Leia sent via R2-D2. Martin fervently hopes someone will actually do this!

▲ To the uninitiated, the dome simply looks like quirky desk furniture rather than a transparent PC

Quick FACTS

- They used R Server to write a program to create images from data
- They also used it to make a program to count YouTube followers
- Veeb ran a contest to win their Pepper Pi
- Entry involved submitting a poem about waffles
- More than a hundred poems about waffles ensued



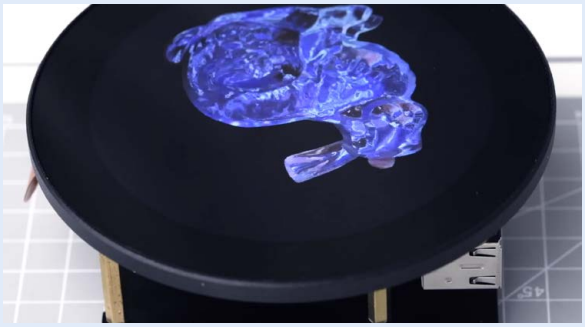
▲ Mounted on the underside of the disc-shaped base for the glass dome, Raspberry Pi 5 plays the images or videos

▼ The duo had fun seeing what they could make appear inside the glass dome

Use your illusion

1. You will need a Raspberry Pi 4 or 5 computer, ideally with 16GB RAM, plus a suitable dome for the illusion, a base, and two small circular mirrors.

2. Set up the source image and mirrors, using the handy reflector shape calculator on Veeb's website: **rpimag.co/pepperpi**. Veeb used MPV (**mpv.io)** and some tweaks to Raspberry Pi OS to include video effects.



3. Further construction details plus the all-important maths (for calculating mirror angles) are shared in this fantastic YouTube video: **rpimag.co/transparentcomputer**.

Atari 2600 Digital Frame

Display photos as blocky 8-bit masterpieces on a classic console. By **David Crookes**

Maker

Nick Bild

Nick is a software engineer, writer, and electronics hobbyist working to build the next big thing you never knew you wanted.

rpimag.co/2600digitalframe

The Atari 2600 is an iconic console dating back to 1977. But it's also a machine that is notoriously difficult to program due to its simple, limited hardware. There's no character ROM, no frame buffer and there's only 76 CPU cycles per scan line. What's more, the console has just 128 bytes of RAM. And yet, for Nick Bild, this is merely a challenge.

"I enjoy pushing this kind of hardware to do things it was never intended to do; things that may not even seem possible to most people," he says. So while, on the face of it, a digital photo frame program doesn't sound like a big deal, it's actually mightily impressive.

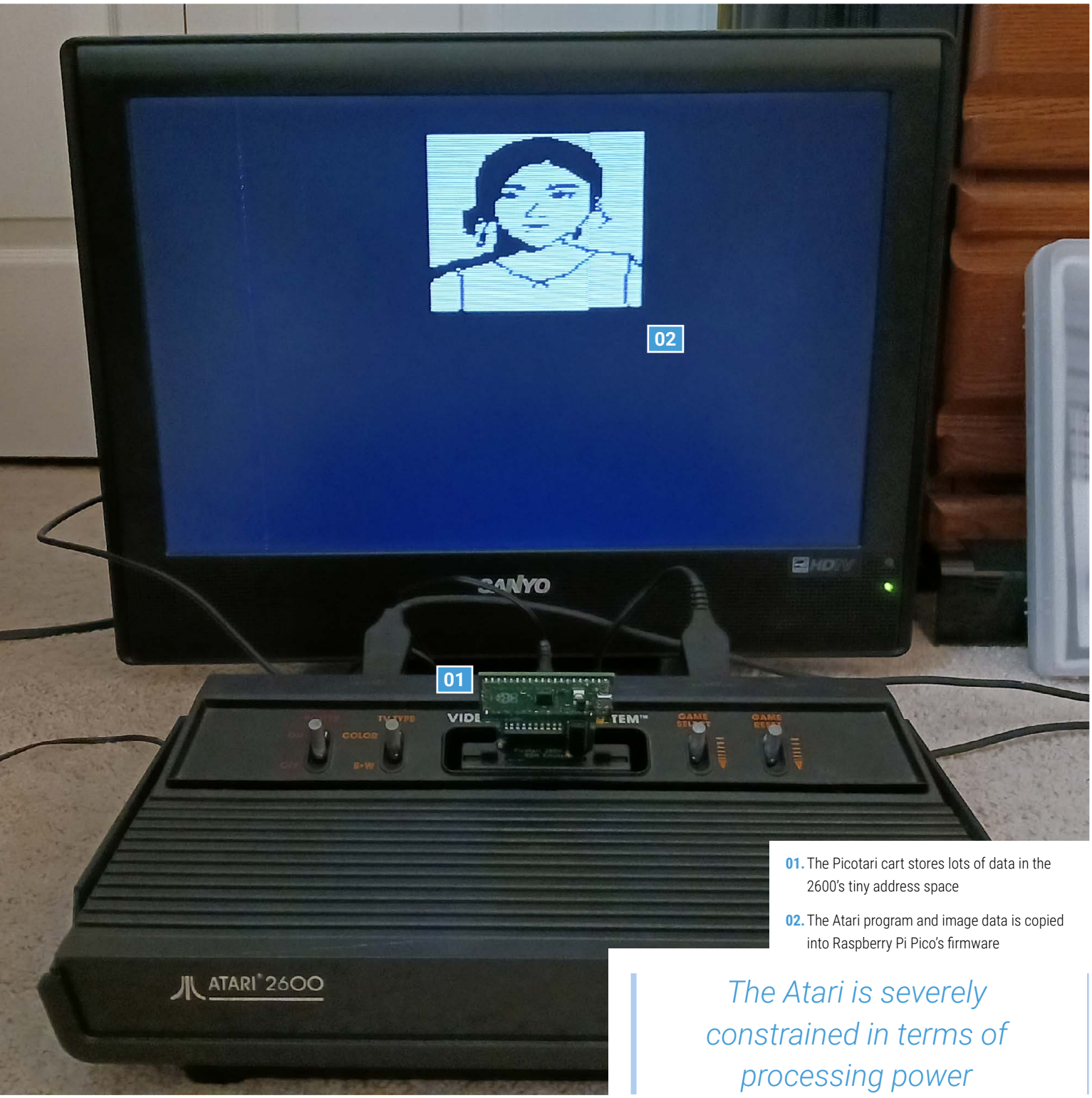
Pixel placement

Nick's program allows photos that have been converted into 8-bit pixel art to be displayed as a slideshow but, since Atari 2600 cartridges can only contain 4kB of data (or 64kB if they're bank-switched), he needed to be creative.

To that end, he's built what he calls a Picotari cartridge with an edge connector that can slot into the console's cart port. It supports a Raspberry Pi Pico microcontroller board which provides two benefits: it adds extra memory to store many images and it allows the running of his customised Atari 2600 ROM emulator so the hardware can make use of them.

The Digital Photo Frame program is then capable of displaying each of those photos, in turn, as a 64×84 pixel image. In doing so, the program is fooling the Atari into thinking only one byte is being used for each picture – it's addressing a specified byte each time another eight pixels are needed to draw a player sprite. Nick uses a trick that employs sprite copies and vertical delays to fit six 8-bit sprites on a single line (giving 48 pixels in total) before drawing another two sprites (an extra 16 pixels) straight after.

"The Atari asks for the data at a given address, and the Pico returns the data that is stored at that address, just like a physical ROM chip in an old cartridge would do," he explains. "But when a special address is requested, the Picotari sequentially returns data from a special storage area outside of what the Atari can normally access. In this way, image data can be supplied byte-by-byte, right as it is needed, to draw the images."



01. The Picotari cart stores lots of data in the 2600's tiny address space

02. The Atari program and image data is copied into Raspberry Pi Pico's firmware

The Atari is severely constrained in terms of processing power

Bit manipulation

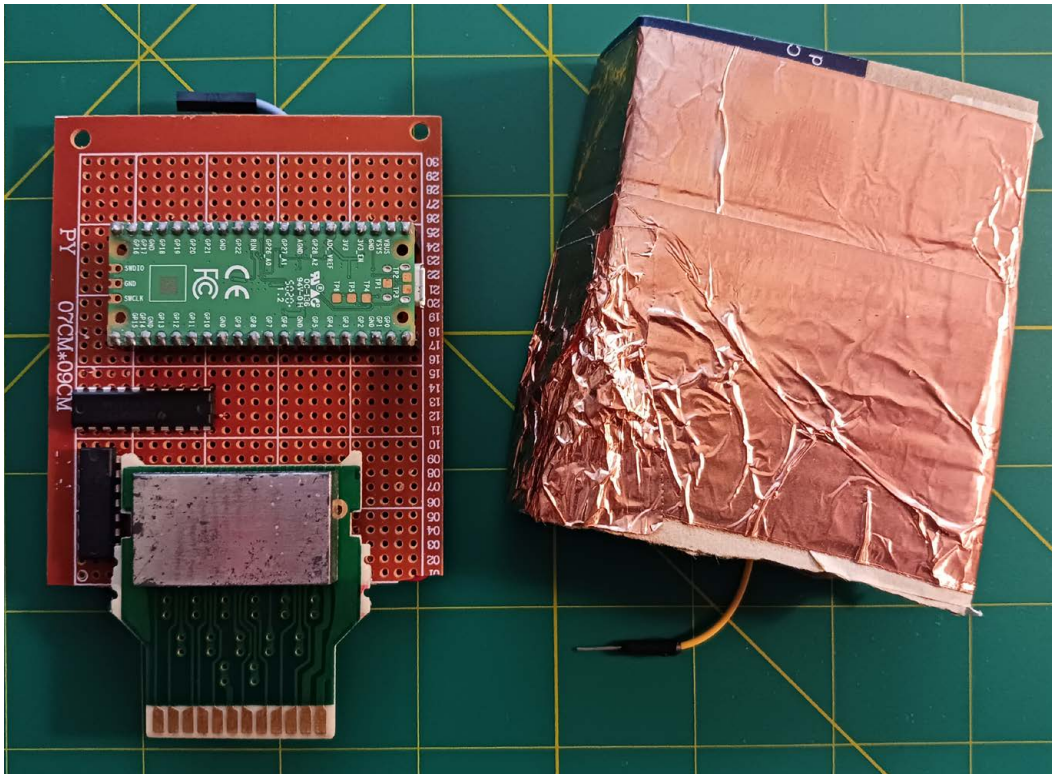
For it to work, the images need to be prepared in advance. This is done on a PC using a JavaScript library, Pixel It (**rpimag.co/pixelit**), with touch-ups with the GNU Image Manipulation Program.

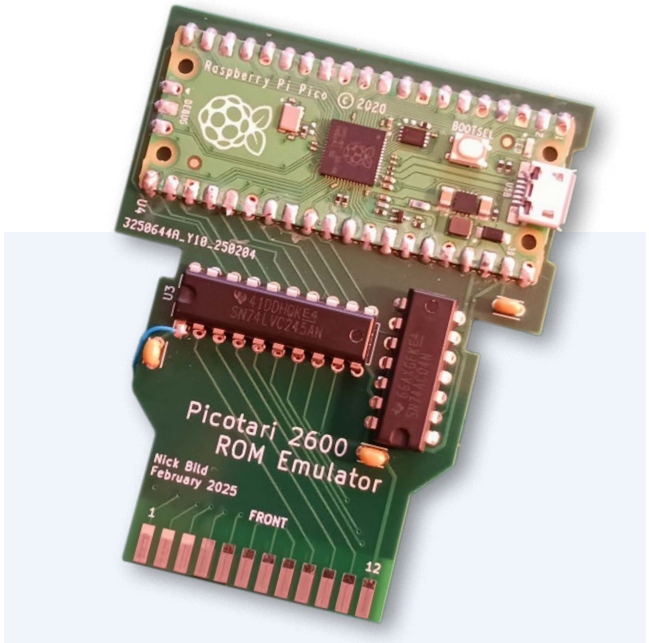
"Aside from lacking in storage space, the Atari is severely constrained in terms of processing power, so the images need to be fed to it in a simple data structure that can drive the display with as few instructions as possible," Nick says. "I experimented with a few tools to give the images the right 8-bit look at a low resolution, then wrote some scripts to prepare the data structures that were to be loaded on to the Pico."

The result is a unique app that truly pushes the hardware to produce instantly recognisable pixel-based photos and Nick has been so inspired, he's now thinking about creating something like a Game Boy Camera for the system. Not that it'll be any easier. "The severe hardware constraints of the Atari makes for difficulties every step of the way," he says.

Quick FACTS

- Photos are displayed as pixel art on an Atari 2600
- With no frame buffer, they're sent one byte at a time
- The program is saved on a custom cart containing a Pico
- Raspberry Pi Pico provides a lot of storage space
- All images have to be in two colours





▲ The prototype cart's breadboard and perfboard caused electromagnetic interference

◀ The custom PCB slots directly into the Atari 2600 cartridge port

WIZnet

W6300

The Ethernet Solution for Faster, Smarter Networking!

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

Key Benefits

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

Where to Use W6300?

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

Performance Test Results

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

WIZ630io 

W6300-EVB-Pico (W6300 + RP2040) 

W6300-EVB-Pico2 (W6300 + RP2350) 

PiEEG

Control Raspberry Pi with brain power alone and learn about the mind. **Rob Zwetsloot** ponders it

Maker
Ildar Rakhmatulin
A neuroscientist with an interest in machine learning and AI and their practical applications.

pieeg.com

An EEG (electroencephalogram – say that three times fast) is, to put it simply, a test to measure brainwaves. It's used to diagnose conditions and illnesses in hospitals, and is usually measured with loads of little discs over your head – a feature you may have noticed on the PiEEG, created by scientist Ildar Rakhmatulin.

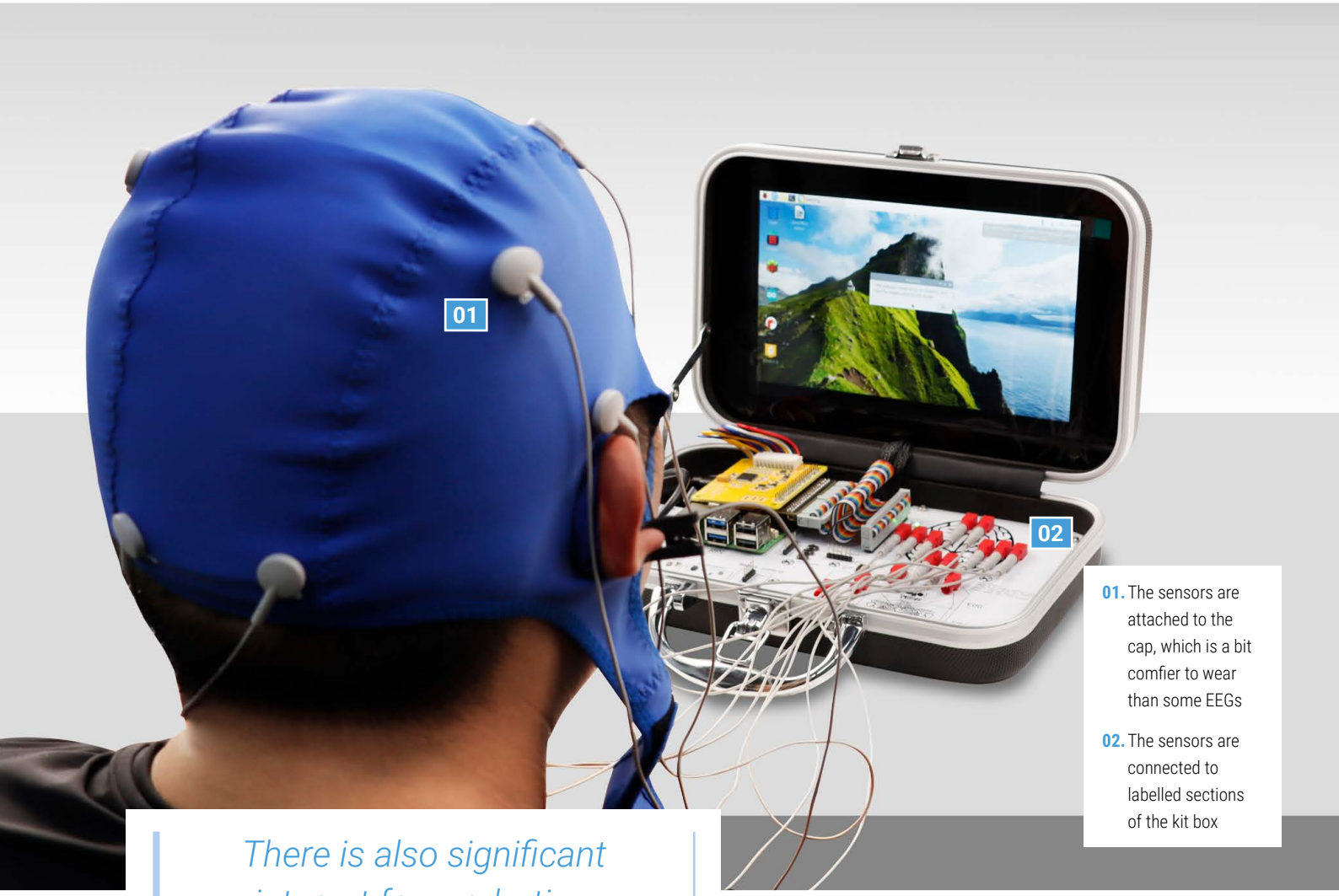
"I started moving to neuroscience about six years ago, when I set out to apply my machine learning skills in a practical way – specifically, to control a robotic arm, which I found to be an exciting challenge," Ildar tells us. "However, I couldn't find a brain-computer interface (BCI) device that met my requirements in terms of price, quality, and openness. So, I decided to develop my own brain-computer interface. This marked the beginning of PiEEG, a low-cost device designed for measuring EEG data using a Raspberry Pi."

PiEEG enables "easy brain signal acquisition, data processing, and feature extraction" for users in what Ildar calls a bio lab in the home. As well as EEG, it's able to record ECG (electrocardiogram), EMG (electromyography, for muscles), EOG (electrooculography, eyes), and PPG (photoplethysmogram, blood) data, and also has other environment sensors. Ildar sees it as a way for folks to learn about how the brain – and body – works.

Mind control

Data is read via the SPI pins on Raspberry Pi and can then be used in Python thanks to the open-source SDK for it, which you can find at **rpimag.co/pieeggit**.

"The PiEEG kit is ideal for anyone interested in brain-computer interface technology and bioscience," Ildar says. "Researchers can easily test ideas beyond the lab, while startups can quickly prototype and validate new concepts in



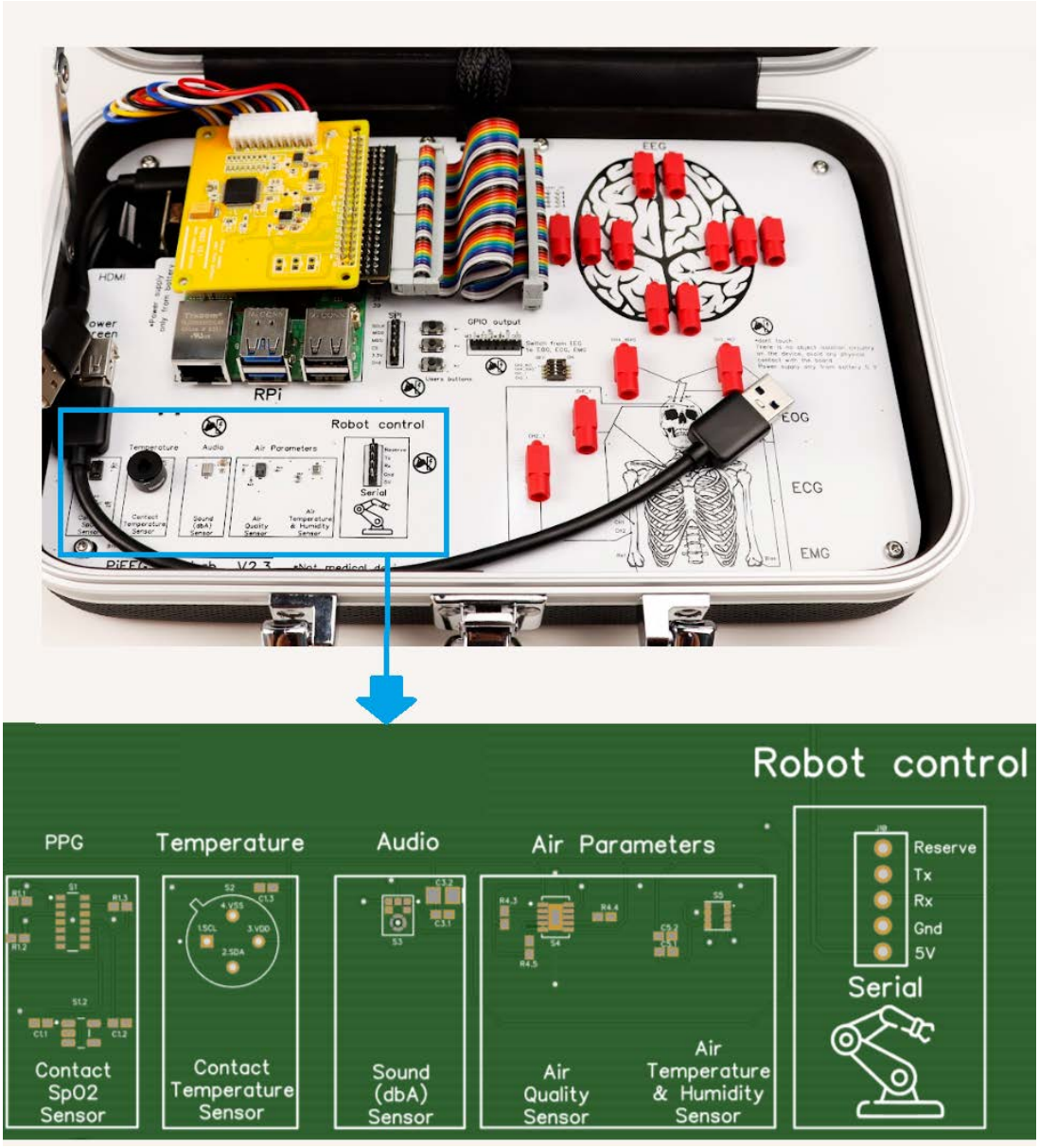
01. The sensors are attached to the cap, which is a bit comfier to wear than some EEGs

02. The sensors are connected to labelled sections of the kit box

There is also significant interest from robotics schools looking to integrate the device for robot control

▼ The new kit version opens up PiEEG for people curious about neuroscience

real-world applications. Machine learning and data scientists benefit from seamless data collection and experimentation with ML algorithms. Students gain hands-on experience in bioscience and observe real-time interactions with the body. Enthusiasts can explore bioscience, understand brain and body functions, and experiment with meditation – whether at home or outdoors. The device can be used not only for learning bioscience, but also for custom projects such as controlling a robotic arm or controlling a laptop."



Controlling robots with thought alone is possible with the right components

Democratising EEG

The PiEEG itself has been on the market for a few years, allowing for fine-tuning of the core project. Ildar recently put out a kit version on crowdfunding website Indiegogo that is easier to use and does not require as much coding experience.

"We announced the device just a week ago and are still gathering public feedback," he explains. "As expected, it has been particularly well-received by researchers and PhD students, many of whom are eager to use it for conducting research outside the lab with ease. There is also significant interest from robotics schools looking to integrate the device for robot control."

Ildar is still working on new abilities for PiEEG: "We are currently developing a system to control games using brain signals based on the P300 paradigm – a challenging task, but we are making steady progress. If successful, this could lead to one of the first laptops controlled entirely by brain signals".

Quick FACTS

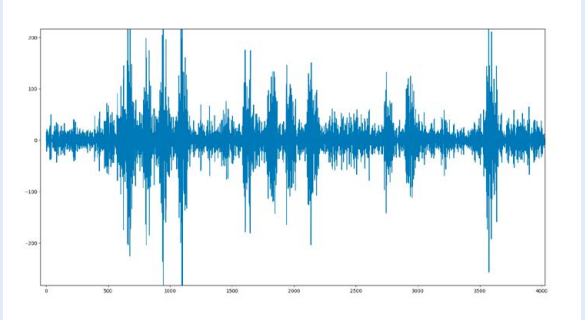
- Ildar is working on connecting PiEEG to ChatGPT…
- …Currently it can describe your stress levels
- Electrical activity in brains has been studied since 1875
- The first human EEG was recorded in 1924
- The kit comes with eight EEG sensors

▼ The cap allows you to easily attach sensors to the right parts of your head

Measuring brainwaves

1. Placing the electrodes on the cap arranges them specifically on your head. You can also use slightly different configurations with more sensors, or placing them elsewhere. These are plugged into the kit using the corresponding port.

2. The PiEEG board works as an 24-bit analogue-to-digital converter, allowing the signals from the EEG sensors to be read by Raspberry Pi.



3. Using the PiEEG SDK and Python examples, you can read the data from the EEG sensors and have them control devices, interfaces, or just help you figure out how to meditate

IINTS insulin pump

A diabetic teen created a Raspberry Pi-based medical pump to see how insulin doses keep him alive. **Rosie Hattersley** is impressed!

Maker
Rune Bobbaers
Rune is a 16-year-old tech enthusiast, programmer, and maker.

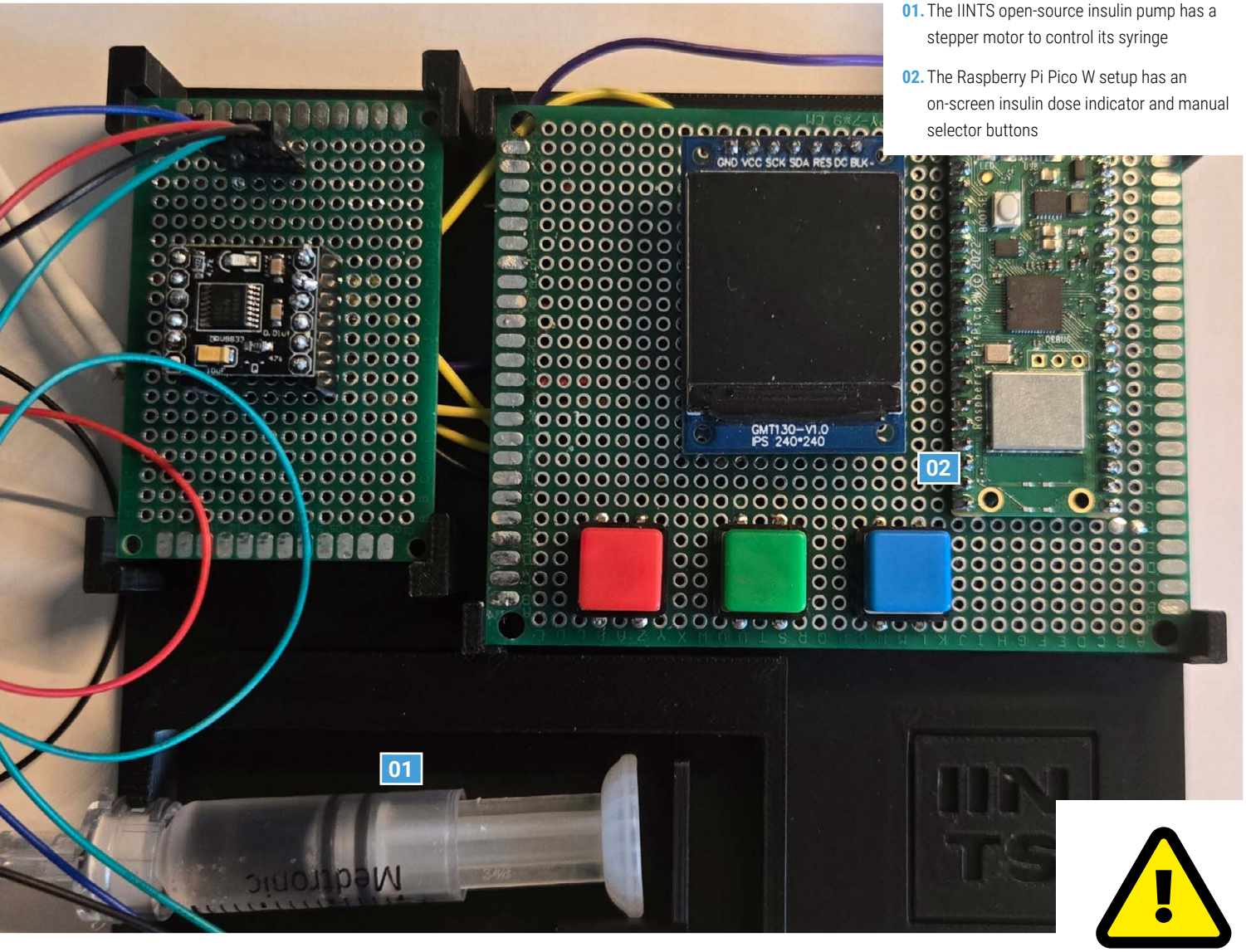
rpimag.co/iintsgit

Teenager Rune Bobbaers was diagnosed with the auto-immune illness type 1 diabetes before he was old enough to start school. Curiosity about how insulin is used to manage his glucose levels and keep his body in check led Rune to investigate how to create his own dosage pump so he could understand how it operates. He was "immediately hooked" as soon as he began using Raspberry Pi 400. "That such a small, affordable computer could do so much really inspired me." He worked on what would become his IINTS insulin pump project during CoderDojo sessions learning MicroPython and using Raspberry Pi Pico W. The now 16-year-old Rune submitted it as a Coolest Project, where it immediately caught the eye of the Raspberry Pi Foundation's judges.

Mastering making

Rune was keen to design and build a fully open-source insulin pump from scratch having long "been fascinated by the medical technology that helps keep me alive, especially insulin pumps". There was never an intention to produce a certified medical product. Instead he was after a personal learning experience as his curiosity gradually turned into a deep interest in technology and programming. Through his local CoderDojo group in Brussels, he's had "the opportunity to turn ideas into real projects, ask questions, and get support from amazing coaches". With a few Raspberry Pi builds under his belt, Rune credits the CoderDojo coaches' support with giving him the confidence to tackle his first full project that combines hardware, software, mechanics, and medical tech all in one.

> *This insulin pump is the result of everything I've learned so far*



01. The IINTS open-source insulin pump has a stepper motor to control its syringe

02. The Raspberry Pi Pico W setup has an on-screen insulin dose indicator and manual selector buttons

Rune set about recreating a pump simply and with affordable components. He chose to use MicroPython and Raspberry Pi Pico W because "it's compact, affordable and powerful enough to handle everything from motor control and display updates to button input and background logic".

The IINTS pump (the acronym means 'insulin is not the solution') can be used to manually select a dose using tactile buttons, with the amount shown on a small display. It can also calculate recommended doses based on a preset insulin-to-carbohydrate ratio. Once the dose is selected, a stepper motor controlled by Raspberry Pi Pico W pushes the plunger of a standard syringe and delivers the insulin. Impressively, Rune has even managed to add a feature whereby a background dose of insulin is continuously delivered, just like the prescription pump he uses to regulate his glucose levels.

Warning!

Do not use

This project is for educational purposes only. It is not a medically approved insulin pump and should not be used to administer insulin. Consult a medical professional for insulin administration advice.

Healthy approach

Rune began simply with a very basic breadboard setup that allowed him to test the screen, buttons, and motor together. Once this was working, he started designing and printing parts to hold the motor and syringe in place. "That's when the real problems began: one of the early 3D-printed parts seemed slightly misaligned, so I mounted it on a metal lathe to try and fix it, only to later realise the issue wasn't with the printed part at all, but with the stepper motor driver! Undeterred, Rune rewrote the code, simplifying large parts of it. He also had to redesign the enclosure, improving its durability and ergonomics as well as making it less bulky.

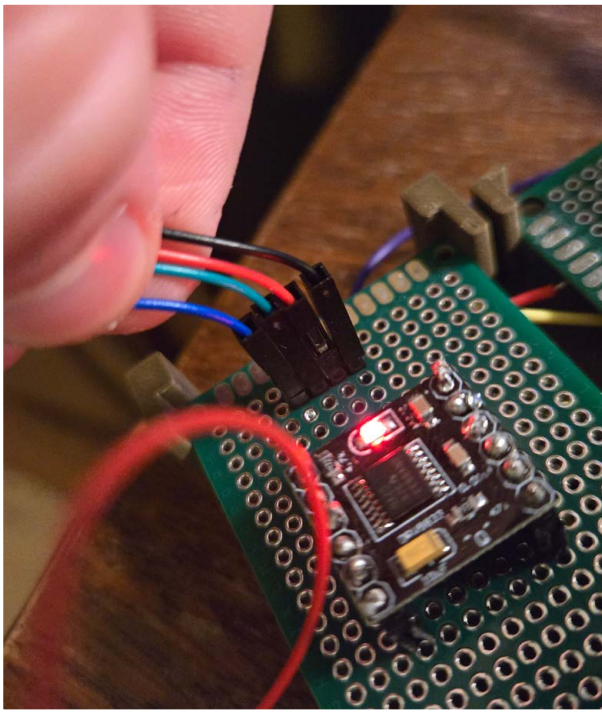
Rune has a sanguine approach to the iteration process. "Each version solved a problem from the previous one and, with every iteration, I learned something new about mechanics, electronics, and system integration," he tells us.

"This insulin pump is the result of everything I've learned so far – and of my passion for understanding the devices that have such a big impact on my life." He's already planning the IINTS Mk II and would love to design a custom PCB to clean up the wiring, make the pump more compact and reliable, and make it easier for others to build their own version.

Anyone inspired to do so can check out his excellent video: **rpimag.co/iintsvid**.

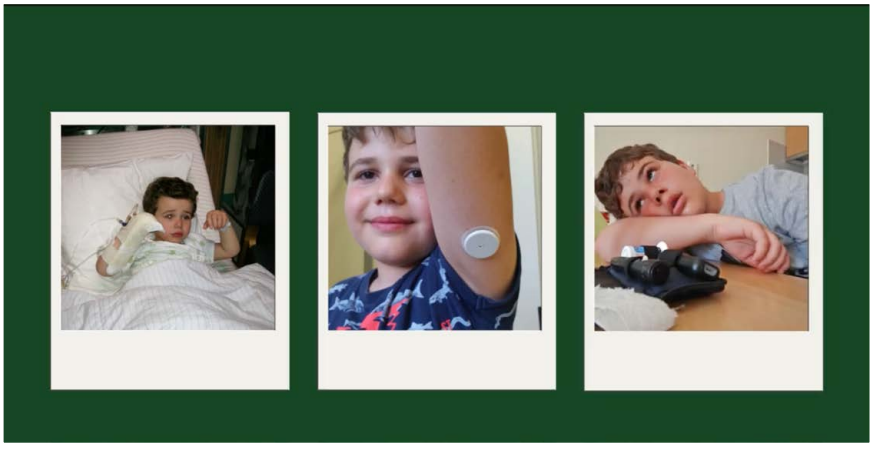
Quick FACTS

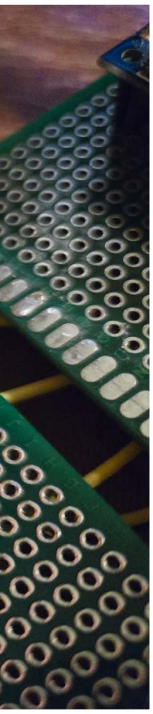
- Rune's curiosity was sparked by the insulin pump on which he depends
- He designed and 3D-printed the mechanical parts himself
- Rune also wrote the MicroPython code to run IINTS
- A CoderDojo mentor gifted him the stepper motor and driver
- He tested and debugged everything in Thonny IDE

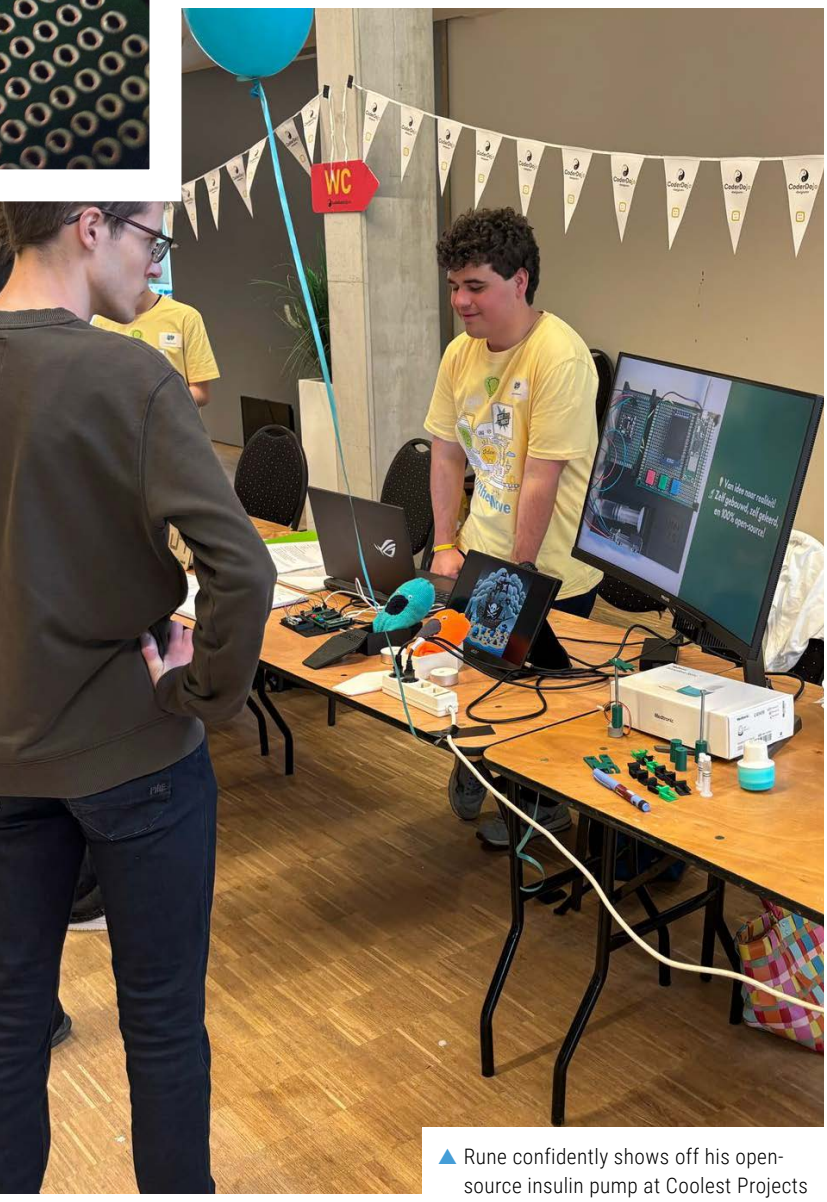


▲ Rune says his "surprisingly precise" insulin pump can deliver micro doses

▼ Rune was fascinated by the medical technology that helps keep him alive

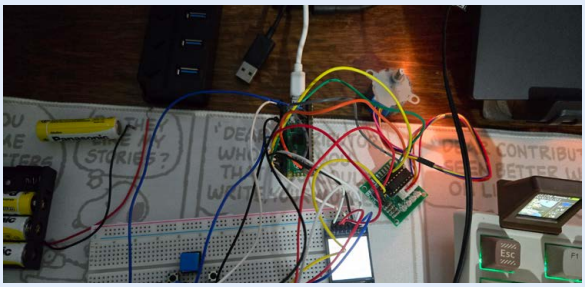




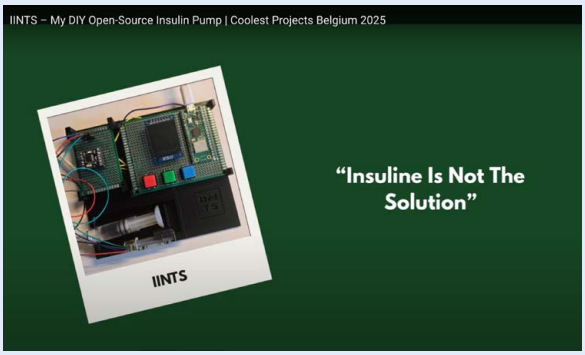


▲ Rune confidently shows off his open-source insulin pump at Coolest Projects

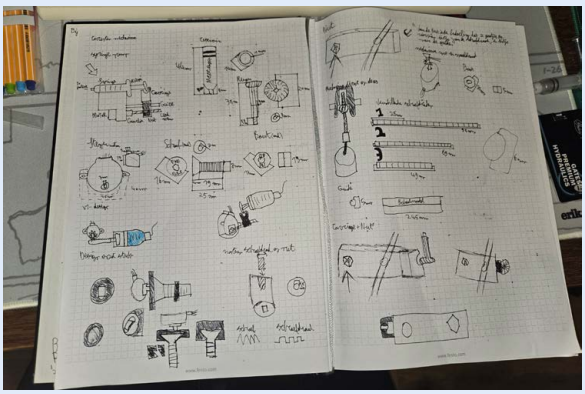
Raspberry Pi Pico-powered pump



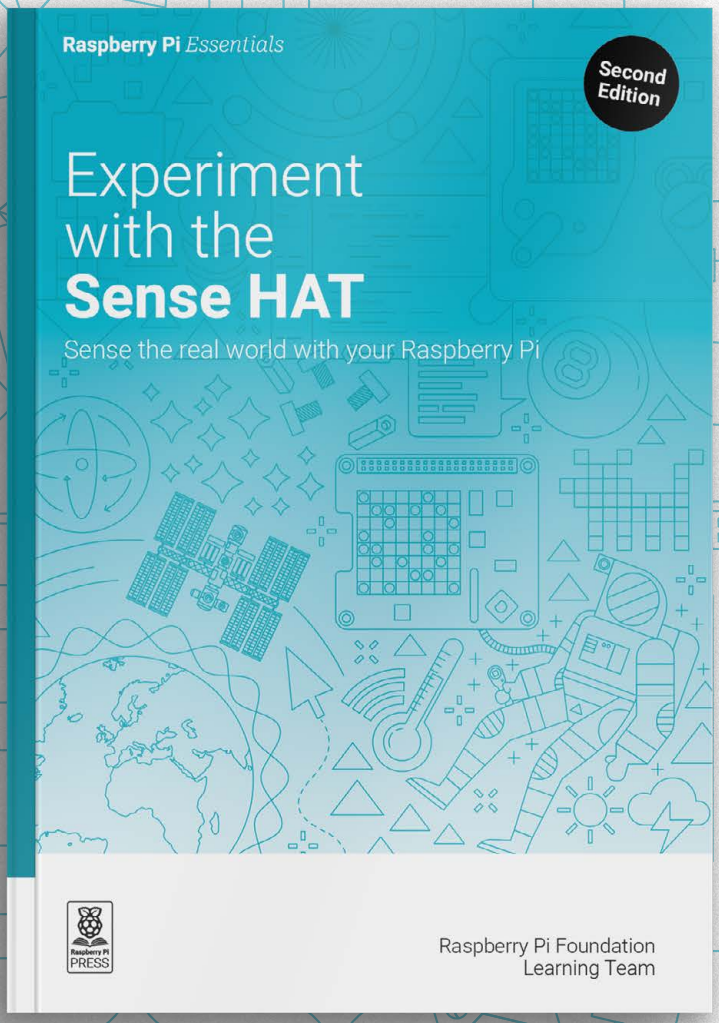
1. The setup is a basic breadboard, Raspberry Pi Pico W, OLED display, stepper motor, push-buttons, and battery, plus a syringe or peristaltic pump to deliver doses.



2. Although created for educational purposes only, Rune provides full instructions, stepper motor control, and code to deliver specific amounts of insulin at **rpimag.co/iintsgit**.



3. Rune's meticulous approach includes sketches, while his MicroPython code even has a bitmap splash screen appear when the IINTS pump boots up.


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

Baby North case

By HardWiredFurry

rpimag.co/BabyNorth

Fractal is a company that makes cases for gaming PCs. They're sleek, beautiful objects that turn your beige box of a computer into something that looks good, with oak, walnut and brass among the smart details on offer. While Fractal doesn't make its cases in a size to fit Raspberry Pi, it has released the files so that enthusiasts can 3D-print their own.

Reddit user HardWiredFurry found a beautiful copper heatsink and was inspired to build their own. This build uses two fans for cooling, as well as the copper heatsink, and is now looking aesthetically pleasing on their desk while hosting a Minecraft server.

▶ Design is subjective, but most can agree when something looks cool

MIDI Blaster

By Luke the Maker

rpimag.co/MidiBlaster

This is the MIDI Blaster: a nostalgia-soaked trip into the digitised music of games past. The idea began when the maker discovered the Roland MT-90S. This is a device that plays MIDI files from floppy disks, and because MIDI is so versatile, it enables the user to speed up, slow down, and otherwise manipulate the sound. The drive on the Roland was beginning to wear out, so Luke needed a replacement. Naturally, he designed and built his own – and put the design files on his Patreon page for free.

The MIDI Blaster uses a Raspberry Pi 4, as the Raspberry Pi 5 doesn't have a 3.5mm audio jack. It's loaded with 'sound fonts', which is a new one on us, but it means that the user can give the music from one game – Final Fantasy VII, say – the sonic characteristics of the music from another game... Ocarina of Time, perhaps. And perhaps the most immersive sonic treat is that you also get the sound of a floppy disk whirring away in the drive.

▲ You can also use the MIDI Blaster as a speaker for anything with a headphone port

Sony Watchman cyberdeck

By Granitsky

rpimag.co/Watchman

Before streaming, before smartphones, before MP3 players, there was the Sony Walkman. This brilliant line of devices played music recorded onto cassette tapes, with each tape typically holding up to 90 minutes of recorded music. They were powered by AA batteries, and enabled the user to listen to their own music wherever they were through a pair of headphones. Times were simpler then, and Sony sold the Walkman in the millions.

From 1982 to 2000 Sony also made a line of pocket TVs, which didn't catch on as much in the UK (who wants to walk around glued to a tiny portable screen, eh?). These devices, collectively called the Sony Watchman, came in many, many variants as screen technology evolved over 18 years of production. What's more, they don't work as TVs today, as the analogue TV broadcast signals have been switched off in favour of digital.

That doesn't mean they're useless, though: Reddit user Granitsky has taken a 1985-era Watchman and grafted on a Raspberry Pi 3B+, a wireless keyboard/mousepad, and a USB battery pack (for the Raspberry Pi 3B+ and keyboard). We love that gorgeous little CRT screen!

▲ According to the maker, this '80s cyberdeck lasts for 'hours and hours' on a combination of AA and USB battery power

3D print

A special, scalable spectacle-case with a flexible 3D structure.

By **Toby Roberts**

rpimag.co/GlassesCase

Around 5000 years ago, something catastrophic happened to humans: we started farming. No longer hunters and gatherers, living off what we could forage from the forests and streams, we succumbed to an innovation from the Middle East that would guarantee us a larger supply of calories for less effort. Urbanisation, spreadsheets, student loans, and mortgages all came from this fateful event.

But recent evidence suggests it's not as simple as that. Examination of grave sites indicates that instead of the Mesolithic locals learning how to farm, they were replaced by far more successful Neolithic farmers. One of the differences in the evidence left by the two populations is skull length. The newcomers with their fancy ability to plant seeds and eat what they grew had smaller skulls, and that's why it's all but impossible to go into an optician's shop today and buy a pair of glasses with a temple length of more than 140mm.

If, like our features editor Andrew, you have a gloriously long skull, you might appreciate this flexible glasses case created by Rorys3D. The maker has made the STL files available on Printables, so if you find that your eyewear doesn't fit the puny standard sizes offered for today's humans, you can scale the design in whatever dimension you need to fit longer/wider/deeper frames.

Other than that, the interesting thing is the flexible flexure. We printed ours on a Bambu Lab Mini using PETG filament. PETG was selected for its improved flexibility compared to PLA, which can be more brittle – making it a more suitable material for a case that may be opened and closed frequently.


HUGO

Home Automation Made Easy

Make your home smarter with Raspberry Pi and Home Assistant

By Phil King

With Raspberry Pi, you can build a cost-effective and flexible smart home setup that's easy to control and customise. A Raspberry Pi paired with Home Assistant – the free, open source home automation platform – becomes the brain of your smart home, enabling routines, voice assistant integration, and remote access, all tailored to your preferences.

First, we show you how to install Home Assistant on your Raspberry Pi and set it up, adding the smart devices on your network. We'll also explain automations and scenes, along with add-ons to get Home Assistant to communicate with more apps and devices. You can even access your Home Assistant system remotely when away from home.

We than explore the vast range of Home Assistant-compatible smart home devices available, ranging from sensors to lights and smart sockets. We also round up some home automation project ideas, including a smart garden.

Finally, we guide you step by step through an example project in Home Assistant: a simple security system with a motion sensor to trigger an alarm and send notifications to your phone. So, get ready to make your home smarter...

Get started with Home Assistant

How to install and access your Home Assistant server

> It supports thousands of devices and services

YOU'LL NEED

- Raspberry Pi 4 or 5
- microSD card, 32GB or more
- Ethernet cable
- At least one compatible smart device: **rpimag.co/hadevices**

Home Assistant offers a powerful alternative to other smart home solutions by focusing on privacy, flexibility, and control. Running locally, it keeps your data private and can still work when the internet is down. It supports thousands of devices and services, enabling seamless integration across brands. Highly customisable, it allows for advanced automations tailored to your needs. As an open-source platform, it's free, constantly evolving through community support, and avoids vendor lock-in, giving you full control over your smart home setup without being tied to a specific company's ecosystem.

Installing it on Raspberry Pi is a quick and painless process. We'll guide you through it and how to start using your new Home Assistant server to monitor and control your smart devices.

01 Prepare Raspberry Pi

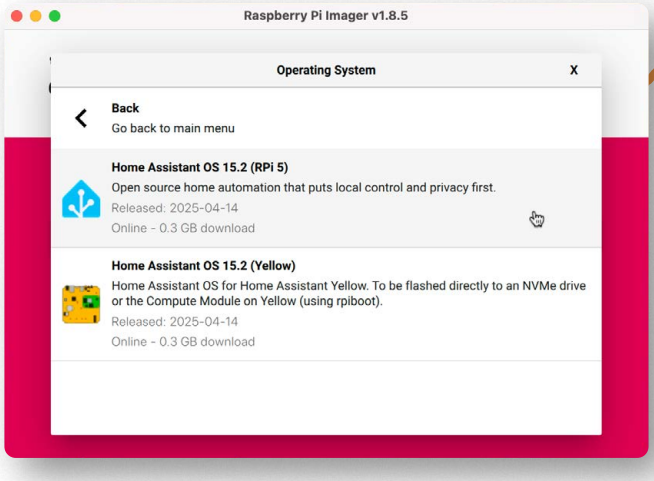
A Raspberry Pi 4 or 5 model is recommended for use with Home Assistant – other models lack the required RAM to ensure system stability. Along with that, you'll need a microSD card with at least 32GB capacity: this will give you sufficient storage for the operating system, database, logs, and potential add-ons. Ideally, the card should be A2 class, for faster read/write speeds.

For initial setup, you will also need an Ethernet cable to connect Raspberry Pi to your home router. You can subsequently opt to switch to a wireless connection if you want to move your Raspberry Pi somewhere else, away from the router (see 'Go wireless' box, overleaf).

02 Install Home Assistant

The Home Assistant OS will be written to your microSD card, so connect the latter to another computer – using a USB card reader adapter if needed. Installation is made easier thanks to the fact that you can use the standard Raspberry Pi Imager tool to write Home Assistant to the card. Download Imager for your computer system (Windows, Mac, or Ubuntu Linux for x86): **rpimag.co/software**.

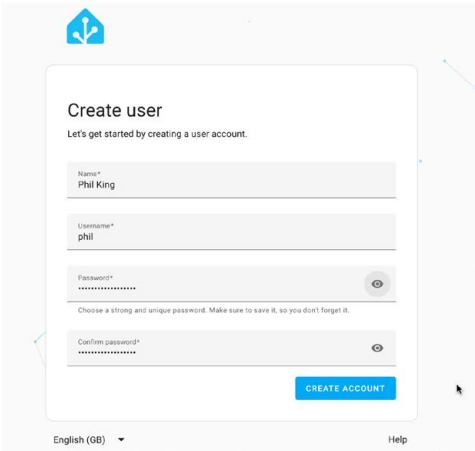
Open Raspberry Pi Imager, click 'Choose Device' and select your Raspberry Pi model to filter the OS options shown. Next, click 'Choose OS', then scroll down and select 'Other specific-purpose OS' > 'Home assistants and home automation' > Home Assistant. Select the version for your Raspberry Pi model. Finally, click on 'Choose Storage', then 'Next', and 'Yes' when the warning dialog appears. Imager will start writing the Home Assistant OS to the microSD card – it will take two to three minutes. Once it's written and verified, you'll get a 'Write Successful' dialog; you can now remove the card from the computer.



03 Connect to router

Take the microSD card with Home Assistant OS installed on it and insert it into the card slot on your powered-off Raspberry Pi. Connect an Ethernet cable to Raspberry Pi's Ethernet port, next to the USB ports, then insert the other end of the cable into an Ethernet port on your home internet router. Power up Raspberry Pi and Home Assistant OS should then start configuring itself.

Give it a few minutes, then open a web browser on another computer or device and point it to **homeassistant/local:8123**. If Home Assistant is still configuring itself, you may see a page with its icon on the top and blue/grey progress bar with three more icons below. It can up to 20 minutes for Home Assistant to be finalise its configuration, but ours finished in about ten minutes. Don't worry about any red exclamation marks appearing during the process.



04 Initial setup

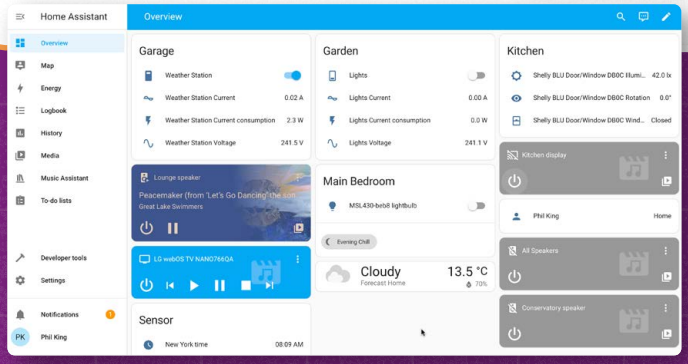
When the Home Assistant server is ready, its web page will prompt you to 'Create My Smart Home' (with a couple of restore options below). Click that blue button to start configuring your smart home system. First, you'll be asked to create a user: enter a name, username, and password. Then click 'Create Account'.

Next, you're prompted to set your location. While your data is stored locally on your Home Assistant installation, an accurate location may be needed for some cloud-based integrations. Enter a postcode or the start of your address and choose from the list, then click Next.

On the next screen, you can opt to share anonymised data to help improve Home Assistant, if you wish. Finally, a screen will appear that may say Home Assistant has discovered some compatible devices on your network. Click on Finish.

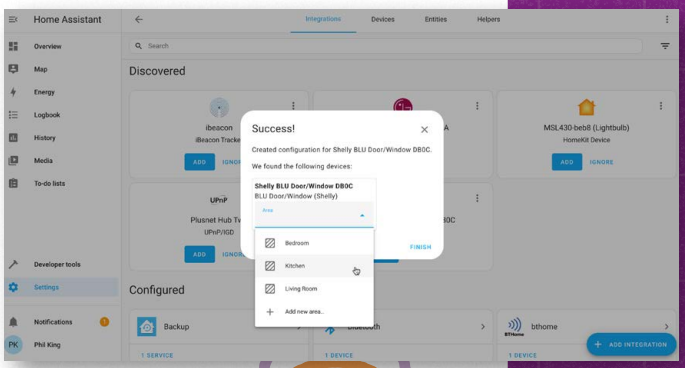
05 Web dashboard

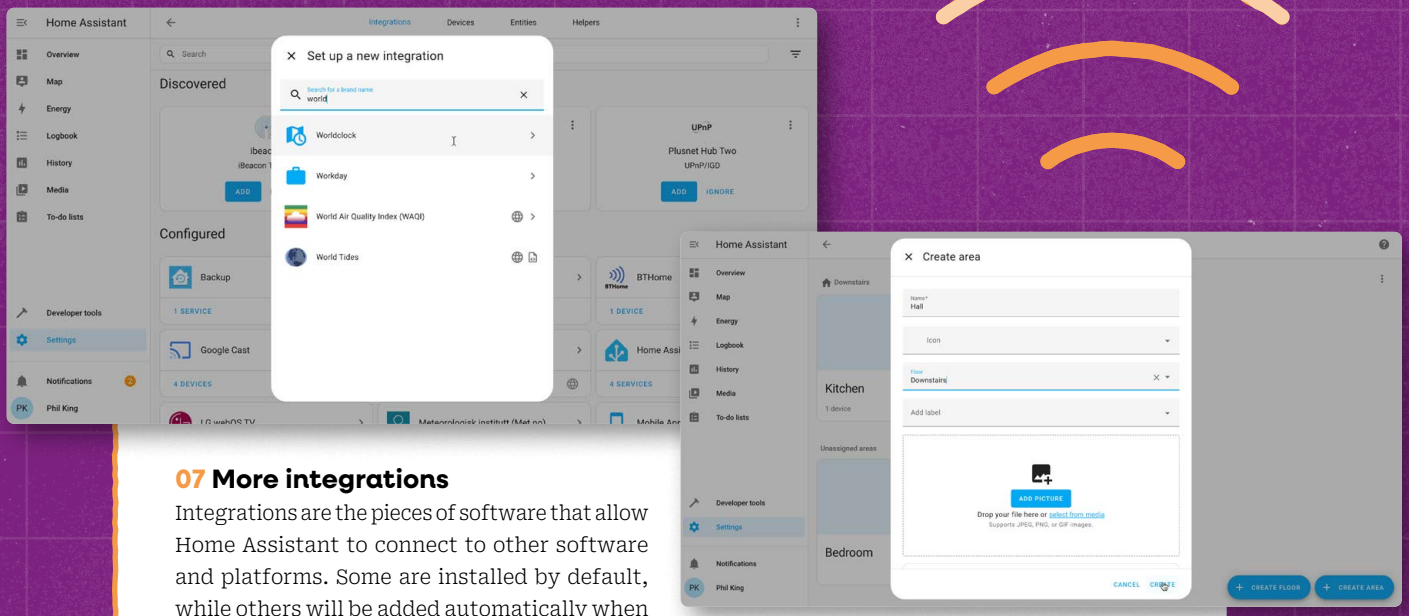
You should now see your Home Assistant dashboard showing any devices that have already been detected automatically on your home network – our Google Nest speakers showed up there, for instance – along with your name and a weather forecast for your location. By default, the dashboard is arranged automatically as you add more devices and entities, although you can create a custom layout later if you want and even change the theme to alter its appearance.



06 Add a device

In the left sidebar of the dashboard, you may see a number next to Notifications; click on that and it will tell you new devices have been discovered. Clicking on 'Check it out' takes you to the Integrations page, which shows icons and names for any new devices under 'Discovered'. You can click 'Add' for one to add it to your system – or 'Ignore' it, which you may want to do for things like your router. Upon clicking 'Submit' to add a new device, you'll see a 'Success!' dialog with the option to add it to an area in your home, using a drop-down menu. Click Finish and the device will be added to your dashboard – click on Overview to see it there.





07 More integrations

Integrations are the pieces of software that allow Home Assistant to connect to other software and platforms. Some are installed by default, while others will be added automatically when you add a device that uses one – for instance, an LG smart TV uses the LG webOS integration. Others may need to be added manually, while some may only be available via the optional Home Assistant Cloud premium service.

To browse the standard integrations available, click Add Integration on the Integrations page. You can then explore a long list of integrations or type in a search term at the top. To find out more about an integration, select it and then click the '?' icon.

As an example, let's add the Worldclock integration. After selecting it from the integrations list, choose a name (e.g. New York time'), time zone, and time format, then click Submit and Finish. It should now appear on the dashboard (Overview).

08 Areas, labels & zones

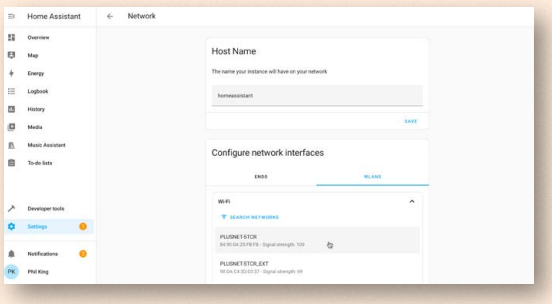
An area in Home Assistant can be used to represent a room or space, such as the garden, in which you can group devices located there. As well as grouping them together on the dashboard, this enables you to address a set of devices in one room when creating automations or scenes, such as turning on all lights in the living room. To add a new area, go to Settings > 'Areas, labels & zones' and click Create Area, then fill in the details (and add an optional picture). If you live in a multi-storey abode, you can also create floors and assign areas to them.

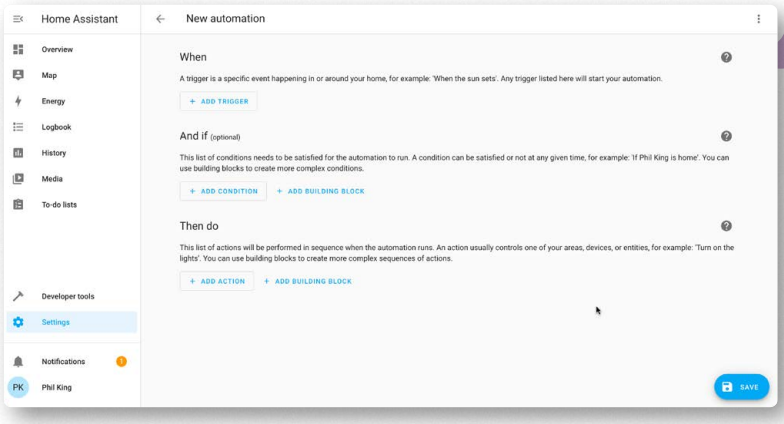
Zones are used to detect whether someone (added to Settings > People) is present in a certain location, typically using their mobile phone to track them. By default, you will have a Home zone – set during initial configuration, but editable by moving the marker on a map – and you can add more zones, perhaps for work or school.

Labels can be used to group related elements – devices, entities, areas, scenes, scripts, and helpers – together, even if they're not of the same type or in the same location. This can be used to filter the list of devices to only those with a certain label.

Go wireless

If you want to move your Raspberry Pi running Home Assistant away from your router, you can switch it from a wired Ethernet connection to a wireless Wi-Fi one. In Home Assistant, go to Settings > System > Network. Select the WLAN0 tab and Search Networks. Choose your router and security method (typically WPA-PSK), then enter your router password and click Save. Power down, unplug the Ethernet cable from Raspberry Pi, then power it up again. Your Home Assistant should now connect to the router via Wi-Fi.





09 Automations and scenes

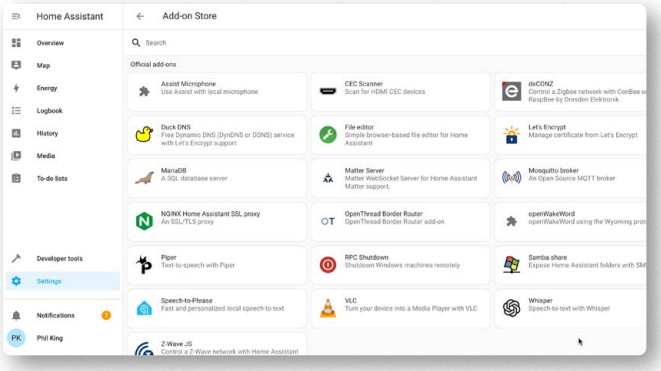
Home Assistant makes it easy to create automations to determine how devices behave when certain conditions are met. For instance, you could create one to turn on smart lights (or lamps connected to smart sockets) at sunset. To create an automation, go to Settings > 'Automations & scenes' and click Create Automation, then select 'Create new automation'. You then need to add a trigger event, plus optional extra conditions that need to be met, and set one or more actions to take. As an alternative to creating a custom automation from scratch, you can use or adapt one of the preconfigured 'blueprint' templates shared by the Home Assistant community.

Scenes enable you to set a state for a group of devices or entities. For instance, you could set a scene for dinner time by dimming lights and playing suitable music on a smart speaker. You can then use the scene as an action when creating automations.

In addition, you can create custom 'scripts' comprising a sequence of actions and/or conditions. These can be created from the GUI menus or by editing the YAML files – which you can also do for automations to fine-tune them.

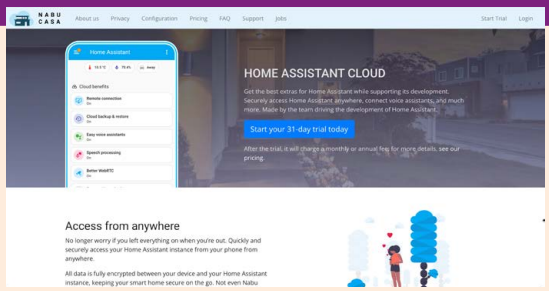
10 Installing add-ons

To customise your Home Assistant setup even further, there's a large range of add-ons that you can install to add extra functionality. Go to Settings > Add-ons and click the Add-on Store button to see the full list, arranged into categories. Official add-ons include a Mosquitto broker for sending MQTT messages between devices, Z-Wave JS to use devices with that wireless protocol (communicating via a Z-Wave USB dongle or HAT on Raspberry Pi), and the Music Assistant music library manager for your offline and online music sources. Just select an add-on and click Install to add it to your Home Assistant setup.



Remote access

While there are other methods, the easiest way to access your Home Assistant server remotely, when away from your Wi-Fi network, is to use the optional Home Assistant Cloud subscription service. This enables you to securely access your Home Assistant from your phone from anywhere. It also makes it easier to get it to work with voice assistants such as Alexa and Google Home Assistant, or use its alternative Assist privacy-focused voice assistant.



Smart devices

Just a few of the devices you can add to your smart home

Sensors

There are all manner of sensors you can add to a Home Assistant setup. They measure everything from temperature and humidity to soil moisture and light levels. Motion and contact sensors can be used for home security, as in our alarm tutorial over the page.

Smart socket

The great thing about a smart socket is that it can make a 'dumb' device smart, enabling you to turn it on and off via Home Assistant. As well as Wi-Fi sockets, there are Z-Wave and Zigbee models.

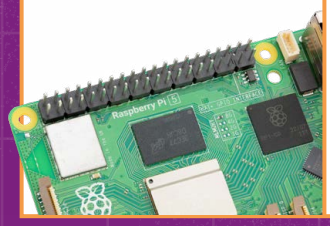
Lighting

Many modern homes have at least a few smart lights or bulbs that can be controlled via a phone app or voice assistant. These can usually be connected to Home Assistant and used in automations.

Motion and contact sensors can be used for home security

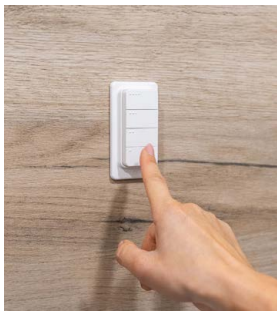
Audio/video

Smart speakers, displays, and TVs are ubiquitous these days. Most are easily connected to Home Assistant via integrations. Alexa speakers are a little trickier to set up, unless you use Home Assistant Cloud.



Raspberry Pi

You can use one or more secondary Raspberry Pi computers linked to Home Assistant, using the Remote GPIO integration to read and control their GPIO pins. There's also a Raspberry Pi Camera integration.



Input device

There are numerous smart buttons – connected via Bluetooth or another wireless protocol – that you can use to control devices with HA automations, with single or multiple button presses triggering different actions.

Wireless protocols

There are four main protocols used by smart devices to communicate with your Home Assistant server, or each other…

Wi-Fi: Offers good range (up to 100m) and high bandwidth, making it ideal for devices such as cameras and smart TVs. The downside is that it's power-hungry – not a problem if the device is mains-connected, however, such as a smart socket.

Bluetooth LE: Uses less energy than Wi-Fi, making it ideal for battery-powered devices. The downsides are lower bandwidth and shorter range: typically around 10m, depending on walls and other obstructions.

Zigbee: Used by certain smart devices, this operates as a mesh network with each device acting as a node, enabling it to cover a large area with energy efficiency. You'll need a Zigbee USB dongle or HAT for your Raspberry Pi to use it.

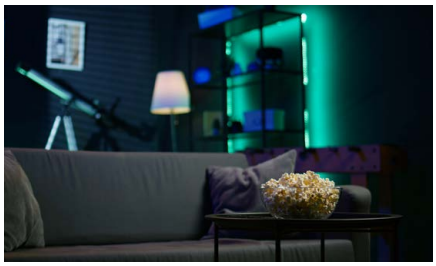
Z-Wave: Like Zigbee, this works as a mesh network between devices, thereby maximising area coverage and power efficiency. A Z-Wave USB dongle or HAT for your Raspberry Pi is required to communicate with such devices.

Automation ideas

Here are some suggestions for HA automation projects

Mood Lighting & Sound

With a few smart lights and a speaker or two, you can set the scene in a room for an occasion such as a romantic meal, party, or movie night. The scenes feature in Home Assistant makes it easy to trigger multiple actions from an automation.



Home Security

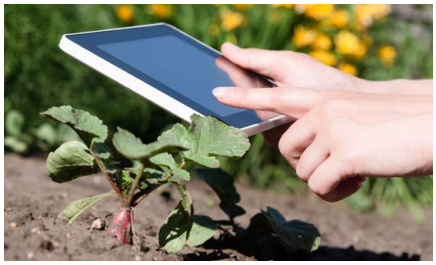
With a few sensors, plus optional cameras, you can create a DIY home security system using Home Assistant. Or maybe just use a motion sensor to trigger the lights to come on when you enter an area, such as a laundry room.

Automated Motion

For the ultimate smart home, you'll want to be able to control motorised items such as blinds, curtains, windows, and garage doors. Home Assistant has the integrations needed and can even work with some robotic lawnmowers!

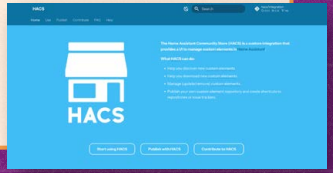
Smart Garden

Your smart home can extend into your garden or other outdoor space. With soil moisture sensors and relay-controlled water pumps, you could create an automated irrigation system to keep your plants healthy.



Community help

Home Assistant has a vibrant community of users who love to share their custom creations: integrations for otherwise unsupported devices, advanced integrations, dashboard cards and themes, scripts, and templates. This is made easier using the HACS (Home Assistant Community Store) custom integration – despite the name, everything in it is free and open source. For more details, visit **hacs.xyz**.



Create a home alarm

How to build a simple security system with Home Assistant

To showcase how to use the powerful automations feature in Home Assistant, we'll build a simple alarm system using a motion sensor and a smart speaker – plus optional mobile phone notifications for when you're away from home.

YOU'LL NEED

- Home Assistant running on Raspberry Pi
- Motion sensor
- Smart speaker, display, or TV
- Window/door sensor (optional)
- Smartphone with Home Assistant companion app

01 Add motion sensor

To start with, we'll set up a motion sensor to detect movement in the room we want to secure – our kitchen in this case. To create a more robust home security system, you could expand this with more sensors to cover other areas of the house.

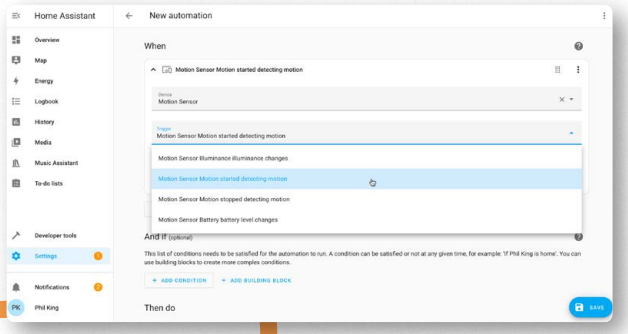
We're using a Shelly BLU Motion Detection Sensor, a passive infrared (PIR) sensor that communicates wirelessly via Bluetooth (**rpimag.co/blumotion**). There are other similar Bluetooth-based sensors, as well as Z-Wave, Zigbee, and Wi-Fi ones. Those longer-range wireless protocols might be preferable if you're having trouble with Bluetooth's limited range (typically around 10m), although it should prove adequate for most homes.

Our Shelly sensor has a beacon mode that broadcasts its status at regular intervals, but for our purposes it's better to use the default regular mode that only does so when the motion detection status changes, as this saves battery power – the sensor uses a standard CR2477 coin cell, which should last a couple of years.

Once you activate the sensor and it detects motion, it should appear as a discovered new device in Settings > 'Devices & services'. Click the Add button to add it to your setup; you can rename it and assign it to an area (in our case, Kitchen). Its illuminance reading and motion detection status should then appear on the dashboard.

02 Automation trigger

Let's start creating our automation for the intruder alarm. Go to Settings > 'Automations & scenes' and click the blue Create Automation button, then select 'Create new automation' in the dialog. As explained in our Home Assistant setup guide, each automation comprise a trigger and an action, along with optional conditions. So, in this instance, we'll set our sensor detecting motion as the trigger.



Under 'When', click the +Trigger button and select Device, then select your motion sensor's name from the Device drop-down list (we called ours 'Motion Sensor'). By default, our sensor trigger came up as 'Motion Sensor Illuminance illuminance changes', so click on that and select 'Motion Sensor Motion started detecting motion'.

03 Audible alert

For our automation action, we'll use a Google Nest speaker to play a text-to-speech phrase. Under 'Then Do', click the Add Action button. Select Media Player > 'Play media'. Select the desired media player from the drop-down list. Click 'Pick media' > Text-to-speech > Google Translate and then type the message (we chose 'Intruder alert!'), select the Language (there are several English accents), and click Finish.

Finally, click the Save button and name your new automation (e.g. 'Kitchen alarm'); you can add an optional area, description, etc. if you want.

To test it out, wave a hand over your motion sensor and you should hear the smart speaker say 'Intruder alert!' There may be a slight delay the first time, but it should say it almost instantly on subsequent triggerings – note that you may need to wait 30 seconds for the motion sensor status to revert to 'Clear' before it will trigger again.

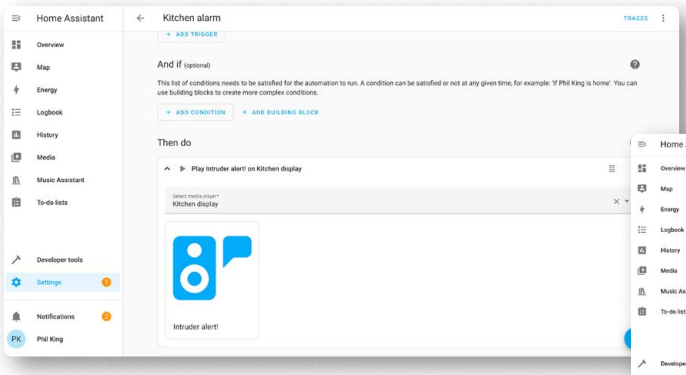
04 Send phone notification

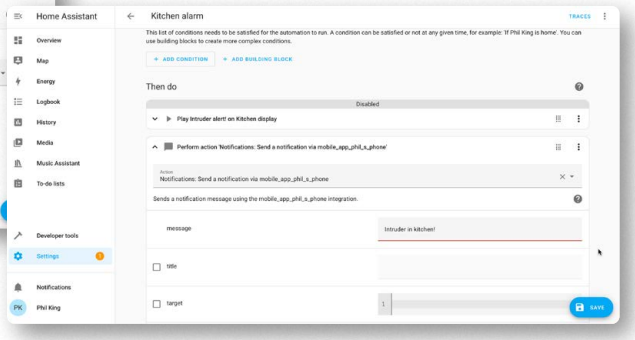
An audible alert is all very well if you're in the vicinity, but what if you're away from home? For this, you can send a notification to your mobile phone. There are various ways to do this, but possibly the easiest is to use the Home Assistant companion app, available for Android and iOS. Install that on your phone and log in to your HA web server; you can then rename the device (optional) and enable location tracking for presence detection (if you want).

You'll then be able to view your HA dashboard. Even though you won't be able to connect to the server when away from your home Wi-Fi zone, you can still receive notifications via the app.

Return to your automation and click Add Action again. Then select 'Other actions' > 'Perform action'. From the Action drop-down list, select 'Notifications: send a notification via mobile_app_[your_phone_name]'. Then type some text into the 'message field'. Finally, click Save.

Test it out by triggering the sensor again. Even if you have Wi-Fi disabled on your phone, you should still receive a notification via the Home Assistant app. If not, double-check that notifications are enabled for the app.





05 Add extra sensors

Now you have a working alarm system, you could add one or more extra sensors to trigger the same action – or for a new automation.

For our system, we added a Shelly BLU Door/Window Sensor (**rpimag.co/bluwindow**). This Bluetooth sensor has two parts: one is attached (via a sticky pad) to the outer edge of a window or door, while the other is positioned next to it on the surrounding frame. So, when you open the window or door, the two parts separate and the sensor's status changes from 'Closed' to 'Open'. It can even estimate the angle of opening. After triggering the sensor, HA discovered it and we added it to the dashboard, as before. We then added it as a trigger, with the 'Window opened' option in our existing automation.

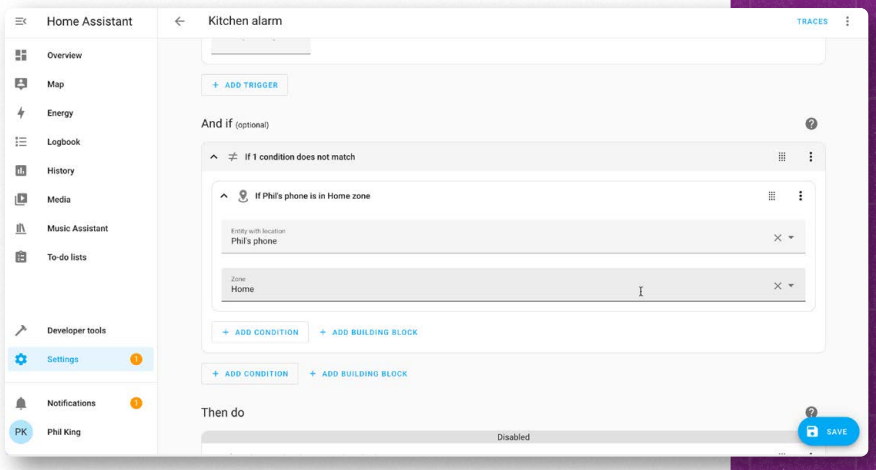
You could add more triggers and actions, as you prefer. Perhaps you might want it to turn on the lights or trigger a camera (HA works with brands such as Ring, as well as Raspberry Pi Camera).

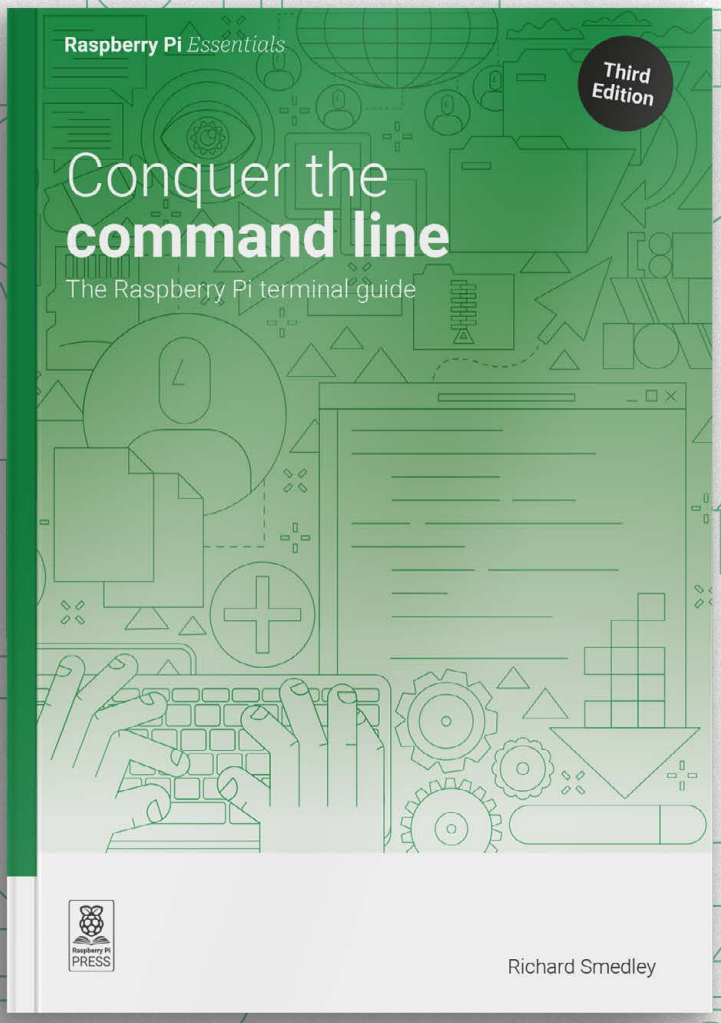
> To fine-tune how your alarm operates, you can add conditions for when it can be triggered

06 Add conditions

To fine-tune how your alarm operates, you can add conditions for when it can be triggered. For instance, you might limit it to certain times of the day or when nobody's home (using mobile phone presence detection). in the automation options, click Add Condition, then 'Time and location'. Select 'Time' to limit triggering to before and/or after a certain time, on selected days.

If you want to use presence detection to only trigger the alarm when you're away from home, you'll need to first add a Building Block and set to 'Not'. Now select Add Condition within that block and choose 'Time and location' > Zone. Then select the Entity (your phone) and the Zone (Home).




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

Part 01

Build a Raspberry Pi classifier

Detect different Raspberry Pi models with Raspberry Pi AI Camera. By **David Plowman**

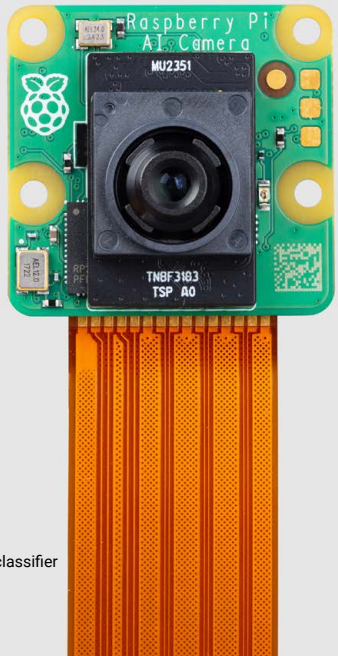
Maker

David Plowman

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

raspberrypi.com

- Raspberry Pi AI Camera contains a Sony IMX500 sensor with AI inferencing ability



In this tutorial, we will use Raspberry Pi AI Camera with a pre-trained model to detect different Raspberry Pi models. Point Raspberry Pi AI Camera at a Raspberry Pi 4 and it will display 'Raspberry Pi 4' on the screen, or 'Raspberry Pi Zero' and so on.

This is a powerful example of how Raspberry Pi can be trained to perform real-time inference of object detection. It's a good way to explore the Raspberry Pi AI Camera API (application programming interface) and the relationship between models and inference.

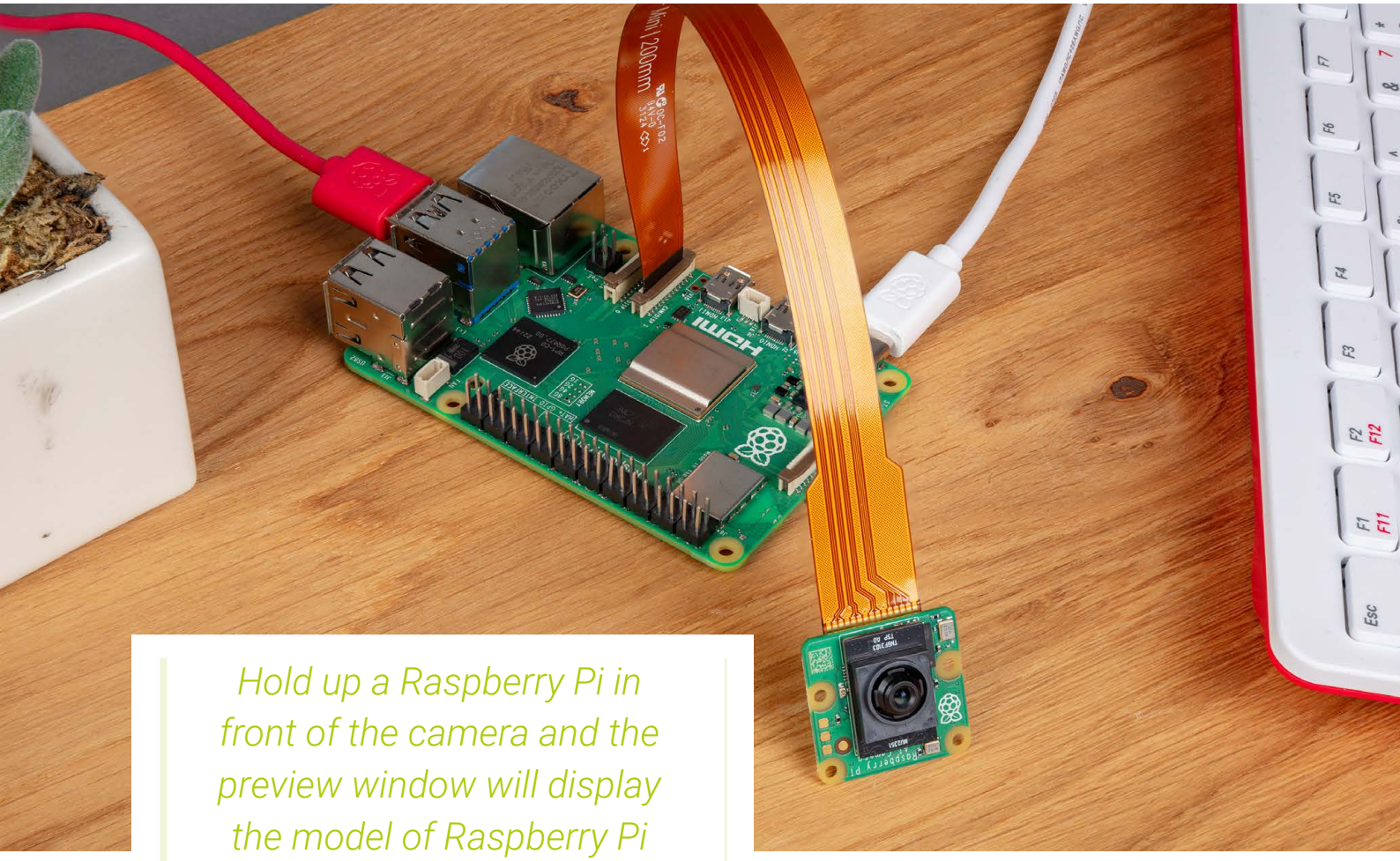
Set up AI Camera

Connect your Raspberry Pi AI Camera to a compatible Raspberry Pi computer (**rpimag.co/aicamera**). We are using a Raspberry Pi 5, but the AI Camera is compatible with all Raspberry Pi models. You can read the AI Camera datasheet for more information (**rpimag.co/aicamdatasheet**).

The Raspberry Pi AI Camera features a Sony IMX500 image sensor that provides low-latency, high-performance AI capabilities to camera applications without needing the AI HAT+ (or AI Kit).

It runs custom neural network models on the IMX500 sensor. Make sure your Raspberry Pi is up-to-date and install the IMX500 firmware:

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



▲ Raspberry Pi AI Camera attached to Raspberry Pi 5

Hold up a Raspberry Pi in front of the camera and the preview window will display the model of Raspberry Pi corresponding to the options in the labels.txt file

This command:

- installs the **/lib/firmware/imx500_loader.fpk** and **/lib/firmware/imx500_firmware.fpk** firmware files required to operate the IMX500 sensor
- places a number of neural network model firmware files in **/usr/share/imx500-models/**
- installs the IMX500 post-processing software stages in rpicam-apps
- installs the Sony network model packaging tools.

Now reboot your Raspberry Pi:

```
$ sudo reboot
```

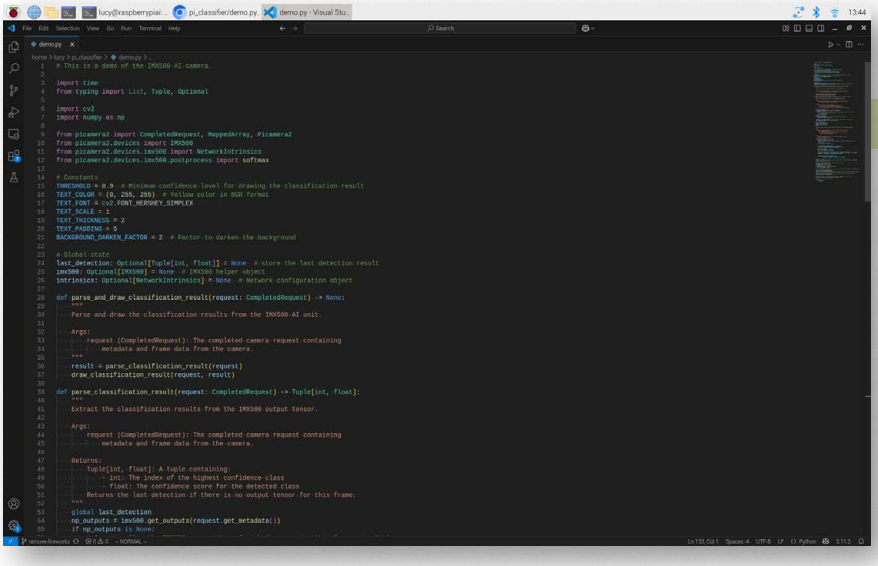
Let's test it is all working. The following command runs rpicam-hello with object detection post-processing:

```
$ rpicam-hello -t 0s --post-process-file
/usr/share/rpi-camera-assets/imx500_mobilenet_
ssd.json --viewfinder-width 1920 --viewfinder-
height 1080 --framerate 30
```

The following command runs rpicam-hello with pose estimation post-processing:

```
$ rpicam-hello -t 0s --post-process-file
/usr/share/rpi-camera-assets/imx500_posenet.json
--viewfinder-width 1920 --view
```

▼ Inspecting the code in Visual Studio Code



Get the code

Now that your AI Camera is set up, you'll need the code. For this tutorial it's best to start by copying the files from GitHub. In terminal, enter:

```
$ git clone https://github.com/davidplowman/
pi_classifier.git
```

Enter the **pi_classifier** directory and take a look inside:

```
$ cd pi_classifier
$ ls
```

Here you will find three files:

- **demo.py** – the Python code file
- **labels.txt** – a text file containing the labels for different Raspberry Pi models to identify
- **network.rpk** – a 'runtime package' file trained to detect Raspberry Pi models using the Sony IMX500 image sensor.

Run the code with Python:

```
$ python demo.py
```

A preview window will appear on the screen, displaying the view from the AI Camera. Hold up a Raspberry Pi in front of the camera and the preview window will display the model of Raspberry Pi corresponding to the options in the **labels.txt** file.

How it works

Let's open up the **demo.py** code in Thonny (or a code editor of your choice) and take a look at it:

```
$ thonny demo.py
```

parse_and_draw_classification_result

This function is given to the camera system as a callback function, so that it gets called automatically every time a new frame arrives from the camera, with us (the application) having to do anything extra. Because we're using the Sony IMX500 sensor, we get not only the usual camera image, but also an output tensor listing the results of running the neural network on the image.

We need to parse the neural network results using `parse_classification_result`, then draw them onto the camera image using `draw_classification_result`.

parse_classification_result

This analyses the output tensor produced by the IMX500 sensor, and decides if the image belonged to one of categories that the network was trained on. There are a couple of things to watch out for.

Firstly, occasionally there isn't an output tensor. This can happen if the neural network accelerator didn't quite finish processing the frame in time. When this happens, we should just reuse the result from the previous frame; another frame, where the tensor is once again available, will be along shortly!

Next, we return the category with the highest confidence value. Some networks may tell us that we should apply a softmax operation to the output values to get a more realistic 'probability' for each class.

draw_classification_result

Here we draw a label indicating the category and confidence value onto the camera image, but only if the confidence value is sufficiently high.

We get access to the actual camera image pixels (not a copy of it) in the usual way, using Picamera2's `MappedArray`. We measure the size of the box that the category label and confidence value will occupy, and darken that part of the image so that the text will stand out better when we subsequently write it on top.

▶ Detecting a Raspberry Pi 3

Finally, we let the script run forever

if __name__ == "__main__"

This section of the code is simply concerned with configuring the IMX500, uploading the neural network firmware file (**network.rpk**), and setting the camera going. There are just a few things to point out.

1. We first create the IMX500 camera helper object. This will upload the network for us, and also tells us which camera number the IMX500 is.
2. The `NetworkIntrinsics` object is a helper for the network, and will recommend an appropriate frame rate that we can use which should match the inference rate of the network on the IMX500.
3. Before starting the camera, we should tell the IMX500 helper that we would like to see a progress bar for the network upload – otherwise it's easy to think it might have got stuck! Recall that the network firmware uploads more quickly on subsequent occasions, as more of the firmware blocks are already cached within the device.
4. Finally, we let the script run forever, and all the camera processing happens in background threads.

demo.py

> Language: **Python**

DOWNLOAD THE FULL CODE:
rpimag.co/piclassifier

```
001. # This is a demo of the IMX500 AI camera.
002.
003. import time
004. from typing import List, Tuple, Optional
005.
006. import cv2
007. import numpy as np
008.
009. from picamera2 import CompletedRequest,
     MappedArray, Picamera2
010. from picamera2.devices import IMX500
011. from picamera2.devices.imx500 import
     NetworkIntrinsics
012. from picamera2.devices.imx500.postprocess
     import softmax
013.
014. # Constants
015. THRESHOLD = 0.9  # Minimum confidence level
     for drawing the classification result
016. TEXT_COLOR = (0, 255, 255)  # Yellow color
     in BGR format
```

```
017. TEXT_FONT = cv2.FONT_HERSHEY_SIMPLEX
018. TEXT_SCALE = 1
019. TEXT_THICKNESS = 2
020. TEXT_PADDING = 5
021. BACKGROUND_DARKEN_FACTOR = 2  # Factor to
     darken the background
022.
023. # Global state
024. last_detection: Optional[Tuple[int, float]]
     = None  # store the last detection result
025. imx500: Optional[IMX500] = None  # IMX500
     helper object
026. intrinsics: Optional[NetworkIntrinsics] =
     None  # Network configuration object
027.
028. def parse_and_draw_classification_result(
     request: CompletedRequest) -> None:
029.     """
030.     Parse and draw the classification
     results from the IMX500 AI unit.
031.
032.     Args:
033.         request (CompletedRequest): The
     completed camera request containing
034.             metadata and frame data from the
     camera.
035.     """
036.     result = parse_classification_result(
     request)
037.     draw_classification_result(
     request, result)
038.
039. def parse_classification_result(request:
     CompletedRequest) -> Tuple[int, float]:
040.     """
041.     Extract the classification results from
     the IMX500 output tensor.
042.
043.     Args:
044.         request (CompletedRequest): The
     completed camera request containing
045.             metadata and frame data from the
     camera.
046.
047.     Returns:
048.         Tuple[int, float]: A tuple
     containing:
049.             - int: The index of the highest
     confidence class
050.             - float: The confidence score
     for the detected class
051.         Returns the last detection if there
     is no output tensor for this frame.
052.     """
053.     global last_detection
054.     np_outputs = imx500.get_outputs(
     request.get_metadata())
055.     if np_outputs is None:
056.         # Occasionally, the IMX500 may not
     have finished processing this frame, in which
057.         # case we should re-use the last
     result.
058.         return last_detection
059.
060.     np_output = np_outputs[0]
061.     if intrinsics.softmax:  # this network
     recommends applying softmax
062.         np_output = softmax(np_output)
063.
064.     top_index = np.argmax(np_output)  # get
     the index of the highest confidence class
065.     last_detection = (top_index, np_
     output[top_index])
066.     return last_detection
067.
068. def draw_classification_result(
069.     request: CompletedRequest,
070.     result: Tuple[int, float]
071. ) -> None:
072.     """
073.     Draw classification results on the
     output frame with a semi-transparent
     background.
074.
075.     Args:
076.         request (CompletedRequest): The
     completed camera request containing
077.             the frame data to draw on.
```

DOWNLOAD THE FULL CODE:

rpimag.co/piclassifier

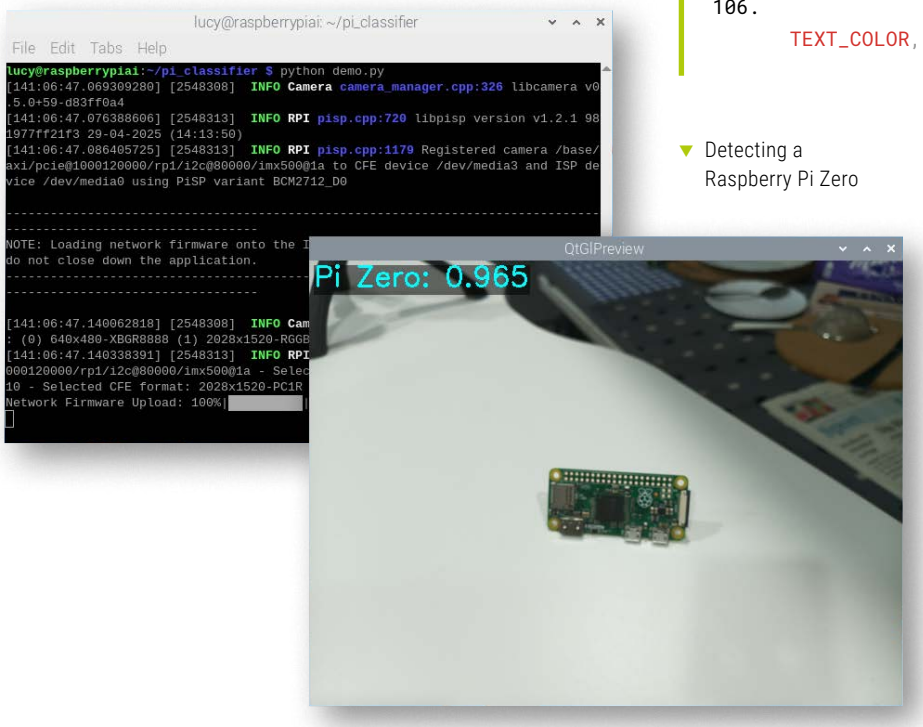
```
        result (Tuple[int, float]): The classification result containing:
            - int: The index of the detected class
            - float: The confidence score

    Note:
        The function only draws results if the confidence score is above
        the global THRESHOLD value.
    """
    with MappedArray(request, "main") as m:
        # Create and validate the text label
        label = intrinsics.labels[result[0]] if result and result[1] > THRESHOLD else None
        if label is None:
            return
        text = f"{label}: {result[1]:.3f}"

        # Calculate text size and position
        (text_width, text_height), baseline = cv2.getTextSize(
            text, TEXT_FONT, TEXT_SCALE, TEXT_THICKNESS
        )
        text_x = TEXT_PADDING
        text_y = text_height + TEXT_PADDING

        # Darken the background where the text will be to make it more readable.
        rectangle = m.array[text_y - text_height:text_y + baseline, text_x:text_x + text_width]
        rectangle[...] = rectangle // BACKGROUND_DARKEN_FACTOR

        # Draw text on top of the background.
        cv2.putText(m.array, text, (text_x, text_y),
                    TEXT_FONT, TEXT_SCALE, TEXT_COLOR, TEXT_THICKNESS)
```



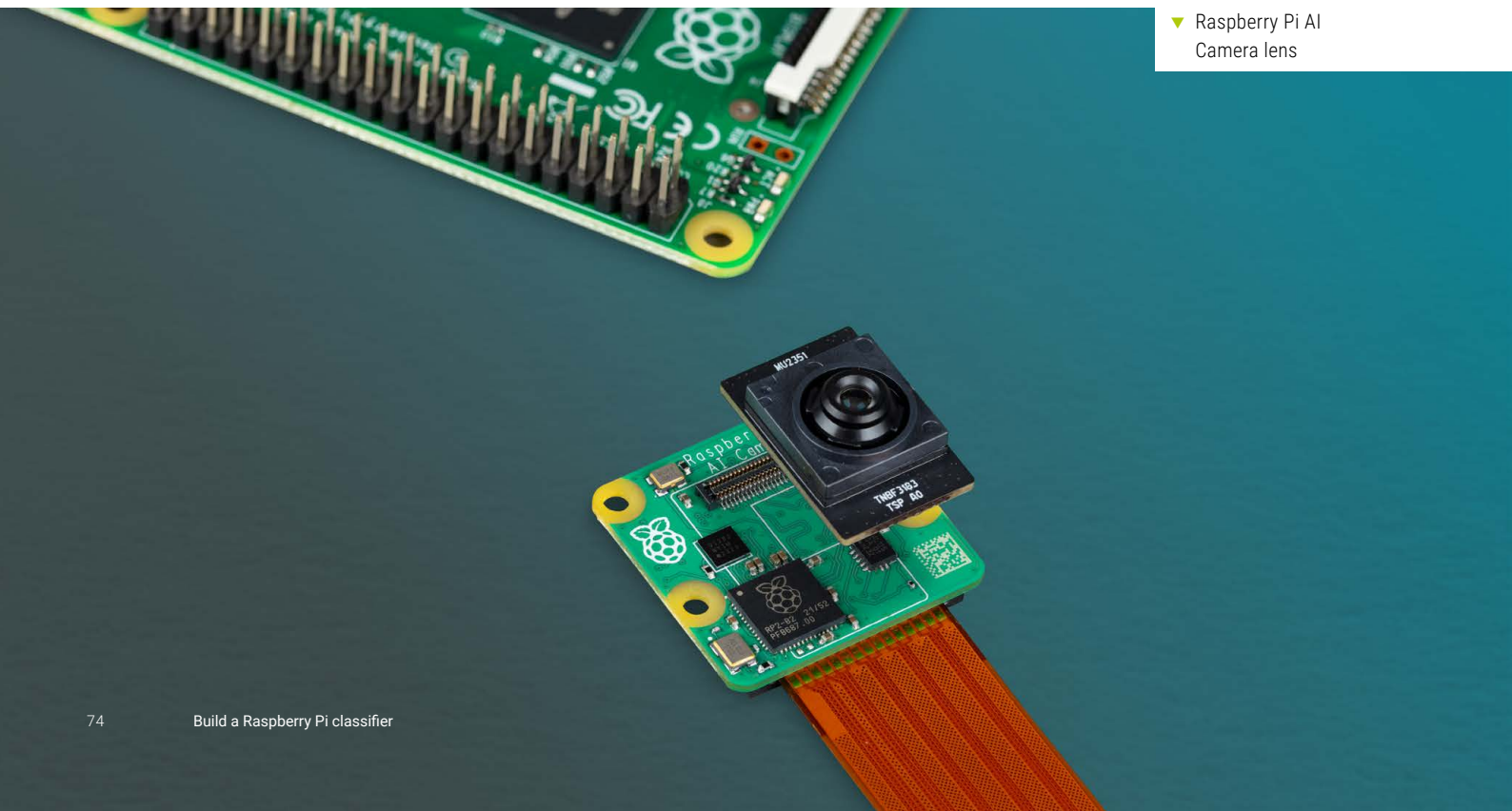
▼ Detecting a Raspberry Pi Zero

A preview window will appear on the screen

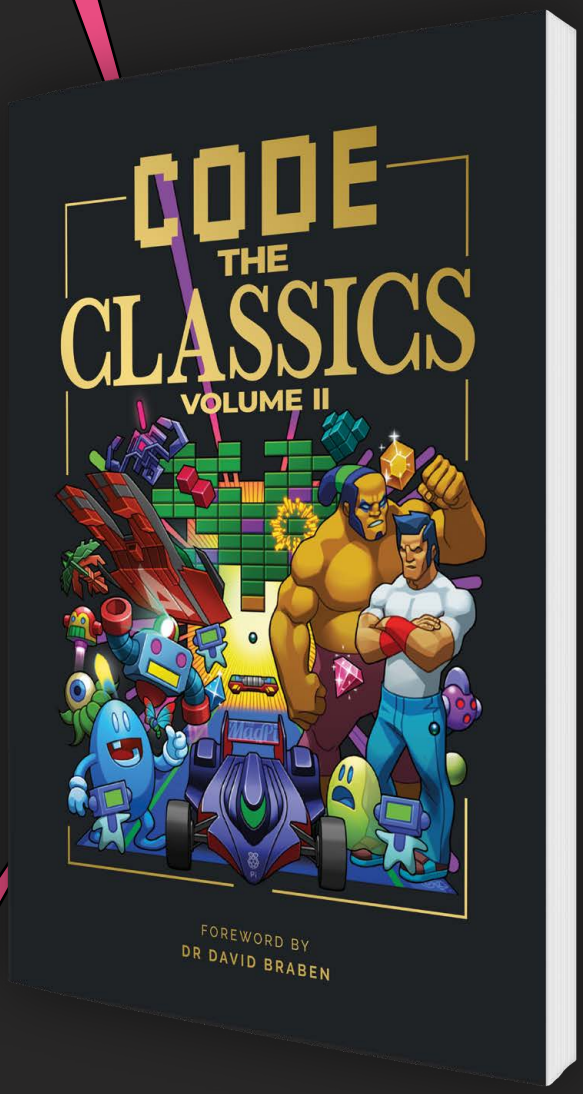
DOWNLOAD THE FULL CODE:

rpimag.co/piclassifier

```
107.
108. if __name__ == "__main__":
109.     # Create the IMX500 helper object first
     so that it gives us the camera number of the
     IMX500.
110.     imx500 = IMX500("network.rpk")
111.
112.     # The intrinsics contains useful
     information about the network.
113.     intrinsics = NetworkIntrinsics()
114.     intrinsics.task = "classification"
115.     with open("labels.txt", "r") as f:
116.         intrinsics.labels = f.read().
     splitlines()
117.     intrinsics.update_with_defaults()
118.
119.     # The intrinsics also contains the
     inference rate of this network, which should
     be a good choice
120.     # for the framerate of the camera.
121.     picam2 = Picamera2(imx500.camera_num)
122.     config = picam2.
     create_preview_configuration(
     controls={"FrameRate":
     intrinsics.inference_rate})
123.
124.     imx500.show_network_fw_progress_bar()
     # Show the progress of the network being
     loaded
125.     picam2.start(config, show_preview=True)
126.
127.     # Set the callback function to handle
     the results of the IMX500 on every frame.
128.     picam2.pre_callback =
     parse_and_draw_classification_result
129.
130.     # Sleep forever while the camera runs.
131.     while True:
132.         time.sleep(1)
```



▼ Raspberry Pi AI Camera lens



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

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

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

Unusual tools: cutting mat glue

A consumable that supercharges your cutting mat

Maker

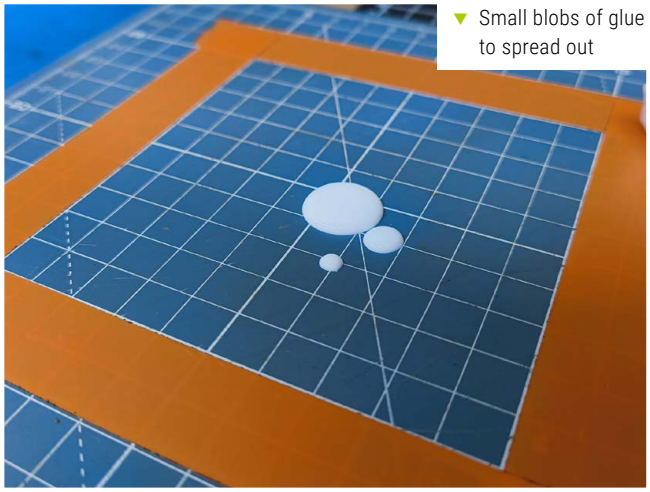
Dr Andrew Lewis

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

lewis-workshop.com

This glue has so many uses that it's difficult to put them all into a single article. Keeping materials in place, stopping tools and small parts from slipping, making fabric or string behave while it's being draped for photographs, and even just making a sticky notice board. Cutting mat glue is possibly the best glue that you never knew about.

If you've ever set out to make a paper model or cut out a fine pattern in a piece of thin material, you'll know that one of the biggest problems you face is the material moving around while you're trying to cut. You can tape the edges, use weights or magnets, pin through the material into a cork sheet, or try any number of other approaches, but you'll always end up with the same problems. Something, somewhere will move. No matter how many times you remember to change the blade on your craft knife, the drag from your blade will pull the material slightly out of shape and something delicate will rip. Curses will ensue, pets and loved ones will flee. Even 1970s working-men's club comedians, sea-hardened sailors, and Gordon Ramsay's book editor will pale at the utterances born from over-concentration, frustration, and wasted time.



▼ Small blobs of glue to spread out

Cutting mat glue comes to the rescue in these sorts of scenario, and it's one of those tools that suddenly becomes an indispensable part of your toolkit. It's a temporary glue that isn't particularly tacky, and doesn't leave a residue on the things that it sticks to. It's available as a spray or as a liquid glue, and both of these work just fine on cutting mats and other smooth surfaces like worktops and glass

QUICK TIP

Be careful using isopropyl alcohol on cutting mats, as some types will melt or lose their painted markings when they are exposed to it.

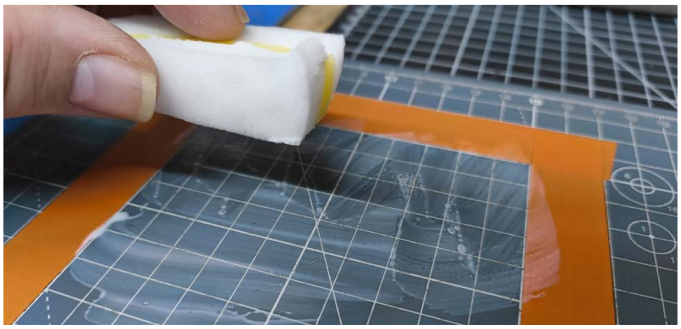
◀ Although this unusual tool is technically a consumable, it gets a special mention because it has the power to upgrade your cutting mat in a way that makes cutting intricate patterns so much easier

sheets. You can use it to hold down flat sheet materials while you work them, and it will also stop small objects like balsa blocks or plastic model parts from moving around while you shave, scrape, or paint them.

A little does a lot

Typically, cutting mat glue comes is small bottles, and it can seem to be expensive when you look at the size of the bottle. Keep in mind that 50ml of liquid will cover quite a large area when spread out with brush or scraper, and one bottle is plenty to cover several large cutting mats with multiple coats of glue. To apply the glue, shake the bottle for a few seconds and then pour a few drops onto your cutting mat. Obviously, the size of the mat will determine how much glue you will need to use, but as a starting point, begin with about a 50p/25¢-sized blob on an A2 cutting board and add more if you need to. If this is the first time you've ever coated a surface with the glue, then the manufacturers will normally suggest that you add two coats, although this can sometimes make the cutting mat too aggressively sticky. One coat of the glue will give a low-tack grab suitable for light paper and similar materials. Two coats are more suitable for getting a firm grip on other materials.

Although the glue is reusable and a single coat of it should last for along time, you'll probably find that it stops being sticky after a while because bits of dust and other dirt settle on the adhesive surface. You can prevent this from happening by covering the mat with a vinyl or silicone sheet when you aren't using it. However, a quick wipe over with isopropyl alcohol is often enough to bring the sticky surface back into working condition. If you absolutely need to, you can reapply another layer of glue to the top of the mat and wait a few minutes for it to dry. If you decide that you no longer want the mat to be sticky, you can wipe the surface clean by rubbing at it with your thumb, or an alcohol wipe.



▲ Apply the glue with a scraper or sponge. You can use a little bit of masking or electrical tape to help keep the glue from ending up in places that it shouldn't and to give you a clean edge. Remove the tape while the glue is still wet

An interesting effect of this glue is that it stops small screws and components from rolling around your cutting mat. Making just a few squares of the mat adhesive is a great way to minimise the chance of small parts disappearing when you're doing a repair.

> *One coat of the glue will give a low-tack grab suitable for light paper and similar materials*

▼ Another great use for this type of glue is as a backing on stencils. The low tack will stick the stencil in place while you apply the paint, but it won't hold the stencil so tightly that you can't get it back off again

Part 02

Meshtastic

Build a standalone solar Raspberry Pi Pico node

Maker

Jo Hinchliffe

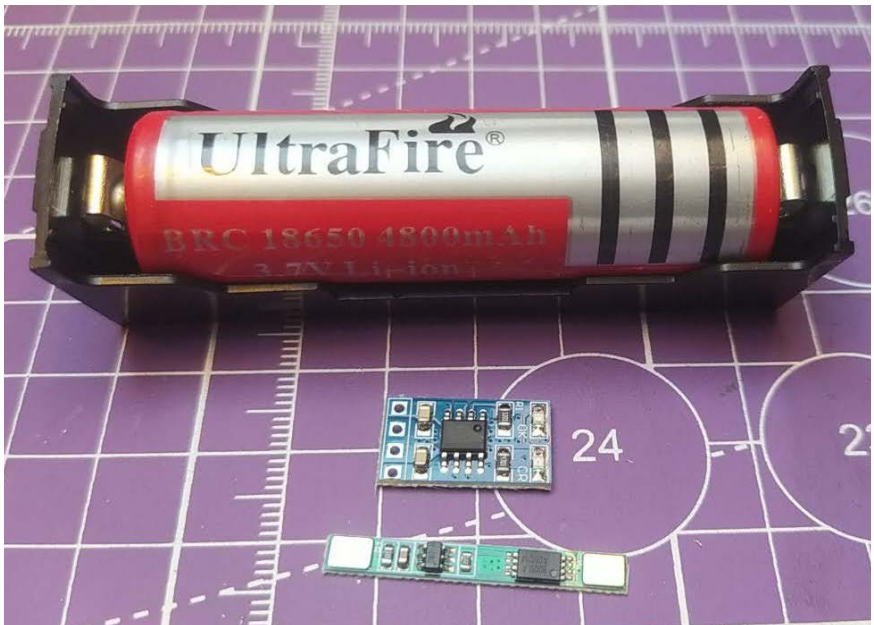
With a house and shed full of lathes, milling machines, 3D printers, and more, Jo is a constant tinkerer and is passionate about making. Obsessed with rockets and robots and much more besides, he often releases designs and project as open source.

concretedog.blogspot.com

In the first part of this guide (in issue 152), we set up a Raspberry Pi Pico W with a Waveshare SX1262 LoRa Node Module. We're going to add some other simple parts to this device and place it in a simple ABS enclosure so it can be deployed outside. This then can be considered a repeater node that we wouldn't directly connect to with the app or anything, but rather, can be placed so it extends range, or solves range problems like line of sight, increasing our mesh network (**Figure 1**).

There are lots of examples online of people building Meshtastic nodes running on batteries and or solar power. Meshtastic on Raspberry Pi RP2040 isn't the most efficient setup, but for this project we aren't aiming for a totally optimised system that can last indefinitely, but rather a device we can deploy quickly that will run for at least a day or two.

As we also aren't too worried about size, having found a pretty large ABS enclosure in a drawer, we can opt for a larger format battery. For a tiny handheld unit we might look at a small single cell LiPo, or perhaps an AA form factor lithium ion battery such as a 14500 cell. We've opted for an 18650 cell. 18650s are very common and come in a range of capacities. Our Ultra Fire cell is listed as 2700mAh. Again, as we have space, we opted to place this cell in a 18650 cell holder rather than spot-welding tabs to the cell permanently (**Figure 2**). This also means that with a spare cell or two, we can increase the uptime of the Meshtastic node by simply replacing the batteries.



▼ **Figure 2:** Our 18650 cell, battery holder, cell protection board, and solar controller that form the basis of our power system

QUICK TIP

If you want to create a very optimised solar Meshtastic node, other MCUs such as the RAK family of Meshtastic boards are very efficient.

In a mountainous region such as here in North Wales, it's easy to walk into valleys or 'cwms' where there is no line of site to any other Meshtastic nodes

▲ **Figure 1:** Our complete standalone node deployed on a hilltop

▶ **Figure 3:** Using a small piece of double-sided sticky foam, we then soldered the cell protection circuit to the battery holder

Battery power

It's good practice to use a cell protection system when using lithium-ion cells and they are available very affordably. The cell protection system will stop the 18650 from being drained too low, with an overdischarge function kicking in and cutting the power output if the cell reaches 2.4V. Likewise, the cell protection board will protect the cell in an overcharging situation and can detect short circuit scenarios.

We used some very sticky M3 double-sided pads to affix the small cell protection board to the side of our 18650 cell holder. The cell protection boards have some large pads labelled B- and B+ and we simply soldered small wires from the device back to the corresponding pins on the battery holder (**Figure 3**).

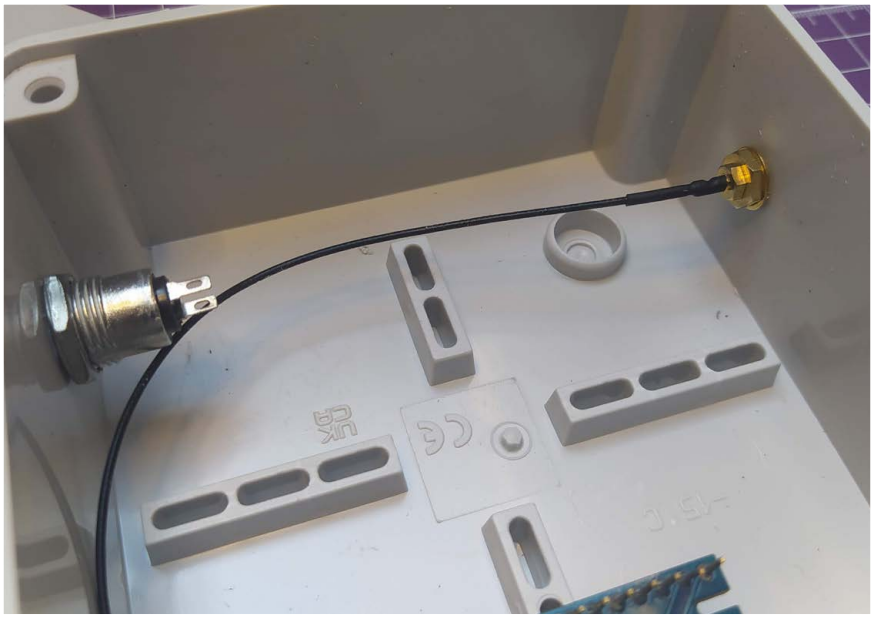


Figure 4: We drilled the enclosure walls to receive a 12mm barrel key switch and the antenna SMA connector

Next, we soldered two pairs of wires to the P+P- terminal on the cell protection board. Obviously it's a good idea to not have the cell fitted at this point in case any of your stripped wire ends accidentally touch, although the cell protection board should stop any damage to the cell. These two pairs of wires will see one set used as the power supply to our Pico H/Waveshare boards via a power switch and the other pair as an input from a solar power charge controller.

For a power switch we used a small key switch. This key switch has an off and on position and, importantly, the key can be removed in either position (**Figure 4**). We simply soldered the switch (after fitting the switch to the enclosure) into the positive wire between P+ and the pin 39 input on the Pico H/Waveshare board. This means we can turn the whole node on and off, but also means that if we deploy the node somewhere in the field, it's unlikely that anyone else could turn the device off and, once the key is removed, it's impossible for the device to be accidentally turned off through movement or a knock. As the Pico H/Waveshare stack has header pins fitted, we opted to simply solder some jumper socket connectors to the power supply wires and could then slot them onto the positive pin 39 and the ground pin 38.

The remaining pair of wires connected to the P+ and P- pads on the cell protection board are wired into our SD05CRMA solar charge controller, which is again an affordable and readily available small board. The solar charge controller only has four pins: a positive and negative input to receive power from a 4–6V capable solar cell, and a positive and negative output for charging an attached single cell. We again soldered two header sockets to the positive and negative inputs and then permanently soldered the charge output to the wires we soldered to the P+ and P- pads on the cell protection system. This means that we can easily wire a solar panel to the device to charge the 18650 cell when required and that the cell can be charged regardless of whether the Meshtastic node is powered on or off.

With our wiring largely complete, we began to assemble our device into the ABS enclosure (**Figure 5**). The enclosure is perhaps a little large, but that makes it easier to work inside and also would allow us to expand the system down the line, perhaps opting for multiple 18650 cells to increase the available runtime. We drilled a 7mm hole for the antenna SMA connector and a larger 12mm hole for the key switch. An additional 7mm hole was also drilled on the underside of the system which would allow us to connect the solar panel (**Figure 6**).

Figure 5: The complete wired-together node assembled in the enclosure

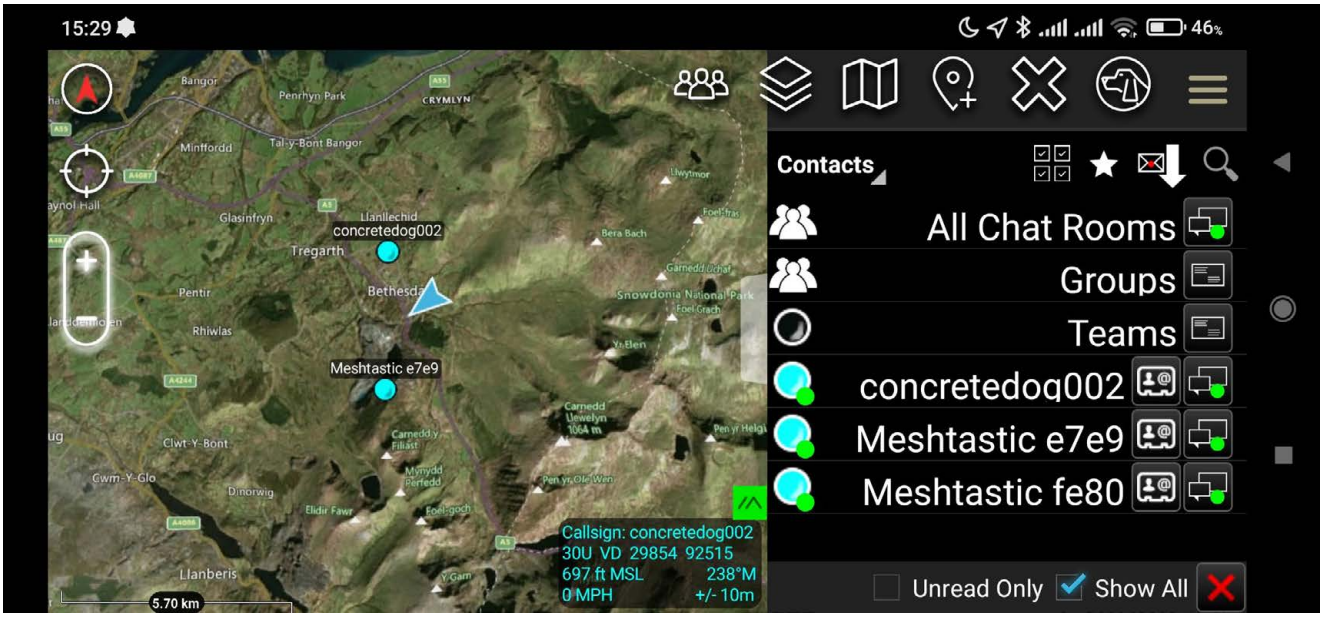
▲ **Figure 6:** Any 4–6V solar cell or combination of cells can be connected into the node, extending the runtime. Here a 5W solar panel originally for a CCTV camera has been used

When it's not in use, we can cover this hole with a piece of tape, or perhaps make a custom grommet to stop any moisture ingress. The Pico and Waveshare module were mounted to the edge of the box with some double-sided foam, as is the solar charge controller. The battery case was fitted with hot glue as it needs to be robustly connected to resist the forces of inserting and removing the cell.

We wanted to be able to quickly deploy the node in the field and so some 3D-printed PETG attachment points were printed and glued to the back of the ABS enclosure. Through this we then added some 6mm shock cord (**Figure 7**). The shock cord arrangement allows the node to be quickly mounted on any vertical pole-like structure – a fence post, for example. In our case, using a trekking pole creates a simple installation for when camping or for impromptu Meshtastic comms sessions taking a break whilst hiking.

◀ **Figure 7:** Some PETG 3D-printed eye mounts were superglued onto the outside of the enclosure. Through these we added some stretchy shock cord, allowing the node to quickly be attached to poles and other objects

▼ **Figure 8:** The main page of the ATAK-CIV Android app, showing numerous Meshtastic devices laid over a downloaded map



Deep in the valley

Mentioning hiking brings us to a couple of example use cases we want to explore with Meshtastic. In a mountainous region such as here in North Wales, it's easy to walk into valleys or 'cwms' where there is no line of site to any other Meshtastic nodes. With a little forward planning and route consideration, it's often really easy to stealthily place a self-powered node such as this one in a position where it can act as a repeater with line of sight into a valley with limited connectivity, but can also see other nodes in the network.

It can also be used as a short distance repeater. For example, in an exposed position such as in **Figure 1**, often the weather and wind might preclude you from sitting in the prime summit position for your node. Placing a node like this, then retreating off the summit/hilltop to escape the weather, you can then have an additional smaller node, with a much smaller antenna, to act as your Meshtastic communication device, knowing that the deployed node is within easy reach.

You can see that a lot of our use cases for Meshtastic are for communications on fun adventures in the hills. This led us to explore another interesting application of Meshtastic, namely plugging Meshtastic into another application call ATAK-CIV (**Figure 8**). The latter is a civilian arm of a tactical military system for managing communications and teams positionally. Whilst it's aimed at military operations, it's a great fit for anything where a team are working in a co-ordinated fashion over a wide terrain. We could see it definitely being of use to mountain rescue teams, but equally, it could be great fun for youth groups, girl guiders or scouts playing wide roaming games in their locality. ATAK-CIV has a huge amount of features and we will only cover the basics, but let's look at getting it installed and setting up Meshtastic as its method of communication.

ATAK-CIV is a civilian arm of a tactical military system for managing communications and teams positionally

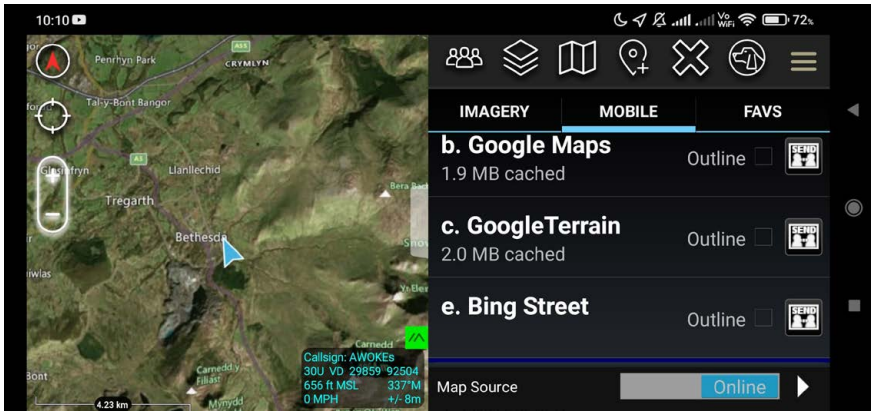


Figure 9: A list of the downloaded, installed, and available maps in our ATAK setup

Figure 10: Configuring the Meshtastic ATAK plug-in

ATAK-CIV

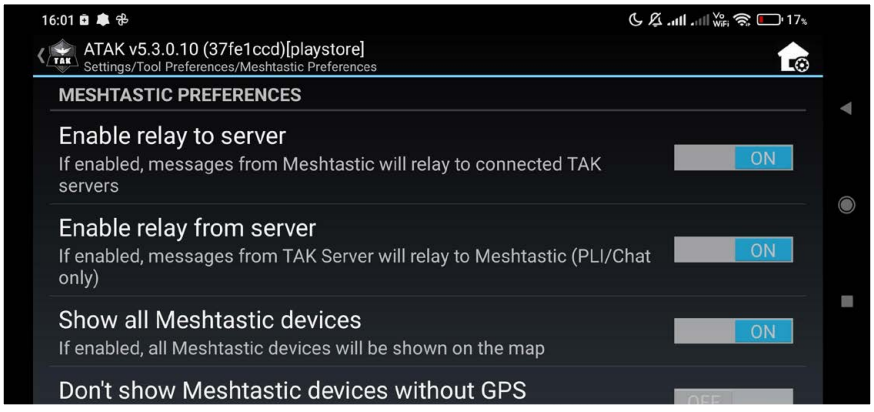
ATAK-CIV is available for Android only and you can download and install it for free from the Google Play Store. You can also find official Android APK files for ATAK-CIV online; so, if you have a de-googled droid device, this is still possible. Next you need to search for the Meshtastic ATAK plug-in APK. This plug-in acts as an intermediary between the Meshtastic app (we installed this when setting up our device in the first part of the guide); once downloaded, click the APK file to install.

Before launching the ATAK-CIV app and connecting it all together, we'll need to download some maps to install to the app. The default map is just a placeholder and when you zoom in at all, it will remain very blurry. You can easily switch between map types in the ATAK-CIV app, so we downloaded two map selections: the Bing map resource pack and the Google Maps resource pack. To do this, navigate to the links on your Android phone, download the files, note the location of your Downloads folder on your Android device (we have one on both our internal storage as well as our SD card, so that caused a moment of confusion).

Now we can launch the ATAK-CIV app. It will ask for lots of permissions and will also ask for the location services to be switched on. In the ATAK-CIV app, click the main menu three-bar icon and then scroll down and click the Import button. Navigate to the map content zip files you downloaded and select them. You'll be given the option to move the file to an ATAK directory – copy the file to a directory or use the file from the current location. We opted to move the files into ATAK. Back in the main screen, you should now be able to click the folded map icon and see a list of your added map options and switch between them (**Figure 9**).

Finally, we need to enable and configure the Meshtastic plug-in. First, fire up a Meshtastic device that you can connect to with the Meshtastic application and check it is connected. Then, in ATAK-CIV, click the menu button and scroll down and click 'Plugins'; you should see the Meshtastic plug-in listed. Click the tick box to load the Meshtastic plug-in and you should see the listing turn green to show the plug-in is loaded. Finally, return to the ATAK menu, scroll to Settings, then Tool Preferences > Specific Tool Preferences > Meshtastic Preferences. There's a lot of settings to explore and tinker with in here, but just to get started, make sure the first three items in the list are enabled and all the others disabled (**Figure 10**).

With all this done, you should now see the ATAK map populate with Meshtastic devices in the area. You will also see an arrow representing you. You can now use the ATAK-CIV app to send messages to other Meshtastic devices, including privately to other Meshtastic nodes with which you have preshared channels. You can also do lots of other fun things. Exploring ATAK, you can do things such as geofencing (so it will alert you if a Meshtastic device in your team moves beyond a specific boundary), allow you to automate sharing map points with all your team members, as well as tracking other publicly viewable Meshtastic devices. We even see that there are ways to connect FPV drones to ATAK and send drones to specific locations!



MIDI on Raspberry Pi Pico

Build Python-powered MIDI musical instruments that run on a Pico

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

robmiles.com

YOU'LL NEED

- Raspberry Pi Pico or Pico W (version 1 or 2)
- M5 Stack Unit MIDI with DIN connector (SAM2695) or Unit Synth
- Switches for keys
- An OLED, a rotary encoder, and a pixel strip for a Pico MusicBox

In this article we'll look at the Musical Instrument Digital Interface (MIDI) and discover how to make interesting instruments. We'll use a self-contained synthesizer unit which responds to MIDI commands. You can find all the example programs and configuration instructions in the GitHub repository for this article: **rpimag.co/picomidi**.

Figure 1 shows the Pico MusicBox, a musical keyboard powered by a Raspberry Pi Pico which uses an integrated MIDI device. You can select voices and sound effects using the rotary encoder. Full instructions for building the device are available in the GitHub repository.

▼ **Figure 3:** The MIDI units are provided with a connecting cable with a socket on each end. You can use male-to-female DuPont cables to connect a socket to pins on Raspberry Pi Pico

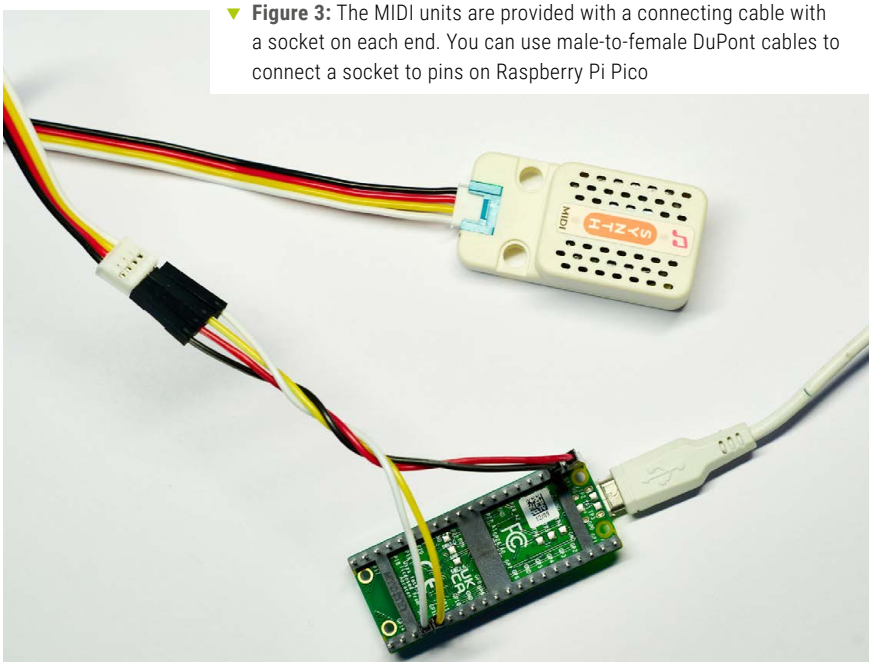


Figure 1: The pixels light up red when the notes are pressed

Figure 2: The MIDI device can also receive signals over MIDI connections

A synth on a chip

Figure 2 shows two devices we can use to create a tiny self-contained music player along with a Pico to control them. The sound devices contain SAM2695 music synthesizer chips that can play 64 simultaneous sounds and provide sound equalizer and reverb effects. The Unit MIDI device on the left of the image (**rpimag.co/m5stackmidi**) has a stereo audio output and can be connected to other MIDI devices. The Unit Synth device on the right (**rpimag.co/m5stacksynth**) has a built-in mono speaker and is designed for free-standing use. Both work in the same way. They receive MIDI messages and create the required sound output.

We'll use a self-contained synthesizer unit which responds to MIDI commands

Figure 3 shows the Unit Synth connected to a Pico. The synth is powered from the VSYS 5-volt output from the Pico and connected to GPIO12 (transmit) and GPIO13 (receive) for serial communication. If you use the MIDI device instead of the synth you can use the same connections, but you will also have to connect a speaker to the 3.5mm stereo output from the MIDI device. Now that we have the hardware sorted, we can consider how the software will work.

QUICK TIP:

`Pull.UP` means that the Pico will turn on a circuit which pulls the input pin high, so that our switch can pull the value down when it is pressed.

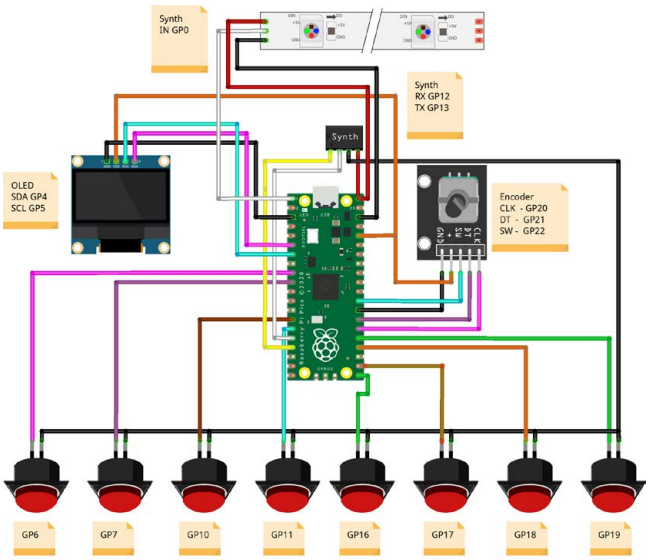


Figure 4: You can use any version of Pico with this circuit. The author used a Pico 2 W so that a later version of the keyboard can be connected to a network

MIDI software

To send MIDI commands from a CircuitPython program, we will use the Adafruit MIDI library. We need to install the **adafruit_midi** folder for the MIDI library and **adafruit_bus_device** folder for the serial port drivers. These can be found in the CircuitPython library bundle at **rpimag.co/circuitpythonbundle**. There are installation instructions in the repository for this project. Once the libraries are installed, we can write a CircuitPython program to play a note. Let's start with a trip to the library:

```
import time
import usb_midi
import adafruit_midi
from adafruit_midi.note_on import NoteOn
from adafruit_midi.note_off import NoteOff
import busio as io
import board
```

The statements above import all the libraries that we are going to use in our program to talk to the hardware and generate MIDI messages. The next thing we need to do is configure the MIDI connection:

```
uart = io.UART(board.GP12, board.GP13,
baudrate=31250)
midi = adafruit_midi.MIDI(midi_out=uart,
out_channel=0)
```

The first statement creates a serial connection. The variable name `uart` stands for Universal Asynchronous Receive/Transmit, which is what serial ports do. The `uart` value is passed to the `adafruit_midi` constructor that creates an object that can receive and transmit MIDI messages.

QUICK TIP:

The MIDI synth device uses a digitised piano waveform to make the sound.

Now that we have a MIDI connection, we can send it messages to play notes:

```
midi.send(NoteOn(60, 100))
time.sleep(0.5)
midi.send(NoteOff(60, 100))
```

The three statements above play MIDI note number 60 at volume 100. There is a gap of half a second between the note being pressed and released. You could get the same result by walking up to a piano, pressing the 'middle C' note and releasing it after half a second. The maximum volume for a MIDI sound is 127, so you would have to press the note quite hard to match the code above. The `NoteOn` and `NoteOff` messages describe the note to be played by giving note number and volume values. You might wonder what volume information means for the `NoteOff` message. This tells the musical instrument how quickly the note was released. This can be used to add 'after touch'. If you reduce the length of the `sleep` between the calls of `NoteOn` and `NoteOff`, you will notice that the sound changes and becomes more 'staccato'. Now that we can make a note, let's make a tune.

AI music

It turns out that ChatGPT (and probably other generative AI systems) will happily try to make MIDI note sequences for you, with varying success. The author used the query "I'm writing a song playback program in Python. I'm storing each note of the song as a MIDI note value and a note duration in beats. Can you make me a list of note tuples for the song 'Twinkle Twinkle Little Star'," and was rewarded with a complete song. However, an attempt to generate "The British national anthem" resulted in a random stream of notes (but the correct lyrics).

Playing a tune

We can create a program that plays a tune by storing some 'tuples' which hold note and duration values.

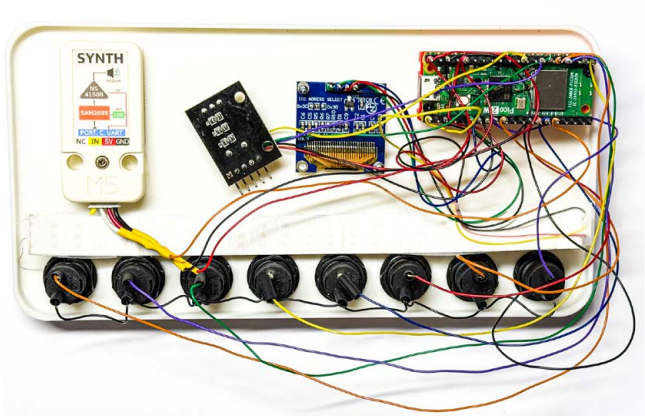
```
twinkle_melody = [
    (60, 1), (60, 1), (67, 1), (67, 1),
    (69, 1), (69, 1), (67, 2),
    (65, 1), (65, 1), (64, 1), (64, 1),
    (62, 1), (62, 1), (60, 2)
]
```

The above statement creates a list of tuples which describe a song. The first value in each tuple specifies the MIDI note to be played. The second gives the number of 'beats' the note should play for. The note at the start of the song is middle C (note 60) and it will be played for 1 beat. There are no prizes for guessing what the song is. Now we can create some code to play it:

```
beat_length = 0.5

for step in twinkle_melody:
    (note,delay)=step
    midi.send(NoteOn(note, 100))
    time.sleep(beat_length*delay)
    midi.send(NoteOff(note, 100))
```

The code above plays the song described by `twinkle_melody`. The first statement sets the speed of playback by specifying the interval between each note and the next. A beat of 0.5 seconds between notes means the music is played at 120 notes per minute. The `for` loop works through each step in the `twinkle_melody` list. The first statement in the loop extracts the `note` and `delay` values from the tuple. These are then used to generate the note-on and note-off messages, with the calculated beat gap between them.



Building a MusicBox

Making tunes from preset note sequences is all very well, but what we really want is a way of entering notes for the Pico to play. We can do this by adding a keyboard. The smallest keyboard that has a usable range of notes has eight keys that can play a complete octave. We can create a keyboard by connecting eight switches to input pins on Pico. Then we can add a rotary encoder, an OLED screen, and some pixels if we want the full MusicBox experience.

Figure 4 shows the circuit for the Pico MusicBox. The notes on the diagram give the Pico pins used by each of the devices. You can leave off the OLED, encoder, and pixels if you just want a simple keyboard.

Figure 5 shows the internals of the music box. The switches were chosen because they were the nicest to press. When a button is pressed, the switch connects an input signal to the ground level, causing the input to read 0 rather than 1. Now we need some code to read the inputs and play the notes.

QUICK TIP:

The SAM2695 chip also has a microphone input, which would be fun if you were making a karaoke machine, but it is not connected in these devices.

Reading keys

We need code that will send a note-on message when the button is pressed and a note-off message when the button is released. Let's start by creating code that connects the program to an input pin.

```
from digitalio import DigitalInOut, Pull
```

The statement above imports `DigitalInOut` and `Pull` from the `digitalio` library. Now we can make a Python variable that represents an input pin:

```
note1_pin = DigitalInOut(board.GP6)
note1_pin.pull = Pull.UP
```

Figure 5: All the elements were connected using wire-wrap wire which was soldered onto the switches and then wrapped around solder pins on the other devices

The statements above create a variable called `note1_pin` which is connected to GPIO6 on the Pico. Our program could read this pin directly, but we are going to use the Adafruit `Debouncer` class to read it. This cleans up the input signal so electrical noise caused by the switch contacts opening and closing will not result in unwanted notes being played.

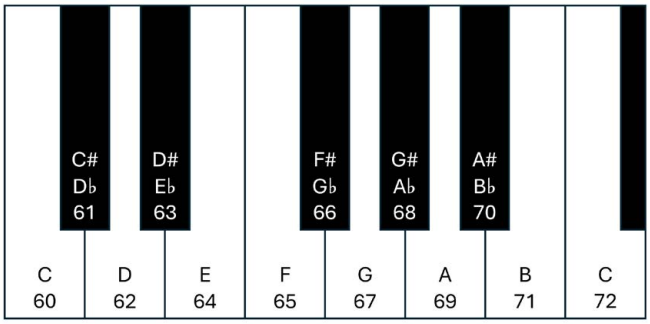
```
from adafruit_debouncer import Debouncer
```

The statement above imports the `Debouncer` class from the `adafruit_debouncer` library. Now we can create a debouncer:

```
note1_debounce = Debouncer(note1_pin,
interval=0.01)
```

The statement above creates a `Debouncer` variable called `note1_debounce`, connected to `note1_pin`. When the program wants to check if a key has changed, it updates `note1_debounce` and then checks the `rose` and `fell` properties on it.

```
while(True):
    note1_debounce.update()
    if note1_debounce.fell:
        midi.send(NoteOn(60, 100))
    if note1_debounce.rose:
        midi.send(NoteOff(60, 100))
```



▲ **Figure 6:** An octave contains twelve semitones

The code above repeatedly `updates note1_debounce` and sends a `NoteOn` message when button falls and a `NoteOff` message when the button rises. If you run this on a device with a button connected to GPIO6, the note middle C will be produced when the button is pressed and fade away when the button is released. Each button will need to produce a different MIDI note when it is pressed so that the user can play a scale of notes. Let's take a quick look at how musical scales work.

A note about notes

Figure 6 above shows the octave starting at middle C on a piano keyboard. The keys are marked with their note names and MIDI numbers. The white keys are identified by a letter in the range A–G. The black keys, called 'incidentals', are named as the sharp (#) version of the white note to their left or the flat (♭) version of the note to their right. Each of the notes has a different frequency. As you go from left to right along the keyboard, the frequency of the note (and therefore the pitch of the sound you hear) gets higher. This pattern of notes is repeated, so the notes D, E and F, along with their incidentals, will continue off the right-hand side of the figure.

The block of notes you see in **Figure 6** is called an 'octave', which comprises a complete set of notes from C to B (note that the C on the right-hand side is the first note of the next octave). When a note goes 'up' an octave (for example from one C to the next), the audio frequency of the note doubles. The range of frequencies of the notes in an octave is divided into twelve equal steps called 'semitones'. So, the step from C to C# is called a semitone, as is the step from C# to D.

This is a bit hard to understand, and some aspects of how music works make very little sense at first. If you played the white notes 'CDEFGAB' in sequence, you would hear something that 'sounds right', even though the gaps between some notes (C and the D) are two semitones and other gaps (E and the F) are one semitone. This sequence of semitone steps is called a 'major scale' and is the basis of most western music. There are other scales in which the gaps between successive notes (i.e. the position of the incidentals) are different. Now we know the MIDI value for each input signal, we can build a proper keyboard.

A touch of class

The keyboard we are going to build will have buttons for the eight notes, so we need a way of mapping eight inputs to the MIDI notes to be played. This looks like a job for a class:

```
class Key():
    def __init__(self, pin, note):
        self.pin = pin
        tmp_pin = DigitalInOut(pin)
        tmp_pin.pull = Pull.UP
        self.debouncer = Debouncer(tmp_pin,
interval=0.01)
        self.note = note
```

The statements above create a class called `Key` which implements one key on the keyboard. The constructor (the method which is called when a `Key` is created) accepts a `pin` value and a `note` value. It creates debouncer and note properties which are stored inside the `Key` value.

```
def update(self):
    self.debouncer.update()
    if self.debouncer.fell:
        midi.send(NoteOn(self.note, 100))
    if self.debouncer.rose:
        midi.send(NoteOff(self.note, 100))
```

The `Key` class also contains an `update` method that is used to update it (the clue is in the name). This updates the debouncer and then checks to see if the input has changed state. If it has, the appropriate MIDI message is sent. Now that we have our `Key` class, we need to make a list of keys on the keyboard.

```
note_descriptions = [(board.GP6,60),
(board.GP7,62),
          (board.GP10,64),(board.GP11,65),
          (board.GP16,67),(board.GP17,69),
          (board.GP18,71),(board.GP19,72)]
```

The `note_descriptions` list contains tuples for each key, giving the pin and the MIDI note for that key. The pins correspond to the connections in **Figure 4**. We can use this list to make all the keys:

```
keyboard = []

for note in note_descriptions:
    (pin,note)=note
    keyboard.append(Key(pin,note))
```

The code above starts by creating an empty `keyboard` list and then works through the `notes` list to create each key. The `pin` and `note` values are extracted and used to create a `Key` which is appended to `keyboard`. Now that we have a keyboard, all the program must do is keep updating it.

```
while True:
    for key in keyboard:
        key.update()
```

QUICK TIP:

A tuple is a Python data type which can hold multiple values. It is rather like a structure in other languages.

The code above repeatedly works through the keyboard, updating each key. The key will check for a change and send the appropriate MIDI message if required. This code works very well. If we want to add more keys to the keyboard, we just have to wire them up and add more entries to the `note_descriptions` list.

A change of voice

The MIDI device always starts with a piano voice, but there are over 100 other voices we can use. To select a voice, we use a MIDI message called ProgramChange:

```
from adafruit_midi.program_change import
ProgramChange
```

The statement above imports the `ProgramChange` class from the Adafruit MIDI library. Now we can use it to select a different voice:

```
voice_number = 9
midi.send(ProgramChange(voice_number))
```

The meaning of MIDI

The Musical Instrument Digital Interface (MIDI) was invented in the 1980s. The MIDI standard describes the hardware that physically links devices and the protocol that transfers musical information. The first MIDI devices used DIN (Deutsches Institut für Normung – German Institute for Standardisation) plugs and sockets with specially assigned pins. The signal paths were electrically isolated, preventing ground loops and reducing noise in audio signals.

Early MIDI devices had IN or OUT connections or both. A device could also provide a THRU connection to pass on input signals, allowing devices to be 'daisy-chained' together. Nowadays MIDI devices are more likely to be connected using a USB connection, but the protocol remains the same. The MIDI 2.0 standard was introduced in 2020, but most devices still use MIDI 1.0.

The two statements above set the value of `voice_number` to 9 (Glockenspiel) and then send a `ProgramChange` message to select this value. After this command, any notes played will use the Glockenspiel voice. The General MIDI standard specifies the voice of each number:

```
midi_instruments = [
    ("Acoustic Grand Piano", 0),
    ("Bright Acoustic Piano", 1),
    ("Electric Grand Piano", 2),
    ("Honky-tonk Piano", 3),
    ("Electric Piano 1", 4),
    ("Electric Piano 2", 5),
    ....
]
```

The list above contains Python tuples which hold the name and number of each voice. A complete version of the list is in the **samples** folder in the repository for this article.

Taking control

Our programs can change the output sound by sending control messages to the MIDI device. This mimics the effect of a control being adjusted on a real device. To send a control value, we use a MIDI message called ControlChange:

```
from adafruit_midi.control_change import
ControlChange
```

There are lots of different things that can be controlled, each of which is identified by a particular control number. Control number 1 simulates the action of a 'modulation wheel' that adjusts the amount of 'vibrato' in the sound being produced. Vibrato is a variation in pitch as a note is played.

```
CC_MODULATION = 1       # Modulation (often
used for vibrato)
midi.send(ControlChange(CC_MODULATION, 127))
```

The above statements would set the modulation to the maximum, leading to a very wobbly sound. The MusicBox device uses a rotary encoder to set control values. Sample code in the repository for this article contains a list of popular control change values.

QUICK TIP:

MIDI channel numbers start from 1 in MIDI documentation, but the CircuitPython channel numbers start at 0. In MIDI documents you will see channel 10 referred to as the drum channel, but it is 9 in software.

Playing the drums

The final thing we need to know is how to play the drums. Drums are handled differently from MIDI notes. A drum doesn't have a particular pitch, it is just a drum. So rather than having a keyboard full of different notes, each key represents a particular drum type. Different note numbers trigger different drums.

```
ACOUSTIC_BASS_DRUM = 35
ACOUSTIC_SNARE = 38
CLOSED_HIHAT = 42
OPEN_HIHAT = 46
CRASH_CYMBAL_1 = 49
RIDE_CYMBAL_1 = 51
LOW_TOM = 45
LOW_FLOOR_TOM = 41
```

The code sample above gives note numbers for popular drum sounds. The drum messages are sent to a different MIDI channel than musical notes. The MIDI standard provides 16 channels which a device can listen on. If you have multiple sound devices and sequences, you can configure each one to listen for commands on a different channel. Drum note messages are sent to channel 9:

```
midi.send(NoteOn(ACOUSTIC_BASS_DRUM,
100),channel=9)
```

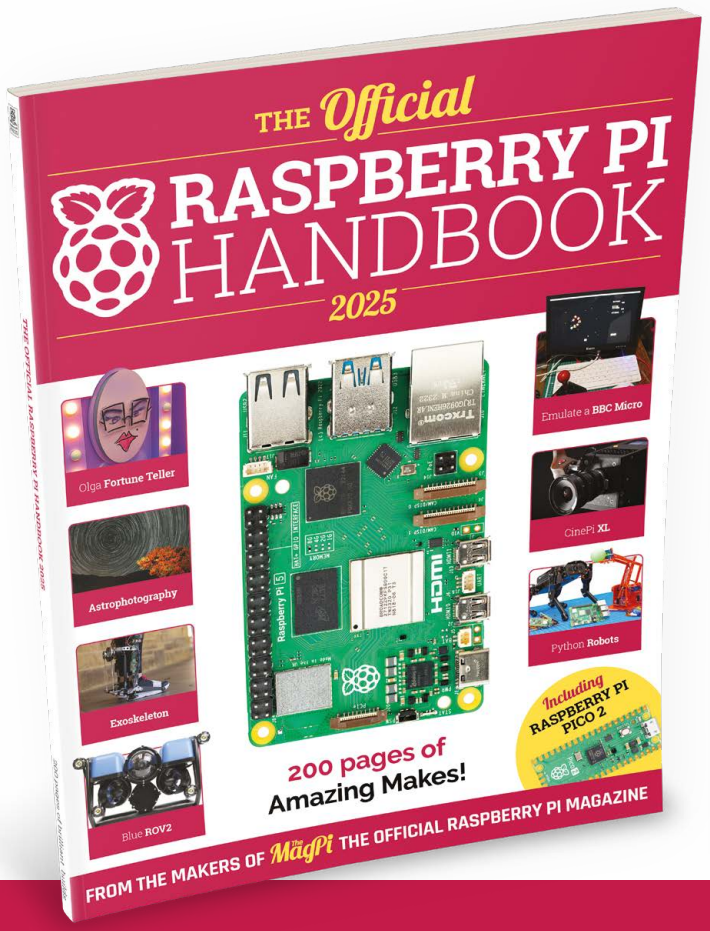
The statement above sends out a rather loud kick drum sound.

A drum sound doesn't need a note-off command because drums are always played as one-shot sounds.

THE *Official* RASPBERRY PI HANDBOOK 2025

200 PAGES OF RASPBERRY PI

- A guide to getting started with every Raspberry Pi board
- All the essential info on the brand new Raspberry Pi Pico 2
- Inspiring projects to give you your next big idea
- Upgrade your emulation with next-gen retro gaming



Buy online: rpimag.co/store

Sewing with LEDs and Raspberry Pi Pico

Why should a handy storage pouch be boring? Bling it up with some light-emitting silicon

Maker

Nicola King

Nicola is a freelance writer and sub-editor. Sewing with a little bit of tech thrown in is hitting the spot this month.

@holtonhandmade

QUICK TIP

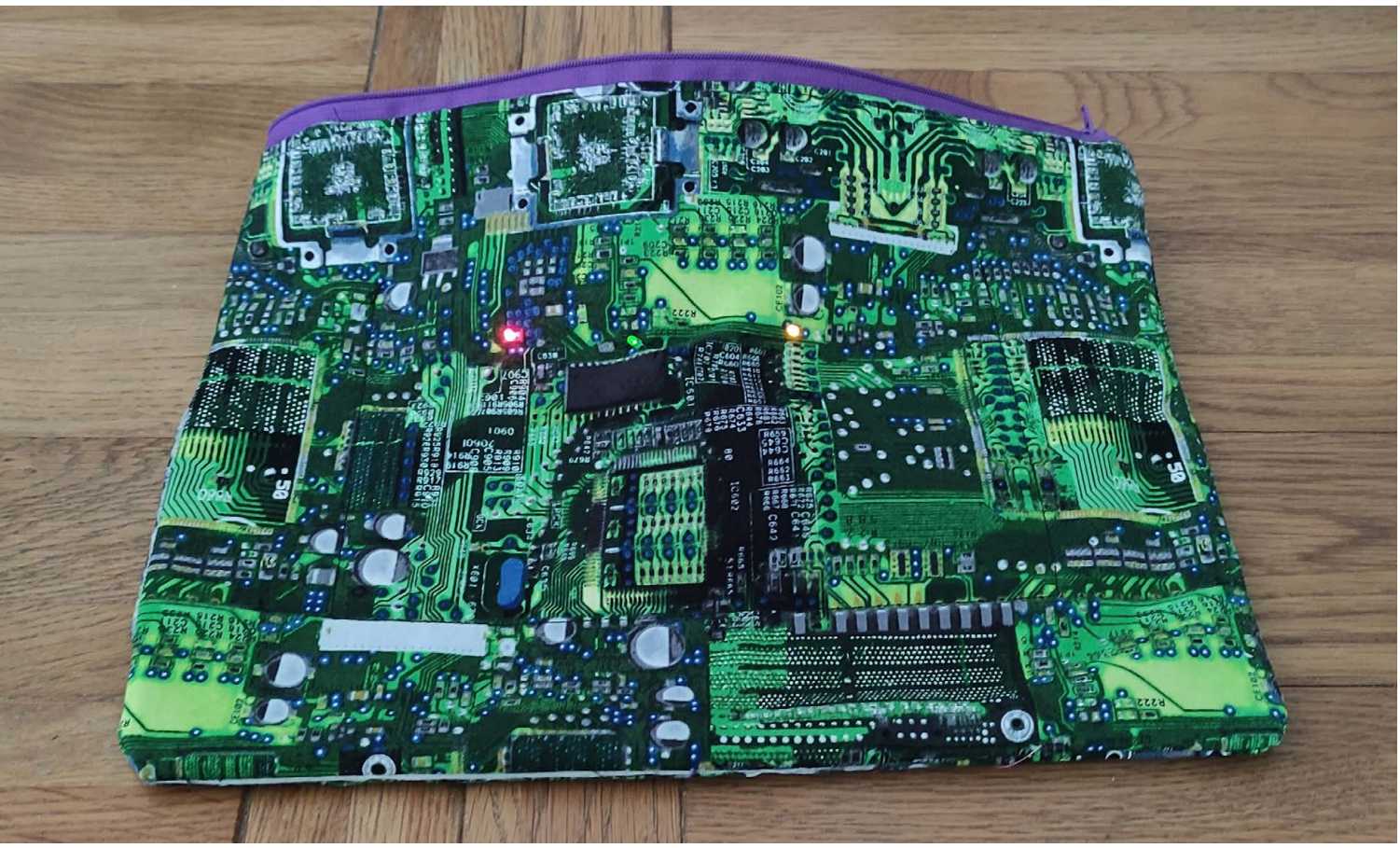
Felt would be another great base fabric to use if you are a beginner – it is stable, very easy to sew through, won't fray, and will easily hold the weight of components.

Combining LEDs into a sewn project is easy to do, and we're not just referring to wearable tech here. You can also craft items that you can actually utilise for a functional purpose and, at the same time, add a little Raspberry Pi and LED luminescence to your design. So, this tutorial takes a look at how to add some LEDs to a simple storage pouch that we are also going to make. In fact, these zipper pouches – which can be made in absolutely any size you prefer – are so uncomplicated to make that we've whipped up quite a number in order to house various components, plus our general odds and ends that need a home.

Sew straightforward

We decided that we wanted a decent-sized pouch, so, we first cut out two rectangles of fabric 25cm by 35cm in our main fabric. Now, a quick word on your choice of base fabric before we move on: your base is important, and we've used cotton backed with a medium-weight fusible interfacing to give it some structure. Cotton is generally a great choice because it is lightweight, flexible, and highly suitable for sewing with electrical components. If your fabric is difficult to sew through, you will have issues, so give this some thought. If you have some lightweight denim on hand, some calico, or even faux leather, they might also be good alternatives.

If you want to add a backing to strengthen your fabric, cut out two pieces in a fusible interfacing the same size and adhere with an iron. Then cut out a 'pocket' in a different-coloured fabric – we cut a rectangle 10cm by 15cm – and back that with interfacing too. The reason we are adding a pocket is so that all of our Pico/LED workings can sit inside it comfortably and separately from whatever else we choose to keep in the pouch. To attach the pocket, fold in the bottom and the two sides of



Cotton is lightweight, flexible, and highly suitable for sewing with electrical components

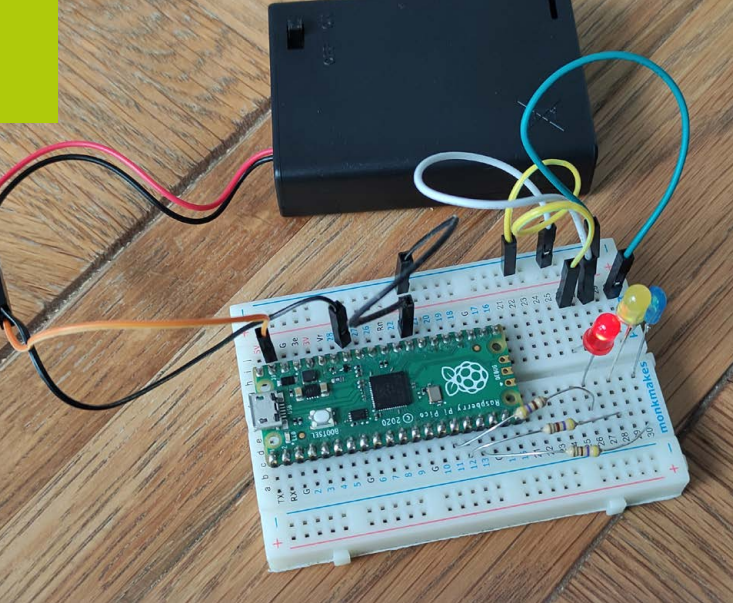
the small rectangle by about 0.5cm, and press, then pin the pocket to the wrong side of one of your pouch pieces. Use your sewing machine to sew the pocket to the main fabric, keeping a consistent seam – 0.5cm works well. As our fabric was dark in colour, the dark thread that we used to sew the pocket blended well into the fabric and is barely visible on the right side of the pouch.

Next, take your zip and place it along the top side of one side of pouch, right sides facing, and pin it. Then sew it with a zipper foot on your sewing machine if you have one. Attach the other side of the pouch to the other side of the zip and sew. Then press the seams out that you have just sewn and top-sew along the length of the zip on the right-side of the fabric to give a neat finish. Next, open the zip halfway, then turn right sides of pouch to right sides (so the inside of the pouch is on the outside) and pin in place. Sew around all three pinned sides in one go, carefully removing pins as you sew.

▲ We had some 'circuit' fabric in our stash that seemed highly appropriate here (**rpimag.co/frumblecircuit**). Just some fabric and a zip, along with our components, meant this was a very low-cost project. Stash-bust and see what you have!

YOU'LL NEED

- Cotton fabric (¼ metre)
- Fusible interfacing (optional)
- Standard cotton thread
- A zip (cut to length required)
- Sewing machine
- Pinking shears (optional)
- Scissors/sewing pins/iron/tape measure
- LEDs (single colour) × 3
- Jumper wires × 6
- Resistors × 3
- Power source
- Raspberry Pi Pico
- Needlenose pliers (optional)



Testing the LED circuit and program on a breadboard before wiring everything directly to Raspberry Pi Pico

Finally, trim loose threads and any excess zip fabric, and use pinking shears to cut away excess from the seams and to ensure that fraying is kept to a minimum. As you opened the zip halfway, you should now be able to turn the pouch so that the wrong side is now inside and then press it out with an iron – one storage pouch ready for use. Note that we have purposely created a very simple design here for the purposes of speed and ease, so we can now concentrate on the electronics. If you have the time, you can line the pouch with another fabric to make the inside tidier.

Power sources

A number of options exist for powering your sewn circuits:

- Coin cell batteries – sewable coin cell holders can be integrated into projects, creating a circuit with some easy-sew LEDs and conductive thread. These batteries are usually around 3V. All of the components are small and inconspicuous while still giving you a great LED effect.
- Battery packs – as we've used in this tutorial, powering Raspberry Pi Pico and the LEDs simply with three AA batteries.
- For a more detailed look at power sources, see the tutorial published in issue 76 of HackSpace magazine: **rpimag.co/hs76**.

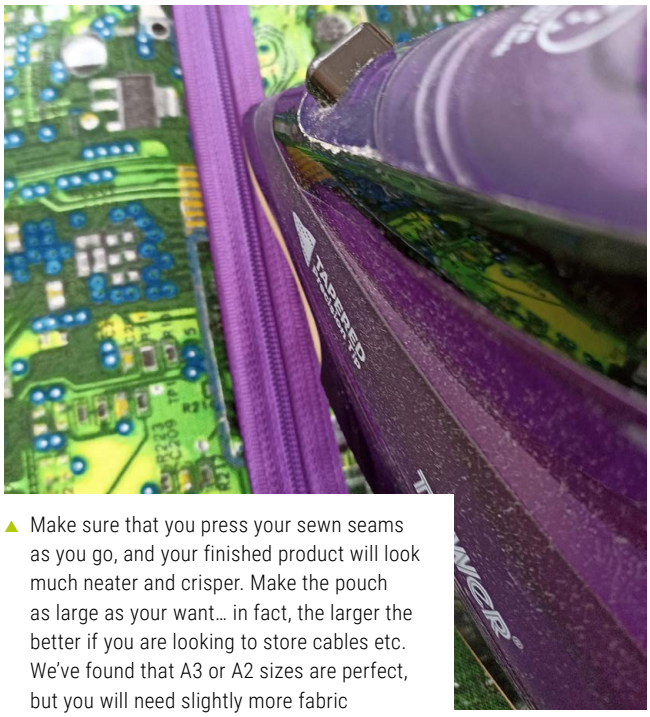
Learn more....

The e-textiles (electronic textiles) field is pretty vast, and there are plenty of sources of inspiration. Here are a few ideas to take you further:

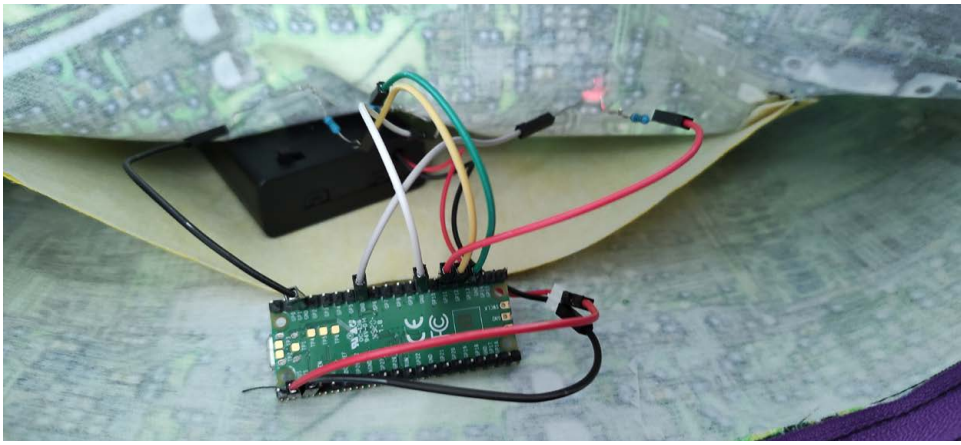
- *Wearable Tech Projects*, a HackSpace publication written by Sophy Wong, was published in 2019, but is still hugely relevant and endlessly inspiring. If you own a copy already, dig it out. If not, you can download the free PDF here: **rpimag.co/wearableprojects**.
- You can find out more about the LilyPad system, a set of sewable electronic pieces designed by Leah Buechley, and sewable electronics, at **sparkfun.com/about-lilypad**. Leah also has a book on the subject, called *Sew Electric*, which is worth a flick through – **sewelectric.org**.
- Interesting work is going on in the subject area of e-textiles. Take a look at this research which is concentrating on making e-textiles sustainable and biodegradable – **rpimag.co/etextilestech**.

QUICK TIP

One way of securing standard LEDs to fabric is through a button – you'll need a two-holed button where the width between the holes is that same as the width between the LED legs. Poke the legs through and perhaps secure the legs with a few stitches using standard thread.



Make sure that you press your sewn seams as you go, and your finished product will look much neater and crisper. Make the pouch as large as your want… in fact, the larger the better if you are looking to store cables etc. We've found that A3 or A2 sizes are perfect, but you will need slightly more fabric



▶ Raspberry Pi Pico, the battery pack, and wiring all fit in an interior pocket to keep them separate from the main bag contents

QUICK TIP

If you use standard single-colour LEDs in your circuit, as we have, ensure you mark the legs so you know which is positive and negative before you flatten/twist them.

Pico programming

For this simple example project, we're just going to light three LEDs in sequence. So, in essence, we just need a program to turn the GPIO pin for each LED on (and off) in turn.

We're using MicroPython for our program – flash it to Raspberry Pi Pico by downloading the latest UF2 file from **rpimag.co/picomicropython**, making sure it's the correct one for your particular Pico model. Then, while holding its BOOTSEL button, connect Pico to a computer via USB to mount Pico as a drive. Now drag the UF2 file onto the Pico drive to install it.

We're using the Thonny IDE (**thonny.org**) to program Pico. In Thonny, click 'Local Python 3' at the bottom right of the window, then select 'MicroPython (Raspberry Pi Pico)'.

We start the program by importing two modules we need.

```
import machine, time
```

We define variables for our three LEDs, assigned to the GPIO pins to which they are connected, as an output.

```
led1 = machine.Pin(11, machine.Pin.OUT)
led2 = machine.Pin(12, machine.Pin.OUT)
led3 = machine.Pin(13, machine.Pin.OUT)
```

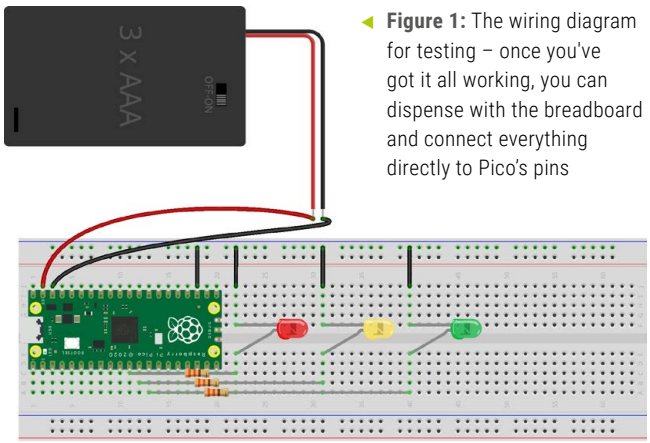
We then create a list of these:

```
leds = [led1, led2, led3]
```

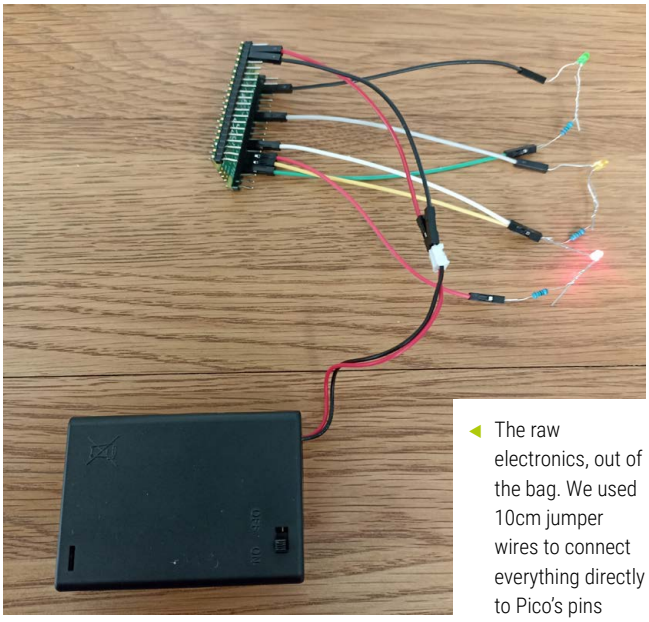
Finally, we create an infinitely repeating `while` loop with a `for` loop nested within it. The latter takes a value from the `leds` list in turn, starting with `led1`, and turns the respective LED on with `led.value(1)`. We set a `time.sleep` delay for how many seconds it will stay on, then turn it off with `led.value(0)`.

```
while True:
    for led in leds:
        led.value(1) # Turn LED on
        time.sleep(0.5) # Delay for 0.5 seconds
        led.value(0) # Turn LED off
```

With the LEDs connected on a breadboard, as in the **Figure 1** wiring diagram, try running the program to see them light up in sequence. To make the program run automatically whenever Pico is connected to a power supply, save it as **main.py** (selecting Raspberry Pi Pico as the location).



◀ **Figure 1:** The wiring diagram for testing – once you've got it all working, you can dispense with the breadboard and connect everything directly to Pico's pins



The raw electronics, out of the bag. We used 10cm jumper wires to connect everything directly to Pico's pins

Once everything is working as expected, you can reconstruct the circuit without the breadboard – after positioning the LEDs where you want in the fabric, poking their legs through it towards the inside of the pocket. (Cotton fabric is very easy to poke the legs through.) Coiling the end of a resistor around the positive leg of an LED should hold it in place, or you may opt to solder it for a more secure connection. We plugged the other end of the resistor into a socket-to-socket jumper going to the respective GPIO pin on Pico; ditto for the negative leg of the LED, connected via a jumper to a GND pin. If your Pico doesn't have pin headers attached, you could just solder the wires directly to the pin holes.

You can easily add extra LEDs to your circuit (and program), but may need to join some ground wires together to connect them all to Pico's GND pins if you run out of the latter. You could also create alternative lighting programs and perhaps fade LEDs in and out using PWM to adjust the brightness.

We're powering our circuit from a 3 × AA battery pack, which supplies around 4.5V, connected directly to Pico's VSYS and GND pins. Pico is fairly voltage tolerant, so this should work fine. If using a Raspberry Pi Zero computer instead, you may well want to use a mini voltage regulator board.

A twinkling tote

The aim of this tutorial has been to offer up a simple idea on how to combine some crafty sewing and LEDs in one project. There are many ways of doing this, so just see this as a starting point and do please take it further. Make a larger tote bag, introduce sewable LEDs such as a LilyPad and maybe some conductive thread, take inspiration from our 'Learn more' box, and create your own sewn and Raspberry Pi or Pico combo project.

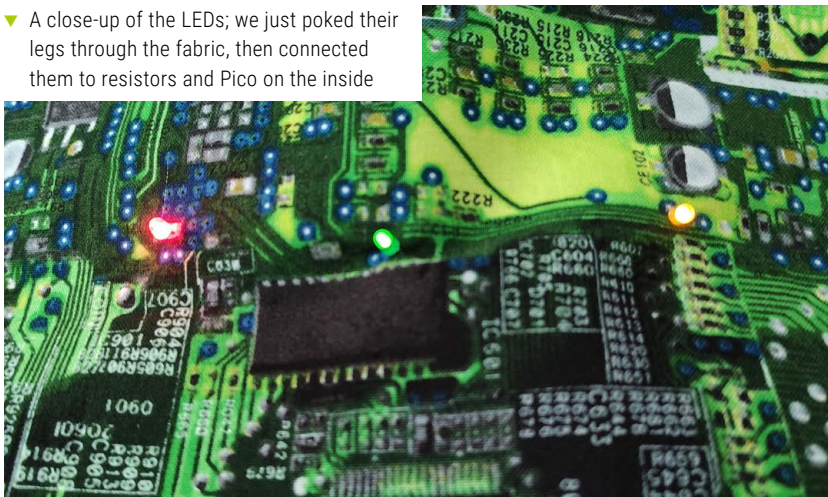
Light-emitting diodes

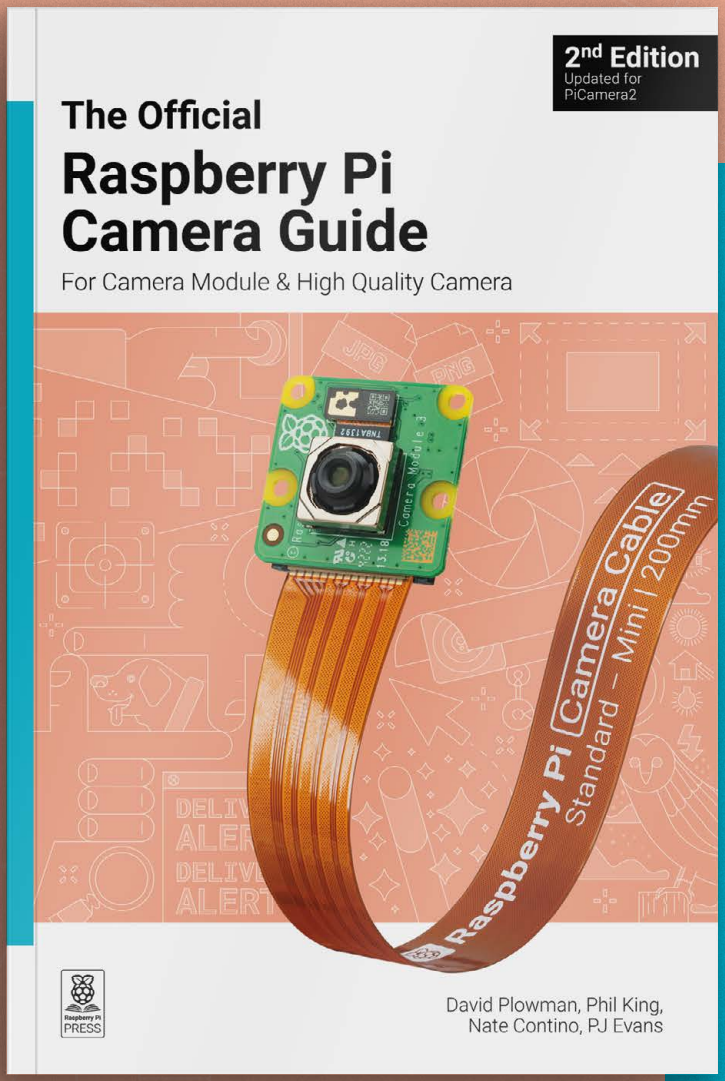
There are a number of options when it comes to choosing which LEDs to use for your project. Let's take a brief look at a few different types which might be suitable here:

- **Sewable LED modules** – these are typically very slim, pre-mounted LEDs designed specifically for sewing purposes, as they usually have a hole in either side that you can get a needle and conductive thread through in order to attach them to a fabric.
- **NeoPixel LEDs** – these are individually addressable, so can be chained together and controlled from a single pin on Raspberry Pi Pico or a Raspberry Pi computer.
- **Single-colour, standard LEDs** – these might be clear, and the light that they generate is often directed and focused. Or, they might be diffused LEDs – the light here is often slightly softer or 'milkier' in nature, and these give a more even and consistent light. Single-colour LEDs usually will have two legs (a long one for the anode or positive, and a short one for the cathode or negative). You can often buy 'super-bright' or 'ultra-bright' versions too, depending on what you are aiming for in terms of brightness levels.
- **RGB LEDs** – these combine three tiny LEDs: red, green, and blue. They usually have four legs (one for each colour, and a shared cathode/anode leg). Essentially, by varying the brightness of each LED element, a wide spectrum of colours can be generated.

You can also purchase addressable flexible strips of LEDs, which are very useful if you want fine control of a lot of LEDs – you can cut the strip to a particular length to suit your needs. Super-flexible 'noodle' filaments are also available, as are strips with power switches, strips with light sensors, and more. Your choice of which to purchase ultimately depends on what you are making, your power source, and the light effect you want to achieve.

A close-up of the LEDs; we just poked their legs through the fabric, then connected them to resistors and Pico on the inside





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

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

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

Part 02

RISC OS for Raspberry Pi

We take a look at the RISC OS Pi's pinboard, and what its onboard icons do

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

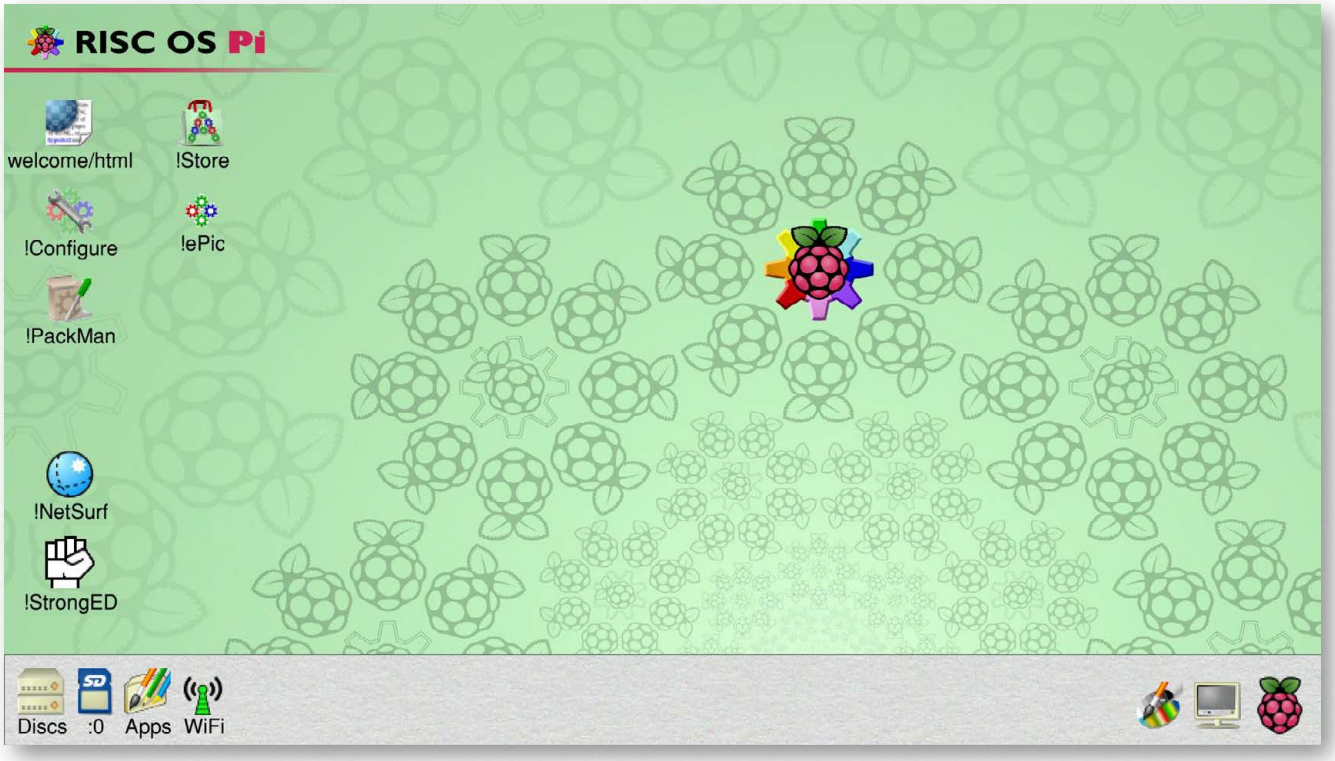
Last issue we covered how to get up and running with RISC OS, installing the operating system on a microSD card for your Raspberry Pi and getting online with the alternative operating system. Now we've done this, let's take a look at the bare-bones basics of how to use it, and the apps that come bundled with it.

The first thing to bear in mind when using RISC OS is it uses all three mouse buttons: left, right, and press the scroll wheel for the middle button. Make sure the controller you use has this feature – the official Raspberry Pi mouse is ideal. In a nutshell, when using the mouse, the left button is 'select' or 'activate', the middle button is 'menu', and the right button is 'adjust'.

Along the bottom of the screen is the icon bar, an on-screen dock showing currently open applications. When you open an app, it appears in this icon bar, with new ones added in the row to the right. If you want to close an app, middle-click on its icon in this bar and choose Quit from the menu.

Gamers will be disappointed to know !PackMan isn't a RISC OS tribute to the Namco coin-op of old

When you boot up your RISC OS Raspberry Pi, two icons are already present on the right-hand side. A Raspberry Pi icon gives access to configuration tools, the task window which shows what's running on the computer and perhaps most importantly, the shutdown option for when you want to shut your Raspberry Pi down before switching off (or restarting, if you wish). Left-click your Raspberry Pi icon to open the Tasks window, or middle-click it for a menu that lets you choose one of these options.

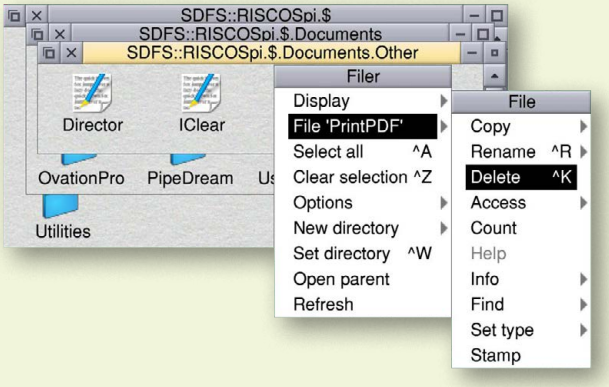


▲ A small edit gives you big icons and text

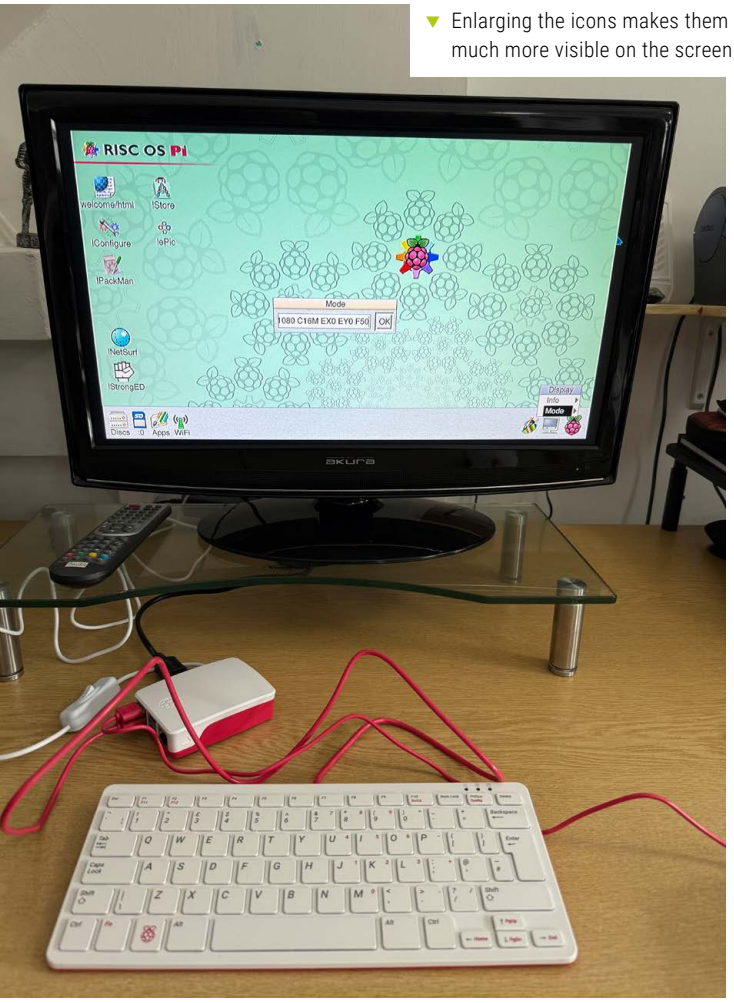
Monitoring the monitor

To the left of your Raspberry Pi icon is the monitor icon. Left- or right-click it for a window showing the number of colours, the output resolution, and the frame rate. You can change any of these factors by clicking the menu icon to their right and choosing a different value (left clicks). Alternatively, middle-click the monitor icon and in the menu, drag the pointer to the triangle next to Mode for a useful little window. This window offers four pieces of information: pixels along the X and Y axes (X and Y), the number of colours (C), and the frame rate (F). You can edit these if you wish. Here's a trick – if you add 'EXO EYO' between the C stat and the F stat, you scale the icons and text by a factor of two. This certainly makes the screenshots in this article clearer, but move the four icons near the top-left corner down the screen before you do it or they'll disappear.

Back to the desktop icons. The one in the top right is called Welcome, and it's simply an HTML document introducing you to RISC OS. Open it (double left-click), read it, chuck it. How do you chuck it? You middle-click on it, hover the pointer over Selection, and then choose Remove in the submenu. This deletes it from the pinboard, the feature that most computer systems call a desktop. If you need to delete a file, folder or app from your system, navigate to it and then middle-click on it. In the



▲ Delete a file, folder or app by middle-clicking on it, selecting the object from the menu, and then clicking on Delete



▼ Enlarging the icons makes them much more visible on the screen

Opening and closing apps

In RISC OS for Raspberry Pi, an app is identified by an exclamation mark in front of its name. It's called a 'pling', for some reason.

To open an app, double-click on it with the left mouse button. The app appears in the icon bar, but an app window doesn't appear on the pinboard. To open the app's window, you have to click on it in the icon bar.

Closing the app window doesn't close the app. It's still there, in the icon bar. To close the app, you have to middle click on its icon in the icon bar, then select Quit.

menu that springs up, the second option listed is a combination of the type and name of the item you clicked on. The first word is App (for an application), Dir (a folder, aka a directory), or File (for a document). The name of the item is then offered in quotes. Whatever it says, drag the menu pointer over this entry and in the submenu, select Delete. You can also use **CTRL+K**, as you can see in the menu.

Apps and Utilities

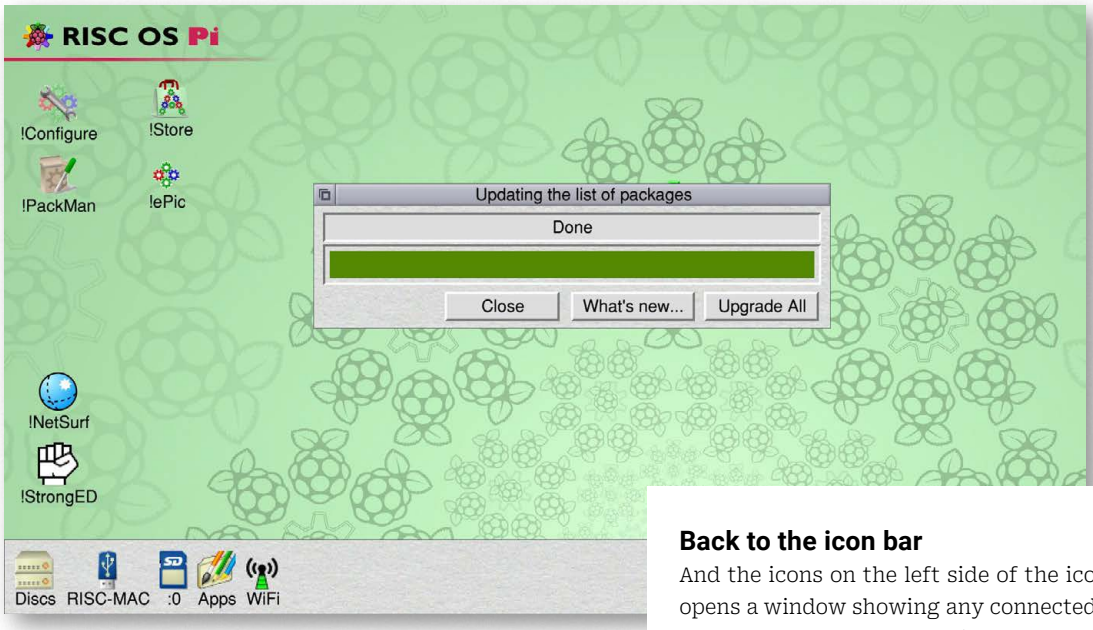
The other icons on the left-hand side of the pinboard are apps. You can get to them using the SD card icon in your icon bar, of course, but having these icons on the pinboard is an easy way of opening what are likely to be the most used apps on your system.

In RISC OS, directories start with an exclamation mark. !Configure is, unsurprisingly, a directory of tools that lets you configure your RISC OS Raspberry Pi to your own requirements. We'll be looking at these tools in a later issue. !Store, equally unsurprisingly, opens the PlingStore, the online shop for RISC OS applications. It has a wide range of paid-for and free software, curated into categories for easy browsing. No doubt we'll be revisiting this feature some time in the future too, but in the meantime, you can take a look at it online on any computer's browser at **plingstore.org.uk** if you wish.

Gamers will be disappointed to know !PackMan isn't a RISC OS tribute to the Namco coin-op of old. In fact, it's an updater utility to help install and maintain applications and other components of a RISC OS computer. It's a good idea to launch and run it every now and again to keep your system up to date and running correctly.

So what's !ePic? According to RISC OS Open Limited, it's "a package of flagship RISC OS software specifically to enhance that already bundled with the base RISC OS Pi distribution". There's some amazing stuff in there. Read all about it at **riscosopen.org**. The icon on the pinboard is a web link that lets you order the ePic pack. Or you can get it using your PC/Mac at **rpimag.co/riscosepic**. The RISC OS ePic SD card costs £50, which is very good value for money considering "the total retail value of the bundled software is around £585".

!NetSurf is the RISC OS bundled web browser. It's popular, but limited – its alternatives are something else we'll be looking at in the future. Finally, there's !StrongED, "a programmer's text editor for RISC OS". Read about it at **stronged.iconbar.com**.



◀ Not a retro game, !PackMan updates the packages in your RISC OS build

▲ An exclamation mark, known as a 'pling', identifies a file as an application

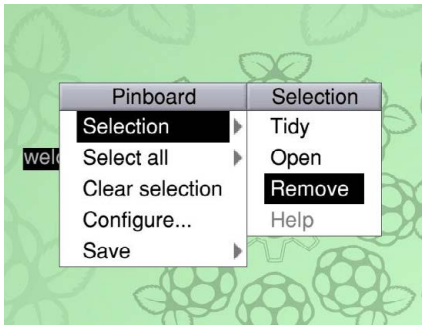
Back to the icon bar

And the icons on the left side of the icon bar? The Discs icon opens a window showing any connected network disks. If you have other computers running RISC OS on your local network, you can share disks between them. This is where they appear. Local disks, such as thumb drives and external hard drives, appear next to here as individual icons.

One drive icon that's always there, of course, is the SD card icon. This is the RISC OS microSD card you slot into your Raspberry Pi instead of the one containing Raspberry Pi OS and is the equivalent of your PC or Mac's hard drive. Click on the SD card icon to open it and explore its contents. There are folders here for your boot files, Apps, Documents, Programming, Utilities, and more. Look out too for 'Diversions', a folder containing games.

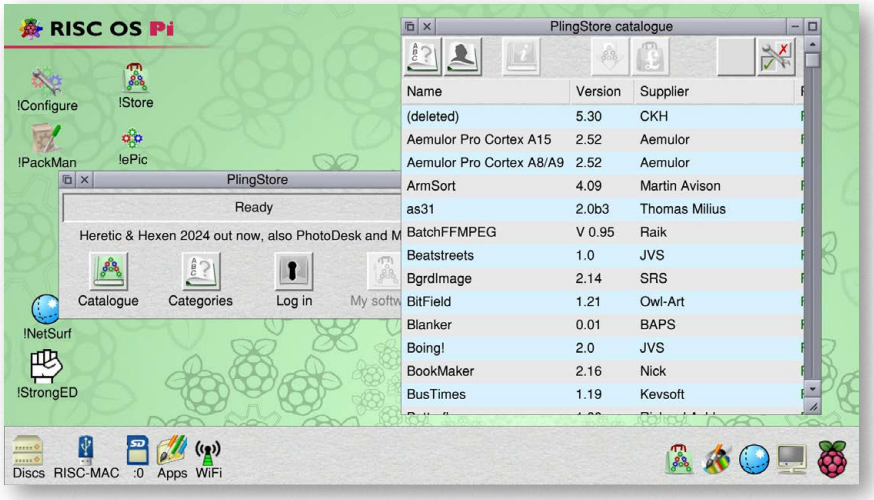
The Apps icon next to the SD card is a quick and easy way of opening an Apps window (which isn't exactly the same as the Apps folder on the top end of your SD card), and the WiFi icon connects you to the internet, which we covered last issue.

So that's it for our whistle-stop tour of the RISC OS pinboard and what it does. Next month we'll take a deeper look at... wait and see!



▲ Remove an icon from your pinboard by middle-clicking and going to Selection > Remove

▶ Download and install software from the PlingStore



Summer Projects 2025

Experience the great outdoors with Raspberry Pi

By indoorsman Rob Zwetsloot

Computing and making is, largely, an indoor hobby. Better access to electricity and generally the Wi-Fi is better too. What if we told you that you can take your projects outside, though? Surely we have now grabbed your undivided attention.

Many people love to improve their lives with a Raspberry Pi project and we're here to show you how you can do so when you leave the confines of your home. You already take a smartphone everywhere, so why not a smart project?

Grab your factor 50 and lightest shirt, and let's have some Raspberry Pi fun in the sun.

Head outside

Touch grass, as they say, in the comfort of your own garden

HeaterMeter

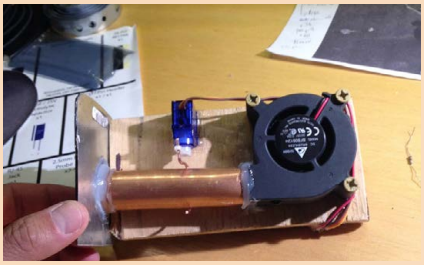
Low and slow

rpimag.co/heatermeter

Who doesn't love a good BBQ? While grilling burgers and sausages will be enough to please many folks, true barbecue heads love to smoke their food for hours on end. This obviously takes a lot more time and effort, but it can be made easier with expensive equipment or open-source thermometer tech running on a Raspberry Pi.

"Temperature data read from a standard thermistor or thermocouple probe is used to adjust the speed of a blower fan motor mounted to the BBQ grill to maintain a specific set temperature point (setpoint)," says creator Bryan Maryland. "A servo-operated damper may optionally be employed. Additional thermistor probes are used to monitor food and/or ambient temperatures, and these are displayed on a 16×2 LCD attached to the unit. Buttons or serial commands can be used to adjust configuration of the device, including adjustment of the setpoint or manually regulating fan speeds."

We can smell the ribs already.



▲ Mount it to your barbecue/smoker, carefully

◀ The little fan that keeps your food delicious

▼ It's even green, so it camouflages itself in the grass

PiMowBot

Automated lawn care

rpimag.co/pimowbot

Robotics and Raspberry Pi go hand-in-hand, and while you can definitely do more traditional outdoor robotics (whatever that might mean to you), you can also put your automaton to work by keeping your garden nicely trimmed. Like a Roomba for your grass, and it even runs on solar power. It's mostly 3D-printed too, which helps a lot in keeping costs down, but also it does require a fast spinning blade, so please be extra careful if you decide to recreate it.

Head into the wild

Ain’t no mountain high enough

GPS Logger

Track your rambles

rpimag.co/gpslogger

A nice summer walk (or hike) is great, and poring over data of what you’ve done is also fascinating. Combine both by building a GPS logger that tracks your position, speed, distance, and even altitude if you need it, and never miss out on tracking your walks/hikes/runs/skateboarding again!

While there are plenty of commercial devices that will do this – even your smartphone – it’s more fun to make your own. You can also attach devices like this to drones, RC boats, and more without having to worry about losing your pocket computer.

▲ Thanks to Raspberry Pi Zero, this device can be very compact

◀ See your route, and your stats

Pi Hiker

DIY geocaching

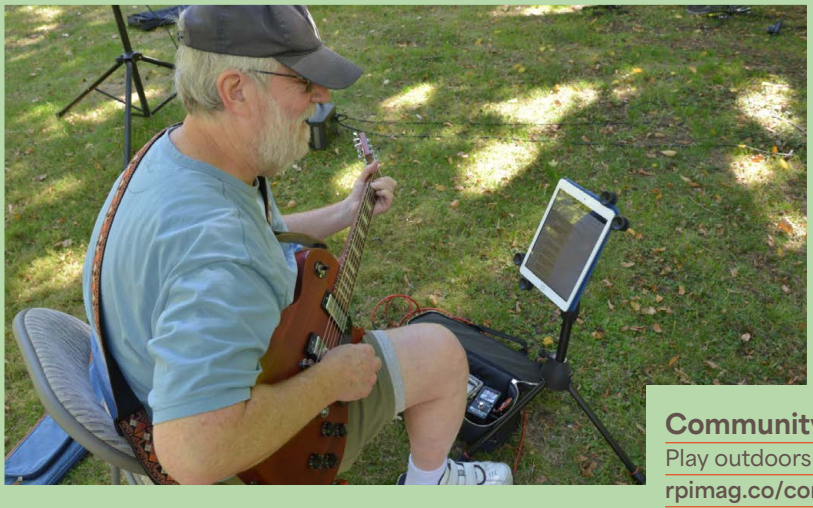
rpimag.co/pihiker

This old-school project (from 2015 in this case) uses a Raspberry Pi 2 to help you track down geocaches – hidden spots around the globe that people have set up with trinkets or curios to look at or trade. It adds a little extra fun to your standard hike.

This device points you in the direction of a geocache in the real world by tracking your movements, then giving a little arrow to point in that direction. Creator Bob Alexander tried to make it kid-friendly so his daughters could use it on their hikes.

▼ The basic interface getting local weather

Points you in the direction of a geocache



◀ Sheet music delivered straight to your tablet, so everyone is on, literally, the same page

▼ This seems like overkill just to power a little wireless network, but many other things need the juice too

Community Jams

Play outdoors

rpimag.co/comjams

This project originally came about during the various lockdowns and safety measures at the start of the COVID-19 pandemic, after what was once a fun jam session indoors became a jam session at the park. To keep track of what music everyone is playing, they use a system over Wi-Fi in their usual location.

Outside, this was not so easy. So maker Bob Steele created a new system using Raspberry Pi to serve Wi-Fi to the musicians, and the notes they could follow along to. What a pleasant way to spend an afternoon.

Cricket scoreboard

Raspberry and cream

rpimag.co/cricketscore

Lovely day for cricket, right? Of course, if you want to keep score you could log it on your phone, but for the true cricket field experience you can't do much better than a handmade board that costs a lot less than an official one.

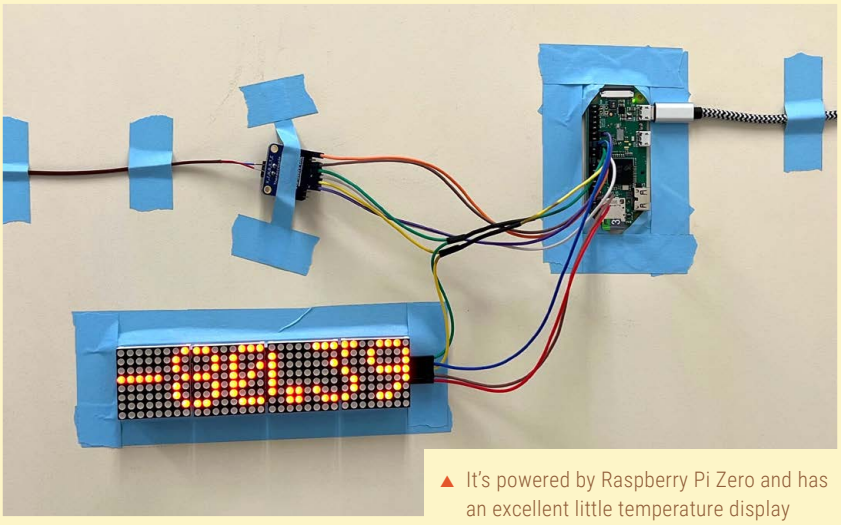
For the Westbury-on-Severn Cricket Club, this involved a big piece of plyboard with laser-cut plastic covers, big LEDs, and a Raspberry Pi that hosts a web server that controls the whole thing. You just connect with your phone and update the score.

▲ The main board in all its finished glory

◀ As we say while playing cricket: howzat!

Head to the beach

Feel the sand between your toes



▲ It's powered by Raspberry Pi Zero and has an excellent little temperature display

Icebox sensor

Cold drink keeper

rpimag.co/fridgesensor

Staying hydrated is important, and keeping water cold in a cooler box is the way to go down by the beach. An ice-cold water is heaven on a sunny day. To make sure you know your drinks are staying cool, it's easy to create a sensor with Raspberry Pi to let you know when it's time to refill the ice.

This project is a little overkill – able to properly check a freezer that needs to stay at -80°C, and it emails you when it's getting too warm. You probably don't need it that cold for cans of pop, though.

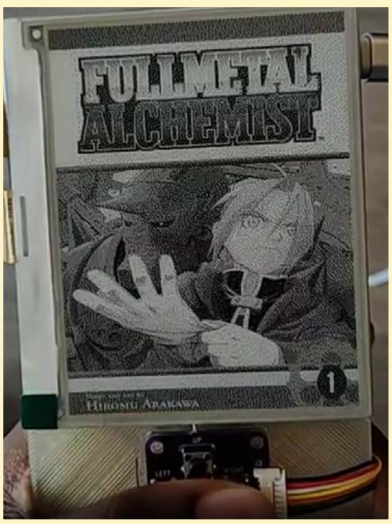
DIY e-reader

Read and tan

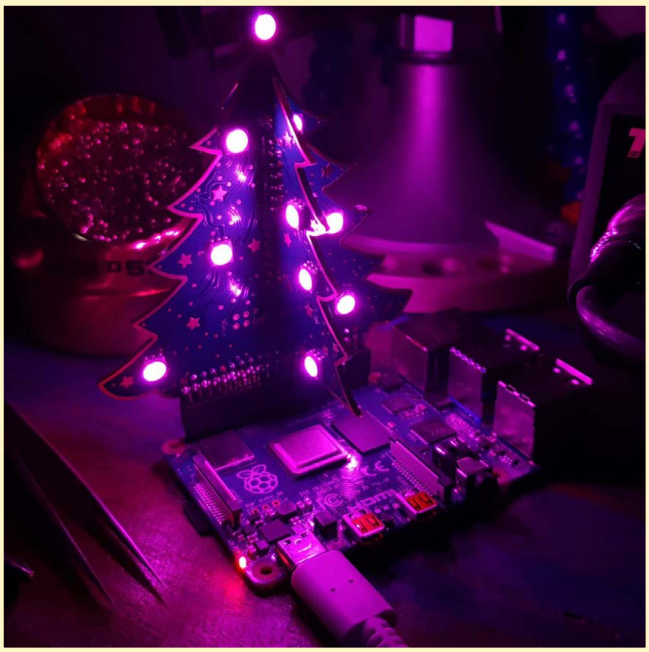
rpimag.co/ereader

Reading on the beach is a time-honoured tradition. A perfect place to relax and catch up on your favourite classic lit, trashy pulp novel, or shojo manga. Books are great, but a good e-reader can hold hundreds or even thousands of novels at once, and with a decent e-ink display you won't have to worry about glare either.

This build using a Raspberry Pi 3 has a 4.2-inch display, which is a little small. However, you can always get a bigger screen and use a Raspberry Pi Zero 2 to make it even more portable. Just plug it into a mobile battery and you're good to go.



▶ You can even read Raspberry Pi Official Magazine on it



Sandcastle lights

It's Christmas somewhere

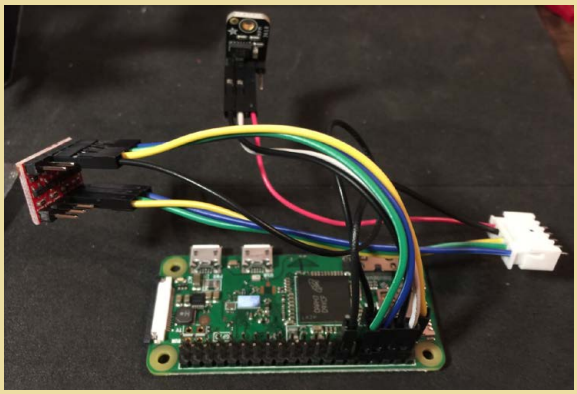
rpimag.co/3dxmastree

Building sandcastles is fun, whatever age you are. Inspired by the LED kits we keep seeing for Lego castles, why not add LEDs to your sandcastles for amazing social media posts?

It'd be remiss of us not to mention that in the Southern Hemisphere, Christmas is in the summer! December there is the perfect time to head the beach and build a Christmas sandcastle, complete with a 3D Christmas tree hooked up to a Raspberry Pi. An even better social media post, we reckon. Aussies, you have six months to prepare for it…

▶ The classic 3D RGB Christmas tree is but one Raspberry Pi-based Christmas LED project

Why not add LEDs to your sandcastles?



Raspberry Pi air conditioning

Shelter in place

rpimag.co/minisplitpi

Want to stay indoors, but finding the summer heat too much? We have you covered with this remote control for a mini-split heat pump – a modern air conditioner and heating unit.

Originally designed to help control when to operate the heat pumps when the main power goes out, it can also get it to switch from hot to cold. With an added thermistor, you could easily control it via temperature with more spots around the house to measure.

▲ This setup is housed within a Lego case

◀ It connects via a special serial port that other remotes would connect to

PicoCalc

The latest handheld offering from ClockworkPi is based around Raspberry Pi Pico. By **Jo Hinchliffe**

ClockworkPi **clockworkpi.com** $75

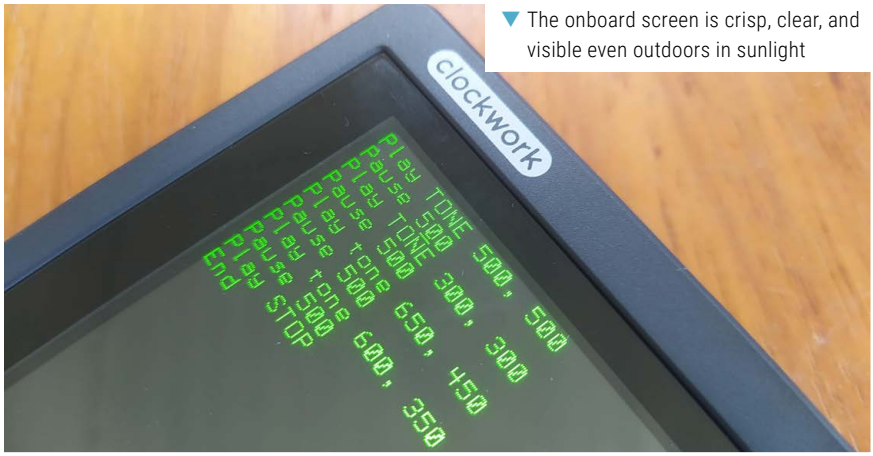
SPECS

PORTS:
USB-C, micro USB, 3.5mm audio out, GPIO header breakout

STORAGE:
32GB SD card

SCREEN:
4-inch IPS screen with 320×320 resolution

POWER:
Up to two 18650 batteries (not supplied)



▼ The onboard screen is crisp, clear, and visible even outdoors in sunlight

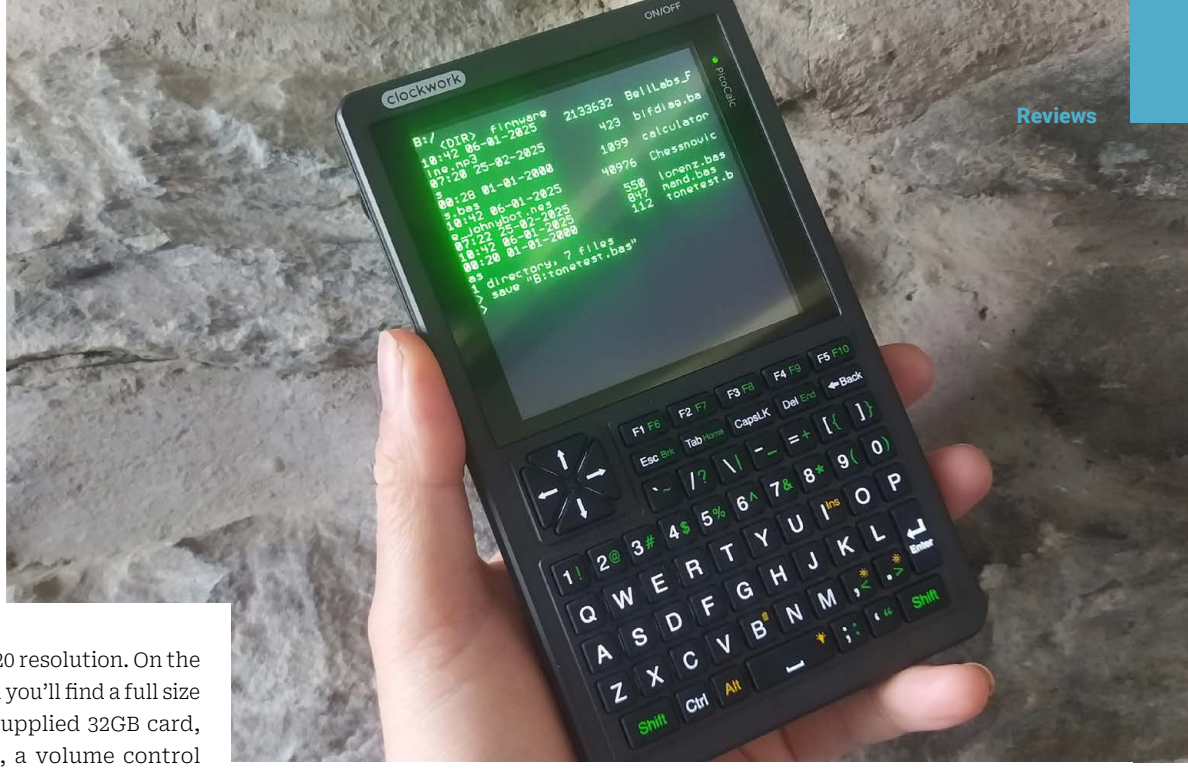
PicoCalc is a fantastic kit that's a step away from ClockworkPi, responsible for previous Devterm and Uconsole products that used Raspberry Pi Compute 4

PicoCalc is a lower-price product built around Raspberry Pi Pico. Whilst on a more accessible price tier, the quality we've come to expect of ClockworkPi kits and devices has not been compromised at all.

If you fit squarely in the Venn overlap of handheld devices, retro computing, and you're a fan of Raspberry Pi Pico, then PicoCalc will be of interest.

The PicoCalc resembles a large vintage scientific calculator, but has lots of modern tech on board. The membrane keyboard is usable as a thumb board with a satisfying key click and is backlit with four levels of light intensity. The keyboard has its own STM32 MCU on board and firmware is available via the PicoCalc repository; ours is running on the stock firmware it arrived with and performs really well.

The screen is a lovely 4-inch IPS affair with a tempered glass cover, so although you still need to treat the device with care, it's reasonably rugged and offers decent



▶ The PicoCalc packs a lot of fun into a chunky handheld form factor

brightness and 320×320 resolution. On the assembled main board you'll find a full size SD card slot with a supplied 32GB card, a headphone socket, a volume control and a horizontal header that is used to access the GPIO from both the core board (Raspberry Pi Pico as standard) or the STM32. You'll also spot the battery socket which is designed for up to two 18650 cells, although it can be powered by a single cell.

15-minute build process

The kit is a pretty straightforward build and the entire PCB is fully assembled. You need to insert a pair of small speaker modules, add the screen and a ribbon cable, and stack up the membrane on the keyboard. It's no more than a 15-minute job with great instructions and the only tool required is a supplied Allen key.

The unit arrives with a generous outer shell which has a neat, recessed power button, so there is no risk of the device powering on in your bag. It also has cutouts for both the main board USB-C connection which can charge the cells and Pico's micro USB for flashing firmware.

The supplied Raspberry Pi Pico H (v1) comes pre-flashed with PicoMite, an excellent BASIC environment built around MMBasic. On the microSD card are a few example BASIC .bas files and with the vast SD card, you have plenty of space to write your own scripts. You can't immediately find masses of documentation on PicoMite on the ClockworkPi website, but the community forum has links to guides and tutorials and lots of hints and tips from others in the community.

The screen is a lovely 4-inch IPS affair with a tempered glass cover

Yes, it will run Doom

While the PicoCalc is a pretty new device, there are already 3D-printable case extensions, people exploring the keyboard firmwares, and there are projects porting lots of other stuff including MicroPython, Lua, Lisp, and more. There are people working on updating versions of PicoMite for the Pico 2 variants and there is an amazing thread on the forum where the Pico is swapped for a pin-compatible Luckfox Lyra board, turning the device into a reasonably fully functional Linux handheld. Rounded out with lots of game emulators, it's really exciting to see where this may end up as the scene expands.

▼ The PicoCalc calculating a Mandelbrot set as an example

Verdict

A fantastic product that's great fun to play with and looks set to grow in functionality. We're knocking one point off for a slight lack of documentation, but that's heavily solved by the community forum.

9/10

ONLY THE **BEST**

Electronics prototyping boards

By **Phil King**

The GPIO (general-purpose input/output) pins on your Raspberry Pi enable you to connect electronic components and sensors as you experiment with building your own circuits on a breadboard. Since the GPIO pins aren't labelled, however, you often need to do a bit of pin counting to find the correct pin to connect to while referring to a pinout diagram. That's where breakout boards can come in useful: they make it easier to connect electronics and focus on building your projects.

Ever since Raspberry Pi was launched in 2012, third-party manufacturers have come up with a variety of breakout boards to help digital makers. They range from the very basic – 'cobblers' that use a ribbon cable to connect the GPIO pins directly to the rows on a breadboard – to more advanced boards with bonus features such as status LEDs, extra connections, input devices, and mini LCD screens.

We take a look at some of the best – including one for Raspberry Pi Pico, along with a standalone RP2350-based board – for exploring electronics prototyping.

Adafruit Pi T-Cobbler Plus

Adafruit | £8 / $8 | adafruit.com

The original T-Cobbler was launched by Adafruit in 2012, the same year as the very first Raspberry Pi! The expanded 'Plus' version works with all 40-pin models (i.e. all modern Raspberry Pi computers), but the concept is exactly the same: it's a handy way of breaking out the GPIO header pins onto a breadboard for the prototyping of electronic circuits.

The T shape enables the supplied ribbon cable to be connected between Raspberry Pi's header and the T-Cobbler without obstructing access to a breadboard. The bottom part of the T features 40 downward-facing pins to fit into the breadboard's holes – either side of its central 'ravine'. So, by inserting the end of a jumper wire to a hole in the same row, in effect you're connecting it to the respective GPIO pin. Not only is this handy, but – unlike on Raspberry Pi – all the pins are labelled, so it's a lot easier to find the one you want.

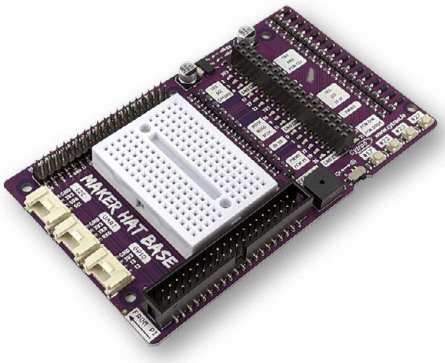
▲ The Pi T-Cobbler is supplied with a ribbon cable

Verdict

An inexpensive way of making GPIO connections a lot easier.

Maker HAT Base

Cytron | £11 / $11 | cytron.io



There's a lot packed into this very useful board

Verdict

Ideal for makers using a Raspberry Pi 400 or 500.

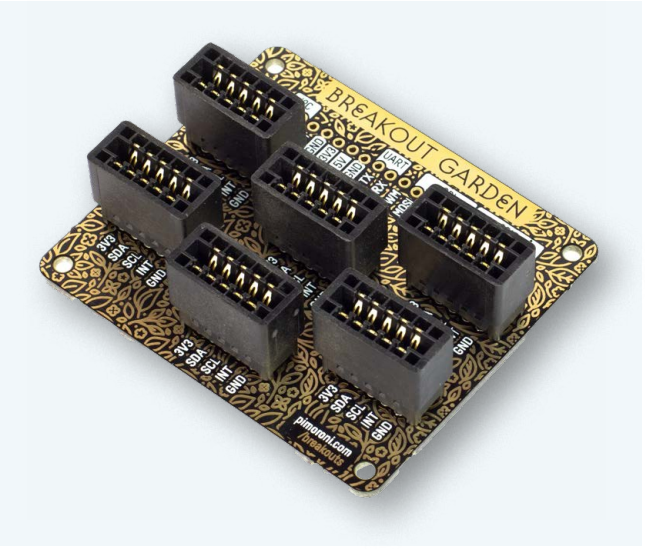
Owners of a Raspberry Pi 400 or 500 needn't miss out when it comes to breakout boards. While you can connect any HAT to its rear-mounted GPIO header, the top of the HAT will be facing the rear – not very user-friendly when you're trying to read the labels for connecting things up to a breakout board.

Designed specially for Raspberry Pi 400 and 500, the Maker HAT Base makes things easier. A supplied short ribbon cable connects it to Raspberry Pi's GPIO header, so the HAT itself can lie flat on your desk or table. You might also find it useful for extending the GPIO header of a standard Raspberry Pi model enclosed in a case.

It's equipped with a good deal of functionality, too, including onboard push-buttons, a buzzer, and a mini breadboard, along with a female breakout header with helpful pin labels and status LEDs. There's also a fully labelled male 40-pin GPIO header.

Breakout Garden

Pimoroni | £13 / $15 | pimoroni.com



There are six slots to insert tiny breakout boards

Verdict

An easier way to add sensors and breakouts to projects.

Adding sensors and other devices to a Raspberry Pi project via a breadboard is all very well, but sometimes you just want to get on with making and coding. In this case, the Breakout Garden makes it a lot easier. The standard HAT version comes with six sockets to insert any of the compatible tiny breakout boards available. There's an ever expanding array to choose from (**rpimag.co/bgcollection**), including all manner of sensors, plus LED matrices, tiny LCD screens, and even thermal cameras.

It really is plug-and play, with each breakout's fixed I2C address recognised by the accompanying software, so it'll detect them correctly even if you move them around. And, since they use a shared I2C bus, this leaves plenty of spare GPIO pins, the most useful of which are broken out by a 20-pin strip of holes.

Other Breakout Garden models are also available: the three-slot Mini and Raspberry Pi 400/500 versions, plus a six-slot Base for Pico.

Adafruit Perma-Proto HAT

Adafruit | From £5 / $5 | adafruit.com

Experimenting with circuits on solderless breadboards is fun, but there may come a time when you want to create something more permanent for a project. This is where protoboard comes in useful, enabling you to solder components and wires to create more reliable and durable circuits.

The Perma-Proto HAT kit takes it one step further by mounting a piece of protoboard on top of Raspberry Pi itself. Compatible with any 40-pin Raspberry Pi computer, the standard version can even be stacked with another regular HAT. Alternatively, choose the version with a built-in EEPROM that you can program to make a setup using Raspberry Pi HAT specs – in effect creating a custom HAT.

Either way, the protoboard features traces on the rear to connect the rows of holes in the same way as a standard breadboard, with a familiar layout. There's also a handy breakout row for the GPIO pins, along with power and ground rails.

Verdict

A great-value way to build more permanent circuits.

▲ Make your own custom HAT with this kit

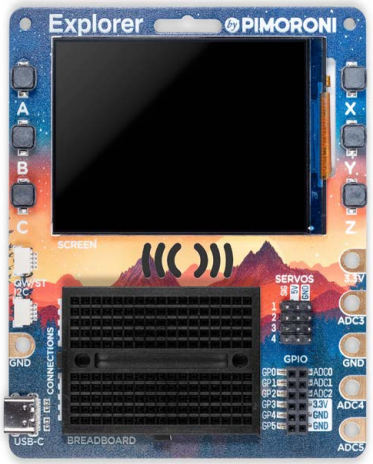
Pimoroni Explorer

Pimoroni | From £34 / $38 | pimoroni.com

◀ An RP2350-based electronics prototyping board that's very expandable

Verdict

A great way to get started with electronics prototyping.



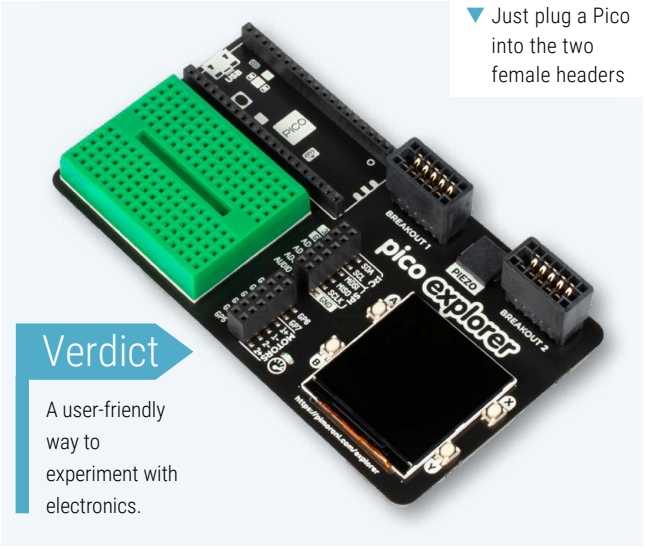
Not to be confused with the classic Explorer HAT Pro (still a great option for Raspberry Pi computers), this standalone Explorer is based around the RP2350 microcontroller chip – as used in Raspberry Pi Pico 2 / 2W, so it's programmable with MicroPython and C/C++.

Billed as an 'electronics project playground', it features a mini breadboard next to a female header breaking out a selection of RP2350's GPIO pins, including three analogue inputs, along with 3.3V and a couple of GND pins. So it's easy to add electronic components to circuits on the breadboard and connect them to the pins. There's also a header to connect up to four servos, along with a couple of Qw/ST ports and six crocodile-clip terminals.

The highlight is a 320×240 2.8-inch LCD screen, ideal for displaying data and graphics. There's even a tiny piezo speaker for basic audio. The Explorer Starter Kit version contains a range of electronics components to get started, including a Qw/ST multi-sensor stick and a couple of motors with wheels.

Pico Explorer Base

Pimoroni | £25 / $28 | pimoroni.com



▼ Just plug a Pico into the two female headers

Verdict

A user-friendly way to experiment with electronics.

Raspberry Pi Pico users aren't left out when it comes to breakout boards. The Pico Explorer Base is one of the best, featuring a mini green breadboard with 170 points, although you could always use a larger separate one if it proves a little cramped for your projects. A mini LCD screen is useful for displaying data and is equipped with four tactile buttons.

A selection of Raspberry Pi Pico's pins are broken out via two 12-pin female headers. Clearly labelled, these include I2C, SPI, ADC, and seven standard GPIO pins, along with two GND connections and one 3V3 power. An Audio pin is connected to the onboard piezo speaker.

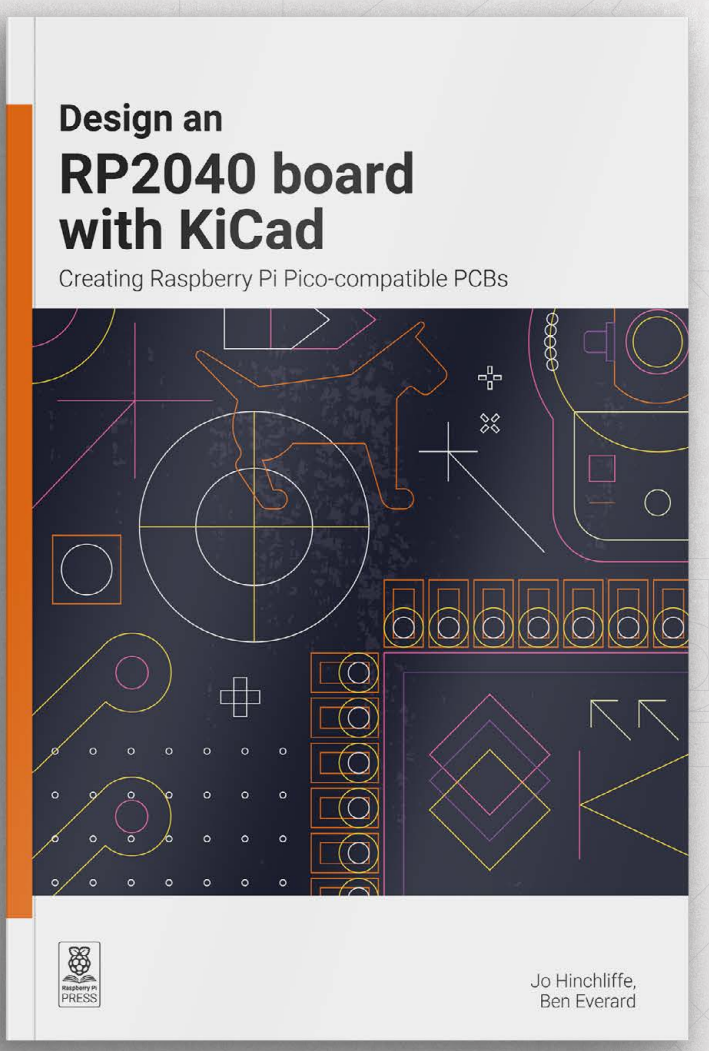
As a bonus, the four remaining breakout pins are allocated to motor connections. Making use of a DRV8833 dual H-bridge motor driver chip, these can deliver 1.5A RMS current output to control two DC motors (or other power-hungry devices such as NeoPixels).

ELECTRONICS KIT

Various | From £6 / $7 | various

While you can buy all manner of electronics components separately, a kit is a more cost-effective option for beginners. There are countless kits available, ranging from the inexpensive entry-level CamJam EduKit #1 to others comprising hundreds of components, such as the SunFounder Super Starter Kit V3.0 for Raspberry Pi (pictured). They typically come with a series of lessons to help you to get started.

▼ This SunFounder kit offers a huge array of components



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

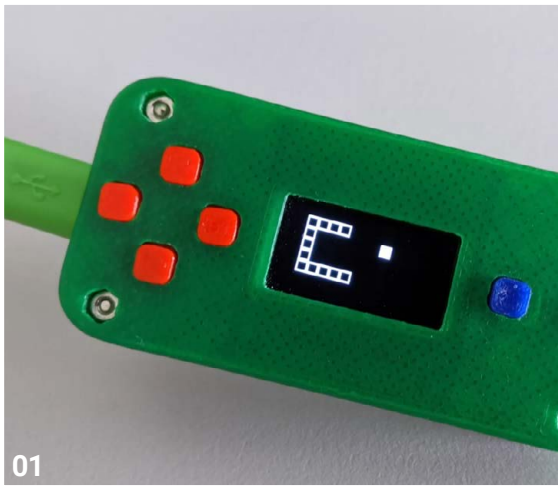
- ***Create a schematic for a microcontroller board using Raspberry Pi's RP2040***
- *Select the right components*
- ***Customise the hardware for your needs***
- *Lay out and route the PCB design*
- ***Prepare your board for manufacture and assembly***
- *Write software to get your design working*

Buy online: rpimag.co/kicad2040

10 amazing:

Tiny Pico projects

Grab your magnifying glass – some of these feel barely visible



Measuring just 51×21mm, there's no doubting that Raspberry Pi Pico is a small microcontroller board. Yet it packs some impressive innards that lend themselves well to some tiny-yet-powerful projects. We love seeing how minute some makes can be. Just as importantly, most Pico projects are also light on price, so it's certainly worth thinking small when you begin to consider what you're going to come up with next.

01. Tiny Game Console

Retro gaming

rpimag.co/tinygameconsolevid

With just enough room for some buttons and a small 0.96-inch display in a 3D-printed case, this cute, compact console can play Tetris and Snake-like games.

02. Doom on a Lego brick

Can it run Doom?

rpimag.co/doomlego

Using Pico's RP2040 chip, this project manages to squeeze a computer capable of playing and displaying the iconic first-person shooter Doom into a replica Lego brick.

03. Solar System Display

Exploring space

rpimag.co/solarsystemdisplay

The Solar System is 18.6 trillion miles in diameter, so it's amazing to see it squeezed into a portable digital orrery with a 1.15-inch Pico Display Pack.

04. Picofract

Explore fractals

rpimag.co/picofractgit

Explore the Mandelbrot set – an intricate shape created by iterating a simple quadratic equation – in a rendering demo that even lets you zoom in/out and scroll.

05. The smallest keyboard

Tiny typing

rpimag.co/smallestkb

Using a custom RP2040 circuit and bespoke PCB, you'll need nimble fingers to bash away at this tiny cap-less keyboard packed with 59 microswitches.

06. PicoZX Handheld

Handheld classic computing

rpimag.co/picozxhandheld

Sinclair wanted to bring computing to the masses with an affordable machine, making this cheap-as-chips tiny computer emulating the classic ZX Spectrum nigh on perfect.

07. Pica

Tiny robots

rpimag.co/pica

Created from a Pico board trimmed up to the crystal, the tiny robot Pica incorporates a couple of 6mm geared motors, a driver, and a battery.

08. PicoStepSeq

Mini music sequencer

rpimag.co/picostepseq

This compact MIDI sequencer has eight steps, an OLED interface, and a rotary encoder, all wired directly to Pico. It outputs audio via a 3.5mm TRS jack.

09. Level crossing lights

Model railways

rpimag.co/levelcrossingvid

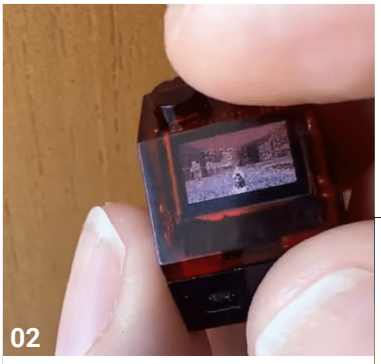
By wiring Pico to a set of tiny level crossing lights for a model railway, this project detects passing trains and determines when it's safe for the model pedestrians to cross.

10. Niki boat

Micro RC boating

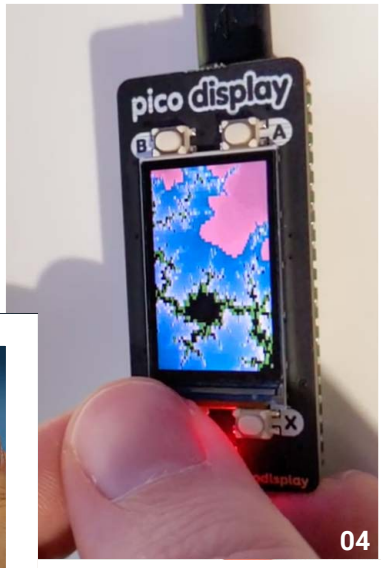
rpimag.co/nikiboatvid

With Pico connected to a dAISy HAT driving a small motor, this tiny RC boat has a body made from recycled contact lens solution bottles.



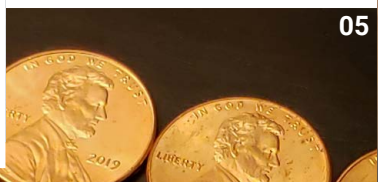
02

03

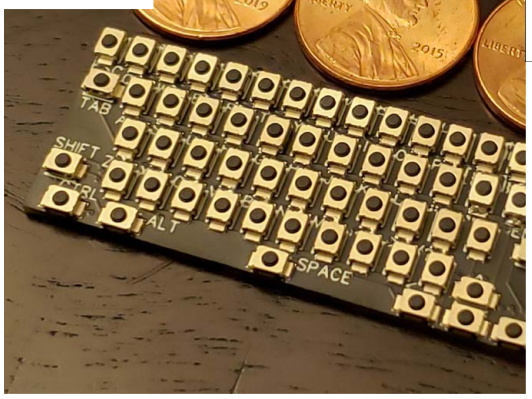


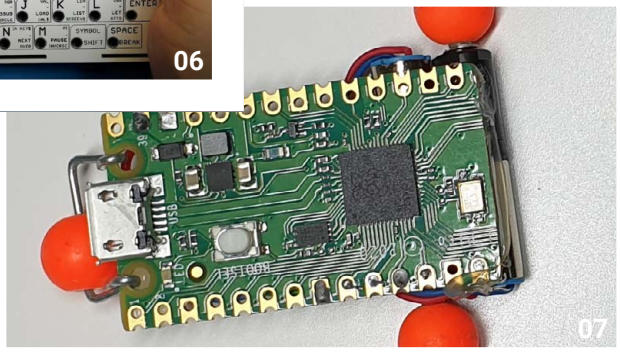
04

06

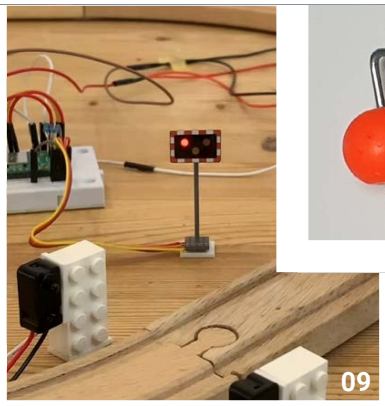


05



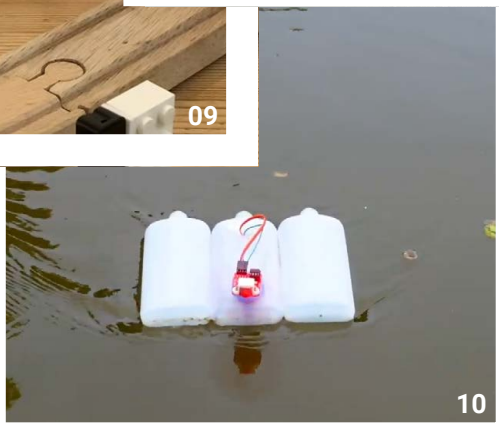


07



09

08



10

Maker Monday

Amazing projects direct from social media!

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

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

01. We could do with a few of these at specific, very old pubs

02. Ah, the tank method of turning: have one side of the treads run faster

03. This will be very useful near the climax of an alien invasion movie, we think

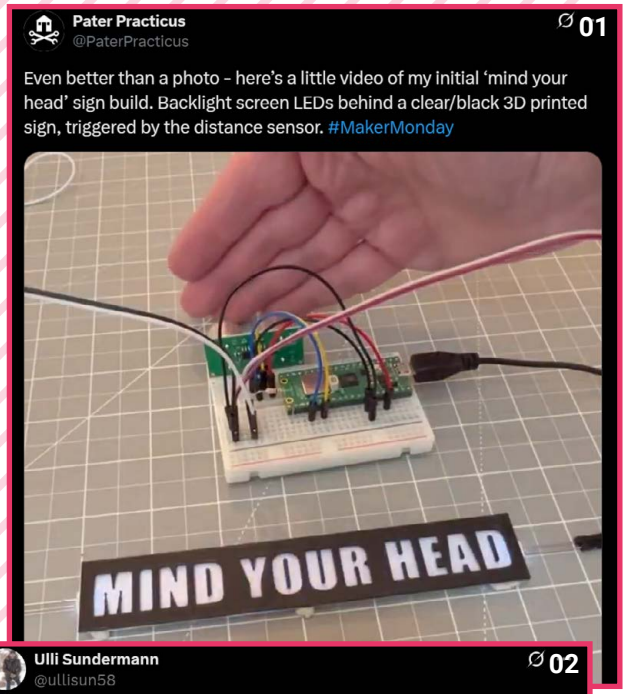
04. This is a very impressive use of Pico and AI coding! LLMs can be quite good for code ideas

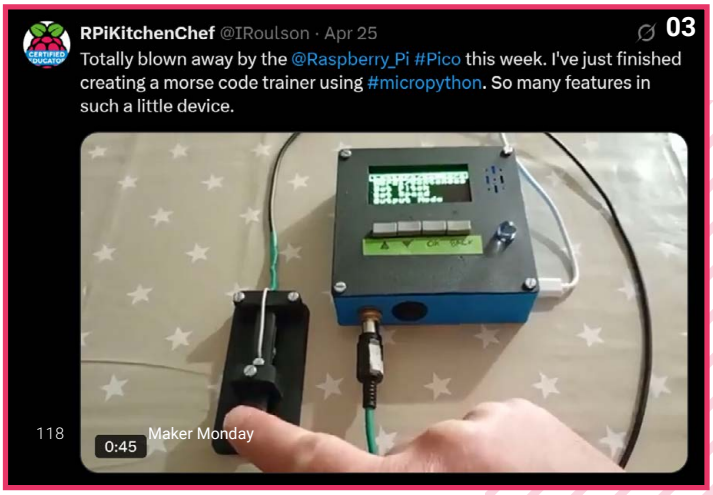
05. The cyberdeck is nearly ready. Maybe next month we'll see the hinge…

06. Dr Footleg spreading more Raspberry Pi knowledge

07. Simple yet elegant. We love the cat on it. We hope it's on a one-second walk cycle

08. We keep hearing about newer/cheaper lidars – we're intrigued



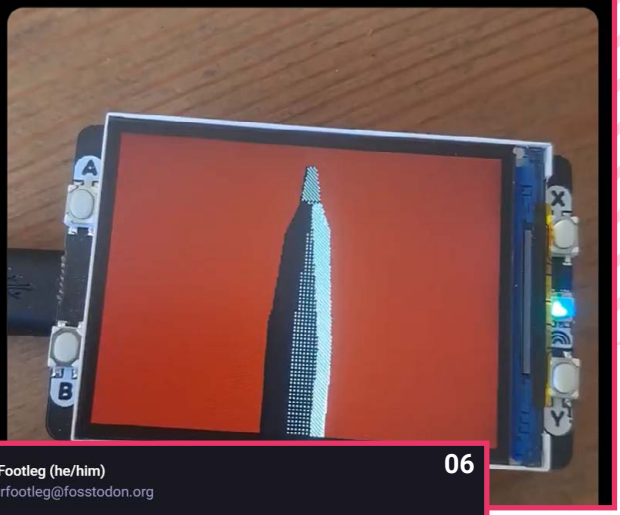


oficinastk
@oficinastk

04

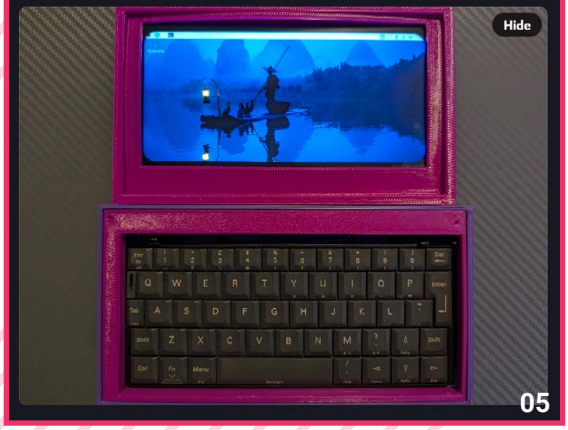
My coding skills are near to none, but recently with the help of a few AI agents, I was able to make a simple 3d engine with dithered based shading that runs on a Pico :)

Geometry is exported from Blender with a py script, also made with AI.



Michael Horne
@recantha@mastodon.social

@rpimag Finished the #3dprinting of my #Psion Pi case. Just need to fix the two halves together! Raspberry Pi 3A+ under the hood. #makermonday



05

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

06

@rpimag I held my monthly free community coding session this weekend in a small rural Suffolk library. We had the best turn out yet, and some really excited and happy new children discovered coding. I know you like pictures, but I didn't want to post one of the children who attended, so here are my own children who helped me set up and pack away all the Raspberry Pi sets. Great to have them learning the skills of volunteering.
#MakerMonday

maresakuのものづくり
@maresaku_mono

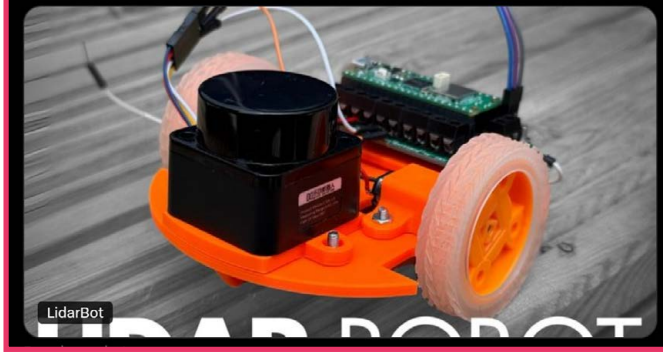
07

This weekend I made a clock using a Raspberry Pi Pico 2W and a circular display(GC9A01).
I also tried displaying pictures of cats on it.
I really like it.
#MakerMonday

Kevin McAleer 🤖 Robot Maker
@kevsmac

08

Good Afternoon! This weekend I created a LidarBot that can see its surroundings using a cheapish Lidar - You can view the objects around it as a radar map within Thonny! Its cute too!



HackRoom – a Raspberry Pi escape room

Making cybersecurity education engaging

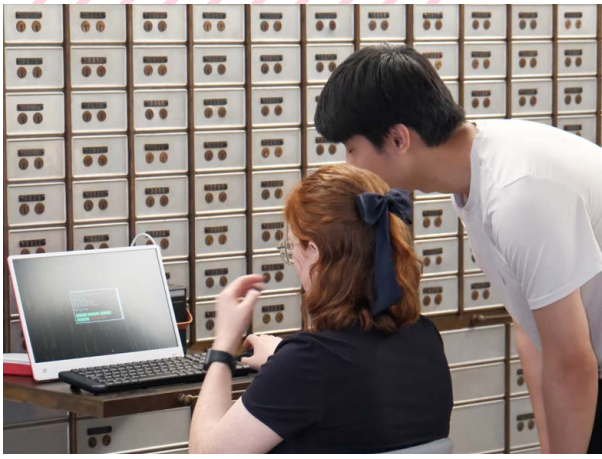
We received an email from Marius Muench recently about an event he is involved in that's running at the time of writing – HackRoom (hackroom.co.uk).

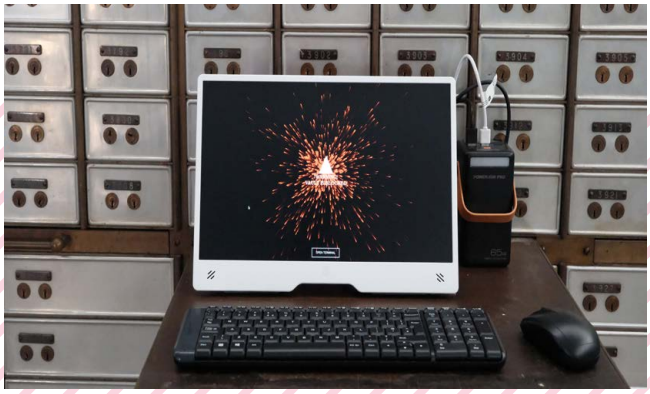
"With HackRoom, we built an interactive escape room, designed as a portable pop-up experience," Marius says. "Our goal is to make cybersecurity education accessible, engaging, and enjoyable for people of all ages and backgrounds."

Immersive challenges are interwoven with story, and Raspberry Pi 5 is used throughout these.

"Regardless of obtaining items from a vulnerable vending machine, finding secrets in a digital art gallery, gaining access to a login terminal of a security guard, or attacking an insecure website – a variety of our puzzles are powered by Raspberry Pi 5s," Marius continues. "They present their contents on Raspberry Pi monitors and touchscreens, which are ideal for a lightweight and portable installation. Distributed over five distinct rooms, the puzzles tie together to create a unique experience with challenges reflecting real-world threats and scenarios in the field of cybersecurity.

The idea is for the event to help highlight cybersecurity threats to attendees and maybe help a few of them to make a career choice.





▲ Hacking your way through an escape room seems like a unique experience

◀ Raspberry Pi 5 is deployed through HackRoom

Crowdfund this

Great crowdfunding projects this month

BerryBot Robot Kit

A Raspberry Pi-powered robot kit designed for kids to learn about electronics and programming. It has various modern functions like linking a smartphone app, block-based programming, and plenty of written lessons in the project book to learn from.

▶ **kck.st/42MfXk0**

CrowPi 3

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

▶ **rpimag.co/crowpi3**

Your Letters

A Raspberry Pi for children

Thanks for the guide to setting up a Raspberry Pi for younger children [issue 153]. I've lost count of the number of gifts for my kids I've had to gently turn down because they're open to the internet. It's the most basic thing, but somehow people can't grasp that the web is not for children! Your writer's points about adapting point size for eyes that can't focus like adult eyes, and getting used to sitting up straight at a desk with a proper keyboard, are also well taken. There are so many toy laptops out there, I can't believe we're setting up the next generation to sit hunched over like prawns.

James via email

Indeed so. Of all the reasons to recommend Raspberry Pi as a home computer, the ability to customise it as you see fit is one of the most compelling. We're used to setting up machines for older relatives, but we rarely give a thought to setting up a machine for the use of a child – other than constant supervision to keep them away from the unsavoury corners of the internet, like Baby Shark videos.

▼ Never mind kids, we know some adults who need a keyboard like this

▲ Meshtastic: brilliant for networking in out-of-the-way places

Meshtastic

I found the Meshtastic article in issue 152 fascinating, so much so that I now have six test nodes around the house!

I have been eagerly looking forward to part 2, mentioned at the end of the first article, so I am disappointed it's not in issue 153 – I even got up early this morning to download the PDF for this!

Will part 2 be published? Have I got to wait another month?

Graham via email

Six Meshtastic nodes sounds like a lot of fun for not a lot of money – we'd well recommend anyone have a go at this brilliant, accessible technology. Much as we enjoy making things work (and making a magazine every month), maths has always been hard, and somehow with the sequencing of the Meshtastic tutorials we managed to take one plus one and make three. However, we've now fixed that scheduling conundrum, and you'll find the promised part 2 of Jo Hinchliffe's guide to Meshtastic in this issue.

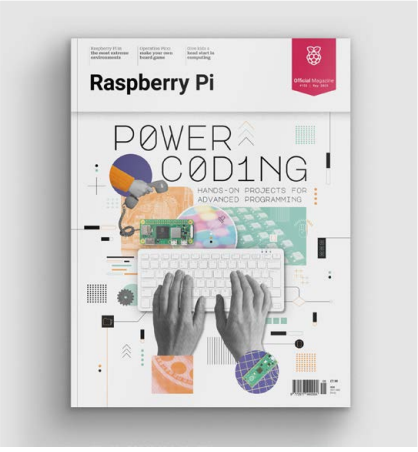
Letter typography

I really liked the font you used for the Power Coding article headlines in this month's issue. Would you share the font name? If not, that's OK. By the way, I really enjoyed the article and found it did help me with my Python skills! It is very hard to come by tutorials that are not for beginners but more advanced coders. So thanks!

Gina via email

We're glad you like it! Coding really is a superpower, but it's not the most visually appealing subject (unless the code in question is controlling something spectacular like a robot). Like every skill, coding needs practice, and practice needs structure and imagination. We wanted to show a few new avenues to explore, with Rust and JavaScript as well as more ways to think about Python. And while it may seem like dabbling in another language may be a waste of time, everything you learn in one language helps you think better in every other language – a bit like how a native English speaker might never know when to use 'whom' instead of 'who' until they learn a bit of German.

Oh, and the font: it's Majör Mono Display.



▲ We're big fans of the constructivist art movement – and we're not the only ones!

Contact us!

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

Community Events Calendar

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

01. Melbourne Raspberry Pi Makers Group Meeting

- **Sunday 1 June**
- **Docklands Makerspace and Library, Melbourne, Australia**
- **rpimag.co/mrpmgm154**

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

02. Riverside Raspberry Pi Meetup

- **Monday 9 June**
- **3600 Lime St, Riverside, CA, USA**
- **rpimag.co/rrpm154**

The purpose of Riverside Raspberry is to share knowledge related to Raspberry Pi hardware in particular, and to promote interest in tech development in the Inland Empire, California, in general. The group is currently meeting on the second Monday evening of each month. While the group is focused on Raspberry Pi specifically, they also cover topics about all kinds of maker technology, as well as having discussions about various programming languages and about electronics in general.

03. CosmoPi

- **Friday 13 June and Saturday 14 June**
- **Lagos State University, Lagos, Nigeria**
- **rpimag.co/cosmopi**

CosmoPi is a hands-on, immersive space workshop designed to ignite your passion for space exploration and technology. Using the versatile Raspberry Pi, participants will dive into the exciting world of space science, learning how to build and program devices that simulate real-world space missions. From tracking satellites to simulating Mars rover missions, this workshop will equip you with the skills to turn your space dreams into reality.

04. Raspberry Jam 2025 – FORRES

- **Thursday 26 June**
- **Juan Octavio School in Gauna, Forres, Argentina**
- **rpimag.co/forres**

This event offers an opportunity for enthusiastic young students to experiment, learn, and share knowledge about developing technological projects using a Raspberry Pi board. The organisers have planned activities for all ages and levels. The day will feature inspiring talks and interactive workshops, seeking to continue expanding the local technological community. This event will be aimed at third-grade students in the afternoon shift.

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

05. Internet of Things with Raspberry Pi Pico workshop

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

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

Official Raspberry Pi Event

Win 1 of 10 KIWI KVMs

The KIWI KVM by Cytrence is a hardware-based screen capture tool that enables you to use another computer to control your Raspberry Pi. It's easy to set up and there's no need for local networking, making it ideal for capturing video, taking screenshots, troubleshooting, and more.

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

Terms & Conditions

Competition opens on **28 May 2025** and closes on **26 June 2025**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from Raspberry Pi Official magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter (X) or any other companies used to promote the service.

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

Reach a global community!

Advertise in
Raspberry Pi Official Magazine

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

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

Take the next step –
advertise with us today!

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

Official Magazine
#155

MEDIA HUB

Regain control of your media playback

On sale 26 June

PLUS!

Training AI models

Powerful photo projects

DIY printed circuit boards!

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 Hinchliffe, Nicola King, Phil King, Dave Lee, Andrew Lewis, Simon Long, Rob Miles, Ian Osborne, David Plowman, 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)

Getting industrious

Putting computers to use.
By **Lucy Hattersley**

I'm often surprised by just how versatile Raspberry Pi is. In the last few weeks I've created an AI-powered Raspberry Pi detector, a Sense HAT data logger, a media player, a retro computer, and a Raspberry Pi radar system. That last one's coming soon – I just need to figure out a use for it.

Whatever you want to make. There's a computer for that...

Raspberry Pi stands at stark contrast to the sealed slabs of glass that people are often glued to. You don't really learn anything from holding a phone or tablet in your hand, and it's no surprise to me that schools are increasingly looking to get smartphones out of the hands of kids.

More and more, we're seeing Raspberry Pi used in industrial settings. It's robust and affordable with rigorous compliance testing, so is a no-brainer for many companies to deploy.

There's a clear path from learning how computers work to putting that knowledge into practice in an industrial setting. Why would anybody ever stare at a phone instead of learning how computers work?

I was at the Moscone Center in San Francisco when the modern smartphone was revealed for the very first time. It was like viewing the future. But as David Cameron famously quipped to Tony Blair: "[you were] the future once". The future now belongs to the kids that own Raspberry Pi computers.

Navigating the new world

We live in what Sir Terry Pratchett taught me to understand as 'interesting times'. It's becoming clear that AI, and in particular LLM (large language models), is eating up the internet and killing off social media platforms.

Now feels like a very good time to put down the phone and pick up a book. As well as making this incredible magazine, the whole team has been pulling together to write more books. So while we are chatting, I'll just plug our new book: *Experiment with the Sense HAT 2nd Edition* (**rpimag.co/sensehatbook**).

> *It's increasingly looking like the next big breakthrough is going to be Agentic AI*

It's starting to look like the next big breakthrough is going to be Agentic AI. This is where an AI is tasked to pursue an objective rather than just responding to prompts. The idea is that software functions like an agent that operates on behalf of a user, and it will have context awareness of its environment and conditions and have memory to learn from past experiences.

I find this development fascinating. Having often thought of technology as a friend, I am intrigued by the idea of Raspberry Pi acting more like a collaborator to work with than a tool. Although there are obvious concerns about safety and ethics with decision-making hardware (Beth's Pink Sentient Switchblade from *Rick & Morty* springs to mind).

The future belongs to those who understand AI and can build it with their own hands – using tools like Raspberry Pi. Tinkering is more than just fun; it's a way to build the world of tomorrow.

Lucy Hattersley – Editor
Lucy is editor of Raspberry Pi Official Magazine and is surprisingly down-to-earth when her head isn't in the clouds.

rpimag.co

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

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

Available at these great Pi stores:

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

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

PiKVM

Remote control **redefined**

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

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

PiKVM V4 Plus

The most feature-rich edition

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

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

Available at the main Raspberry Pi resellers

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

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

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

HiPi.io

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

List of official resellers by country: