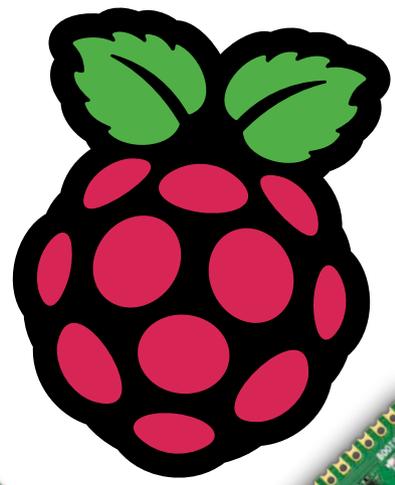


BUY IN PRINT **WORLDWIDE** MAGPI.CC/STORE

The *MagPi*



Issue 116 | April 2022 | magpi.cc

The official Raspberry Pi magazine

HIDDEN HACKS

FOR RASPBERRY PI

GET **EVEN MORE**

OUT OF YOUR

MICROCOMPUTER

TIPS
FOR EVERY
MODEL!



CAPTURE THE NIGHT SKY WITH ASTROPHOTOGRAPHY

MECHANICAL CYBER-DECK

LEARN JAVA WITH RASPBERRY PI

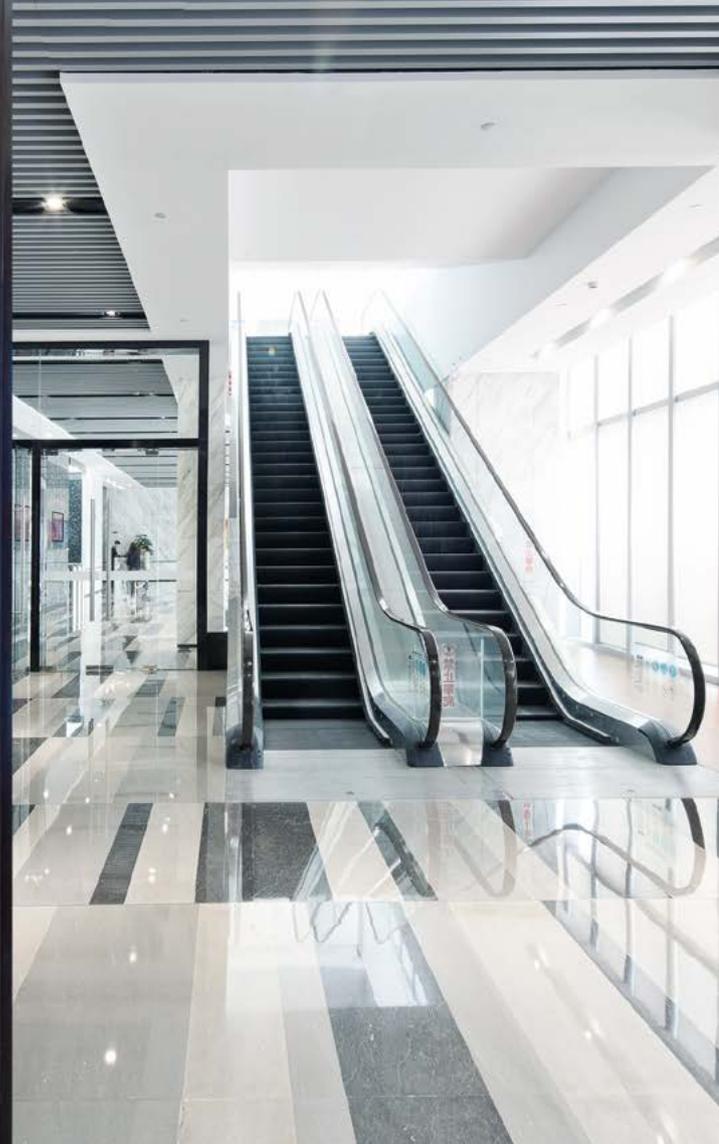
TOP 10 RASPBERRY PI ROBOTS

REVIEWED!

8BITDO ARCADE STICK WEATHER HAT BADGER 2040

GLOBAL DELIVERY
magpi.cc/store

Raspberry Pi PRESS



Escalate your controller design in style

Fully configurable universal case system- the perfect fit for all your application needs

Every panel on the UCS enclosure range can be customised enabling you to create your own individual stylish design easily.

With optional wall, desk & DIN rail mounting adaptors the range is available in 2 colours, 4 sizes & 2 heights ensuring you'll find the right enclosure for your next project.

For more information visit <https://phoe.co/ucs-rpi-uk>

WELCOME

to The MagPi 116

We've spent a lot of time tinkering with Raspberry Pi and its wonderful Linux-based operating system, Raspberry Pi OS. As a result, we've discovered a rich seam of hacks, tweaks, and interesting tips that would be plain rude not to share. So, this month's lead feature is Hidden Hacks (page 32), where we reveal some of the best things we've learnt when using Raspberry Pi.

Looking upwards, Rob has an interest in astrophotography, and the Shoot for the Stars feature (page 66) explains all about capturing the night sky. All you need is Raspberry Pi and a powerful - but not necessarily expensive - camera.

April 2022 marks the 40th anniversary of another favourite computer: Sinclair's amazing ZX Spectrum. One maker has taken Raspberry Pi Zero and squeezed it inside a Spectrum's cassette tape (page 8). Another is using Raspberry Pi to scan barcodes on retro games and play them in a retro games store.

One more thing: if you've ever wanted to learn about the ARM processor, be sure to look at the start of a new set of tutorials by Stephen Smith (page 46). Fascinating stuff.

Hope you enjoy this issue!

Lucy Hattersley Editor



EDITOR Lucy Hattersley

Lucy is editor of *The MagPi* and never did get the hang of Z80 machine code.

@LucyHattersley



E-PAPER DISPLAYS

More details, please check Waveshare website:
<https://www.waveshare.com/epaper>

Multi-Size

From 1.02" to 13.3"

Display Port

USB / SPI / I80 / HDMI

Other Features

Support Partial Refresh / Flexible

Grey Scale

2-16

Display Color

Black / White; Black / White / Red; 7-Color

Support Device

Raspberry Pi / Pico / Jetson Nano / Arduino / STM32 / ESP32 / ESP8266 /
PC / Android and iOS Phone (with NFC)

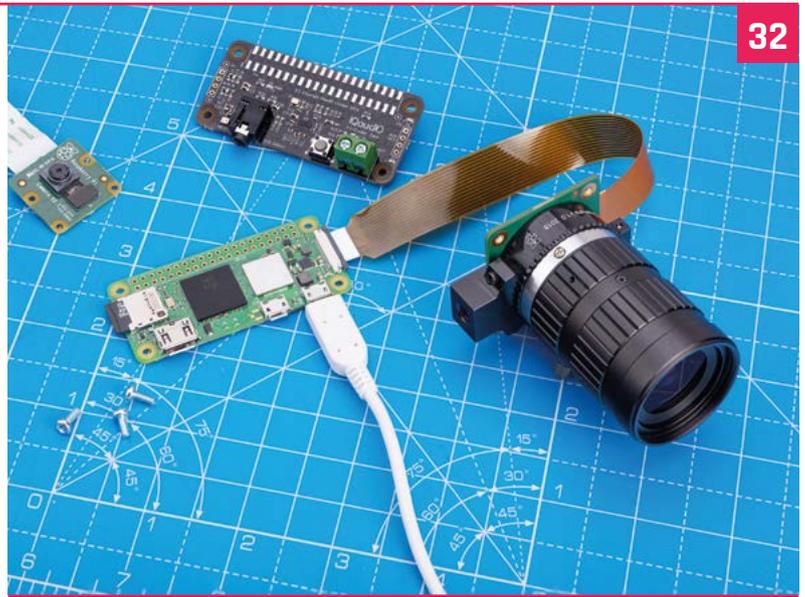


Contents

► Issue 116 ► April 2022

Cover Feature

32 Hidden Hacks for Raspberry Pi



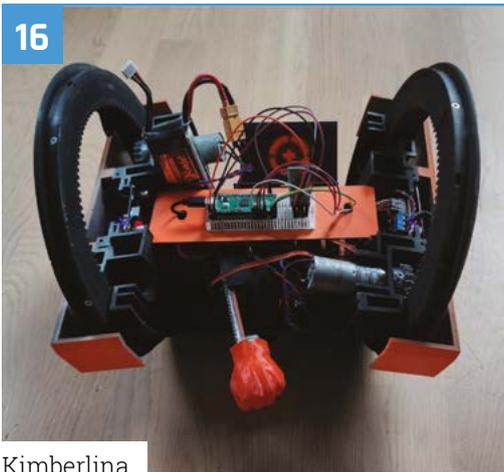
Regulars

- 28** Case Study: Elevated Materials
- 92** Your Letters
- 97** Next Month
- 98** The Final Word

Project Showcases

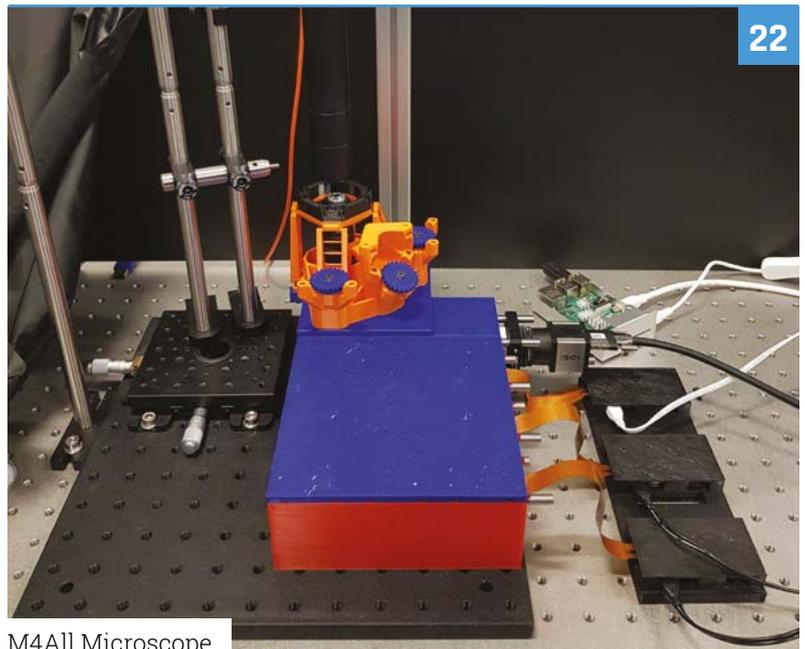
- 08** ZX Spectrum Raspberry Pi Cassette
- 14** Penkesu Computer
- 16** Kimberlina Droid
- 20** Time Machine Radio
- 22** M4All Microscope
- 26** Retro Barcode Scanner

16



Kimberlina

22



M4All Microscope

The MagPi magazine is published monthly by Raspberry Pi Ltd. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send US and Canadian address changes to The MagPi magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

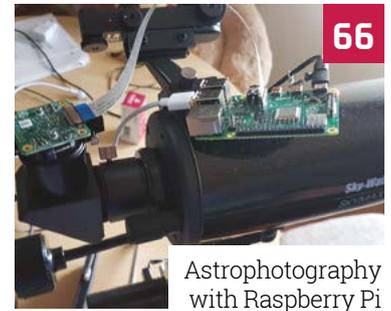
Tutorials

- 46** Learn ARM Assembly
- 52** Digital Audio Workstation
- 56** LED patterns with web GUI
- 60** Pico Hot Wheels Racetrack

The Big Feature



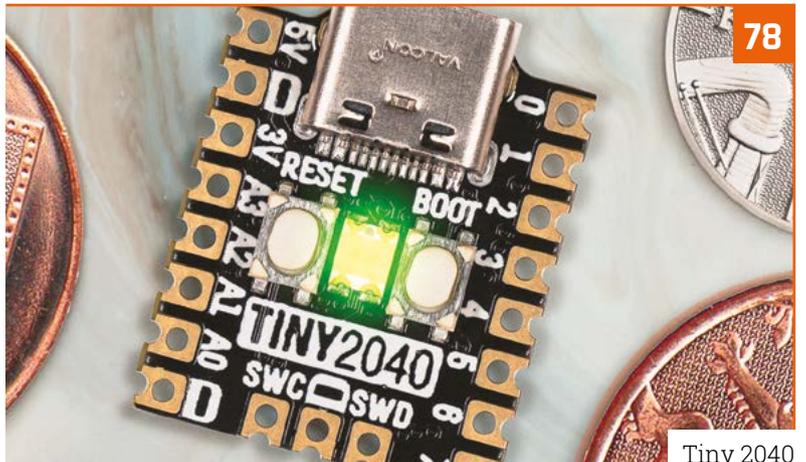
Learn ARM Assembly



Astrophotography with Raspberry Pi



Astrophotography with Raspberry Pi



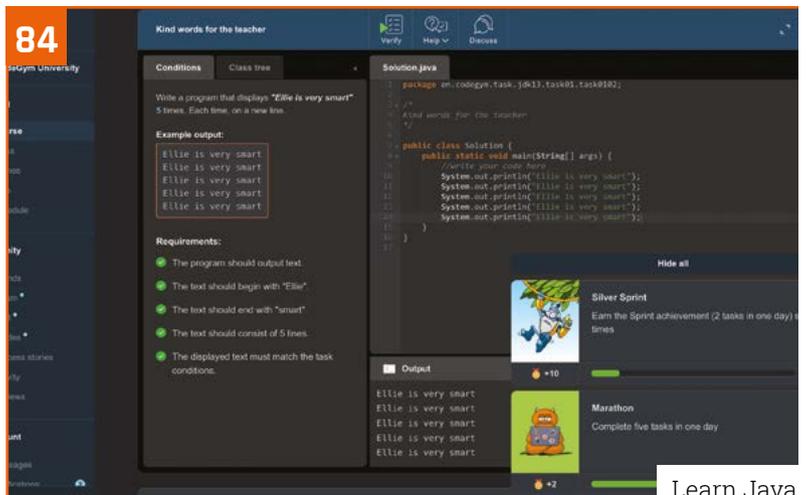
Tiny 2040

Reviews

- 74** 8BitDo Arcade Stick
- 76** Weather HAT
- 78** Tiny 2040
- 80** Badger 2040
- 82** 10 amazing robot projects
- 84** Learn Java

Community

- 86** Mike Cook interview
- 88** This Month in Raspberry Pi



Learn Java

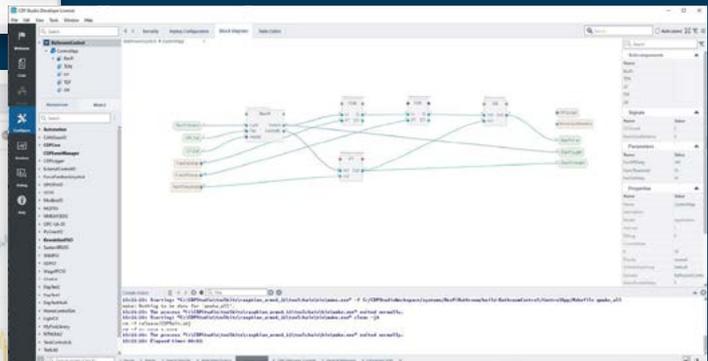
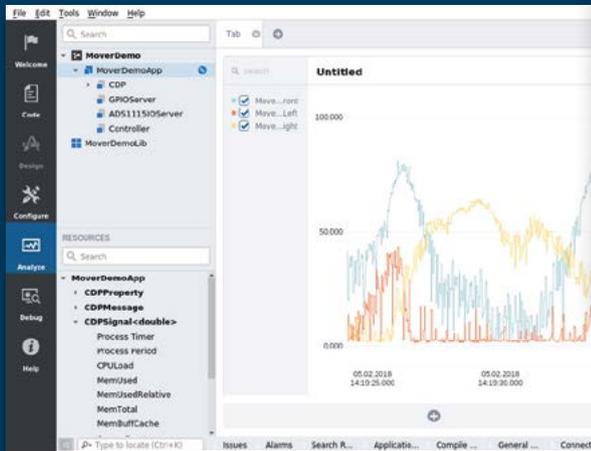
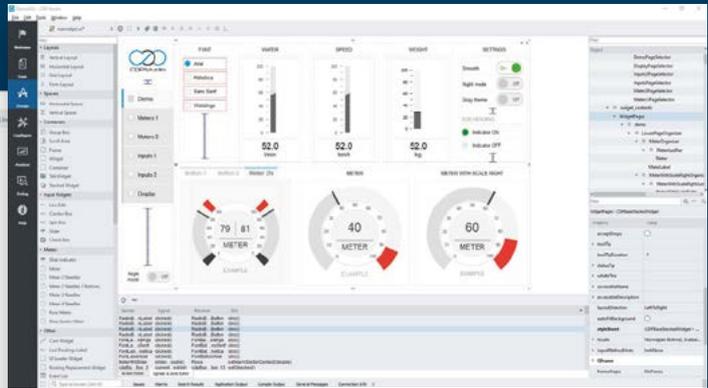
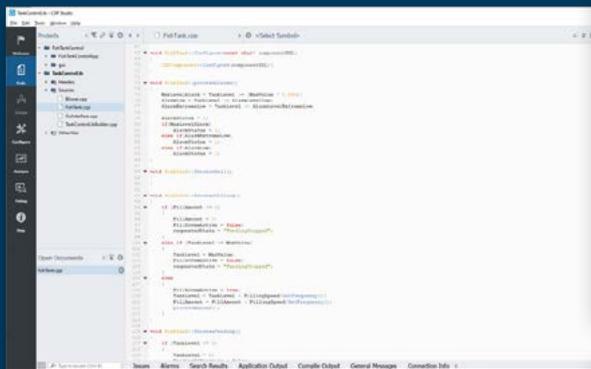
WIN
1 OF 5

8BITDO PRO 2 CONTROLLERS

95

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

Code



Configure

Analyze

With RPi 4 and
Bullseye support

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

CDP Studio, a great software development tool for your home projects. Build systems for Raspberry Pi, use C++ or NoCode programming, open source libraries, out of the box support for GPIO, I2C, MQTT, OPC UA and more. Create beautiful user interfaces. Built for industrial control system development, **FREE for home projects.**

cdpstudio.com

Tel: +47 990 80 900 • info@cdptech.com

CDP Technologies AS // Hundsværgata 8, 6008 Ålesund, Norway



ZX Spectrum Raspberry Pi Cassette

Between jobs, one maker decided to push their Raspberry Pi skills and make a portable ZX Spectrum Raspberry Pi. **Rosie Hattersley** approves



Stuart Brand

Stuart aka JamHamster is “an IT Root Cause Engineer for work and an avid (some would say obsessed) tinkerer in my off hours!”

@RealJamHamster

Stuart Brand was between jobs and decided to concentrate on pushing his skills by building Raspberry Pi projects: “I headed to the garage and embraced my inner nerd!” exclaims the maker of the ZX Spectrum Raspberry Pi Cassette. “I wouldn’t have had a clue how to build any of this stuff before lockdown. It goes to prove that you never know what you’re capable of until you give it a go.”

Stuart’s first computer, a ZX Spectrum, has a special place in his heart, so a Raspberry Pi project based around one seemed ideal. “They’re still great machines!” he says of the beloved computer which celebrates its 40th birthday this April.

Stuart loves repairing and running real hardware as well as emulations and thought “it would be

nifty to see if I could fit an entire ZX Spectrum emulator into a cassette tape shell.” He now uses his ZX Spectrum Pi Cassette as a ‘pick up and play’ device whenever he fancies “a quick bash at some old school gaming.”

Learn as you go

Prior to this project, Stuart had several retro makes under his belt and had made a tape emulator for an Arduino-based ZX Spectrum +2 that acts like a multi-cart tape. “Putting a whole Spectrum in a tape shell was the next logical step and an interesting challenge,” he says. Being tight for space, he chose Raspberry Pi Zero W. He loves the fact both ZX Spectrum and Raspberry Pi’s ARM processor were developed in Cambridge.

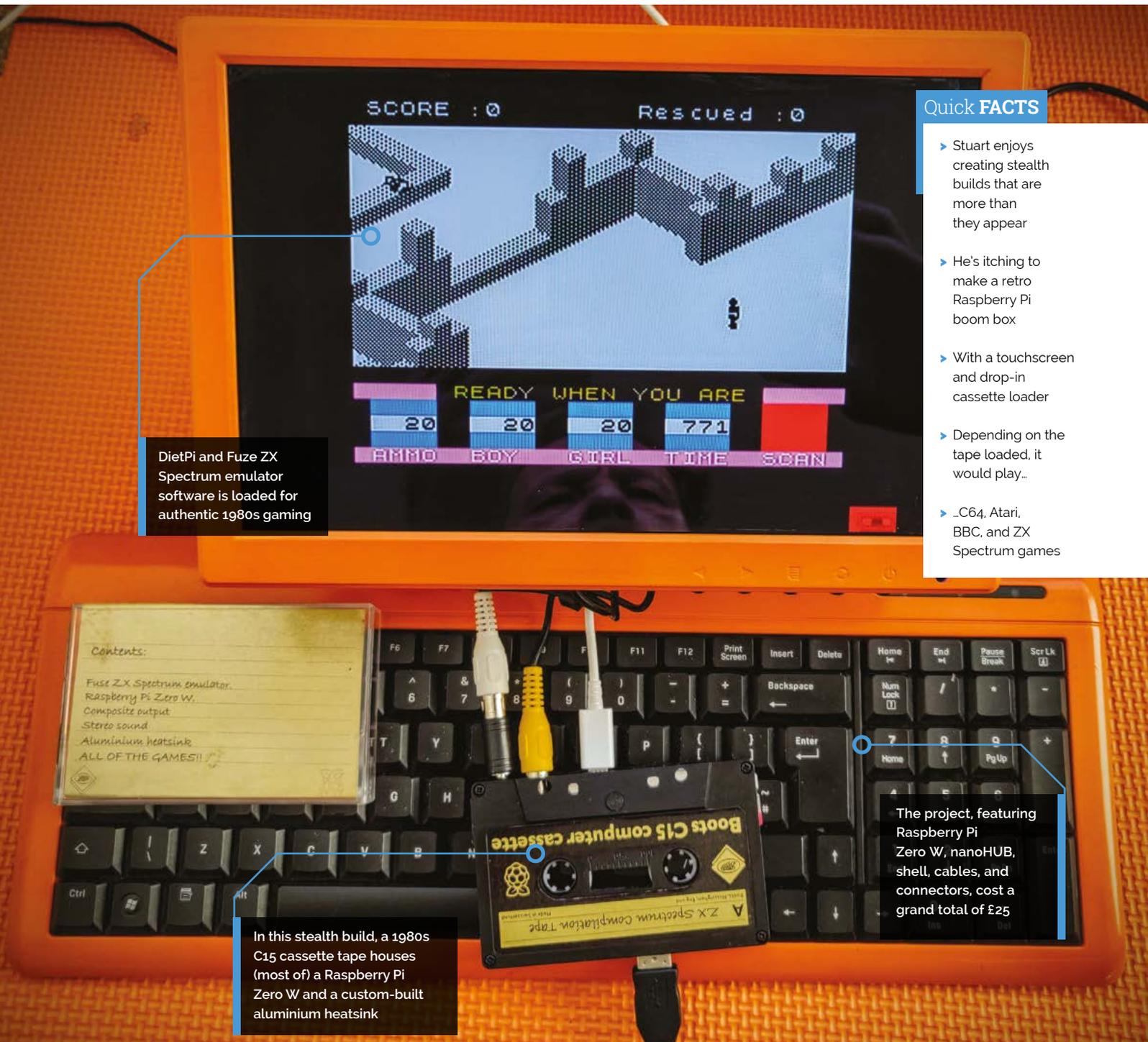


▲ The handcrafted heatsink fits beautifully inside the repurposed cassette tape

“ Stuart wrapped Raspberry Pi in foil and went at it with a Dremel ”

Despite this, he describes himself as a haphazard tinkerer with little electronics experience, who plans everything in his head. “I don’t have any schematics to share,” he apologises, “and never measure anything.” However, he makes paper mock-ups of everything he’s planning, largely to ensure it all fits. A veteran of small case builds, Stuart cautions other wannabe makers to leave far more room for cables than you think you’ll need. He also admits to treating his Raspberry Pi collection rather roughly: “even though they have been abused and tortured, they still keep running.”

Stuart assembled the ZX Spectrum Raspberry Pi build from what he had to hand. He took a sheet of scrap metal and used a bandsaw to fashion a crude



DietPi and Fuze ZX Spectrum emulator software is loaded for authentic 1980s gaming

Quick FACTS

- ▶ Stuart enjoys creating stealth builds that are more than they appear
- ▶ He's itching to make a retro Raspberry Pi boom box
- ▶ With a touchscreen and drop-in cassette loader
- ▶ Depending on the tape loaded, it would play..
- ▶ ..C64, Atari, BBC, and ZX Spectrum games

Contents:

Fuze ZX Spectrum emulator
Raspberry Pi Zero W,
Composite output
Stereo sound
Aluminium heatsink
ALL OF THE GAMES!!

In this stealth build, a 1980s C15 cassette tape houses (most of) a Raspberry Pi Zero W and a custom-built aluminium heatsink

The project, featuring Raspberry Pi Zero W, nanoHUB, shell, cables, and connectors, cost a grand total of £25

shape for what would act as Raspberry Pi Zero W's heatsink. A Dremel, needle files, and fine-grit sandpaper were used to finesse the shape.

Getting it taped

Stuart bought a job lot of cassette tape seconds: "Boots C15 were the cassettes I used for storing my programs back in the '80s; it was an obvious choice"

– for which he designed and printed new labels. "Cassette shells make for a great form factor," says Stuart, "I started with a plain black spare cassette shell and used a small hand file and side cutters to remove the plastic supports in preparation for fitting the heatsink."

The 5 mm interior of the C15 cassette tape meant something would have to give: fitting a Zero W



- ▶ Stuart has form making retro gaming builds using Raspberry Pi
- ▶ Customised cassette labels complete the 1980s look



Alert!
Warranty voiding

This project involves cutting off the GPIO pins in order to fit Raspberry Pi Zero W inside a cassette tape. This risks damaging Pi Zero W and invalidates your warranty. Wrapping it in aluminium foil while cutting is a sensible precaution. Cut carefully.





“ I lost some GPIO ports, but it was well worth it to get the tape looking right ”

inside involved cutting out a section to nestle under the reels and “preserve the illusion” – not something inexperienced makers are advised to tackle. Stuart has eight similar builds under his belt, hence his confidence. He wrapped Raspberry Pi in foil and “went at it with a Dremel.” Surprisingly, it survived. “I lost some GPIO ports, but it was well worth it to get the tape looking right.”

Configuring the DietPi and Fuze ZX Spectrum emulator took lots of tweaks before Stuart was able to get them to boot in an acceptable time frame. “I eventually got it to boot in 16 seconds. The full-width heatsink meant I could safely overclock Zero W and saved another couple of seconds,” he says.

His next challenge: a 1980s boom box with drop-in cassettes that boot up and play games from different iconic home computers. We like his thinking! [W](#)

▲ Stuart's cassette edition of the ZX Spectrum atop an original Spectrum keyboard

Make a tape



- 01** Install DietPi (dietpi.com) and the Fuze ZX Spectrum emulator (fuse-emulator.sourceforge.net) on your Raspberry Pi and set it to autorun at startup.



- 02** Create a heatsink following the contours of the cassette case, and avoiding the spool area. Wrap Raspberry Pi Zero W in aluminium foil and very carefully cut off the GPIO port section where it will prevent the spool wheels fitting.



- 03** Position Raspberry Pi Zero W in the cassette shell, then fit the ports. Stuart used GPIO sound and an RCA connector for composite video out and added a shutdown button on the front of the tape, and then hooked it up to a shutdown script.

Where the World Innovates



FREE SHIPPING
ON ORDERS OVER
£33 OR \$50 USD*



0800 587 0991
DIGIKEY.CO.UK



2,200+ INDUSTRY-LEADING SUPPLIERS | 100% FRANCHISED DISTRIBUTOR

*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of \$18.00 USD will be billed on all orders of less than \$50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA



English is not your mother tongue?

MagPi is also available in German!



Subscribe to the German edition of MagPi and get a Raspberry Pi Pico with headers and a cool welcome box **FOR FREE!**

Use the coupon code **115PicoDE** on www.magpi.de/115



Penkesu Computer

Pop Raspberry Pi in your pocket and compute on the go with Penk Chen's stunning mini retro computer. **David Crookes** takes a look



Penk Chen

MAKER

Penk is a software engineer and a toy hobbyist. He refers to himself as a digital nomad from Taiwan and lives in Tokyo with Ruha Cheng and two cats, Musashi and Kojiro.

[penkesu.computer](#)

Electronic organisers were once all-the-**rage, especially among businessmen in the 1990s. Roughly the size of a spectacles case, they allowed people to keep a check on their daily activities.** But while Penk Chen appears to have created his own cool, minimalist version of one of those devices, inspiration for his project came from elsewhere.

Penk was looking to build upon his work with mechanical keyboards. “I’d learned not only how to build them from scratch, or mould and cast my own key caps, but also how to use a laser engraver to melt pigment onto key caps to print my own legends,” he says. “I recall staring at those low-profile key switches and thinking to myself, this could make a great retro-style computer”.

There were two things he wanted above all: compact physical dimensions and a nostalgic theme. “Back in the 1990s, gadget pencil cases were extremely popular among kids in Asia,” he says. “I figured if I were to pay homage to the past, what could be better than recreating the

magical moment of opening a pencil case?” This thinking led to the project’s name: Penkesu is the Japanese word for ‘pencil case’, while happening to reference the maker himself.

Key to success

Identifying the appropriate components proved relatively easy. “Since this project started with the keyboard, finding the other parts was fairly straightforward,” Penk affirms. At first, this meant looking for a screen to match the size of his 48-key mechanical keyboard, and it led him to choosing a 7.9-inch touchscreen display.

He then opted for a Raspberry Pi Zero 2 W computer. “It was a natural fit,” he says. “Looking back at its predecessors, you will find that there’s virtually nothing to complain about except to marvel at the wonders of modern technology.” The rest of the components included a LiPo battery, some diodes, and a keyboard-controlling Arduino Pro Micro, but designing a 3D-printable case was the greatest challenge.

Making a good case

“The weight distribution was not an easy task,” Penk explains. “The screen is mounted in a rather top-heavy fashion, so it took me some time to properly design the chassis and place counterweights so that it would not tip over when opened at its maximum angle.” This was achieved by gluing a thin metal sheet to the base. “The end result is a low-centre-of-gravity, stable, and satisfyingly clicky device,” he adds.

Interestingly, the project also uses a set of replacement hinges designed for a Game Boy Advance SP console, making the device easy to open and close while adding extra retro goodness.



▶ Penk says the spray paint colours had him “stumped”. “Using beige feels a bit like ‘cheating’ but thankfully it turned out fine,” he says



The Penkesu has been designed with just three internal components: Raspberry Pi Zero 2 W, a LiPo battery, and an Adafruit PowerBoost 1000

The Waveshare 7.9-inch widescreen LCD display has a resolution of 480 × 1280 and a HDMI connection



This keyboard – called Koda and designed by 'larrbo' – doesn't have a top set of numbers. It is released under a Creative Commons BY-NC-SA licence (magpi.cc/oddrocket)

Quick FACTS

- ▶ A ribbon cables carries the HDMI signal
- ▶ It's designed to have as few parts as possible
- ▶ It folds to protect the screen, like a mini-laptop
- ▶ The keyboard uses an ortholinear layout
- ▶ The Arduino runs the QMK keyboard firmware

- ▼ The mini-computer runs Raspberry Pi OS and makes for a cool retro-style handheld device

“What could be more nostalgic than using genuine console parts from the 1990s?”

“What could be more nostalgic than using genuine console parts from the 1990s?” he asks. It was then a matter of connecting everything up – in this case, largely plugging the components together – and getting Raspberry Pi to run Raspberry Pi OS for an instant mini-machine.

“Everyone knows that when it comes to presenting your build to the interweb community, cable management is the most important thing, and I’m terrible at soldering,” he laughs, before adding that in a world of supply chain disruption, it also made the project more viable. “I’m now experimenting with the mouse keys feature of QMK firmware to see if I can rely on the keyboard more when using Raspberry Pi OS,” he says, explaining that he’s not finished with this project just yet. [M](#)



Kimberlina Droid

Selin took inspiration from Star Wars when designing a Raspberry Pi Pico-powered battle bot. She talks strategy with **Rosie Hattersley**



Selin Ornek

Selin lives in Istanbul and fits building robots and giving TED talks about STEM around her schoolwork. Her mission is to use AI for good.

selinoid.com

▼ Selin and Kimberlina take to the battleground and prepare to vanquish less speedy foes

In *Star Wars: Attack Of The Clones*, a convoy of tanks with enormous wheels that bear down inexorably on their enemies was a fearsome sight. The concept seemed ideal to 15-year-old robot builder Selin Ornek when she needed ideas for a new battlebot. Having already made quite a name for herself in the world of coding, Selin was likely to have plenty of competitors keen to take her down, which partly explains why she named her battle bot Kimberlina in tribute to a comedy character from TV show *Full House* that she watched in lockdown. Kimberlina may have a quirky name, but her strike is deadly! Once you know something about Kimberlina’s maker, this won’t be a surprise.

Selin is an accomplished coder and Raspberry Pi robot builder. She began coding aged eight – first Scratch, but later Python, Java, and C++ – and designed her first robot at ten.

Her interest in coding began after Selina interviewed a family friend, who is a mechanical engineer, about building a robot dog as a means of bringing her much-loved dog, Korsan, back to

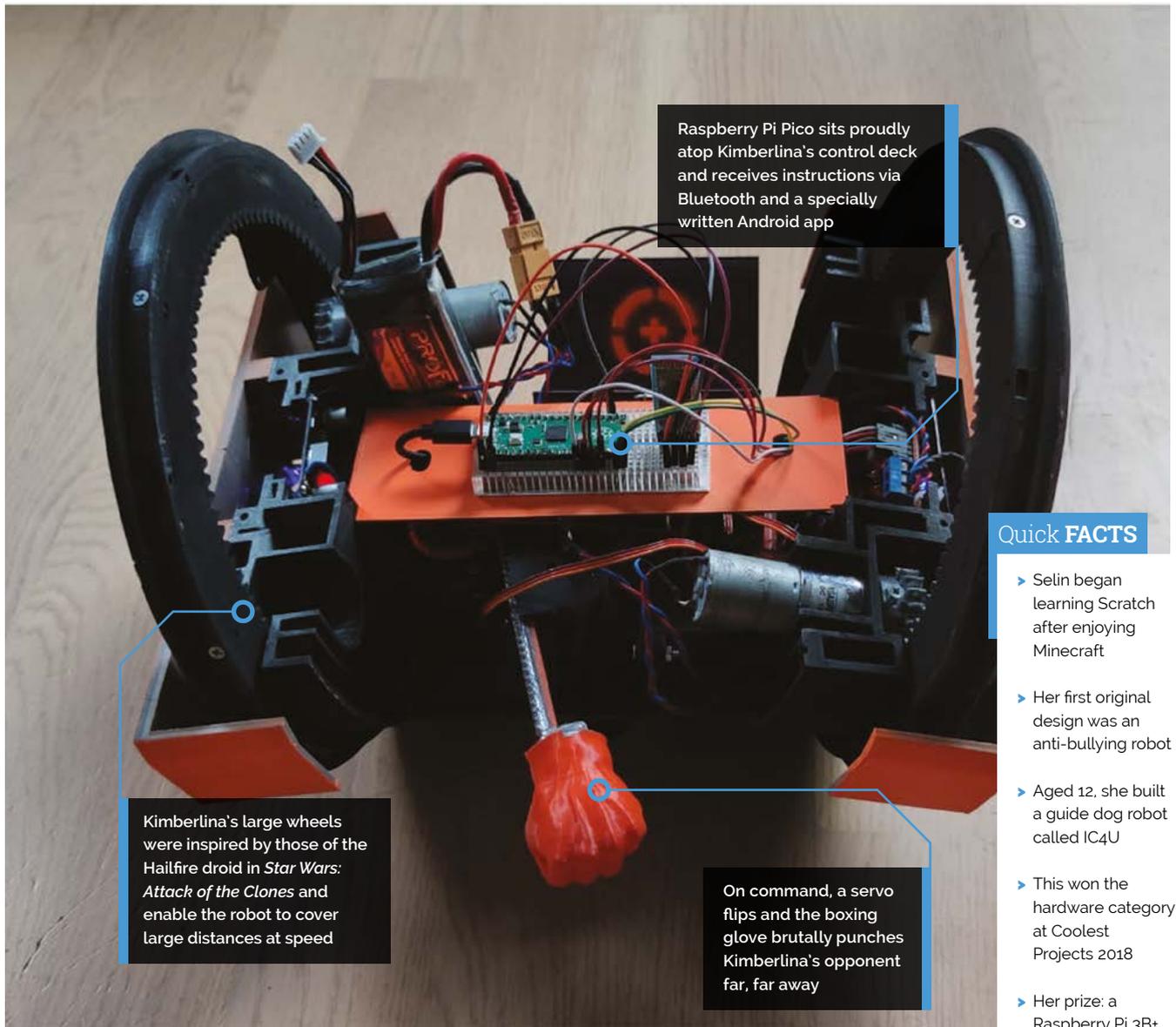
life. His advice to learn coding and robotics, plus Scratch-based lessons that helped her class learn English, set Selin on her robot-building journey. She has now built six robots, including two that act as guide dog robots for blind people. The original version of this, the Arduino-based IC4U, won Selin her first prize at Coolest Projects International in 2018. “I won first place in the Hardware Category at Coolest Projects, and one of my prizes was a Raspberry Pi 3B+.” The other was a Google AIY kit – presented to her by Raspberry Pi’s very own Eben Upton. “After this, I started to use Raspberry Pi in all my projects.”

With a firm focus on using technology for good and demonstrating that an interest in computing is gender-free, Selin won the Aspiring Teen category in the 2021 Women In Tech Global Awards (magpi.cc/witawards).

Firm foundations

Kimberlina began life as a cardboard prototype, which helped Selin decide on the parts she was going to use. The robot seemed a great opportunity to try out Raspberry Pi Pico for the first time, since she needed a microcontroller that could run motors, servos, and receive Bluetooth signals. Selin used the wheel design from her previous robot and designed other parts in Autodesk Tinkercad. “Because the wheel has a very small contact point with the ground, I had to find a way to balance the robot when it accelerates forward and backwards, so I placed steel ball bearings with enough space to move within the robot,” she says. The contest in which Kimberlina was to compete threw up challenges too, stipulating maximum weight and speed limitations. “The weight was a challenge after the decision to use ball bearings, but choosing the right motors was the key to [having] a fast-moving and turning robot.”





Raspberry Pi Pico sits proudly atop Kimberlina's control deck and receives instructions via Bluetooth and a specially written Android app

Kimberlina's large wheels were inspired by those of the Hailfire droid in *Star Wars: Attack of the Clones* and enable the robot to cover large distances at speed

On command, a servo flips and the boxing glove brutally punches Kimberlina's opponent far, far away

Quick FACTS

- ▶ Selin began learning Scratch after enjoying Minecraft
- ▶ Her first original design was an anti-bullying robot
- ▶ Aged 12, she built a guide dog robot called IC4U
- ▶ This won the hardware category at Coolest Projects 2018
- ▶ Her prize: a Raspberry Pi 3B+

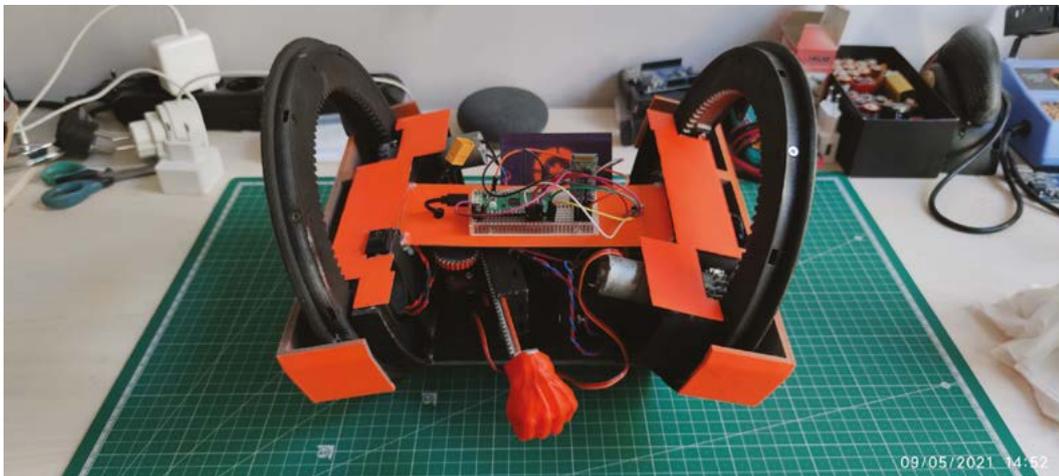
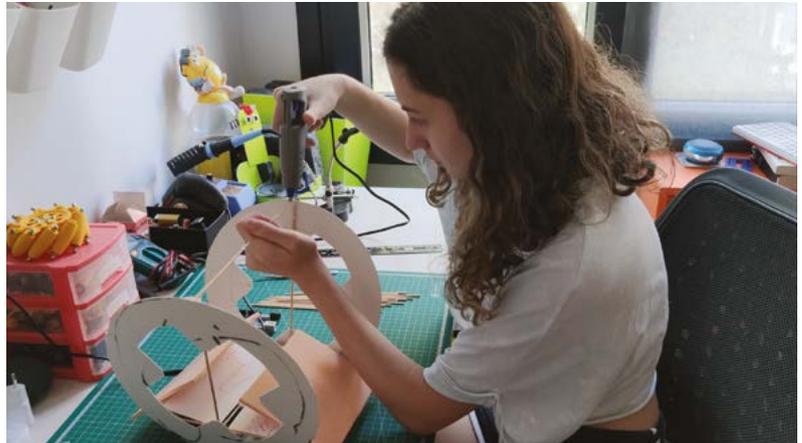
“ Selin's competition strategy was to move quickly and remain agile avoiding contact ”

Decisions made, Selin 3D-printed and assembled Kimberlina's body, using the time available while the body parts were printing to design a mobile app in MIT App Inventor to control the droid over Bluetooth. She used MicroPython to code Raspberry Pi Pico and the hardware connected to it. Once the coding was complete and the robot was assembled, it was time to test it. “I was lucky,” says Selin of how her meticulous planning and experience played out.





◀ iC4U2 is a Raspberry Pi robot guide dog, one of six robots Selin has designed to date



▲ Having built five other robots, Selin knew exactly what circuitry to use for Kimberlina

◀ A powerful punch and speedy wheels make Kimberlina a fearsome robot opponent



◀ Kimberlina being weighed prior to the competition



“ Kimberlina is controlled via Bluetooth from a mobile application ”

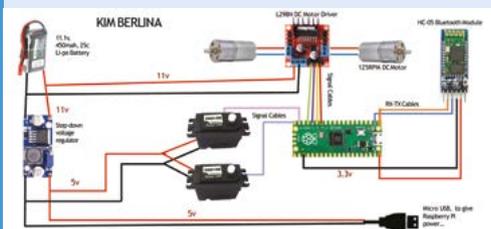
▲ Selin won first place in the Hardware Category at Coolest Projects in 2018

“Usually my robots do not work immediately. I come across a problem and sometimes even burn a part, but fortunately, Kimberlina worked on the first try.”

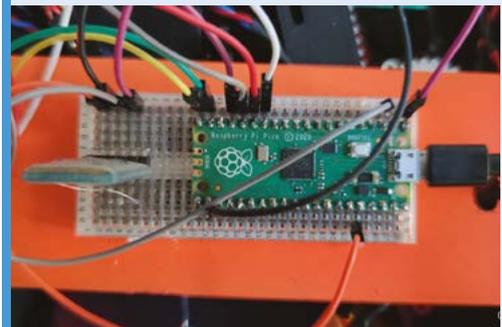
Speed and control

Kimberlina is controlled via Bluetooth from a mobile application Selin built using MIT App Inventor. A servo at the front controls the flipper to flip an opponent, while the servo at the back is attached to a lever that pushes the opponent away. “Kimberlina has great balance thanks to steel ball bearings that help centre it. This means that when it gets a hit or accelerates fast, it doesn’t roll over,” explains Selin. “My competition strategy was to move quickly and remain agile, avoiding contact, and make the other robots fall from the platform without touching them,” she adds. The mighty wheels of Hailfire, coupled with the portable power of Raspberry Pi Pico, proved an unbeatable combination. [M](#)

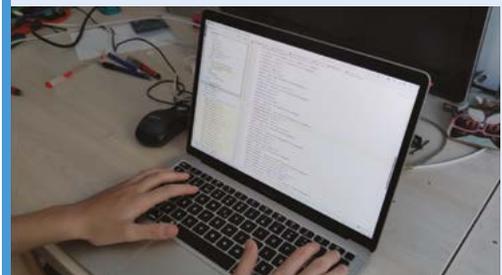
Build a bruising battlebot



01 Sketch out your design and make a cardboard prototype to help you work out the overall dimensions, how everything will be powered, and the connections needed.



02 Use CAD software to design the robot’s body, leaving space for Raspberry Pi Pico and any batteries or cables, then 3D-print or otherwise build the casing.



03 Use the web component of MIT App Inventor to create a means of controlling your battlebot. Set Raspberry Pi’s URL as the destination and add `:1880/mit`. An alternative is to use Node-RED and control Raspberry Pi directly.

Time Machine Radio

We've seen many a retro radio project, but this one is a personal dip into the past. **Nicola King** enjoys a melodic journey through time



George Edwards

George is a computer science graduate and a new father. He is genuinely surprised that any computer-based tech actually works!

magpi.cc/alephzero

When New South Wales resident George Edwards wanted to make a Christmas present for his dad, he had the perfect idea. As his father owned a beautiful 1938 Austin car that he often exhibited at classic car events, George thought that making him a vintage radio to go with it, that played the songs from bygone eras to add a little authenticity, would be a great gift.

"I had thought a little music box which played content from the 1930s and 1940s would be an excellent addition to the car when it's on display," explains George. "After putting some thought into making one myself, the idea of having content from each decade in the 20th century somehow seemed more naturally appropriate."

Radio refit

So, George set to work on a project that would take him around four months. For the hardware, he gutted a vintage-look NR-3013 radio bought from Amazon, salvaging a few components like

the speakers, one of the potentiometers, and the lithium battery.

A Raspberry Pi Zero 2 W equipped with a Pimoroni Audio Amp SHIM (with built-in DAC) provides analogue audio out to the speakers. "The finished project has two working potentiometer knobs, one for the volume and one for 'tuning' the decade," says George. "They're both connected to a MCP3008 [ADC] and then [Raspberry] Pi." An Adafruit PowerBoost 1000C acts as both a charger for the lithium battery and a power supply for Raspberry Pi.

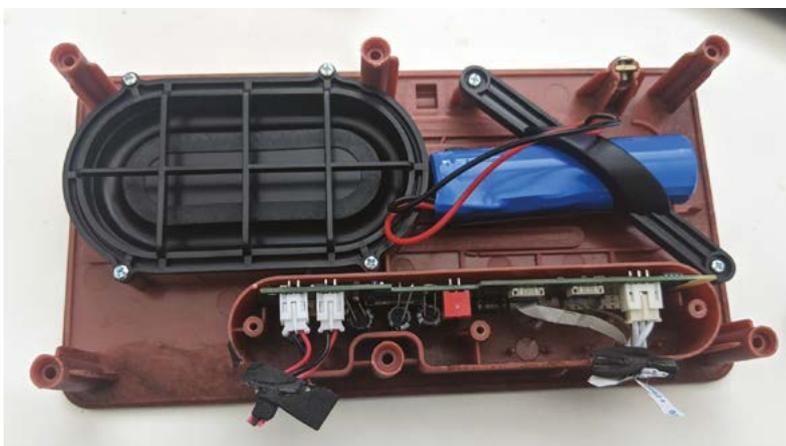
Regarding the software, Raspberry Pi boots up to a Python script which is infinitely looping. "At each loop it does three things: changes the channel if someone has 'tuned' to a different decade; plays the next song if it's appropriate to do so; and then sleeps for 50 milliseconds so we don't smash the CPU. Rinse and repeat ad infinitum."

To ensure that each tune gets a fair chance of being played, George has worked it so that once a song is selected, it is moved from an 'unheard' array to a 'heard' array, "effectively removing it from the pool of songs to be plucked from. Once all songs have been heard, they get moved back and the process starts again."

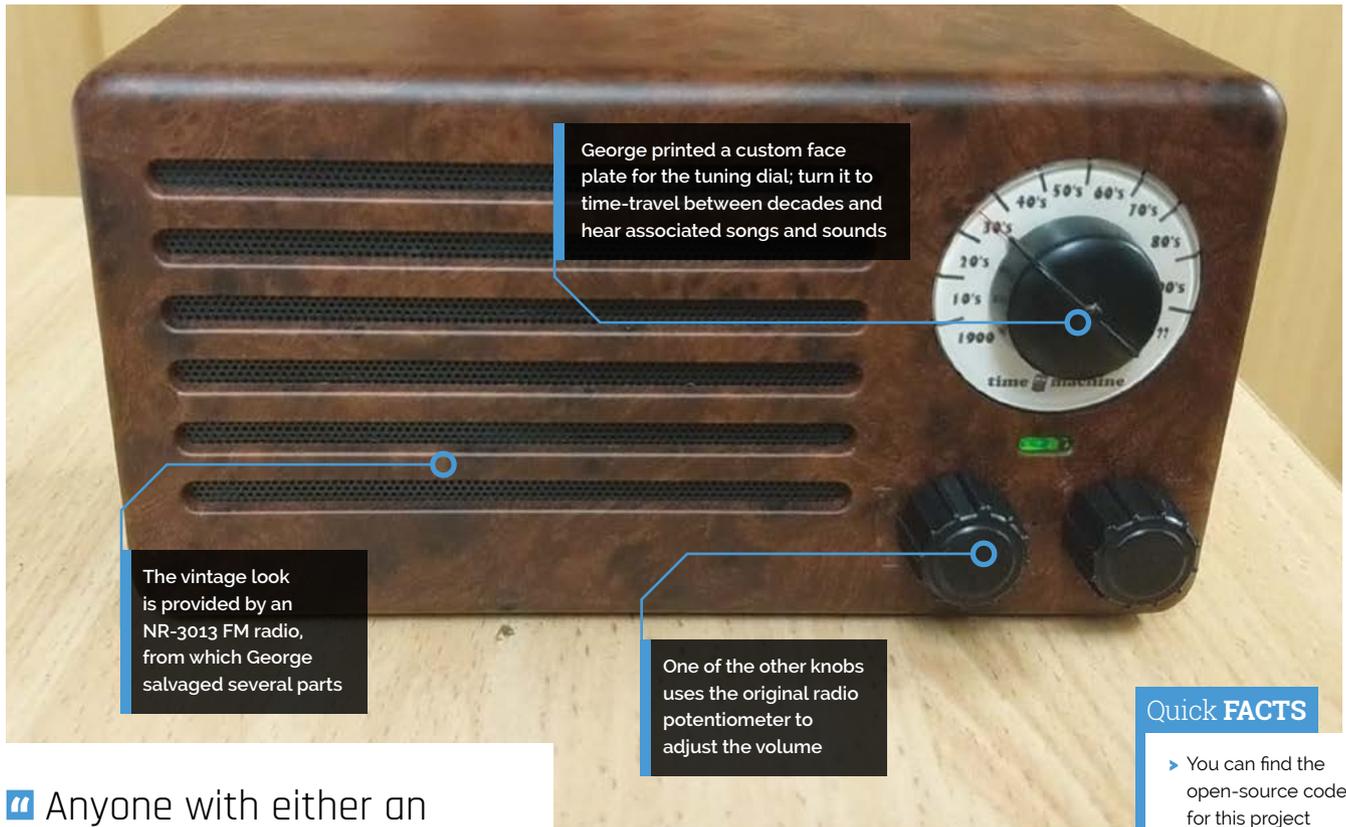
Doctor Who?

Whenever the dial is turned to move between decades, the sound of *Doctor Who*'s TARDIS is heard. "We used to watch *Doctor Who* together when I was a kid, and so using the TARDIS sound to 'move through time' to another decade seemed appropriate," says George.

Another lovely touch is the inclusion of a "secret decade" within the radio's setup, which contains some personal content that George's uncle had digitised from old tapes. "They're recordings of my late grandfather who died when my father was a young man and whom I've never met," he explains.



▲ George replaced the controller board (shown here) from the radio's back panel with a Raspberry Pi Zero 2 W, but kept the lithium battery and speaker



George printed a custom face plate for the tuning dial; turn it to time-travel between decades and hear associated songs and sounds

The vintage look is provided by an NR-3013 FM radio, from which George salvaged several parts

One of the other knobs uses the original radio potentiometer to adjust the volume

Quick FACTS

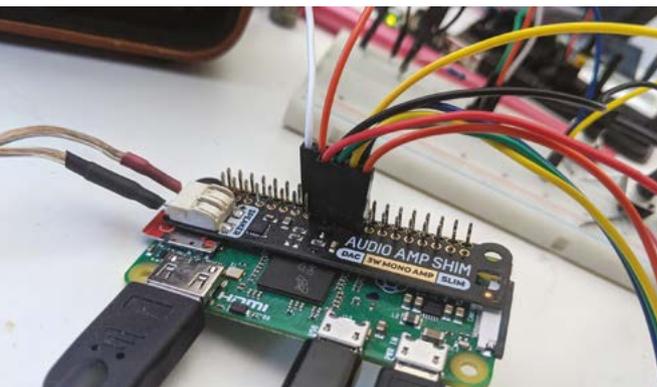
- ▶ You can find the open-source code for this project at magpi.cc/tardisradio
- ▶ The tunes come from Australia's National Film and Sound Archive
- ▶ His Dad loves his gift, listening in his garage as he tinkers
- ▶ George has since used a Raspberry Pi to also make a web-streaming baby monitor...
- ▶ ...and he's generously shared the code here: magpi.cc/babycamgh

“ Anyone with either an interest or rudimentary understanding of electrical engineering and/or microelectronics could make one ”

He did the same for another version of the radio that he made for his father-in-law, as he managed to include some audio of his late father too in that radio's hidden decade.

George thinks making a time-travelling retro radio is within the grasp of many. “Anyone with either an interest or rudimentary understanding of electrical engineering and/or microelectronics could make one,” he urges. “Provided you've got Googlefoo you should be right, and I would encourage all who are interested to give it a crack. Anyone reading this can get in touch at georgeedwardsmail@gmail.com, and I'll be happy to answer any questions.”

With that generous encouragement, have a go, personalise your decades as George has done, and enjoy a trip to yesteryear. [M](#)



▲ An Audio Amp SHIM is used to output analogue audio to the original radio speaker



- ◀ George's father owns a beautiful Austin classic car, so he thought the TARDIS Radio would complement it perfectly

M4All

Making microscopes accessible and affordable by open-sourcing it. **Rob Zwetsloot** takes a peek



Gemma Cairns

A NanoBioPhotonics PhD student at the University of Strathclyde, developing imaging technologies.

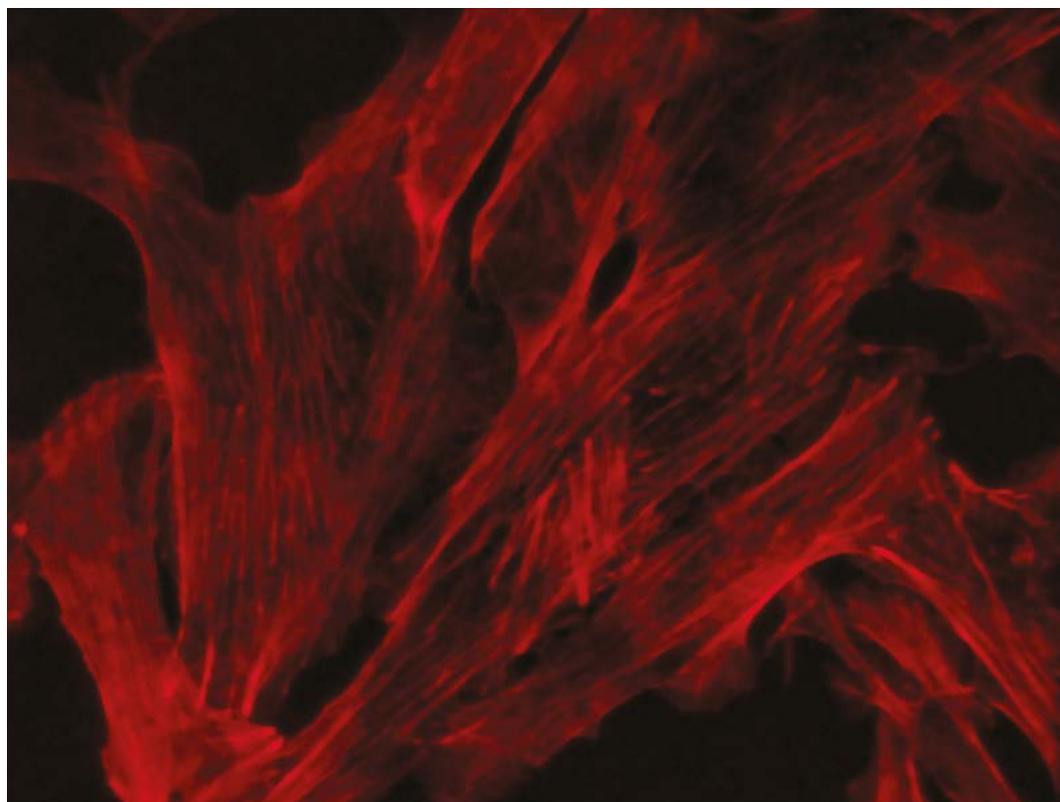
magpi.cc/m4all

It's not too hard to understand that Raspberry Pi cameras can be used for microscope builds – you just need the right lenses to focus the image on a small enough object. With a bit of custom code and software paired with it, you can then use it to view a tiny, hidden world.

“We have developed M4All, a fully open-source, 3D-printable, and modular microscopy system, which enables wider accessibility to advanced

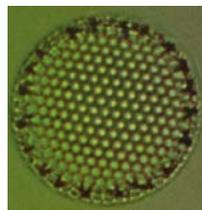
imaging techniques at a much lower price point than commercial microscopes,” Gemma Cairns, one of the students working on this project, tells us. “We integrate Raspberry Pi components into our M4All systems for microscope control, image capture, and data analysis.”

The team, the NanoBioPhotonics group at the University of Strathclyde, have been using their imaging expertise to “investigate the mechanisms



▲ This is actin, a protein that forms the cytoskeleton of cells, illuminated with fluorescence

▼ Another diatom shot, with another amazing pattern



that macrophages (a type of immune cell in the body) employ to remove respiratory pathogens”, pathogens that, among other things, can lead to pneumonia or meningitis.

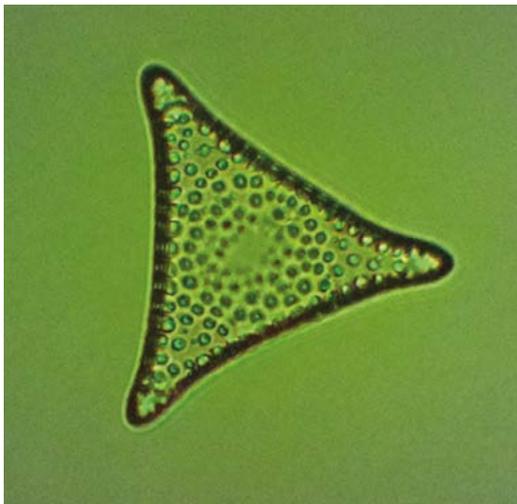
“From my experience using research microscope facilities to image macrophage samples, I realised there can be limitations with accessing commercial microscope systems, not only with costs but also time limits and flexibility,” Gemma explains. “A high-end research microscope can cost over £250,000 and so access to it can be limited and charged per-hour! In the past few years there has been a growing community of research groups working on open-source microscopy hardware and software solutions. Seeing their work inspired us to

“ We wanted to create a system that can be used by a wide level of expertise ”

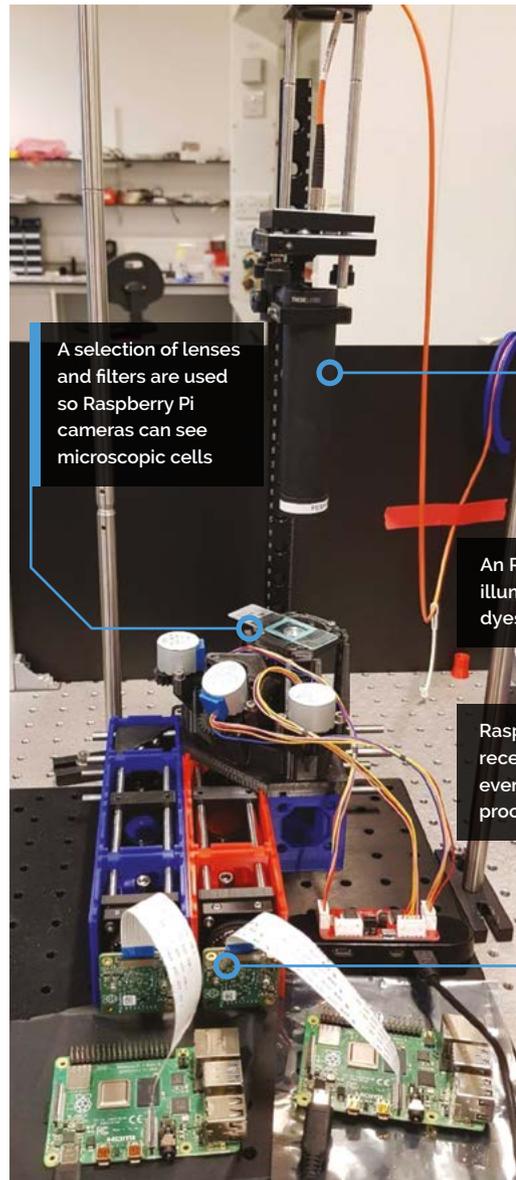
create M4All, which directly addresses our imaging requirements for macrophage research, but also contributes to the community where the system can be adapted for many other applications.”

Raspberry Pi microscope

The team opted for Raspberry Pi because they wanted something low-cost and small, and Raspberry Pi’s Camera Modules and HQ Cameras, along with its GPIO pins for controlling motors and



▲ This is a diatom, a single cell of algae which has silica cell walls which create an amazing pattern



A selection of lenses and filters are used so Raspberry Pi cameras can see microscopic cells

An RGB laser is used to illuminate the different dyes in the samples

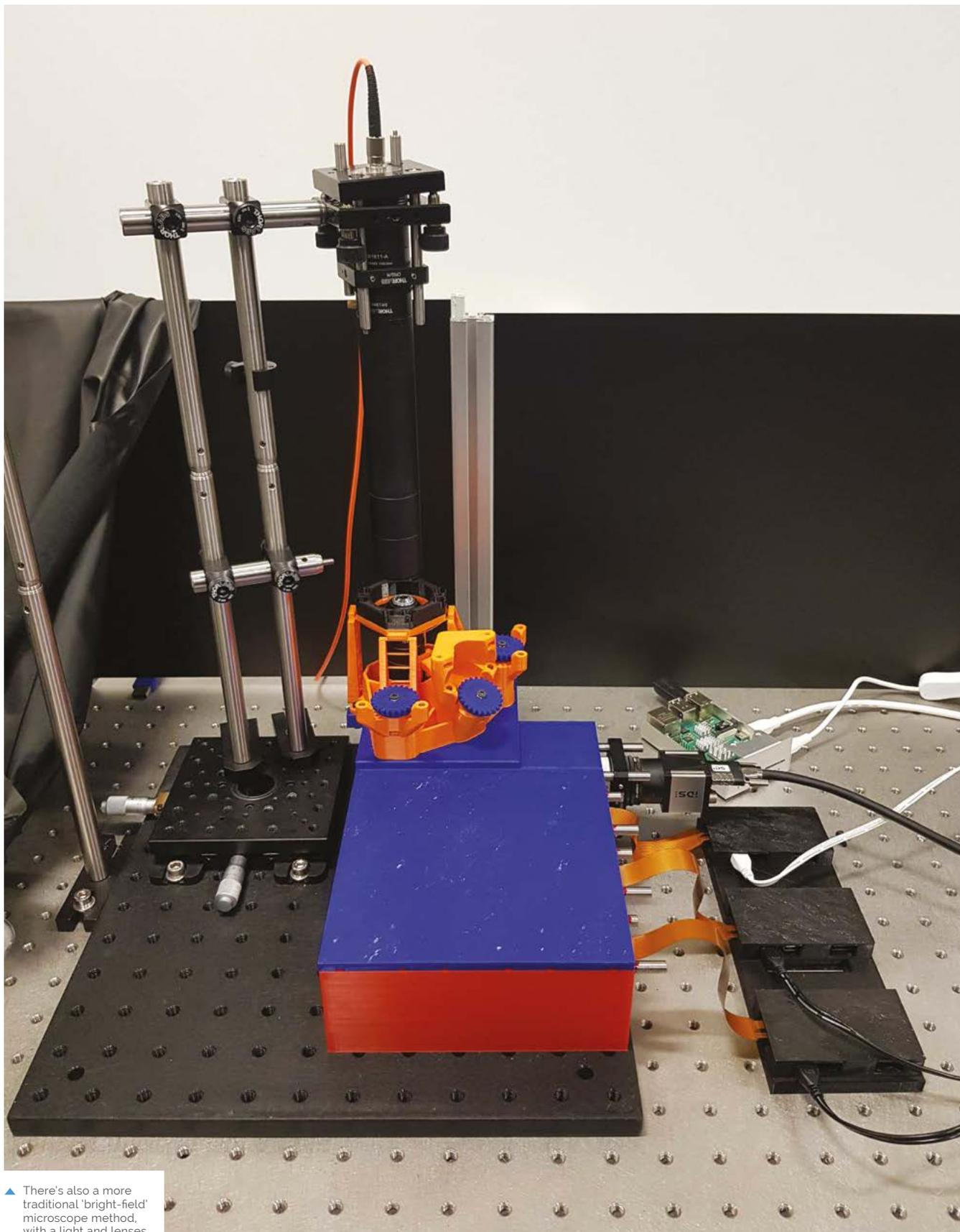
Raspberry Pi is able to receive the images and even do some post-processing on them

Quick FACTS

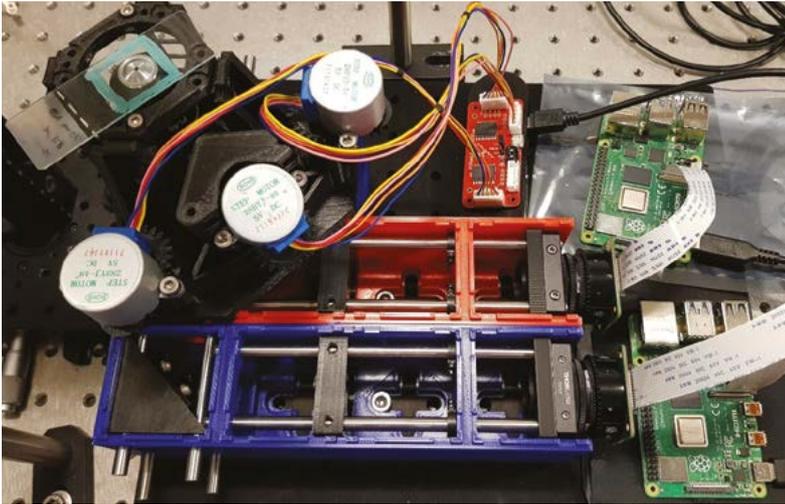
- ▶ M4All uses fluorescence microscopy that involves fluorescent dyes...
- ▶ ...which are illuminated by different light and then filtered
- ▶ The next stage is testing live cell imaging in a cell culture incubator
- ▶ OpenFlexure was created by the University of Bath
- ▶ The ultimate aim is to see real-time interactions with bacteria and other cells

LEDs, made it perfect as the heart of M4All. The next step was just making it all fit together.

“Developing the designs for the 3D-printed parts has been a very iterative process to achieve their final form,” Gemma says. “We wanted to create a system that can be used by a wide level of expertise, but also has the stability and capability for quantitative biological imaging (that is, images where the brightness of each pixel can be trusted and compared with nearby pixels or other regions of the image to perform numerical analysis). The modular cubes are printed as monolithic parts to form the optical path, and the inserts are designed to minimise the number of degrees of freedom for alignment. The only optical alignment that needs to be carried out is setting up and focusing the lenses for imaging.”



▲ There's also a more traditional 'bright-field' microscope method, with a light and lenses



As an open-source project, it also integrates other open-source bits into it, like OpenFlexure. OpenFlexure allows the image to be separated in a specific way and focused on the camera installed in Raspberry Pi.

Small goals

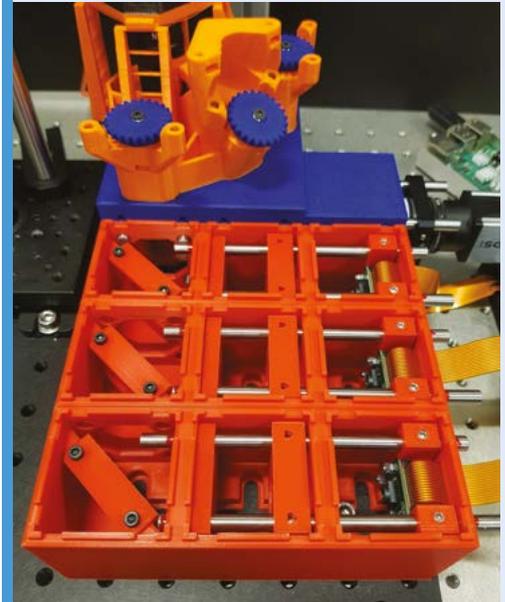
“We have been able to image various structures inside of cells using fluorescence microscopy, such as mitochondria, bacteria, and actin, with good resolution and with a signal-to-noise ratio which allows us to obtain quantitative information on what’s happening in the cell,” Gemma reveals. “This will allow us to study interactions with bacteria inside of macrophages at a much lower cost than commercial systems. Then, if interesting data arises, more specialised microscopy systems can be used for further analysis based on what the M4All microscopes have found. We have also imaged diatoms in bright-field which are single cells of algae. Their cell walls are made of silica, which forms very intricate and cool patterns.”

You can head to magpi.cc/m4all for instructions on how to build your own system, and start doing your own microbiology.

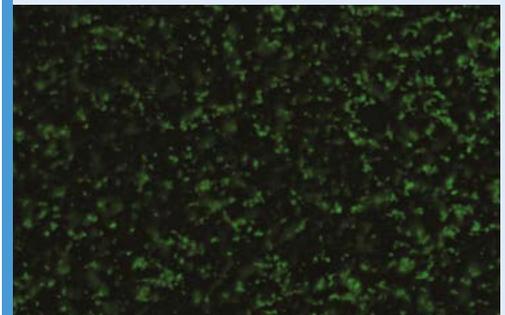
“Having a flexible platform also means that we can test out new designs and applications in collaboration with a huge range of researchers – from biochemists or marine environmental scientists!” 

▲ To get the right focus and filtering, there are a lot of motor-controlled lenses to manipulate

See the sights



01 You can choose between two different methods of microscopy – bright-field or fluorescence. You’ll need to prepare your setup slightly differently for each style, although both use a similar series of stepper motors and lenses.



02 For fluorescent microscopy, you’ll need to add fluorescent dye to the sample. This will react to different wavelengths of light.



03 You can then observe and analyse your results via Raspberry Pi as it captures and processes the image.

Retro barcode scanner

Forget plug-and-play! Neil Thomas and Chris Harris have created a scan-and-play system as part of a replica retro video game shop, as **David Crookes** explains



Neil Thomas

Neil is a retro tech YouTuber who has been running his channel as his full-time job since 2019. His first computer was an Amstrad CPC464!

magpi.cc/rmccave

For the past five years, Neil Thomas has indulged his passion for classic technology by presenting a popular YouTube channel called RMC – The Cave.

Tens of thousands of viewers watch him bringing the past back to life (literally, in some cases, by repairing old machines). But success ended up creating a small problem for Neil: what to do with all of the retro consoles and computers he's ended up collecting.

"It felt a shame to go to so much effort in restoring and repairing these classic machines only to hide them away in a cupboard," he says. "A pipe dream began to form that I might one day be able to open an exhibition space for retro fans to come and enjoy them and, through hard work

and incredible generosity from viewers, here we are, on the cusp of opening up to the public."

Hey MiSTer

Located on the top floor of an 18th century mill in Chalford in Gloucestershire, The Cave, as the space is called, includes lots of games consoles and computers from the past 40 years, as well as classic arcade machines. But one of the stand-out sections is a replica retro video game store where visitors can pick a game off a shelf, scan its barcode at a kiosk, and quickly start playing – a system that has a Raspberry Pi 3 computer at its heart.

"We'd just created new hardware called the MiSTer Multisystem which was powering the kiosk in the shop, but choosing a game to play on it required me to open the kiosk and manually select one, and that meant there would really only be one playable game all day," Neil recalls. "I wanted to make it more fun with a front-end menu that wouldn't confuse visitors, and then realised the room itself would make the perfect physical front-end. It didn't need to be a menu on the screen!"

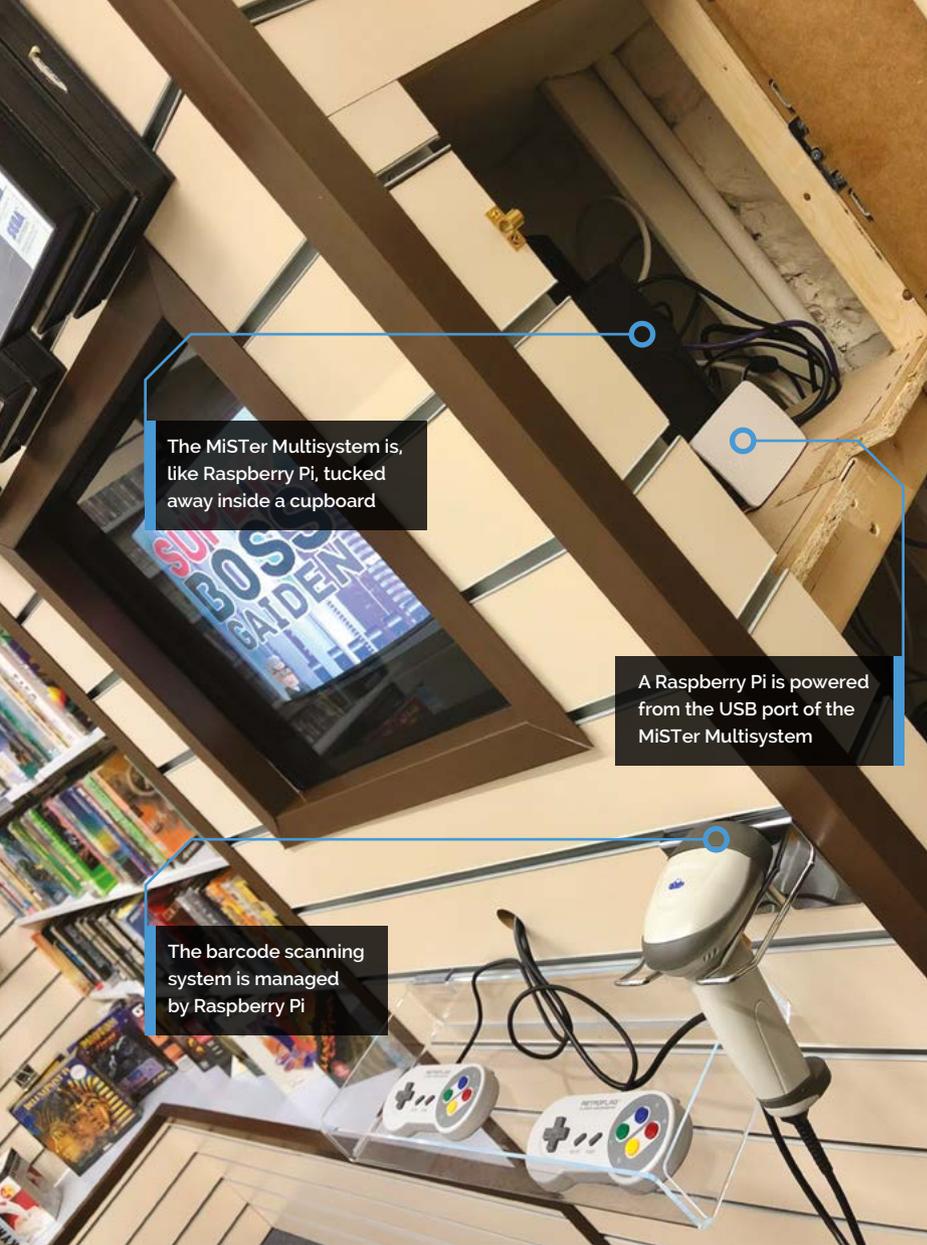
At first, Neil and his friend Chris Harris considered using a Raspberry Pi Camera Module to scan the barcodes. "But we found that environmental changes in light and mounting it behind the smoked perspex of the kiosk made it less than 100% reliable," Neil says. "We then switched to a handheld barcode scanner which not only works great but also fits the theme of a retro shop very well."

Space invader

Raspberry Pi runs a program called Barcode Rattler, created by Chris. It processes information from the scanner connected to Raspberry Pi via the USB port, and the information is sent via a secure network connection to the MiSTer gaming



▶ Games are loaded using a barcode reader powered by Raspberry Pi, called Barcode Rattler (magpi.cc/bcrattler)



The MiSTer Multisystem is, like Raspberry Pi, tucked away inside a cupboard

A Raspberry Pi is powered from the USB port of the MiSTer Multisystem

The barcode scanning system is managed by Raspberry Pi



▲ Visitors can just take a game from the shelf and scan at this kiosk which is inside a replica retro video game shop

Quick FACTS

- ▶ The games play on a CRT monitor
- ▶ One button turns the whole system on
- ▶ A database of barcode data was created
- ▶ The games play via a MiSTer Multisystem
- ▶ ROM sizes limit games to 8- and 16-bit titles

device – a highly accurate system well-loved by retro fans that emulates machines using an FPGA chip rather than software.

“Python waits for a keyboard event which it then reads and looks up a barcode in a CSV file,” Neil explains. “The barcode scanner behaves as a keyboard device so, when it scans a code, it sends

“ People seem to enjoy zapping the games and they love the ‘beep’ of the barcode gun ”

the string to Raspberry Pi as if you’d typed out the numbers on the keyboard. If it finds a matching barcode in the CSV, it will send SSH commands to the MiSTer to start the correct system core and load the game via a utility called MiSTer Batch Control.”

The result is a fun physical user interface that has gone down well with those who have tested it.

“People seem to enjoy zapping the games and they love the ‘beep’ of the barcode gun,” Neil says. It’s also become his favourite room. “It’s a time-warp where you can forget the real world for a day and be swept away in a wave of carefree nostalgia,” he adds. “What could be more nostalgic than revisiting the video game shop from your childhood?” [M](#)



◀ The MiSTer Multisystem was created by Neil in collaboration with the electronics firm Heber, and allows for accurate hardware-based emulation

CASE STUDY

Elevated Materials

Fine process control for space-grade waste recycling. By **Rosie Hattersley**

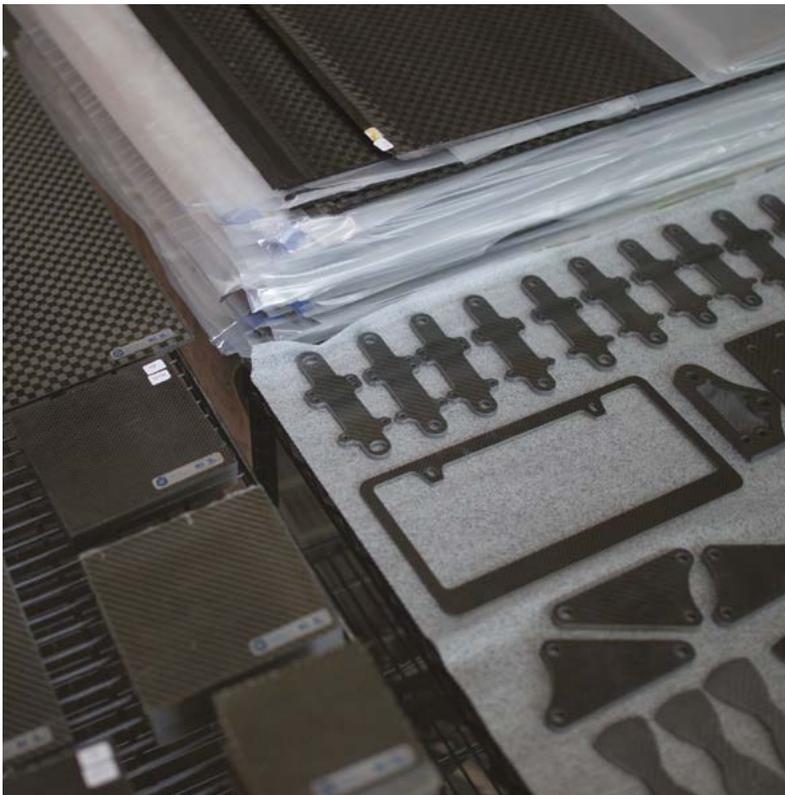
The idea that grew into Elevated Materials (elevatedmaterials.com) came about when founder Ryan Olliges – then seconded to the Rocket Propulsion Lab at the University of Southern California – was tasked with sourcing

carbon fibre offcuts that he and fellow students could use for their designs. He soon found he had more waste material from the space and aeronautical industries than the lab could use, and his first business used a heat press to repurpose the material to produce extremely robust, streamlined, lightweight skateboards.

The project was a commercial success and, having proven the possibilities of secondary-use carbon fibre, Olliges and colleagues set up Elevated Materials to address in earnest the need for a responsible alternative to landfill for this industrial by-product. They ramped up their recycling efforts, bringing in the technology needed for precise control of the finely tuned heat treatment that is required to repurpose each consignment.

THE CHALLENGE

Most of Elevated Materials' commissions involve waste produced during the manufacture of rockets and spacecraft, high-tech vehicles that really need the strength-to-weight ratio that carbon fibre offers, explains Jaysen Harris, the company's co-founder and Tech Development VP. These waste materials come from the manufacturer with bonding resin already introduced into the carbon fibre weave. Bonding together in multiple layers and orientations is what gives carbon fibre products their strength, but the composition of each product that Elevated Materials receives from each client is different. Consequently, each consignment needs a different heat treatment to process it for reuse.



Importantly, the design patterns embodied in the waste carbon fibre is destroyed in treatment. This is critical to preventing its composition from being reverse-engineered: a key part of the service Elevated Materials provides is to make sure nothing persists that could potentially let a client's competitor know the temperatures, resin amounts, and composition used.

Sophisticated temperature control and detailed logging are essential.

“ Raspberry Pi provides detailed report logging so processes can be analysed and replicated ”

THE SOLUTION

Elevated Materials uses a heat press and technology developed in-house together with Raspberry Pi 3B+ as the process-controlling computer to bring in multiple temperature and pressure values, ensuring they are adhering to the manufacturers' specifications of the material, with exactly how much heat to apply and how quickly to increase it.

Raspberry Pi provides detailed report logging so processes can be analysed and replicated. Later in the recycling process, some of the treated and pressed material is precision-cut into smaller sizes and custom parts with a Raspberry Pi-controlled file server and connected CNC router.

WHY RASPBERRY PI?

Tech Development VP Jaysen Harris credits the precision afforded by Raspberry Pi as a defining feature of Elevated Materials' offering. “Raspberry Pi is constantly taking values and evaluating where it's supposed to be, making any changes and turning on and off the power.”

Other available solutions were inferior: “There are products out there that do temperature control, but we would not have gotten such an easy-to-use, thorough logging capability.”



Alternatives would also have been more expensive and, given the level of sophistication required, “it would have been inaccessible at that level of startup,” he says. “We just wouldn't have been able to do it.”

THE RESULTS

Elevated Materials is the only company in southern California offering a carbon fibre upcycling service; the minute control offered by the Raspberry Pi controller at the heart of each of its heat presses allows it to offer more services than competitors as well as greater precision – assets that win them more customers.

Innovation, key to the company's success so far, is something Raspberry Pi's broad product range continues to support. Harris is investigating further use of Raspberry Pi computers, particularly the Compute Module line, which he envisages as the brain of their heat presses. Meanwhile, Raspberry Pi 4 offers upgrade possibilities for CNC routers, and Raspberry Pi Camera Modules could provide valuable monitoring. Raspberry Pi digital signage is used to explain the process to visitors. “We are constantly iterating,” Harris says.

To date, Elevated Materials has kept more than 200,000 lb (approx 90,000 kg) of carbon fibre out of landfill. [M](#)

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%



Subscriber Benefits

- ▶ **FREE Delivery**
Get it fast and for FREE
- ▶ **Exclusive Offers**
Great gifts, offers, and discounts
- ▶ **Great Savings**
Save up to 35% compared to stores

Rolling Monthly Subscription

- ▶ Low monthly cost (from £5)
- ▶ Cancel at any time
- ▶ Free delivery to your door
- ▶ Available worldwide

Subscribe for 12 Months

£55 (UK) £90 (USA)
£80 (EU) £90 (Rest of World)

Free Raspberry Pi Zero 2 W with 12 Month upfront subscription only (no Raspberry Pi Zero 2 W with Rolling Monthly Subscription)

📞 Subscribe by phone: **01293 312193**

📧 Subscribe online: **magpi.cc/subscribe**

✉ Email: **magpi@subscriptionhelpline.co.uk**

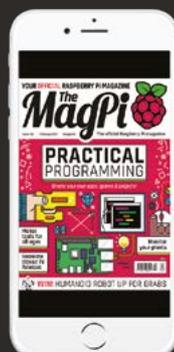
JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero 2 W

WITH YOUR FIRST
12-MONTH SUBSCRIPTIONSubscribe in print
today and get a
FREE computer!WORTH
\$15

- ▶ A full Raspberry Pi desktop computer
- ▶ Learn to code and build your own projects
- ▶ Make your own retro games console, media player, magic mirror and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.



 Buy now: magpi.cc/subscribe

SUBSCRIBE
on app stores

From **£2.29**

 Available on the
App Store

GET IT ON
 **Google Play**

HIDDEN HACKS FOR RASPBERRY PI

Get more from your microcomputer with these clever tricks By **Lucy Hattersley**

Raspberry Pi is an incredibly versatile computer that delivers much more power and functionality than its small stature suggests. Now, with ten years of solid development behind it, Raspberry Pi computers and Raspberry Pi OS are packed with activity.

Hidden in amongst all the hardware and software are some real gems of tricks that you can use to learn more about computing, make better projects, and get far more out of Raspberry Pi than you ever thought possible.

In this feature, we're going to unearth hidden hacks and handy tricks that every talented Raspberry Pi owner should know.



RASPBERRY PI OS TRICKS

Get under the hood of Raspberry Pi OS

TIP!

```
pi@raspberrypi400:~$ sudo adduser lucy
Adding user 'lucy' ...
Adding new group 'lucy' (1001) ...
Adding new user 'lucy' (1001) with group 'lucy' ...
Creating home directory '/home/lucy' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for lucy
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
pi@raspberrypi400:~$ users
pi pi
pi@raspberrypi400:~$
```

Set up a new user

As a security measure, you should set up a custom user name, rather than using the default 'pi' user that has been used in Raspberry Pi OS (and previously Raspbian) since the beginning.

Get on board with improved security and create your own new user account.

Create new users in Raspberry Pi OS with `adduser` and the desired username (in this example, 'izzy'):

```
sudo adduser izzy
```

You will be prompted for a password for the new user, izzy. Enter a password for security reasons. Terminal will ask for the Full Name, Room Number, Work Phone, Home Phone, and Other information. Press **RETURN** to keep these blank and enter 'y' to 'Is the information correct?'

When you create a new user, they will have a home folder in `/home/`. Upon creating a new user, the contents of `/etc/skel/` will be copied to the new user's home folder. You can add or modify dot-files such as the `.bashrc` in `/etc/skel/` to your requirements, and this version will be applied to new users.

Overlay File System

Overlay File System keeps Raspberry Pi OS in a fresh state. With Overlay File System enabled, any changes you make are saved to RAM instead of storage, and are lost when you reboot.

This is useful for using Raspberry Pi in a public setting where you want people to be able to use the operating system without making any permanent changes to it. Such as with a Raspberry Pi set up in a store.

Open Raspberry Pi Configuration and choose the Performance tab. Click Configure next to Overlay File System and enable User Overlay.

The write-protect boot partition enables you to lock down the boot partition for even more security.

Take a look at the kernel

Raspberry Pi kernel is stored in GitHub (github.com/raspberrypi/linux). It follows behind the main Linux kernel. Raspberry Pi takes long-term releases of the kernel, which is mentioned on GitHub, and integrates the changes into Raspberry Pi kernel.

There is also a 'next' branch that contains an unstable port of the kernel. After extensive testing and discussion, the next branch becomes the main branch.

Switch users

The `su` command (related to `sudo`, which we'll look at later) is used to 'switch user'.

```
su izzy
```

...will switch between your current user and izzy. Use `whoami` to see which user you're currently using.

```
whoami
```

...will return 'izzy'. You'll be in the `pi` folder still so use `cd` to switch to izzy's folder.

```
cd /home/izzy
```

Use `ls` and you'll see the files created by `/etc/skel`.

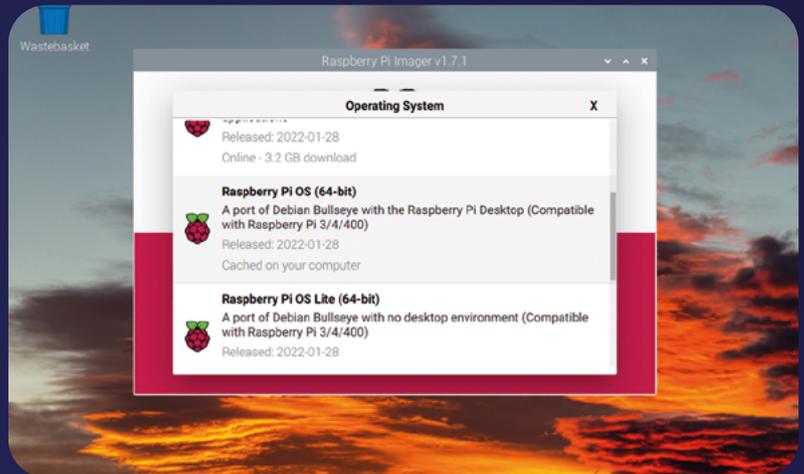
Sudoers list

The default user on Raspberry Pi OS is a member of the sudo group, which enables it to perform elevated actions (as root) when you use the `sudo` command.

To add your new user to the sudo group, use:

```
sudo adduser izzy sudo
```

Now they'll also be able to use `sudo`.



Move to 64-bit

If you've been running Raspberry Pi OS for a while, the chances are you're using the 32-bit version. It's time to double-up to the latest, 64-bit edition. This came out of early beta testing last month, and can now be found inside Raspberry Pi Imager (magpi.cc/imager). If you have a Zero 2 W, Raspberry Pi 3, Raspberry Pi 4, or Raspberry Pi 400, then make the switch to Raspberry Pi OS (64-bit). Read Gordon Hollingworth's blog for a detailed write-up (magpi.cc/64bit).

TIP!

Check out sources

APT keeps a list of software sources on Raspberry Pi in a file at `/etc/apt/sources.list`.

"Chances are you're using the 32-bit version of Raspberry Pi OS"

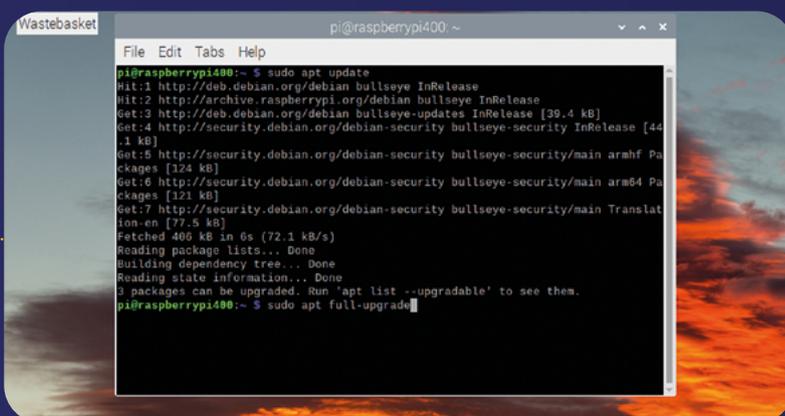
Updating Raspberry Pi

It's best to update Raspberry Pi OS using the built-in Software Updater tool. You should do this frequently. It's important to keep Raspberry Pi OS up-to-date for security purposes, and also to ensure you are running the latest version, which will contain fewer bugs.

It is also possible to update Raspberry Pi OS using APT (Advanced Packaging Tool) in Terminal, but make sure to do a full upgrade (rather than just a regular upgrade).

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

Using `full-upgrade` (over just regular `upgrade`) ensures that APT picks up any dependency changes that have been made.



TAKE CONTROL OF YOUR RASPBERRY PI

Take control of your Raspberry Pi
from across a network

Copy files with SCP

After you have connected to Raspberry via SSH (or VNC), you'll soon wonder how to copy files to and from a remote computer on the network. There are a few different ways to copy files, and we generally prefer to use SCP (secure copy).

While in Terminal on your source computer, you use the `scp` command with the SSH login details and file location (separated by a colon). It looks like this:

```
scp <user@IP>:path/to/file path/to/destination
```

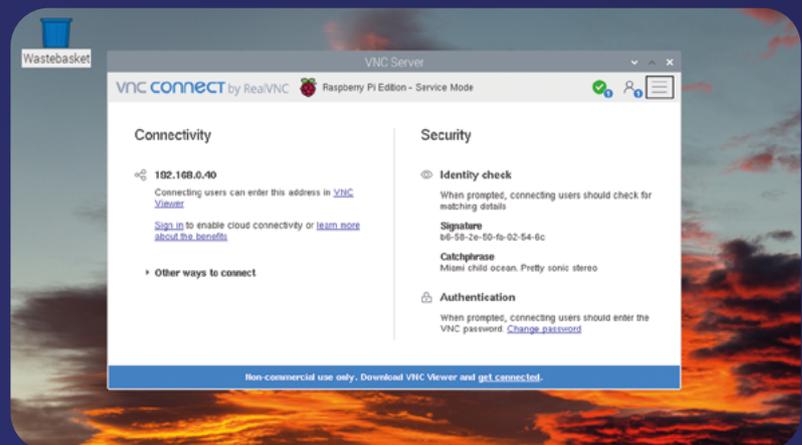
Let's say we want to copy a file called `test.txt` from the Desktop on our Raspberry Pi to the Desktop on our remote computer; we would enter:

```
scp pi@192.168.0.40:/home/lucy/Desktop/test.txt /home/lucy/
```

We find it less wordy to navigate to the destination folder on our computer, and then use a dot `'.'` as the destination to copy the file to the current location

```
cd ~/Desktop
scp pi@192.168.0.40:/home/lucy/Desktop/test.txt .
```

If you copy a lot of files, you can install FTP (File Transfer Protocol software) like FileZilla (filezilla-project.org) and use it to transfer files across computers on your network.



Remote control: VNC

VNC enables you to remotely access Raspberry Pi from another computer. VNC is a graphical desktop system that lets you view the desktop of Raspberry Pi OS from a window in another computer.

To do this, you'll need to activate server software in Raspberry Pi OS and install viewer software on your other computer (this can be a Mac, PC, or Linux computer, or another Raspberry Pi).

Raspberry Pi OS comes with VNC Server built in. All you need to do is turn it on. Open Raspberry Pi Configuration (menu > Preferences > Raspberry Pi Configuration) and choose Interfaces. Set VNC to On and click OK.

The VNC Server icon will appear in the menu bar. Click it to open the VNC Server window, and you will see the IP address underneath.

You will most likely need to install VNC Viewer on the other computer (magpi.cc/vncviewer). Once installed, open the IP address of your Raspberry Pi into the text bar of VNC Viewer and press **RETURN**. Enter the Username and Password for Raspberry Pi OS and click OK.

You will now be able to remotely access Raspberry Pi OS desktop in a window on your remote computer. Take a look at magpi.cc/vnc for more information.



Set Headless Resolution

When accessing Raspberry Pi OS remotely, without a monitor attached, you will find the desktop defaults to a tiny 640×480 resolution. There is an easy way to fix this. Open Raspberry Pi Configuration and click Display. Set Headless Resolution to a higher setting (it will need to be compatible with your monitor). Click OK and you will be prompted to restart your Raspberry Pi. When it reboots, you'll be using the new resolution.

“Your Raspberry Pi computer’s hostname is well worth getting to know”

Set a custom hostname

Your Raspberry Pi computer’s hostname is well worth getting to know. The hostname is a label that can be used to identify and locate your Raspberry Pi on a network. By default, all Raspberry Pi computers have the hostname ‘raspberrypi’. You can connect to a Raspberry Pi using the hostname and a ‘.local’ suffix instead of the IP address.

```
ssh pi@raspberrypi.local
```

If you have more than one Raspberry Pi, it’s a good idea to start renaming them. Open the Raspberry Pi Configuration tool under Raspberry Pi menu > Preferences and choose the System tab. Change the text field in ‘Hostname’ to your custom name and click OK. You will need to restart Raspberry Pi OS for the change to take effect.

Remote control: SSH

SSH (Secure Shell) is the simplest way to access a Raspberry Pi on your network. The downside is that it works via the Terminal and is command-line based. You need to activate SSH on your Raspberry Pi first. Launch Raspberry Pi Configuration from the Preferences menu and choose Interfaces. Set the radio button next to SSH to Enabled and click OK.

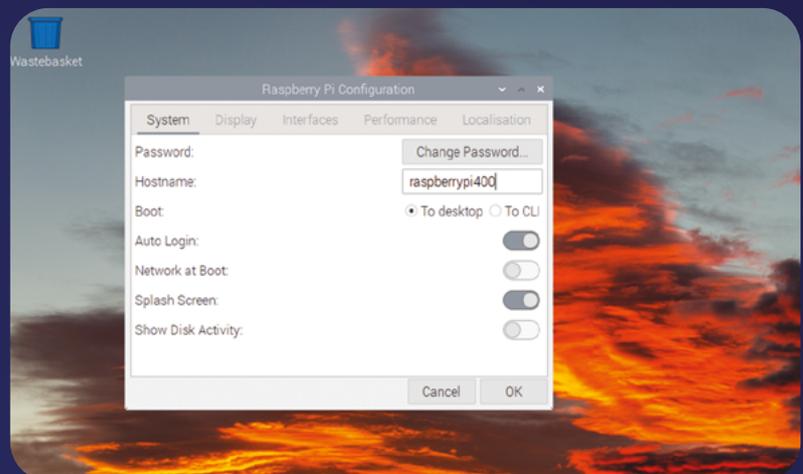
You’ll need the IP (Internet Protocol) address for your Raspberry Pi, which you can see by hovering your mouse over the network icon in the menu bar. It will be a sequence of four numbers separated by dots. Ours is: ‘192.168.0.40’ and the username of Raspberry Pi, which on our computer is ‘pi’.

Open a Terminal app on a macOS or Linux system, or PowerShell on a recent edition of Windows. Many Windows users use a Terminal app called PuTTY (putty.org).

To connect, you need to enter `ssh` `<user>@<ip address>`. For example:

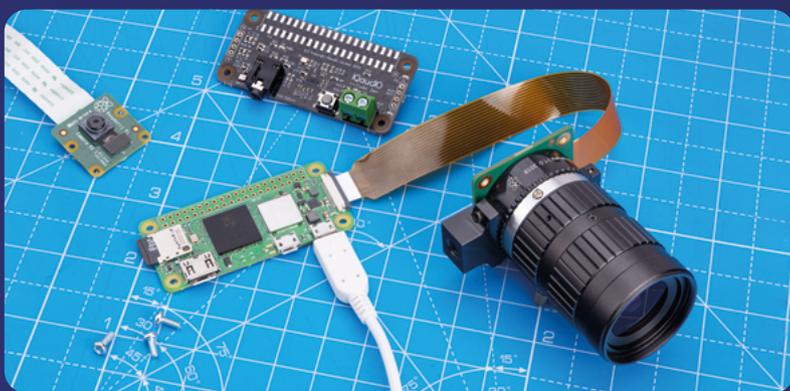
```
ssh pi@192.168.0.40
```

You will be prompted for that username’s password of the remote computer (the Raspberry Pi you are connecting to). Enter it and respond with ‘yes’ to the security prompt. You will only see this prompt the first time you attempt to log in. You will now be signed in to your Raspberry Pi.



GET HEADLESS WITH RASPBERRY PI OS

Desktop? Where we're going we don't need desktops



When you first set up Raspberry Pi, you generally connect a screen, keyboard, and mouse and navigate Raspberry Pi OS using the graphical user interface.

This isn't the only way to use Raspberry Pi though, and you can install Raspberry Pi OS Lite (64-bit). This is a port of Raspberry Pi OS with no desktop environment – everything takes place in the Terminal.

There are a lot of advantages to running Raspberry Pi OS in headless mode. Especially if your project is going to be used remotely, without a screen attached (say, as a weather monitor or security camera). Once these projects are set up, you'll only access them remotely via SSH, so it makes sense to set them up as light as possible. The download is only 0.4GB, so it fits on smaller microSD cards.

Thanks to a recent update in Imager, it's much easier to add custom settings to Raspberry Pi OS during the installation procedure. You can now add custom hostname and wireless LAN settings, as well as activate SSH for remote access. This makes it possible to fully set up a headless Raspberry Pi without ever having to attach a keyboard and monitor.

Use Imager to set up a headless Raspberry Pi

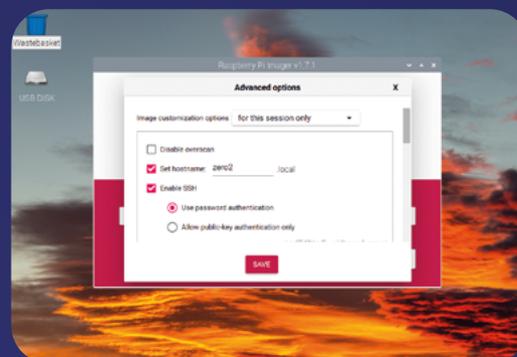
01 Install Imager

Imager is an app for Windows, macOS, and Linux that makes it easy to set up Raspberry Pi OS and other operating systems. View installation instructions at magpi.cc/imager. Open Imager and attach a microSD card to your computer.

Click on Choose OS and select Raspberry Pi OS (other) and Raspberry Pi OS Lite (64-bit) if you are using a Raspberry Pi 3, 4, 400 or Zero 2 W; or 32-bit if you are using an earlier computer.

02 Set the hostname

Choose Storage and select your microSD card (we're using a 4GB microSD card as we don't need the full 8GB required by the full desktop OS). Before starting the process, click on the Advanced Options cog in the bottom-right. Here you can set up a range of options for the install, which will be very helpful when running headless mode. Start with ticking 'Set hostname', and change it to something unique on your network; we're changing from the default 'raspberrypi' to 'zero2'.



03 Connect to the network

Check the 'Set username and password' and enter a password. Now check 'Configure wifi' and enter your SSID (wireless network name) and wireless LAN password. Finally, set the locale settings to your location. Click Save when you're done.

Click Write and Yes to record the Raspberry Pi OS Lite operating system to the microSD card. And Yes again to the warning and enter your password if required.

04 Connect via SSH

Wait till it's finished and transfer the microSD card to your Raspberry Pi. Now boot up and Raspberry Pi will connect to your local wireless network. And you can connect to it via SSH from Terminal on another computer. Wait for Raspberry Pi to boot and use the custom hostname (that you set up in Imager) to connect to it. You can locate the IP address using:

```
ping zero2.local
```

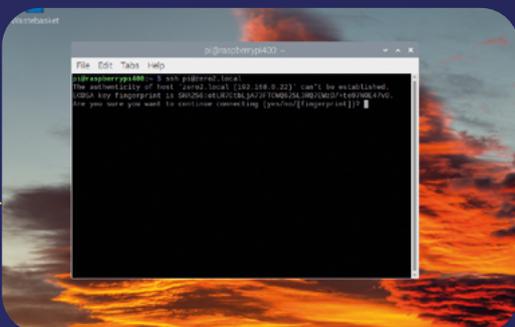
And connect to the computer using SSH and the hostname:

```
ssh pi@zero2.local
```

Or via the IP address (substituting the numbers for Raspberry Pi on your network):

```
ssh pi@192.168.0.22
```

From here on in, you can access your Raspberry Pi via SSH on another computer. This setup is ideal for Raspberry Pi projects that don't use a keyboard and screen, and which embed Raspberry Pi in a mobile or hard-to-reach location, such as a robot or security camera.



“There are a lot of advantages to running Raspberry Pi OS in headless mode”

Decode the LEDs

Raspberry Pi boards have a red power light and a green status light. If Raspberry Pi fails to boot or has to shut down, the LED will flash a specific number of times to represent the situation. Use this table to decode the LEDs.

Long flashes	Short flashes	Status
0	3	Generic failure to boot
0	4	start*.elf not found
0	7	Kernel image not found
0	8	SDRAM failure
0	9	Insufficient SDRAM
0	10	In HALT state
2	1	Partition not FAT
2	2	Failed to read from partition
2	3	Extended partition not FAT
2	4	File signature/hash mismatch – Pi 4
3	1	SPI EEPROM error – Pi 4
3	2	SPI EEPROM is write protected – Pi 4
3	3	I2C error – Pi 4
4	4	Unsupported board type
4	5	Fatal firmware error
4	6	Power failure type A
4	7	Power failure type B

NEXT-LEVEL TERMINAL

The joy of text with Raspberry Pi's
command-line interface

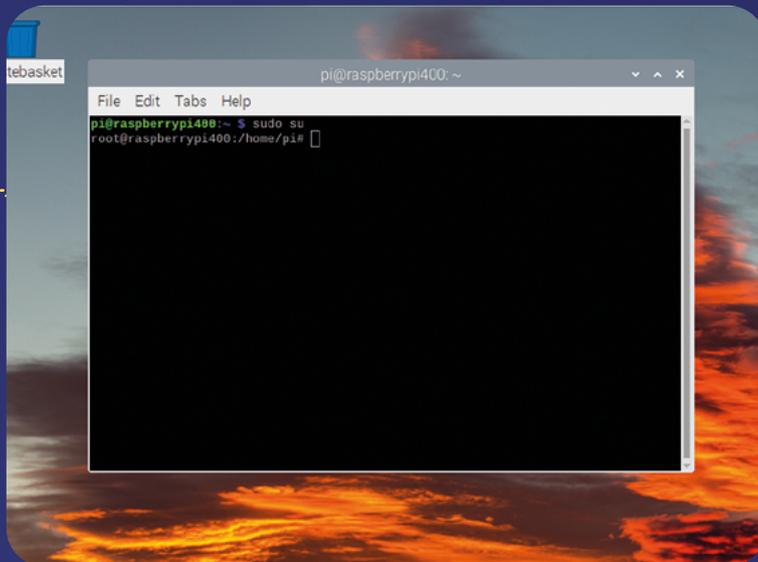
Switch to root

An elevated `su` command on its own switches to the root user. Enter:

```
sudo su
```

...to run a shell as the root user. Every command you enter will be as a superuser, so you do not need to use `sudo` or enter a password. This can be helpful with complex installations that require multiple sudo privileges. It's a bad habit though, and you should only use `sudo su` when required and exit from the root shell as soon as possible (enter `exit`).

You can learn more about sudo in Raspberry Pi's documentation (magpi.cc/sudo).



Config.txt

The `config.txt` file is read by the GPU before the ARM CPU and Raspberry Pi OS Linux is initialised. It is located on the first (boot) partition of the microSD card alongside the `bootcode.bin` and `start.elf` files.

It is possible to view the values of `config.txt` without opening it, using `vcgencmd get_config <config>`. For example:

```
vcgencmd get_config arm_freq
```

...will return the CPU frequency set in `config.txt`. You can also use `int` and `str` to see all values stored as integers and strings.

```
vcgencmd get_config int
vcgencmd get_config str
```

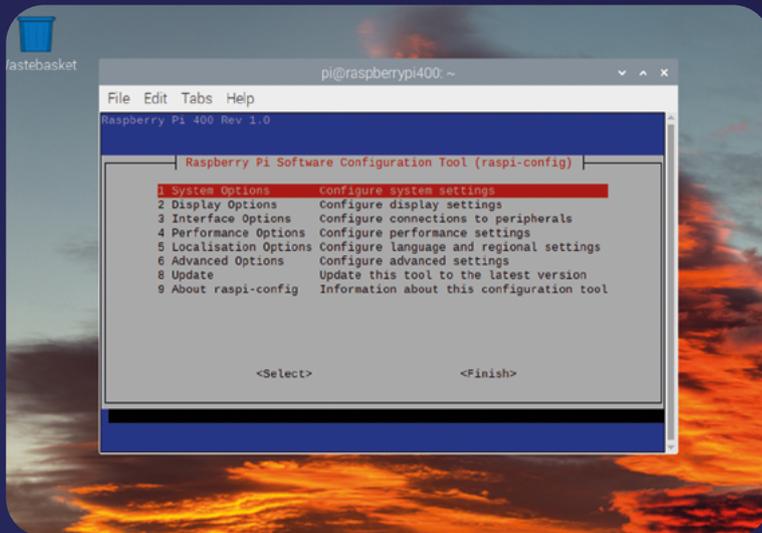
You can edit `config.txt` using:

```
sudo nano /boot/config.txt
```

It is also one of the files you can edit manually by removing the microSD card from Raspberry Pi and attaching it to another computer (macOS, Windows, or Linux).

Any changes you make to `config.txt` will only take place after you reboot Raspberry Pi.

Take a look at the official Raspberry Pi documentation (magpi.cc/configtxt) to discover more about the settings in `config.txt`.



Go back in time

If you want to go the other way, you can install earlier shells.

```
sudo apt install tcsh
sudo apt install csh
```

Then open into them with `csh` or `tcsh`. They are a lot more limited compared with the default Bash shell, and `csh` (C Shell) doesn't have tab autocomplete or history. Use `exit` to back out and back to your default shell.

Raspi-config is your friend

When Raspberry Pi first came out, `raspi-config` was used widely to adjust settings without diving into the underlying configuration files. This handy tool can be accessed from Terminal with:

```
sudo raspi-config
```

These days it's easier to use Raspberry Pi Configuration (under menu > Preferences) when in the desktop interface. But if you're using Terminal, headless, or accessing via SSH knowing `raspi-config` can be a game-changer. See magpi.cc/raspiconfig for more info.

Behind Raspberry Pi Configuration and `raspi-config` is a text file called `config.txt`. This sets many settings for Raspberry Pi OS.

“Change your default shell from Bash to Zsh”

Change default shell

You can change your default shell from Bash to Zsh (or earlier shells) using:

```
chsh -s $(which zsh)
```

You will be asked to enter your admin password. And changes only take effect on restart:

```
sudo reboot
```

Open a Terminal window and you will be greeted with Z Shell.

To get back to Bash, enter:

```
chsh -s $(which bash)
```

...and reboot.

Tab completion

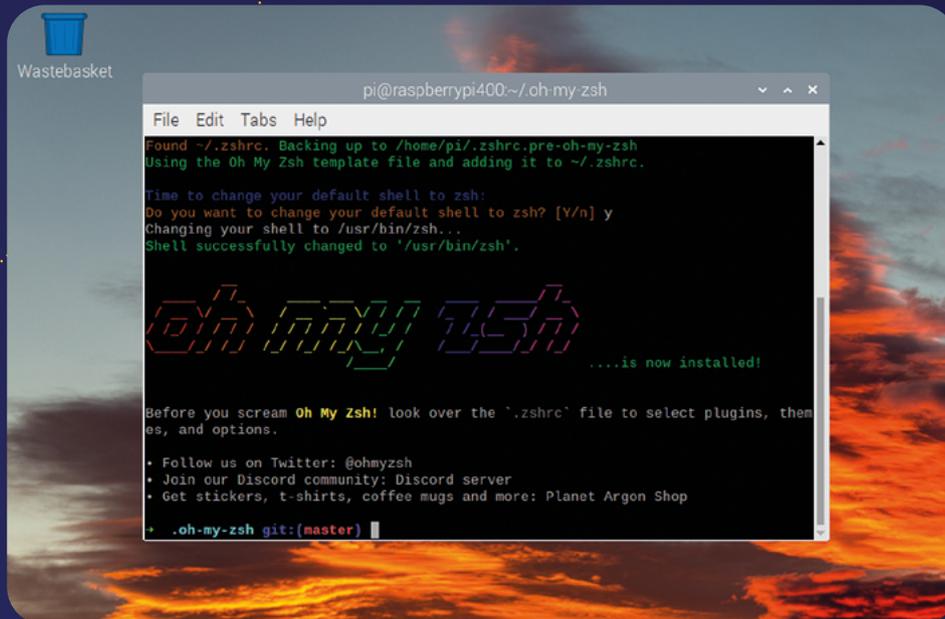
It's a basic hack but if you don't know it, tab completion is a game-changer. When using the Terminal, simply start typing a command or directory location and press the **TAB** key to complete it. For example, to navigate to the **Documents** folder, open a Terminal window and enter:

```
cd Doc
```

...and press **TAB**. Hey presto, it automatically fills up the rest. If there's more than one location, it'll show you the options and let you carry on entering letters. Try:

```
cd D
```

...And press **TAB** and you'll see 'Desktop/ Documents/ Downloads'. Carry on entering letters until you can autofill the command. You can use **TAB** multiple times in each command to fill out commands, files name, directory paths, and much more.



Experiment with Zsh

Terminal uses Bash (Bourne Again Shell) by default, which provides many of the advanced features such as tab completion and brace expansion. However, it's not the final word in shell technology and you can switch to the more recent Zsh (Z Shell). This is an expanded Bourne shell. ZSH has a bunch of modern features.

Oh My Zsh

Oh My Zsh (**ohmyz.sh**) is an extension to ZSH which enables you to take things even further, with a large number of alias shortcuts, themes, and plug-ins that enable you to add a huge range of functionality to the shell – everything from integrated web search to advanced GitHub functionality.

Install Oh My Zsh using a script (also available on the website):

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Take a look at the documentation if you want more information on Oh My Zsh (magpi.cc/omzwiki), and bookmark the cheatsheet (magpi.cc/omzcheatsheet). There are some great features in Oh My Zsh: you can use **1**, **2**, **3**, and so on to return to preview directories. Also, be sure to make use of Oh My Zsh's extensive plug-ins (magpi.cc/zshplugins).

These feature everything from web search integration to *Hitchhiker's Guide to the Galaxy* quotes.

```
sudo apt install zsh
```

Then start Zsh with:

```
zsh
```

“Zsh has a bunch of modern features”

The first time you run Zsh, you will be offered Z Shell configuration options. This sets up the `~/.zshrc` file (the configuration file). For the first time, we recommend using option ‘2’, which populates the

`.zshrc` file with recommended settings. Z Shell has a colourful interface and a number of useful features.

For example, in Zsh you now get the new **take** command, which combines **mkdir** and **cd** so you can make a directory and enter it with a single command:

```
take test
```

...is equivalent to:

```
mkdir test
cd test
```

A new **where** command joins **which**, and shows all locations of a file on Path. To get back from Zsh to your usual Bash shell, enter:

```
exit
```

...And you'll be back to Bash. Take a look at the Zsh website (zsh.sourceforge.io) for more information and documentation.

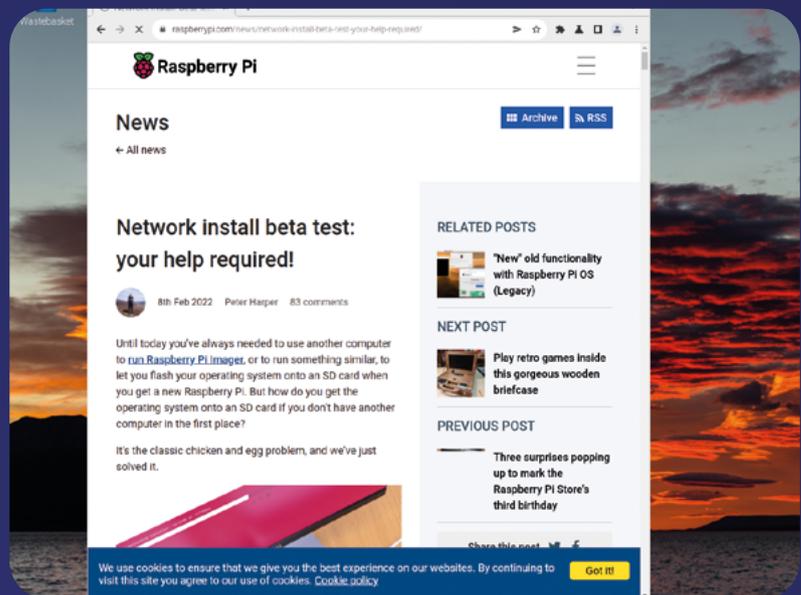
RASPBERRY PI HARDWARE HACKS

Get more from your boards with these physical hints

Portable power

One of the easiest ways to take Raspberry Pi portable is to use a mobile phone charging bank. These sticks typically use a USB connection to hold power, and you can plug them into Raspberry Pi just like your phone charger.

It's a hack though – if you want a more secure solution, it's possible to power Raspberry Pi computers using batteries (both alkaline and lithium). Devices such as this LiPo Battery HAT for Raspberry Pi (magpi.cc/lipobatteryhat) combine a power bank management chip with a Li-polymer battery. Be careful when using lithium-ion batteries to enclose Raspberry Pi and the battery inside a case.



Network install (beta)

Until recently you've needed to use another computer to set up Raspberry Pi. Typically using Raspberry Pi Imager.

A recent development at Raspberry Pi is looking to fix this with a Raspberry Pi bootloader that implements network installation. If Raspberry Pi is connected to the internet with an Ethernet cable, it will download Imager from the internet and enable you to flash a microSD card or USB drive.

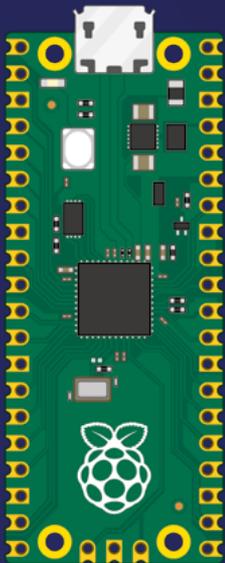
You'll need to update the bootloader on Raspberry Pi and the service is currently in Beta. But it's well worth investigating to see the future of Raspberry Pi OS installation (magpi.cc/networkinstall).

Switch to an M.2 drive

The traditional storage for Raspberry Pi is a microSD card, and you can get a huge performance boost by swapping over to an M.2 SATA drive. This can provide ten times the storage transfer speed.

If you have a Raspberry Pi 4, consider getting an Argon M.2 case (magpi.cc/argononem2). This neat case houses an M.2 SATA blade in the bottom part of the case, and Raspberry Pi 4 in the top (repositioning and expanding the ports).

If you have a Raspberry Pi 400, it's possible to gain an equivalent speed boost with an M.2 SATA drive inside a USB enclosure (magpi.cc/m2sata).



Pico pinout

Look closely at the GPIO pins on Raspberry Pi Pico and you'll detect that some of the pads are round, while other are square. The square solder pads are ground, making it easy to identify them. Take a look at the Raspberry Pi Pico pinout document (magpi.cc/picopinout).

“Look closely at the GPIO pins on Raspberry Pi Pico”

Overclock Raspberry Pi

Recent updates have enabled Raspberry Pi 4 and Raspberry Pi 400 to run at a clock speed of 1.8GHz. Combined with the impressive 4GB RAM (and 8GB on some Raspberry Pi 4 models), this enables Raspberry Pi to act as an effective desktop computer.

It's possible to take these Raspberry Pi models further, though, especially if you combine them with effective cooling. We've had a Raspberry Pi 4 with a cooling solution running at 2.147GHz.

Take a look at our 'How to overclock Raspberry Pi' tutorial (magpi.cc/overclock).

Add a reset button

It's possible to add a reset button to Raspberry Pi Zero 2 W. It has a test pad on the back marked RUN. Connect this to ground and you will reset Raspberry Pi, and this can be used with a push button to create a reset button.

You can discover the location of the RUN (and other test pins) at magpi.cc/zerotestpads.

It is also possible to connect a similar reset button to Raspberry Pi Pico (magpi.cc/picoreset). This makes it a lot easier to test out prototypes with the microcontroller.

Learn ARM assembly language with Raspberry Pi



Stephen Smith

Stephen is a retired software developer who has written three books on ARM assembly language programming. He is a member of Sunshine Coast Search and Rescue and enjoys mountain biking, hiking, and running.

magpi.cc/stephensmith

Code a small assembly language program and then compile, execute, and debug it

If you've ever wondered how the ARM CPU at the heart of a Raspberry Pi works, or how to write the most efficient code possible, then assembly language is the place to start. Raspberry Pi OS includes everything you need with the pre-installed GNU Assembler (gnu.org) and GNU Debugger. In this tutorial, we will show you how to write simple code in our favourite text editor as we explain assembly language instructions, and you'll learn about the architecture of the ARM CPU.

01 A small program

Create a folder called **Simple** to hold your program. Open Geany (menu > Programming > Geany Programmer's Editor) and enter the listing of **simple.s** (see magpi.cc/learnassembly1). Anything after an '@' sign is a comment. Assembly language statements are in the form of:

```
Label:          opcode  operands @ comment
```

You can use any text editor, and Geany Programmer's Editor is installed with Raspberry Pi OS. Geany has simple colour coding in the editor and allows you to build and execute your program. Save the file as **simple.s** in the **Simple** folder you created.

02 Create the makefile

You can enter the build steps manually in a Terminal prompt, but it is easier to have a makefile automate the process. This makefile creates a debug build, which includes information to help us use gdb (GNU Project Debugger) in a

later step. Although the makefile is a textfile, the indentations must be made with tab characters and not spaces. Copy, type in, and save the makefile, as **makefile** with no file extension, to the **Simple** folder.

03 Build the program

In a Terminal window, **cd** into the **Simple** folder, and enter the **make** command:

```
cd Simple
make
```

Or, in Geany, choose Build > Make from the menu bar.

04 Execute the program

In the Terminal window, type:

```
./simple
```

Or, in Geany, select Build > Execute from the menu. You should see 'Hello World!' printed in your Terminal window. Yay, we've written our first complete assembly language program.

05 Examine the machine code

Enter the command:

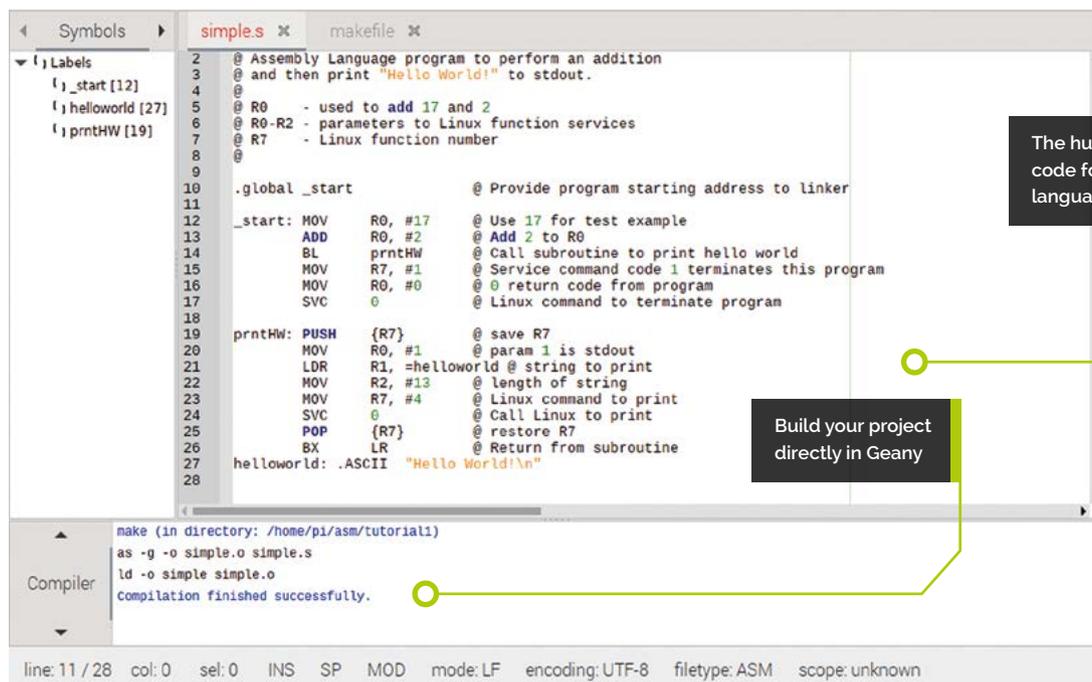
```
objdump -s -d simple.o > simple.lst
```



Alert!
32-bit only

This tutorial requires Raspberry Pi OS (32-bit) and will not work with Raspberry Pi OS (64-bit).

magpi.cc/osimages



The screenshot shows the Geany IDE with two tabs: 'simple.s' and 'makefile'. The 'simple.s' tab contains assembly code with comments. The 'makefile' tab shows the output of the 'make' command, indicating successful compilation.

```

2  @ Assembly Language program to perform an addition
3  @ and then print "Hello World!" to stdout.
4  @
5  @ R0 - used to add 17 and 2
6  @ R0-R2 - parameters to Linux function services
7  @ R7 - Linux function number
8  @
9
10 .global _start          @ Provide program starting address to linker
11
12 _start: MOV  R0, #17     @ Use 17 for test example
13         ADD  R0, #2     @ Add 2 to R0
14         BL   prntHW    @ Call subroutine to print hello world
15         MOV  R7, #1     @ Service command code 1 terminates this program
16         MOV  R0, #0     @ 0 return code from program
17         SVC  0         @ Linux command to terminate program
18
19 prntHW: PUSH  {R7}     @ save R7
20         MOV  R0, #1     @ param 1 is stdout
21         LDR  R1, -helloworld @ string to print
22         MOV  R2, #13    @ length of string
23         MOV  R7, #4     @ Linux command to print
24         SVC  0         @ Call Linux to print
25         POP  {R7}     @ restore R7
26         BX   LR        @ Return from subroutine
27 helloworld: .ASCII "Hello World!\n"
28
make (in directory: /home/pi/asm/tutorial1)
as -g -o simple.o simple.s
ld -o simple simple.o
Compilation finished successfully.

```

The human-readable code for our assembly language project

Build your project directly in Geany

Inspect the file with `cat`:

```
cat simple.lst
```

The file `simple.lst` contains lines like:

```
0: e3a00011 mov r0, #17
```

The bytes `e3 a0 00 11` are the machine code that our code was assembled into. The numeral `11` is hexadecimal for `17`. The register is `0` and the `e3a` part indicates which instruction we are executing and some options for that instruction. Each assembly instruction compiles to a single 32-bit word. A word is the natural size of data for a particular processor, in this case it is 32 bits or 4 bytes in length. This is characteristic of RISC CPUs, allowing them to quickly load and decode each instruction. Even though the `R0` register is 32 bits, we can only load a small subset of values, since there are only 16 bits available in this instruction's encoding.

06 Debug the program

We haven't talked about what each statement does, so for the next few steps, we'll debug through the program and talk about

what each statement does. In the Terminal window, enter:

```
gdb simple
```

07 Set a breakpoint

The first statement in the program is:

```
.global _start:
```

This is a label marked as global. When Raspberry Pi OS loads a program, it starts executing at the global label `_start`. We want the debugger to stop at the

“ We'll debug through the program and talk about what each statement does ”

beginning of our program when we execute it, and a convenient way to do this is to set a breakpoint at this location. All `gdb` commands have short forms to save typing, in this case `b` is short for breakpoint. Next, we can execute the program by typing the `run` command abbreviated as `r`. `gdb` will stop at the `_start` location, displaying the assembly language instruction that will execute next:

```

pi@raspberrypi:~/asm/tutorial1 $ gdb simple
GNU gdb (Raspbian 10.1-1.7) 10.1.90.20210103-git
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from simple...
(gdb) b _start
Breakpoint 1 at 0x10054: file simple.s, line 12.
(gdb) r
Starting program: /home/pi/asm/tutorial1/simple

Breakpoint 1, _start () at simple.s:12
12  _start: MOV    R0, #17    @ Use 17 for test example
(gdb) i r r0
r0          0x0            0
(gdb) s
13      ADD    R0, #2      @ Add 2 to R0
(gdb) i r r0
r0          0x11           17
(gdb) s
14      BL     prntHW     @ Call subroutine to print hello world
(gdb) i r r0
r0          0x13           19
(gdb) i r pc lr
pc          0x1005c       0x1005c <_start+8>
lr          0x0            0
(gdb) s
prntHW () at simple.s:19
19  prntHW: PUSH  {R7}     @ save R7
(gdb) i r pc lr
pc          0x1006c       0x1006c <prntHW>
lr          0x10060       65632
(gdb) s
20      MOV    R0, #1      @ param 1 is stdout
(gdb)

```

▲ Debugging our simple program in gdb

```

(gdb) b _start
Breakpoint 1 at 0x10054: file simple.s, line
12.
(gdb) r
Starting program: /home/pi/asm/tutorial1/
simple

Breakpoint 1, _start () at simple.s:12
12  _start: MOV    R0, #17    @ Use 17 for
test example

```

08 Examine the registers

In 32-bit mode, Raspberry Pi's ARM CPU has 16 registers in each CPU core. There are twelve general-purpose registers and four registers with specific purposes. To see all the registers, enter:

```
(gdb) i r
```

This is short for 'info registers'. This instruction also shows the two program status registers that you test to create conditional logic. To see a specific register, add the register's name, such as:

```
(gdb) i r r0
r0          0x0            0
```

Notice the value of R0, as the first instruction we execute will load a value into R0.

09 Single step through the addition

To execute a single assembly language instruction, enter the `step` command into gdb (`s` for short). The ARM processor uses what is called a load/store architecture. That means you need to load data from memory into the CPU to operate on it, then store the value back to memory when you are done.

The `MOV` instruction is one way to load data into a register; in this case, the value is embedded in the instruction. Next, add 2 to R0 using the `ADD` instruction. Enter the `step` and `info` commands below to see this in action.

```

12  _start: MOV    R0, #17    @ Use 17 for
test example
(gdb) s
13      ADD    R0, #2      @ Add 2 to R0
(gdb) i r r0
r0          0x11           17
(gdb) s
14      BL     prntHW     @ Call
subroutine to print hello world
(gdb) i r r0
r0          0x13           19

```

10 Jump to a subroutine

The program contains a subroutine to print the string 'Hello World!' to the Terminal window, to execute this `prntHW` subroutine, we use the Branch and Link (`BL`) instruction. This routine saves the return address in the link register (LR) and sets the program counter (PC) to the start of the routine. To see this, we will check the LR and PC registers, single-step the `BL` instruction, and then see the effects on these registers.

```

(gdb) i r lr pc
lr          0x0            0
pc          0x1005c       0x1005c
<_start+8>
(gdb) s
prntHW () at simple.s:19
19  prntHW: PUSH  {R7}     @ save R7
(gdb) i r lr pc
lr          0x10060       65632
pc          0x1006c       0x1006c
<prntHW>

```

The LR has been set to one instruction after the BL instruction and the PC is now the first instruction in the subroutine.

11 Single-step through the subroutine

The subroutine starts with a PUSH {R7} instruction to save the register R7 to the stack. Functions and subroutines are responsible for restoring the values of registers R4-R12. Before returning the subroutine calls POP {R7} to restore the saved value from the stack.

“ Call the Linux write command to output a string to stdout ”

Call the Linux write command to output a string to stdout. This call takes three parameters that are placed in registers R0, R1, and R2. The function call number to indicate a write call is placed in register R7. If you call a Linux service from any programming language, this is the code that is eventually executed. The Linux function call is initiated with the SVC (Supervisor Call) instruction. This is a software interrupt and a convenient way to switch execution to the Linux kernel via an operating system interrupt handler.

```

19 prntHW: PUSH  {R7}      @ save R7
(gdb) s
20      MOV   R0, #1      @ param 1 is
stdout
(gdb) s
21      LDR   R1, =helloworld @ string to
print
(gdb) s
22      MOV   R2, #13     @ length of string
(gdb) s
23      MOV   R7, #4      @ Linux command
to print
(gdb) s
24      SVC   0           @ Call Linux to
print
(gdb) i r r0 r1 r2 r7
r0          0x1          1
r1          0x1008c     65676
r2          0xd         13
r7          0x4         4

```

simple.s

> Language: **Assembly Language**

```

001. @
002. @ Assembly Language program to perform an addition
003. @ and then print "Hello World!" to stdout.
004. @
005. @ R0   - used to add 17 and 2
006. @ R0-R2 - parameters to Linux function services
007. @ R7   - Linux function number
008. @
009. @
010. .global _start          @ Provide program starting
    address to linker
011.
012. _start: MOV    R0, #17    @ Use 17 for test example
013.          ADD    R0, #2    @ Add 2 to R0
014.          BL     prntHW   @ Call subroutine to print hello
world
015.          MOV    R7, #1    @ Service command code 1
    terminates this program
016.          MOV    R0, #0    @ 0 return code from program
017.          SVC    0         @ Linux command to terminate
    program
018.
019. prntHW: PUSH   {R7}      @ save R7
020.          MOV    R0, #1    @ param 1 is stdout
021.          LDR    R1, =helloworld @ string to print
022.          MOV    R2, #13   @ length of string
023.          MOV    R7, #4    @ Linux command to print
024.          SVC    0         @ Call Linux to print
025.          POP    {R7}      @ restore R7
026.          BX     LR       @ Return from subroutine
027. helloworld: .ASCII "Hello World!\n"

```

**DOWNLOAD
THE FULL CODE:**



magpi.cc/learnassembly1

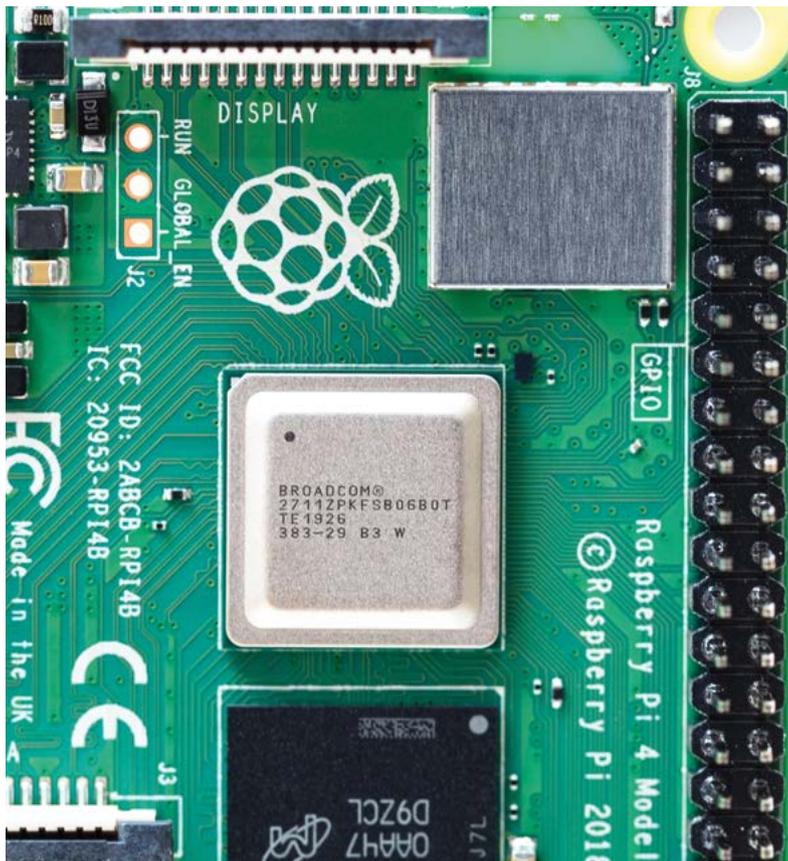
makefile

> Language: **MAKE**

```

001. simple: simple.o
002.          ld -o simple simple.o
003.
004. simple.o: simple.s
005.          as -g -o simple.o simple.s

```



▲ Get to the heart of Raspberry Pi and program the CPU with ARM assembly

```
(gdb) s
Hello World!
25    POP    {R7}           @ restore R7
(gdb) s
prntHW () at simple.s:26
26    BX    LR             @ Return from
subroutine
(gdb) s
```

“ Congratulations, you have written your first assembly language program ”

The **LDR** instruction loads the address of the string to print into register R1. Memory addresses are 32-bits, and each instruction is 32-bits long, so how does this work since the instruction also holds the operation code and the register number? If you look at the **simple.lst** file (from Step 5), you will notice this instruction was changed to:

```
ldr    r1, [pc, #32]
```

Which is a program counter relative address. The address is calculated by adding a small constant to the current value of the program counter to get the correct 32-bit address. This is an example of the GNU Assembler helping us deal with one of the tricky aspects of programming RISC processors.

12 Exit the program

After the subroutine returns, we are done. To exit our program, we call a Linux service to terminate the program. The Linux service number (1) goes in R7, and our program’s return code is placed in R0. When the **SVC** instruction is executed, Linux terminates the process, which gdb reports.

```
15    MOV    R7, #1         @ Service
command code 1 terminates this program
(gdb) s
16    MOV    R0, #0         @ 0 return
code from program
(gdb) s
17    SVC    0              @ Linux
command to terminate program
(gdb) s
[Inferior 1 (process 2248) exited normally]
```

To exit gdb, press **CTRL+D**. Congratulations, you have written your first assembly language program. Why don’t you try making changes, for instance change the **ADD** instruction to **SUB** or **MUL**? Perhaps change what is printed or see what happens if you change the program’s return code. The only way to truly learn a new programming technique is to experiment and learn by doing. 📖

Stephen’s stuff

Stephen has written three books on assembly language programming. The first, *Raspberry Pi Assembly Language Programming*, is the place to go for a deeper understanding of the topics touched on in this tutorial. The second is *Programming with 64-Bit ARM Assembly Language*, and the third is *RP2040 Assembly Language Programming* for the Raspberry Pi Pico.



RETRO GAMING

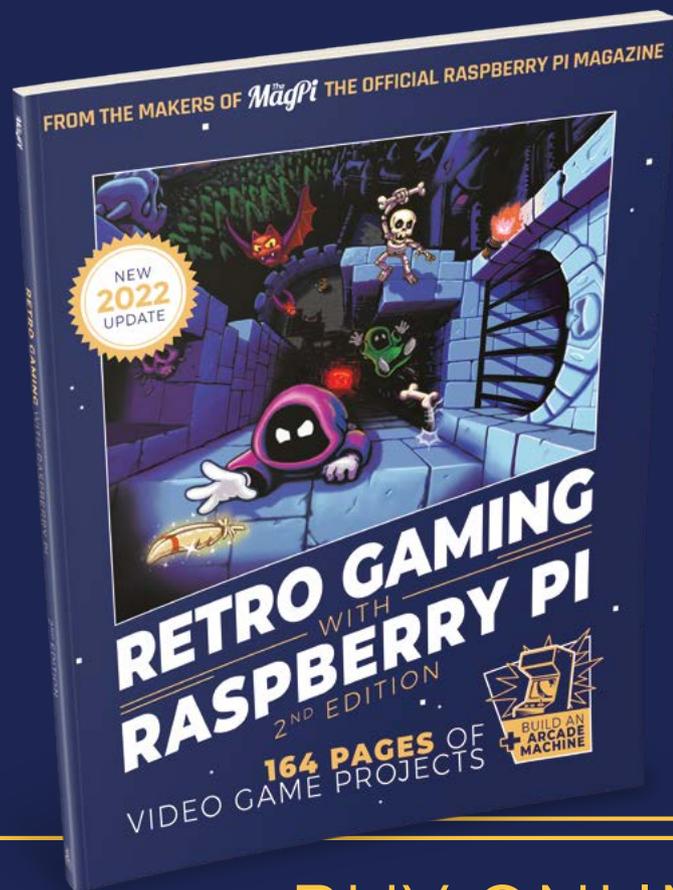
WITH

RASPBERRY PI

2ND EDITION

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games. Build your own games console or full-size arcade cabinet, install emulation software and download classic arcade games with our step-by-step guides. Want to make games? Learn how to code your own with Python and Pygame Zero.

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



BUY ONLINE:
magpi.cc/store

Set up a digital audio workstation



K.G. Orphanides

K.G. now has enough GM/GS MIDI devices to create an entire dungeon synth orchestra.

@KGOOrphanides

You'll Need

- ▶ mt32-pi project from issue 114 magpi.cc/114
- ▶ Reaper reaper.fm
- ▶ USB MIDI keyboard magpi.cc/vortexwireless2

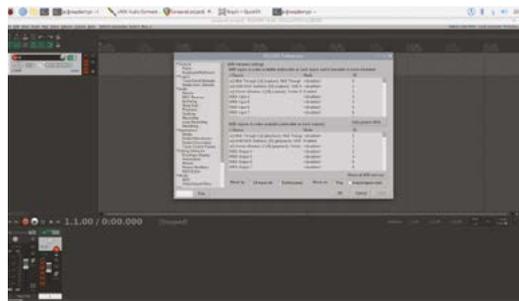
Build a powerful digital audio workstation setup for MIDI composition with Reaper, Raspberry Pi OS 64-bit, and a dedicated mt32-pi external synth

This month, we're using our mt32-pi MIDI synth from issue #114 (magpi.cc/114) to make music. We'll use a Raspberry Pi 4 8GB model with Raspberry Pi OS 64-bit as our main computer, running our digital audio workstation (DAW) software, and connect our mt32-pi to it using two sets of USB MIDI adapters, as discussed in the previous tutorial. We built our mt32-pi with a 4GB Raspberry Pi 4 and a HiFiBerry DAC+ACD audio HAT, but any mt32-pi build will do the trick, as will an off-the-shelf external synth like the Roland MT-32 and SC-55 we used in issue 111 (magpi.cc/111).

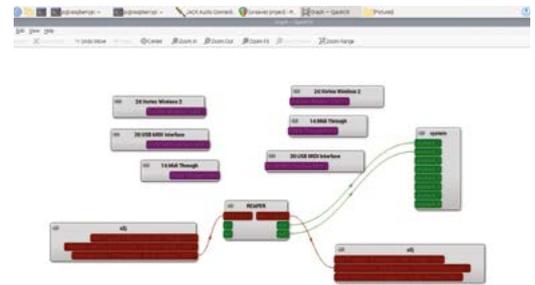
01 Connect your audio hardware

Connect the usual monitor, keyboard, and mouse to a Raspberry Pi that you'll be using as your DAW computer. Plug a USB MIDI adapter into any of its USB ports. If you have one, connect a USB MIDI keyboard.

Next, connect mt32-pi's audio outputs (preferably on a dedicated audio HAT) to your speakers, and plug in a USB MIDI adapter. Connect the MIDI IN cable from each adapter to the MIDI OUT cable of the other, and vice versa. Remember that some USB MIDI adapters, such as the industry-standard Roland UM-ONE mk2, label their MIDI IN as 'Connect to MIDI OUT' and their



- ▶ Enable your desired MIDI inputs and output devices in Reaper's preferences



▲ The red azjmidid blocks give you one JACK MIDI port for every ALSA sequencer port. Configuring Reaper's inputs and outputs will automatically connect them in QjackCtl

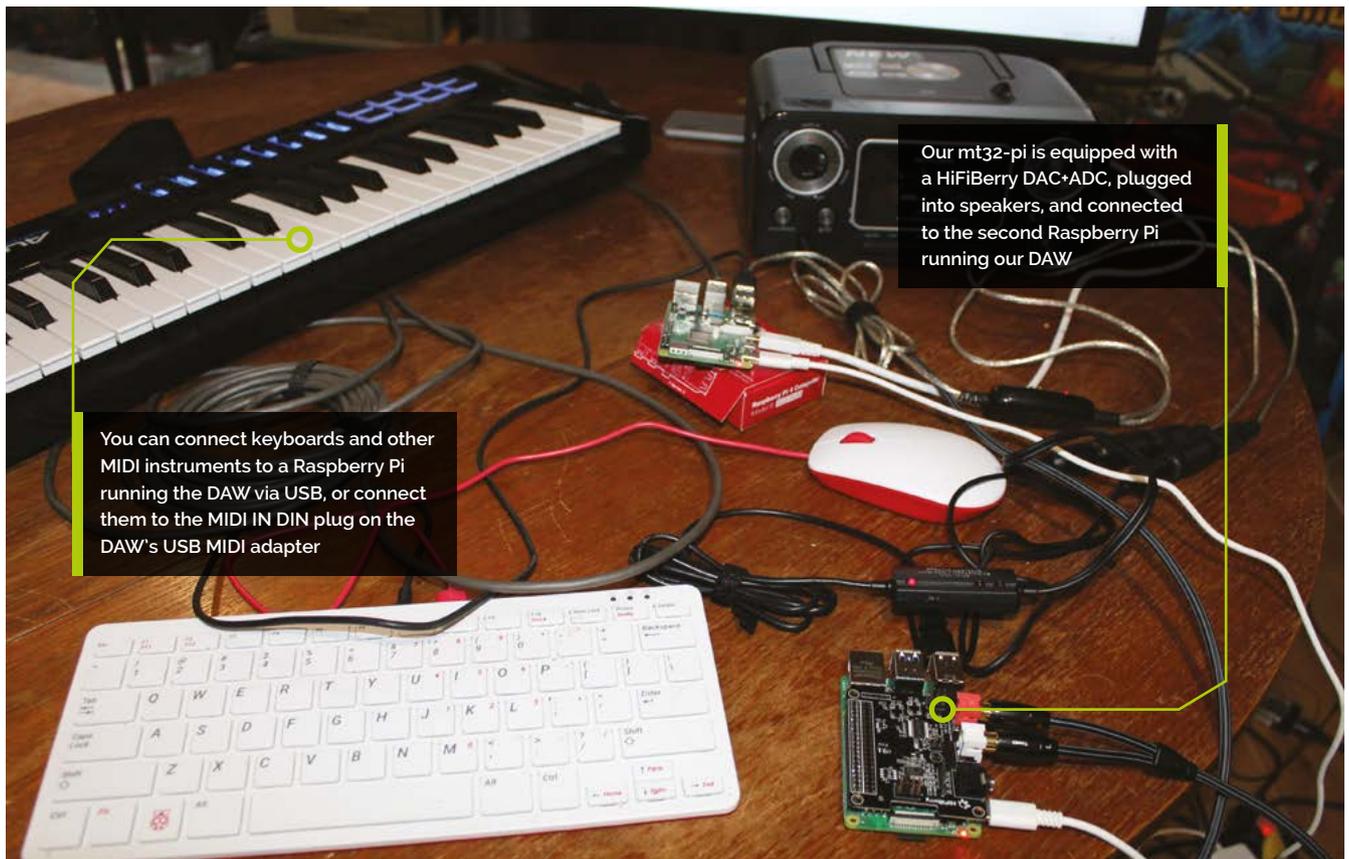
MIDI OUT as 'Connect to MIDI IN'. Connect power to both Raspberry Pi computers.

02 Download and extract Reaper

Reaper is outstanding, professional-grade digital audio workstation software. It's free to try for 60 days, but a single-user, multi-device licence costs \$60 and is well worth getting if you find the software useful. Go to reaper.fm, click Download, and select the appropriate version for your OS – that should be Linux aarch64 if you're using Raspberry Pi OS 64-bit, and Linux armv7l if you're using a 32-bit version. Create a folder in your home directory called Software, copy the downloaded file to it, and extract it via the right-click menu.

03 Install dependencies

Although you can run or install Reaper straight away, we're going to install some dependencies and optional components first. Open a Terminal and type:



```
sudo apt install jackd qjackctl vlc ffmpeg
lame a2jmidid
```

Most of these will probably already be installed, but JACK in particular is an important optional component, particularly if you want to handle PCM audio in real time, as well as MIDI. We'll use the a2jmidid package to automatically bridge ALSA's MIDI sequencer ports to JACK.

04 Install Reaper

Open a Terminal, **cd** to the folder you copied Reaper to and run its installer. For our 64-bit build, that looks like:

```
cd ~/Software/reaper_linux_aarch64
sh install_reaper.sh
```

Reaper's installer gives you a few different options. Type **R** to try it out without installing it. Type **I** to install it in either **opt** or **~opt** – you'll probably be prompted for your root password during this process if you choose the former. Or type **A** to add desktop integrations and file associations for the Reaper executable in its current path.



◀ QjackCtl can automatically start and kill a2jmidid, getting you ready to make music faster

We're going to press **I** to install it, then **1** to put it in **/opt/REAPER**, **Y** to install desktop integrations, **Y** to create relevant symlinks, and **Y** again to proceed.

05 Configure JACK

Type **qjackctl** at the Terminal to load GUI controls for the JACK low-latency audio. Press Start on QjackCtl, then open a fresh Terminal and type **a2jmidid -e**. In QjackCtl, click Graph to see your MIDI and PCM audio connections. Disconnect everything by pressing **CTRL+C** and then clicking Disconnect in the toolbar. Test your keyboard and mt32-pi by dragging a connection from your keyboard's a2j capture interface – 'Vortex Wireless 2 [32] (capture)' in our screenshots – to mt32-pi's

Top Tip

Other voices

We're using our mt32-pi with Roland SC-55 GM/GS voices, but more GM-compatible SoundFonts can be found at magpi.cc/mt32soundfonts.

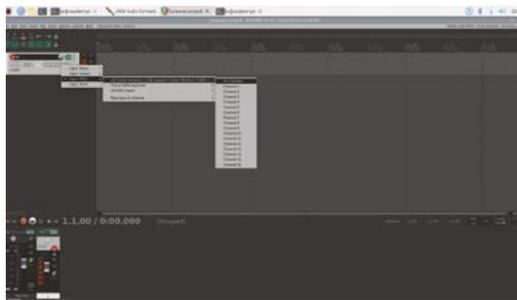


▲ Channel your inner 1980s band member with a Vortex Wireless 2 keytar

a2j playback interface, which will be something like our ‘USB MIDI Interface [28] (playback)’. Make sure your speakers are on and play a few notes. Select everything again and disconnect.

06 Configure Reaper

Start Reaper from Raspberry Pi’s Sound & Video menu. When you start Reaper, you’ll be informed that it isn’t free software and that you should register it after 60 days. Close this by clicking Still Evaluating, and continue. Go to Options > Preferences and select MIDI Devices under Audio. All your a2j (ALSA to JACK) MIDI ports should be here. Double-click on the entry for your MIDI instrument under the top inputs section and click ‘Enable input from this device’. Click OK. In the outputs section below, double-click the entry for your USB MIDI interface and tick ‘Enable output to this device’. Click OK. You can also enable input

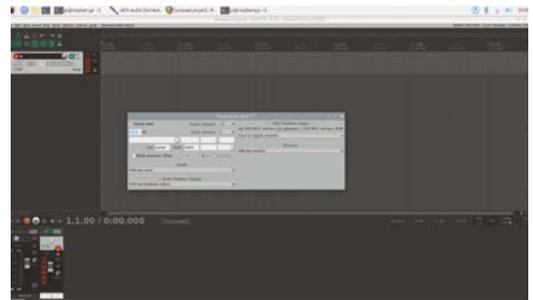


▲ You can select a MIDI input – or even specific channels – for each Reaper track. This also makes it easy to use multiple MIDI input devices

or output from the right-click menu of each device. Click Apply and OK.

07 Prepare a MIDI track

Right-click in the blank area at the left-hand side of the screen and select Insert new track. You can also press **CTRL+T** to do this. Click the round red record button on the track you’ve just created to arm it, then left-click where it now says ‘Input 1: not connected’. Mouse across your a2j MIDI keyboard entry and select ‘All channels’. Now when you press a key, a yellow level meter should flicker on the channel. We’re now at the point where we can record some MIDI, but we still need to hear what we’re playing. In the mixer panel at the bottom of the screen, you’ll see a mixer to go with our track. Click the stripy Route button and under MIDI Hardware Output, select the a2j USB MIDI Interface entry.



▲ Reaper MIDI outputs are most easily assigned on a per-track basis. Composers who use multiple MIDI tracks often make a template file with MIDI outputs already assigned

08 Make a test recording

Make sure the playhead (the position indicator on the timeline) is at the beginning. Press the red record button on the main control panel above the mixers and play something on your MIDI instrument. This should default to a general MIDI piano sound, which you should hear in real time as you play. Click the red button again to stop when you've finished, and click Save All when asked which files you want to save. Note that this doesn't save your entire project – press **CTRL+S** to do that. Now press play – right next to the record button – to listen to your work. Press stop when it gets to the end, and return the playhead to the beginning.

09 Control your MIDI instruments

Make sure the playhead (the position indicator on the timeline) is at the beginning. Press the red record button on the main control panel above the mixers and play something on your MIDI.

10 Automatically launch a2jmidid with JACK

Having established that everything works, it would make life a lot easier if all our audio daemons launched together. Open QjackCtl's Setup menu and go to Options. Tick 'Execute script after Startup' and add the following to its text entry box:

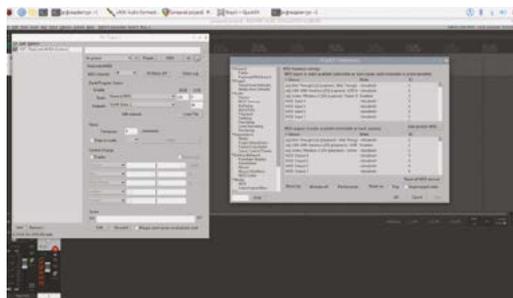
```
a2jmidid -e &
```

And in the same way, add the following command to 'Execute script after Shutdown':

```
killall a2jmidid
```



▲ Reaper looks a little austere by default. Adding themes will help



◀ The ReaControlMIDI FX plug-in allows you to command your external hardware MIDI synth to change voices

11 Add PCM audio recording

As a fully-fledged DAW, Reaper can also record analogue instruments as real-time PCM audio, but you'll need a USB audio interface such as a Focusrite Scarlett to connect microphones, guitars or other instruments to Raspberry Pi, as it lacks any audio input and its integrated BCM2835 analogue audio output isn't production-quality, either. If you want to export a project with both MIDI and PCM tracks, you can use the FluidSynth SoundFont synthesizer (fluidsynth.org) to render MIDI tracks into PCM or use an external mixing desk to route all audio to a recording device. Raspberry Pi isn't hugely powerful, which limits simultaneous multi-track recording, but it is absolutely silent, making it a useful studio device.

12 Why an external MIDI synth?

Raspberry Pi 4, particularly the 8GB version running a 64-bit OS, packs in a lot of power for its size and price. However, audio processing can be resource-intensive. While we could get the same sounds by running FluidSynth or a FluidSynth-based VST on our DAW system and routing everything through that, you're making Raspberry Pi do more work than it has to. Using mt32-pi – or any vintage or modern external MIDI synth – takes some of the load off. This comes in particularly handy if you're recording PCM audio alongside MIDI. [\[7\]](#)



▲ A range of comfortable and even colourful themes are available at Reaper's website

Top Tip

Pay the Reaper

If you find Reaper's professional-grade recording environment useful, don't forget to register it at reaper.fm.

CDP Studio: Trigger LED patterns with a web GUI



Phil King

WRITER

Long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

You'll Need

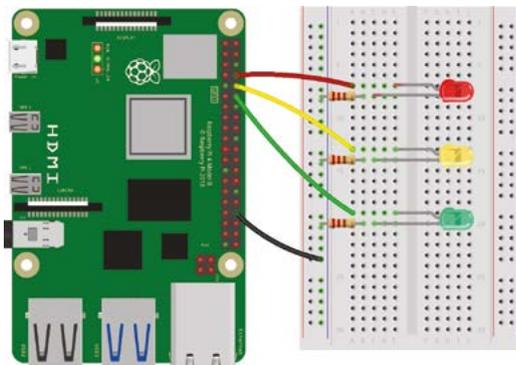
- ▶ Linux or Windows PC
- ▶ CDP Studio cdpstudio.com/getstarted
- ▶ Raspberry Pi
- ▶ Raspberry Pi OS (Bullseye or Legacy version, 32-bit)
- ▶ 3 × LEDs
- ▶ 3 × 330 Ω resistors
- ▶ Jumper wires

▶ **Figure 1** The wiring diagram for the LEDs circuit

Use the low-code, block-based environment of CDP Studio to create and deploy a Raspberry Pi project

An ‘out of the box’ software development tool, CDP Studio is used by numerous companies to build industrial control, automation, and edge systems. Yet, it’s fairly easy to get to grips with its low-code programming environment. Better yet, you can deploy projects to a Raspberry Pi – this is often used for testing and prototyping purposes in industrial settings.

In this introductory tutorial, we’ll guide you through the process of creating an app to flash LEDs connected to a Raspberry Pi in different patterns, selectable on a web GUI. We’ll mainly be using CDP Studio’s Block Diagram mode, connecting various preset block components, although we’ll also do a tiny bit of hard coding in C++ to create a new custom function.



01 Install the software

Visit cdpstudio.com/getstarted and download the free non-commercial version for Linux or Windows. During installation, make sure you select both Raspberry Pi ARMv8 32-bit (Debian 11) and Raspberry Pi ARMv6 32-bit (Debian 10) components, along with the one already ticked for your host PC. You will then be able to deploy projects to any Raspberry Pi model, using Raspberry Pi OS Bullseye or Legacy version, by selecting the appropriate toolkit in CDP Studio.

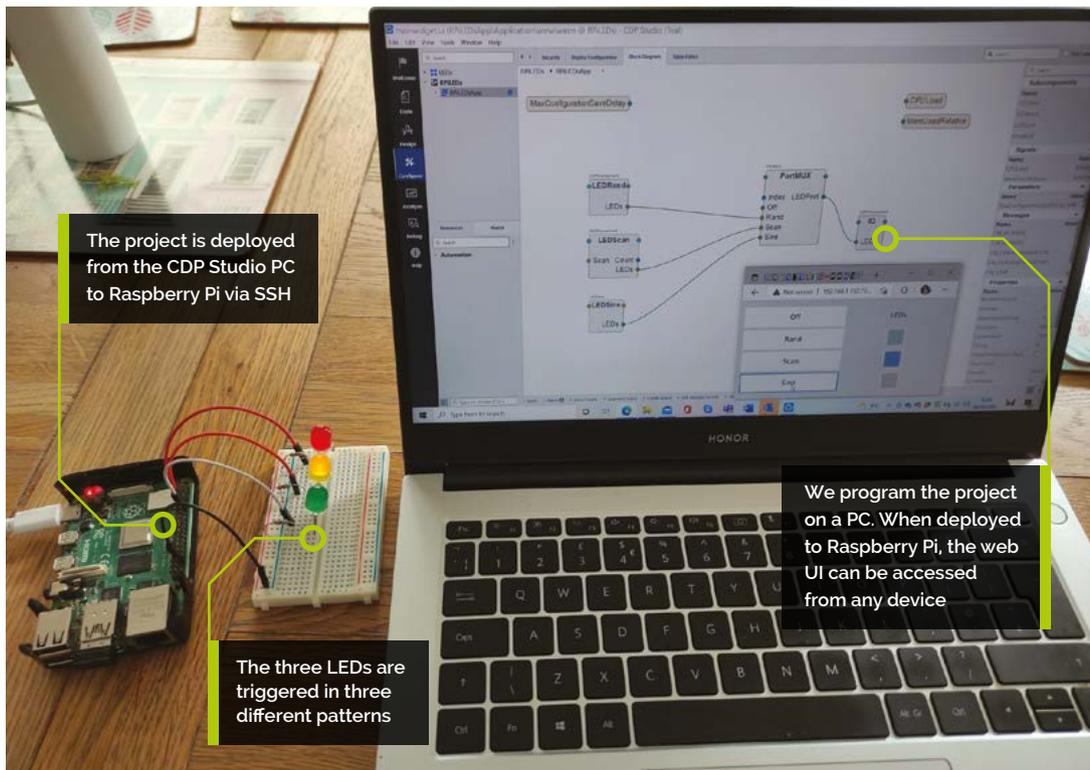
02 Create a library

For this project, we’ll create a library to house a custom function. Components created in libraries can be used easily in other projects.

Go to Create New > Library and enter a name for it (we called our library ‘LEDs’). Right-click your library in left panel and select Add New. Select the CDP Port Model in the dialog and click Choose. Click Next, then enter name for it (LEDPort in our case), then click Next and Finish.

At the bottom of the **LEDPort.cpp** C++ code file that appears, right-click **UserProperty** and select CDP > Change, then add a name for it (LEDPort). Change the Type to bool, then click OK. This will auto-create an **LEDPort.h** header file.

In **LEDPort.cpp**, we need to add extra properties for two more LED patterns. Add a new line under



Top Tip

Random issue

When the project is deployed on a Windows PC, the three AddRandom blocks will set the same value at the same time, turning all three LEDs on/off together. When deployed to Raspberry Pi, however, each individual LED will toggle randomly.

LED1. Create, then right-click and select CDP > Add > Property. Name it LED2 and change its Type to bool, and Routing Type to Periodic. Tick the box for Create Another. Click OK, then name this one LED3; untick Create Another, then click OK.

Finally, we need to build the library. Right-click the LEDs library name in the left panel and select Build, then Save All.

03 First LED pattern

For our first pattern, we'll create a custom component. Switch to Configure mode and right-click your LEDs library name in the left panel, then Add New > CDP Component Model. Click Choose, then Next; in the Component Model parameters, give it a class name (LEDSine), and change the Frequency to 2 (cycles per second). Click Next, then Finish.

The **LEDSine.cpp** file will be auto-created. Right-click in its code, then select CDP > Add > Port. In the Port type drop-down, select LEDS::LEDPort (the one you created). We'll name it 'LEDs'.

Under `/* Write your code here */` in **LEDSine.cpp**, add the following (indented) code:

```
LEDS.LED1 = (counter==0);
LEDS.LED2 = (counter==1);
LEDS.LED3 = (counter==2);
counter += increment;
```

```
if (counter==0 || counter ==2)
    increment*=-1;
```

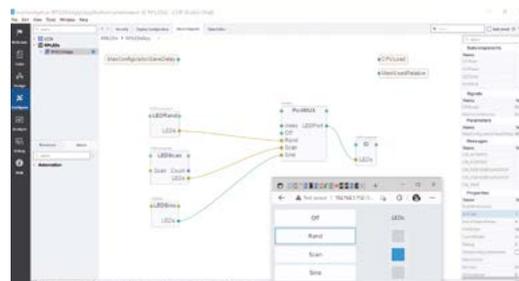
In the **LEDSine.h** header, under `LEDPort LEDs;`, add these two lines to create the **counter** and **increment** variables:

```
int counter = 0;
int increment = 1;
```

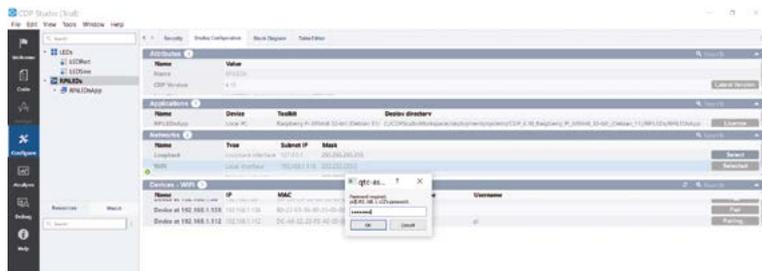
Build the LEDs library again. In Configure mode, under LEDs in the Resources panel, we now have the LEDSine component available.

04 Main block structure

Right-click in the left panel and select Create New > CDP System. Give it a name (RPILEDs), click Next twice, then Finish. In the Block Diagram tab of Configure mode, you'll now see the main block for the project. Double-click



◀ The main block diagram for the LEDs app in CDP Studio, with the three LED functions on the left. The web GUI is also shown here in a browser window



▲ Pairing CDP Studio with a Raspberry Pi on the wireless network via SSH. It will also need its `/etc/security/limits.conf` file modified – see magpi.cc/cdprpisetup

it (or select `RPiLEDsApp` in the left panel) to see inside. We can now start adding program blocks.

In the bottom left Resources panel of Configure mode, open the `CDPCore` category and drag a `CDPComponent` into the diagram. Drag another `CDPComponent` underneath it. Then open the LEDs category in Resources to find your `LEDsSine` component; drag it under the two `CDPComponent` blocks.

Click on LEDs on the `LEDsSine` block and untick Input in the right-hand panel, to make it an output. Select `LEDsSine` in the LEDs library. In the Table Editor tab, click the right arrow next to LEDs (under Ports) and untick Input there too.

Back in the Block Diagram view for `RPiLEDsApp`, open `CDPCore` in Resources and drag a `PortMUX` block into the diagram. This acts as a multiplexer for the three LED patterns, avoiding complex, messy individual wiring.

Drag another `CDPComponent` to the right of `PortMUX`. Right-click and rename it `IO`. This block will house our GPIO server.

Now drag over a `CDPPort` (under `CDPCore`) into both `CDPComponent` blocks. Select it in each block and untick Input in the right-hand panel, to change it to an output.

05 PortMUX inputs

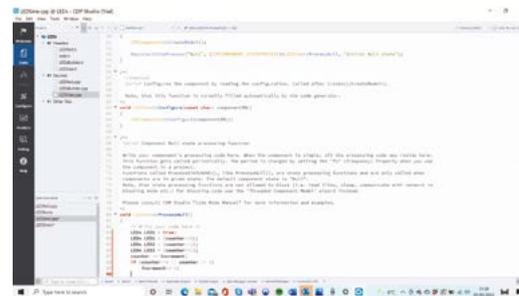
Now to add more inputs to our `PortMUX` block. From LEDs in Resources, drag an `LEDPort` into the `PortMUX` block. Select it and, in the right-hand panel, rename it 'Off'.

From `CDPCore`, add three `CDPPorts` to the `PortMUX` block – these are for our three LED patterns. Rename `CDPPort` to `Rand`, `CDPPort1` to `Scan`, and `CDPPort2` to `Sine`. From LEDs in Resources, drag an `LEDPort` into `PortMUX`; untick Input in the right-hand panel.

It's time to make our first connection: drag a wire from the LEDs output of the `LEDsSine` block to the `Sine` input on `PortMUX`.

06 IO block

Add a `CDPPort` to the `IO` block and rename it LEDs, then connect the LEDs output of `PortMUX` to it.



▲ We need to add a few lines of C++ code for our custom function's `LEDsSine.cpp` file

Now to add our GPIO server for Raspberry Pi. Double-click the `IO` block, then open `GPIOPinIO` in Resources and drag a `GPIOServer` into `IO`. With `GPIOServer` selected, right-click `GPIOPin` in Resources and select `Add Multiple`. Change its name to `LED1` and increase `Count` to 3. Click `OK` to create three outputs – `LED1`, `LED2`, `LED3`; change them all to inputs.

Select the LEDs block in `IO`. Open `CDPCore` in Resources and drag three `PushConnection` outputs into the LEDs block. Rename them `LED1`, `LED2`, and `LED3`. Now connect the `LED1`, `LED2`, `LED3` outputs of LEDs block to the `LED1`, `LED2`, `LED3` inputs of the `GPIOServer` block.

We need to enter the GPIO pin numbers for our circuit's LEDs (Figure 1). Select each of the `LED1`, `LED2`, `LED3` inputs on the `GPIOServer` in turn and set its `Nr` value (right panel) to the GPIO pin number: 14, 15, and 18 respectively.

07 More LED functions

Back in `RPiLEDApp`, rename the top `CDPComponent` to `LEDRand`, and the other `CDPComponent` to `LEDScan`. Rename both their `CDPPorts` to LEDs. Connect them up to the `Rand` and `Scan` inputs on `PortMUX`.

We'll create both of those functions using existing elements and no hard coding. Open `LEDRand`, select the LEDs block, and add three `PullConnection` inputs. Rename them `LED1`, `LED2`, and `LED3`.

From Automation in Resources, drag an `AddRandom <unsigned char>` component to the left of the LEDs block. Change its `Upper` value to 2 in the right-hand panel – it will generate values that will be interpreted as a 1 or 0, to turn an LED on or off.

Connect its `Out` output to the `LED1` input on the LEDs block. Let's create two more `AddRandom` blocks. Right-click the existing `AddRandom` block and `Copy and Paste` into a blank area of the diagram, then select `Add`. Repeat this process to add a third `AddRandom` block. Connect the two new blocks' `Outs` to the `LED2` and `LED3` inputs respectively on the LEDs block. In the right-hand

Top Tip

Deploy on PC

Before deploying the project on Raspberry Pi, you may want to test it by running it on the local PC, in which case the web GUI is at `127.0.0.1:7689/index.html`.

panel, change the `fs` (i.e. frequency) of `LEDRand` to 2 (or whatever you prefer) to slow it down.

08 Scan function

Our `LEDScan` function flashes the LEDs in sequence, using bit shifting. We don't have room here to describe its creation in detail, but you can recreate it or copy it directly (by holding **SHIFT** while dragging over it) from the downloadable project (magpi.cc/github). If you do copy it over, make sure to copy the three `LED1`, `2`, `3`, connections (right panel) from the `LEDScan` block into the LEDs block you created earlier for your `LEDScan` function, then delete the copied-over `LEDScan` block.

In your `LEDScan` function, select `Count` on the left and untick `Input`, then connect the `Out` of the `MUX` block to it. We need to loop it back to the start, so right-click `Count` and `Copy Path`. Select the `In` of the `ShiftLeft` block and paste the path into the `Routing` field.

Finally, check the values (right panel) are the same as those in the downloaded project version for the `ShiftLeft`, `GT`, and `MUX` blocks. Change the `LEDScan` function's `fs` property to 2 to slow down the sequence.

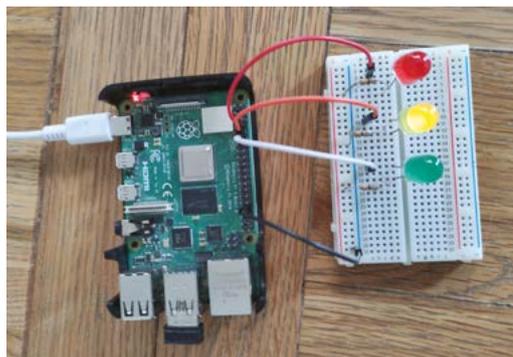
09 Web GUI

Let's add a web GUI to control the program. Right-click `RPiLEDsApp` in the top-left panel and `Add New > Add Web UI`. Use the default options in the dialog.

Switch to `Design` mode to start designing the UI. Delete the existing text box, then drag a `Grid Layout` element into the canvas. Drag four `Buttons` into it. Drag a `Lamp` (representing an LED) to the right of the second button from the top. Then drag two more `Lamps` to the right of the lower buttons. Drag a `Label` to the right of the top button; double-click it and change it to 'LEDs'. Double-click the buttons and label them: `Off`, `Rand`, `Scan`, and `Sine`.

Select the `Off` button. In the bottom-right panel, scroll down to `cdpCheckedRouting`. Here you'll need to input the routing for the button so it works. To do so, return to the `RPiLEDsApp` diagram, right-click `Index` on `PortMUX`, and `Copy Path`. Back in `Design` mode, paste it into the `cdpCheckedRouting` field for the `Off` button. Tick `checkableByEmitValue`.

For the `Rand` button, tick `checkableByEmitValue` and change `valueToEmit` to 1; paste the path (as before) into `cdpCheckedRouting`. Repeat this



▲ A simple LED circuit is connected to Raspberry Pi's GPIO pins

for the other buttons, with `valueToEmit` set to 2 and 3 respectively.

Now to route the Lamps. In the `Block Diagram`, select the `LEDPort` output on `PortMUX`, then right-click `LED1` (right panel) and `Copy Path`. In `Design` mode, select the top `Lamp` and paste the path into its `cdpStyleRouting` field. Select each of the other Lamps, then paste the path into `cdpStyleRouting` and change the last digit to 2 and 3 respectively.

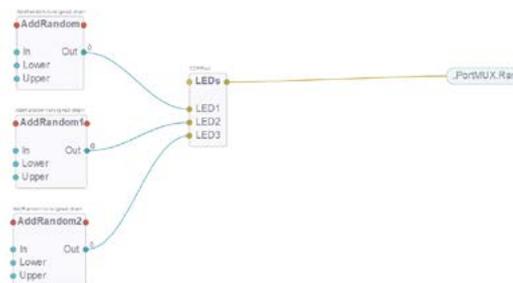
10 Ready to deploy

In `Configure Mode`, select the `LEDs` library and the `Deploy Configuration` tab. Tick the box for `Raspberry Pi ARMv6` and/or `ARMv8` under `Toolkits`. Rebuild the library.

Make sure your Raspberry Pi has `SSH` enabled and you've modified its `/etc/security/limits.conf` file – see magpi.cc/cdprpisetup for details. Wire up the LEDs circuit as in **Figure 1**.

To find your Raspberry Pi from `CDP Studio`, in `Deploy Configuration` for `RPiLEDsApp`, click `Select` next to `WiFi` under `Networks` to show devices. Enter `pi` as the `Username` and click `Pair`, then enter Raspberry Pi's password.

Choose the appropriate `Toolkit` for your Raspberry Pi OS version; under `Device`, select your Raspberry Pi's IP address. Right-click `RPiLEDs` in the left panel and select `Run & Connect`. The web UI can be found at your Raspberry Pi's IP address followed by `:7869/index.html` – for example, ours was `192.168.1.112:7689/index.html`. You can now click the UI buttons to change the LED pattern. 🎇



◀ The `LEDRand` function comprises three `AddRandom` blocks, one for each LED

HOW

L

ast February, my children rushed in with a copy of the latest HackSpace magazine for me, excited because of the gold lettering and free gift on the cover.

The gift was a microcontroller, and my children's enthusiasm was a catalyst for me to come up with a project that would appeal to and involve them. This would be the first project that I had done with a microcontroller. I picked the Python route as it seemed the closest step from BASIC that I had used many years ago at school. I worked through the tutorials published in HackSpace issue 39 (hsmag.cc/issue39), and these gave me a good understanding of how to flash the Pico with Python and connect components. If you are starting out with the Pico, I recommend this as a great place to begin.

With small children, we have an abundance of toy cars and building bricks and, whilst rolling cars down improvised tracks built from books and supported by DUPLO, I thought it would be great if we could set up a race track and use the Pico to determine the winner. After research on YouTube, watching videos of Hot Wheels cars being raced down roof sheets and guttering, I was left wondering how I could transport such a large track. I needed to find a more portable solution, and inspiration arrived in the form of plastic conduit. I took a trip to the local electrical wholesaler to find

By Robert Hamilton-MacLaren

I
MADE

RASPBERRY PI PICO HOT WHEELS DRAG RACE TRACK

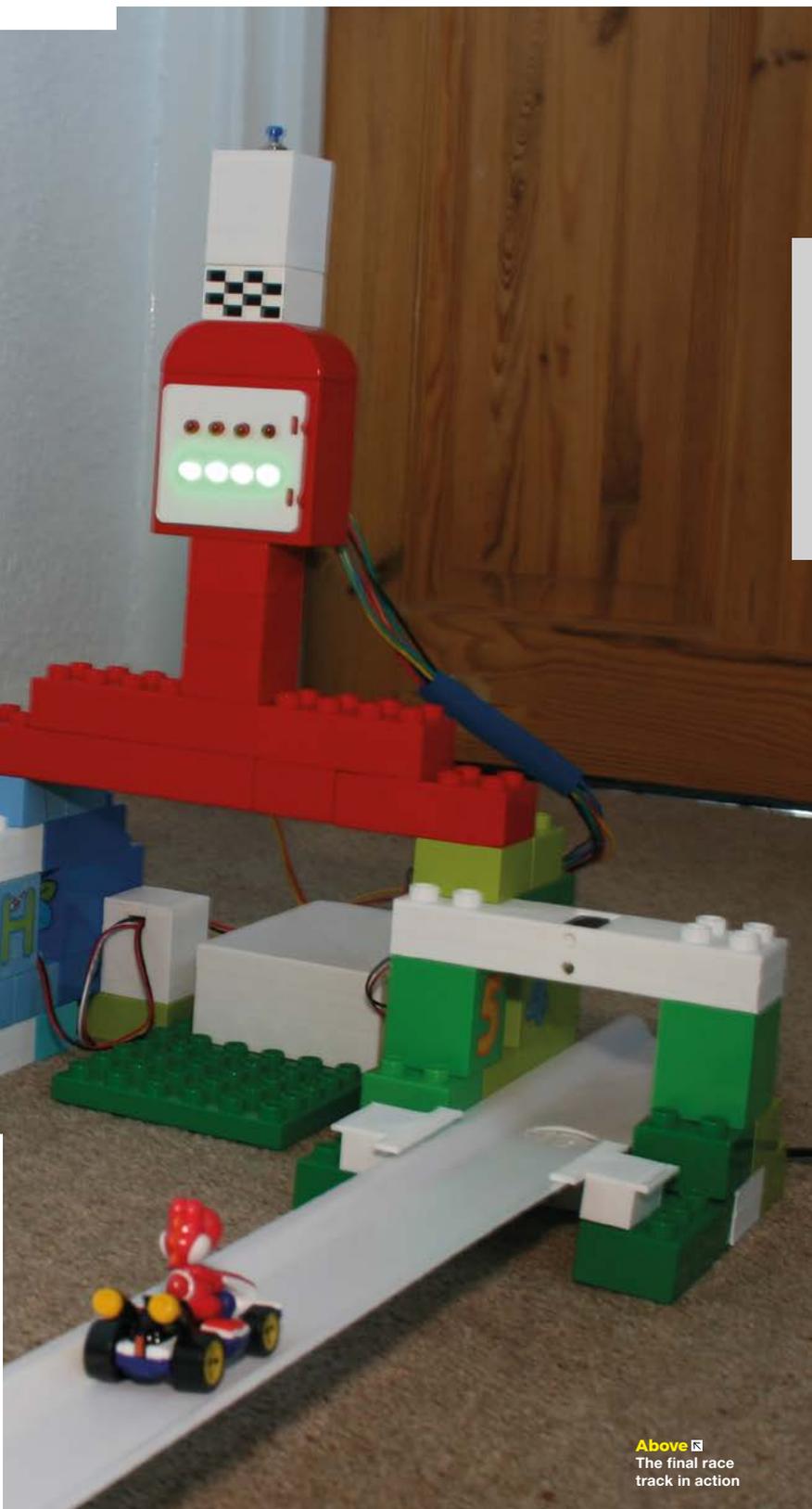
A microcontroller for micro racing

What I used

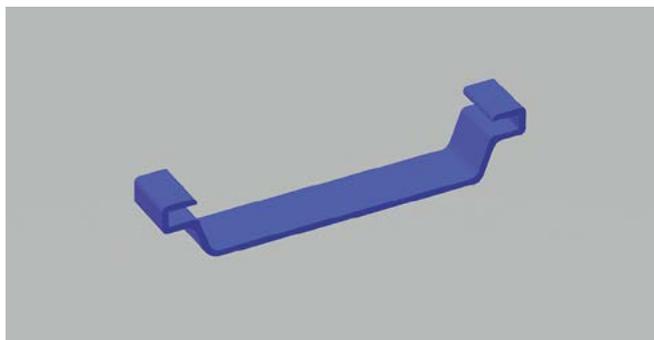
- Raspberry Pi Pico
- 4 × red LEDs
- 4 × green LEDs
- 8 × 330Ω resistors
- Transistor and resistor
- Piezo buzzer
- 2×3mm break beam sensors
- Push-to-make button
- Electrical wire
- Stripboard
- Race track
- Toy cars

the optimum width for the cars to run along (38mm) and the 2m lengths easily flexed to fit into the car. Hot Wheels track would be another option but, for a fair race, the competing tracks would need to be identical and quite a large collection would be needed. A 3D-printed adapter could probably be made to incorporate loops and jumps to add to the excitement.

In terms of components and materials for the build, this is a low-cost project (£30 or less, depending on components you already have). Of course, you will need a soldering →



Above  The final race track in action



Above ♦
Custom brick to hold and connector to join the track

iron and a computer to program the Pico, but other parts can be adapted or improvised using whatever you have available. I chose to house the sensors in 3D-printed parts that fit to DUPLO because I could save a lot of printing time. All that was needed was a few bespoke bricks. Thingiverse proved a great starting point with bricks that could be modified to fit the chosen sensor. On Thingiverse, 'klamowk' has published a customisable DUPLO-compatible brick:

[thingiverse.com/thing:4815088](https://www.thingiverse.com/thing:4815088). The required size can be entered, which takes the hard work out of making a brick that will fit to others. Other benefits of using DUPLO are that you don't need much space to store the completed project because it can easily be disassembled, and the layout can be varied every time.

Don't be put off doing a similar project at this point if you haven't got access to a 3D printer – you could easily mount the sensors in wood or even cardboard.

THE TRACK

The track on its own is quite flexible so does need supporting. I designed a track connector that slides over two pieces of conduit to join them. If you don't have a 3D printer, the track could easily be supported and joined by fixing it to a batten (this would be very rigid, but not so easy to collapse to store). DUPLO towers were used at intervals to support the track along its length up to the finish line, with custom bricks holding it in place. I got the children involved in building the towers, and we were having fun racing cars even before I'd started on the electronics.

BREAK BEAM SENSORS

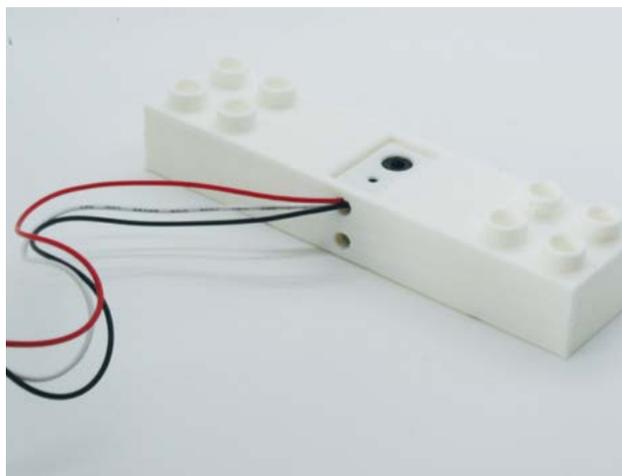
We tried timing the races manually, and the children realised how difficult it was to tell by eye which car had won the race. Browsing the available sensors, I came across the Adafruit Breakbeam sensor ([hsmag.cc/BreakBeam](https://www.hsmag.cc/BreakBeam)), which will operate at the Pico's 3.3V (certainly over the distance needed for the project). There are two parts (an infrared LED and a sensor) and, as both components are the same size, I was able to design a brick that could be used for both and know that they would be perfectly aligned. I chose to place the sensor vertically above and below the finish line. The conduit track can easily be drilled to allow the beam to pass through without compromising the car's path. Once connected to the Pico, the pin is high until the beam is broken. I set up a trigger event to capture the pin going low (see **Code 1** listing). The code run by the trigger event is really simple: if an `in_a_race` flag is set to

“THE TRACK ON ITS OWN IS QUITE FLEXIBLE”

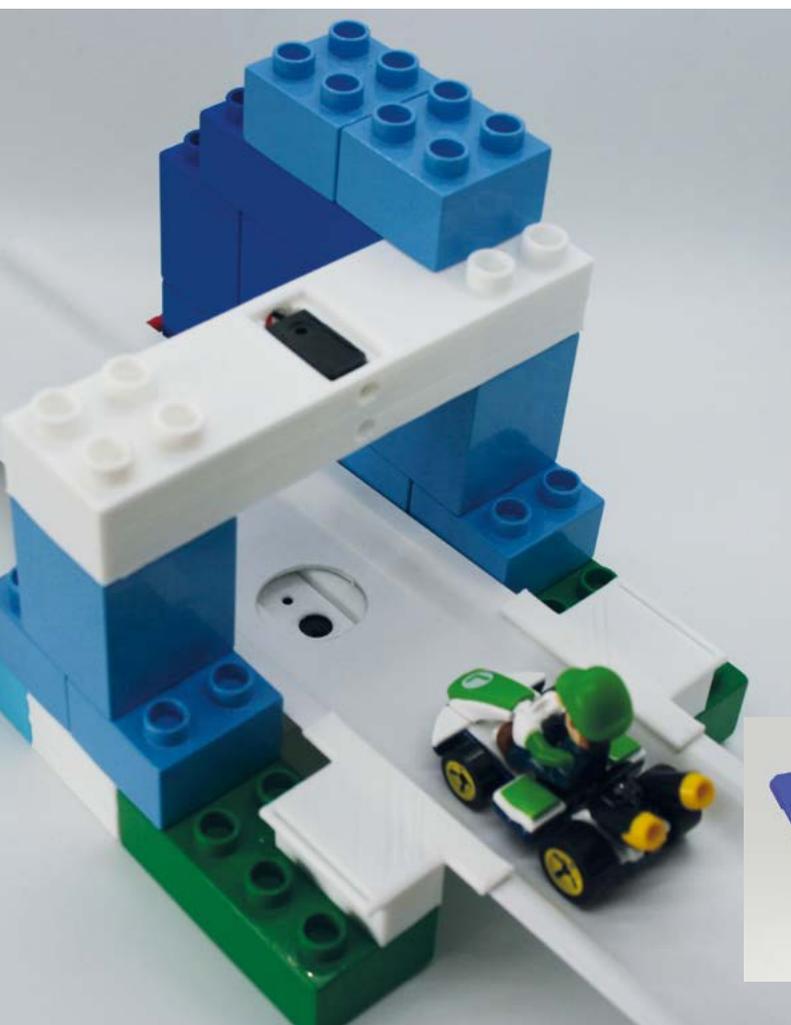


HackSpace

This tutorial is from HackSpace magazine. Each issue includes a huge variety of maker projects inside and outside of the sphere of Raspberry Pi, and also has amazing tutorials. Find out more at [hsmag.cc](https://www.hsmag.cc).



Left ♦
Light and Sensor fits the same sized 3D-printed housing, ensuring they align

**CODE 1**

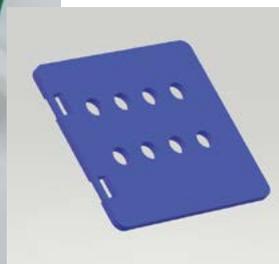
```
breakbeam1 = Pin(5, Pin.IN, Pin.PULL_UP)

def beam1broken(pin):
    #set beam broken flag only if in a race
    global beam1_broken
    if in_a_race == True:
        beam1_broken = True

breakbeam1.irq(trigger=Pin.IRQ_FALLING, handler=beam1broken)
```

Left  Assembled finish line, ready for track to be fitted

Below  Modified DUPLO window to house LEDs



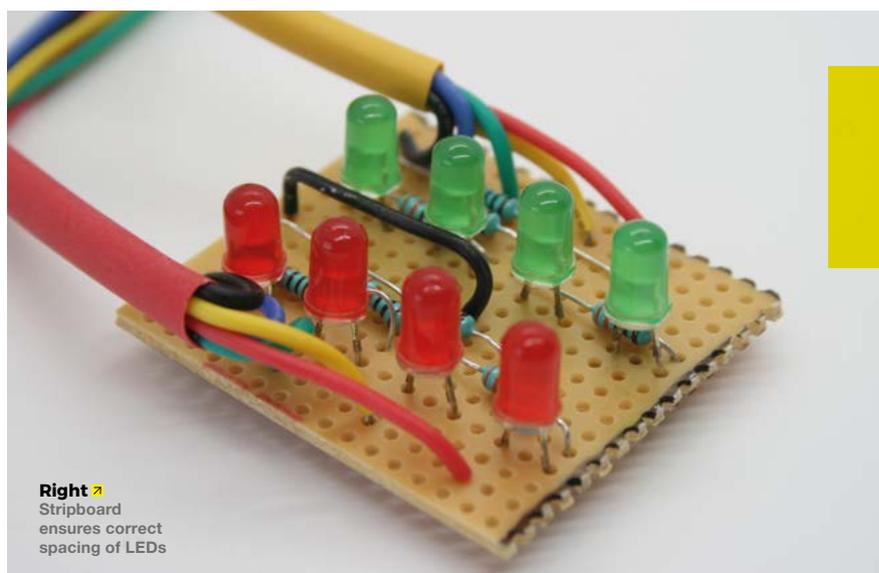
I did start looking at options such as adapting an SPI module which would have only required a few wires. I opted to keep things simple, and connected each LED to a pin on the Pico as there were plenty available for this project. The eight LEDs were soldered onto a stripboard with appropriate resistors, as this ensured the correct spacing for the printed door. The finished circuit fitted nicely into the window casement.

We thought it would be fun to have an audible start to the race, but the first piezo buzzer I used was far too quiet. I purchased a louder one on eBay, which I thought had a built-in transistor to operate (as there is not enough current to power direct from the pin). After testing, it turned out there was no transistor, but once I wired it up with an external one, it provided a good loud beep. >

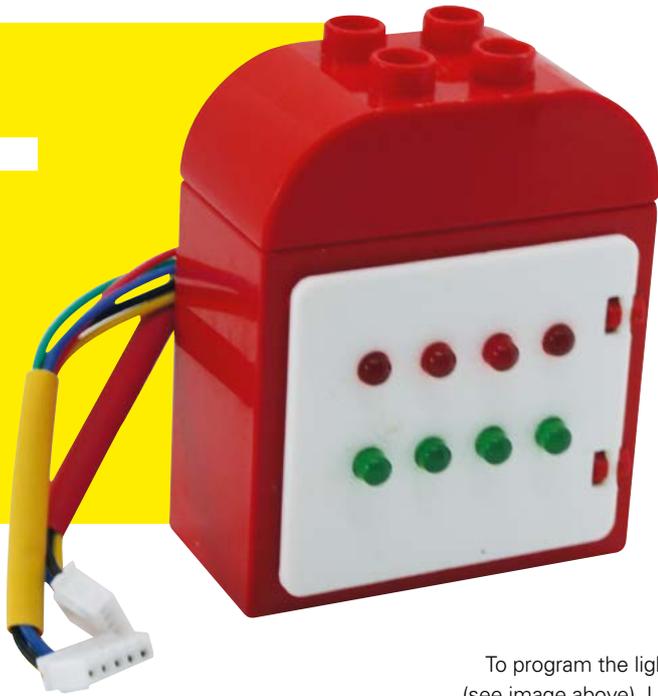
True, then the **beam1_broken** variable is set to **True**. The rest of the code is within the main loop so that the trigger code is run as quickly as possible. Initially, I had thought I would need to deal with multiple signals as the car passed through the beam. But actually, I haven't had any trouble with false triggers and I just need to know the first instance of the beam being broken.

STARTING LIGHTS

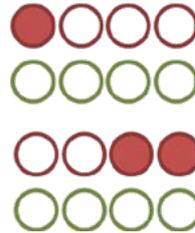
Rather than a basic stop/go light, I wanted to create the feel of a race track starting grid with a series of red and green lights similar to those seen on Formula 1. I started designing a brick to house the lights, but realised a DUPLO window brick combined with a 3D-printed front panel for housing the starting lights would considerably cut down the printing time and, of course, a blank was available on Thingiverse [thingiverse.com/thing:2888128](https://www.thingiverse.com/thing:2888128).



Right  Stripboard ensures correct spacing of LEDs



End of the race



Above ⬆
Red lights alternate to show 1st and 2nd place

Left ⬅
Completed starting lights

connectors make the project easier to take apart, but are not essential. I added some extra connections to facilitate future upgrades. The circuit was covered with a 3D-printed DUPLO brick-size case. A downside of using

DUPLO for the construction is that there is nowhere to route the cables. If repeating this project, I would have built the starting lights and control box into one enclosure to reduce the number of exposed wires.

To program the light and beep sequence (see image above), I created a starting light routine which can be called from the main loop each time a race is started.

The race is started (or reset) by pressing a button mounted in a printed custom DUPLO brick. In the future, I would like to change this to a remote control to reduce the number of wires.

The lights also perform a second function; they are used to show first and second place at the end of the race, with the red lights flashing accordingly to indicate the winner (see End of the race, above).

THE CODE

The main code is a loop within which there is a sub loop that runs while a race is taking place. This sub loop looks to determine if an event has been triggered by a beam being broken. If it is the first occurrence, it is recorded as the winner, and the second occurrence finishes the race and lights display the final race positions.

The setup can be run connected to the computer, but as the intention was to have this as a standalone toy, I saved the code on the Pico as **main.py** so that it would run each time the circuit was powered. I used a USB power adapter, but a battery would make this even more portable.

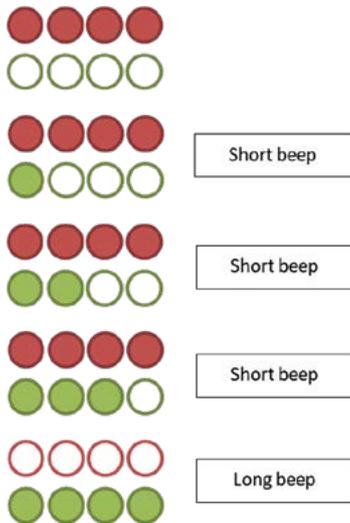
IN ACTION

After some initial testing, the track was put to use at a party. Children quickly got to grips with how to operate and interpret the results. Initially they were just racing cars, but soon they wanted to answer questions like, "Which types of cars were fastest?" and "Does the same car always win?"

FUTURE IDEAS

In terms of construction, there are plenty of potential enhancements, such as ramps and jumps or even a car launcher. These could be 3D-printed or made from cardboard. An upgrade to the electronics would be to display the race finishing times (the full code already stores the times in variables which could be output to a display). Older children might enjoy the challenge of modifying

Starting Sequence

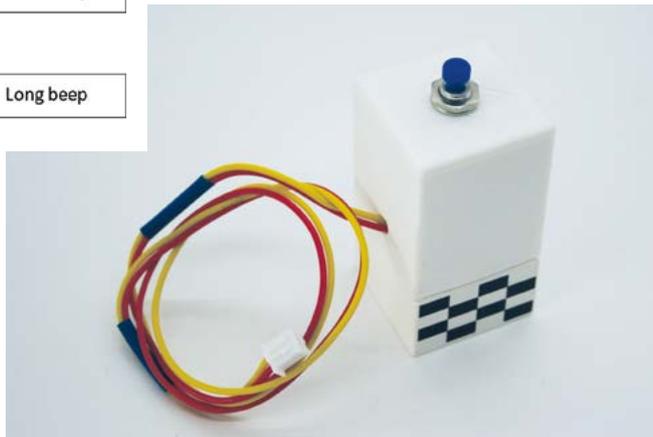


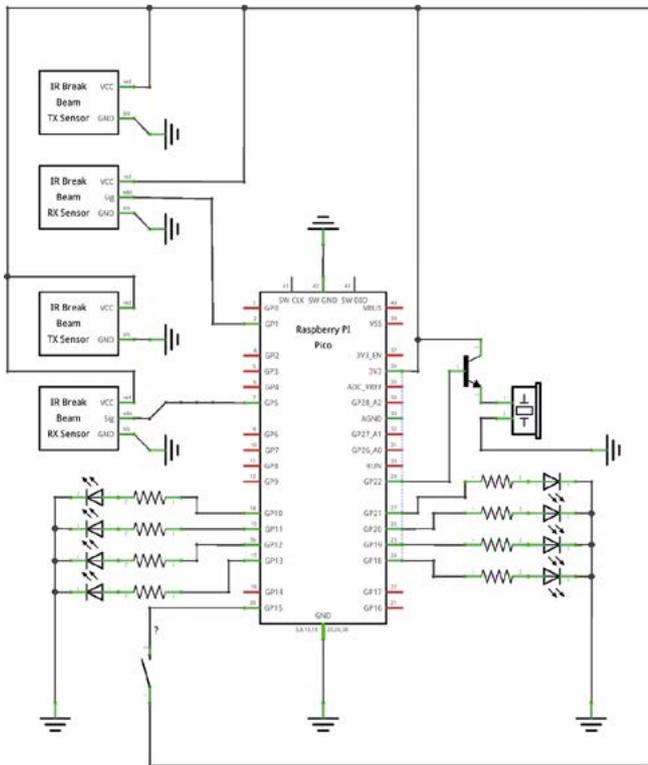
Above ⬆
Starting sequence

Right ➡
Start and stop button printed DUPLO brick

BRINGING ALL THE PARTS TOGETHER

After initially testing the various parts on a breadboard, I decided to make the wiring more permanent using stripboard. The JST





the code further. For example, a relay race where as many cars as possible are sent down the track in two minutes, or a time trial which uses both of the sensors on the same track and measures the time between the two to calculate the speed.

Above ⚡
The schematic for the circuit

Below ⚡
Race track in action

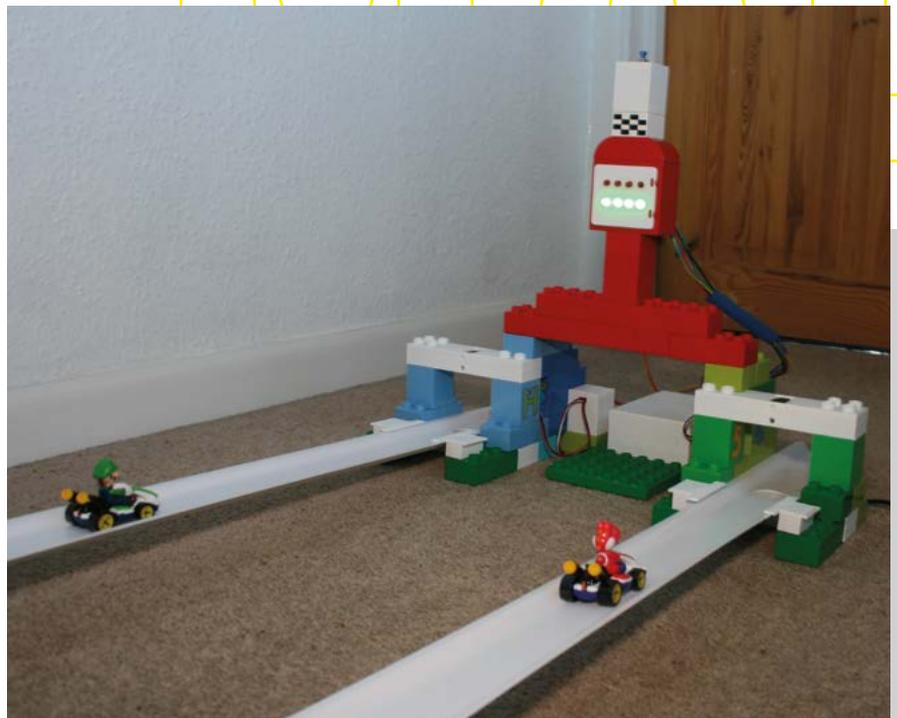
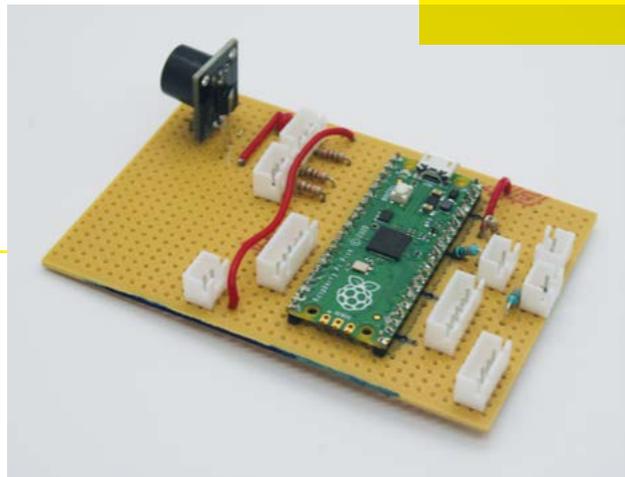
CONCLUSION

The Pico Drag Race Track has provided us with hours of entertainment, and the ease with which it can be packed up, transported, and reassembled has meant it has made regular appearances at family gatherings. This project has developed my 3D printing, electronics, and programming skills and has also been fun to build, especially as I've been able to involve my children with so much of the process. It is an accessible and versatile project that can be adapted in many different ways (if cars aren't your thing, how about modifying this for a marble run?). I've put the code and 3D models on a Facebook group, [hsmag.cc/PicoRace](https://www.facebook.com/hsmag.cc/PicoRace), should you want to create something similar. □

PSEUDOCODE

- When button is pressed to start the race
- Call the starting lights and beeps routine
- Enter the in-race loop
 - First car past the finish line – record the winner
 - Second car past the finish line – the race is finished
- When race has finished, exit the race loop
- Display the winner
- Repeat

Below ↓
JST connectors allows quick assembly with extras for expansion



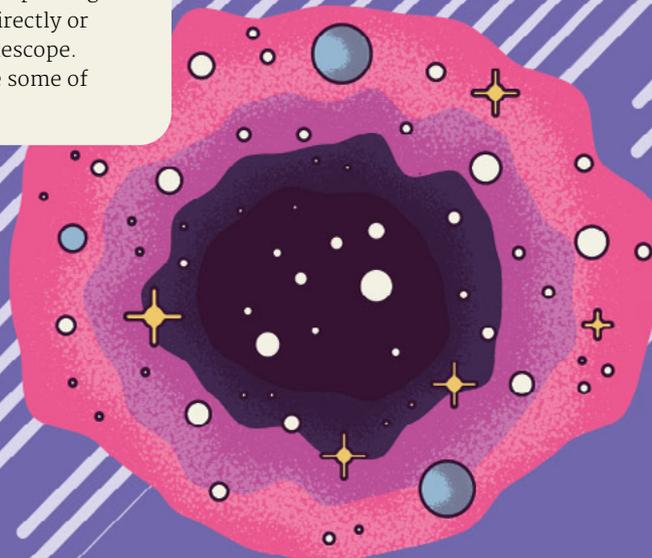
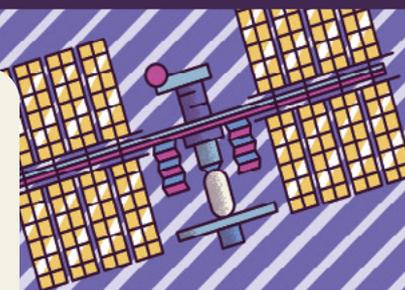
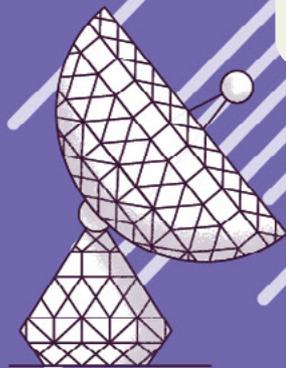
Shoot for the stars

Astrophotography with Raspberry Pi. By **Rob Zwetsloot**

Space is very cool. Stars, planets, nebulae, and the Moon are very pretty as well. However, getting a good photo of the night sky can be difficult without specialised equipment, even if you own a telescope yourself.

Thanks to Raspberry Pi and its powerful camera functionality, and the open-source night sky trackers you can install to it as well, you can quickly improve your astrophotography – whether that’s peering up into the sky with the camera directly or going through a high-powered telescope.

Space is very cool. Let’s capture some of that cool.



Low-tech space views

Get a better look without investing in tech

While electronics and telescopes will help you get some really great views and shots, you can use these low-tech methods to get a better look than with the naked eye.

DSLR camera

While these are higher tech, they're not particularly designed for astronomy. Still, with a tripod, as long a lens as you can find, and a long exposure time, you can get some great photos. Check out this guide for more info on how to do it: magpi.cc/astrodslr.



Credit: Jacek Halicki CC BY-SA 4.0



Credit: Jacek Halicki CC BY-SA 4.0

Binoculars

While there are binoculars that are more specifically designed for stargazing, any decent pair can be pointed up when it's night. You'll get better views of the Moon, actually see Mars, and even potentially see some of the moons of Jupiter if you look at the right time of year.

Creating a space camera

Different ways to capture the cosmos

You can take incredible photos with the Camera Module or High Quality Camera and Raspberry Pi using a few clever tricks with lenses, exposure, and maybe even a telescope.

Raspberry Pi Moon Camera

James Mitchell ([raspberrypi.com/news/jamberslin/](https://www.raspberrypi.com/news/jamberslin/))

The Raspberry Pi Moon Camera is something we featured way back in issue 45. What makes this system quite unique is the use of a custom, 3D-printed lens adapter that allows you to attach traditional lenses to a Raspberry Pi Camera Module – long before the High Quality Camera came along with interchangeable lenses. In this case, they were Canon EF (EOS) lenses.

The shots James got were quite remarkable, and only used the default settings: “I’m embarrassed to say I left it on auto. In fact, the command I used had no modifiers at all – `raspistill -o moon.jpg`. What I got really was a happy accident. Could I improve the picture with changing some settings? Absolutely!... I think I need to learn more about how the optics work. Maybe I could adjust the distance between the lens and the sensor to improve the sharpness.”

Interestingly, the most expensive part of a project like this is the lens.

magpi.cc/45

You'll have to use the legacy camera library or legacy OS: magpi.cc/legacy.



▲ Using a 3D-printed lens adapter, you can use powerful lenses with a standard Raspberry Pi Camera Module



▲ A great picture of the Moon captured with the camera



Cheap All Sky Camera

Jippo

While this camera may look a little low-tech, it actually uses a lot of smart techniques to not only see more of the sky, but automatically take photos of anything bright that appears in it. It makes use of a HQ Camera and a 180-degree CS lens beneath an acrylic dome to protect it from the elements.

It uses software called Meteotux Pi, written by Jippo's amateur astronomy friend Jani, which does a lot of the heavy lifting, allowing the whole setup to stay relatively cheap.

"After Meteotux Pi has taken the images, a Python Shell script starts to generate star trails and normal video from stacked images," Jippo tells us. "This script uses ImageMagick and FFmpeg and when they are ready, [it] moves those images and videos to my network drive via WiFi where I can check them."

magpi.cc/allskycam



▲ This image of a satellite makes us think the images from a meteor shower would be fantastic

▶ A very unassuming setup for a fairly cool astrophotography project



Hubble Pi

Santiago Rodriguez

This is quite the beefy piece of kit, combining a telescope and a HQ Camera with some custom software to control image taking. It's no wonder that it was created by a physics student like Santiago.

This setup doesn't have a dedicated star tracker, but Santiago loaded Raspberry Pi with KStars, which gives a live view of the night sky that he could use to find celestial objects. He's able to control exposure, ISO, and shutter speed – although there are some limitations to that shutter speed as it's digital. Still, he's able to use multiple exposures to approximate a long exposure.

magpi.cc/hubblepi



▲ While a small telescope, it's still great for looking at the night sky

◀ An extreme close-up of the Moon, possible thanks to the telescope

Official Moon Shots

Greg Annandale, whose photos you'll find on the desktop of Raspberry Pi OS, took this incredible photo of the Moon using the HQ Camera, with quite the lens attached, that allowed for a 1400mm focal length.



Powering a telescope with Raspberry Pi

Using Astroberry Server on a telescope



1 Hardware setup

Computerised telescopes and telescope mounts either make use of USB or RS-232 connectors for hardware to interact with it. For USB, you may just need to provide external power, as well as plugging in to Raspberry Pi, but for RS-232 you may need more specialist hardware, such as the Pegasus USB Control Hub: magpi.cc/pegasususb.

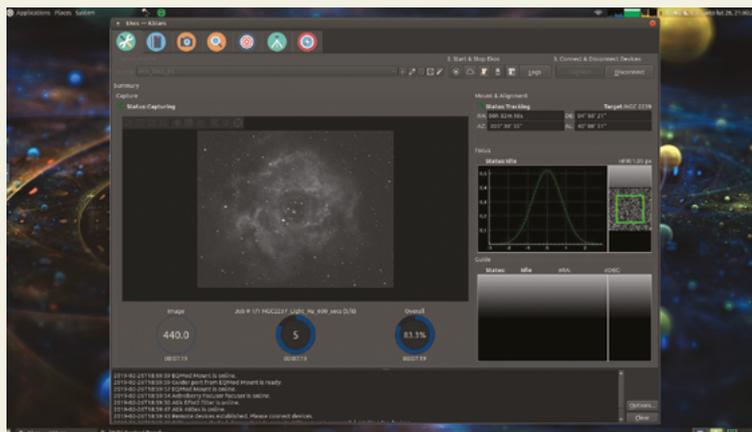
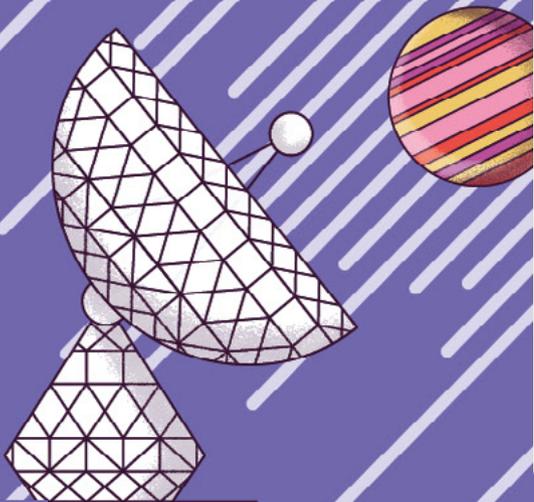
2 Install Astroberry Server

From astroberry.io you can find the image to download to install Astroberry Server using Raspberry Pi Imager (magpi.cc/imager). After a first boot, you can connect via the Astroberry wireless hotspot, using `astroberry` as the password, and head to astroberry.local in your browser. You'll be able to configure the system from here, including giving details on your telescope hardware to make it easier to control via INDI. Refer to the Astroberry Server documentation for more info.

3 Using Astroberry

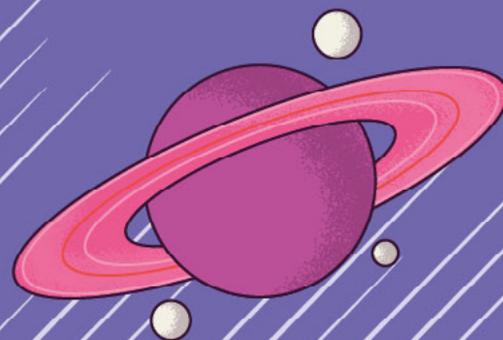
Astroberry comes with a variety of software that allows you to track celestial objects and take photos – such as KStars which comes with specific Astroberry drivers, several pieces of imaging software for image stacking and tweaking exposure.

Connecting a Raspberry Pi Camera Module or HQ Camera to the viewfinder will allow you to use the telescope and automated systems for fantastic images – just make sure to weather-proof it!



A year in the night sky

What to look out for while stargazing over the next year



SPRING

Lyrids Meteor Shower

This meteor shower is created by dust particles from a comet, and will be mostly visible around the Lyra constellations

22-23 April

Mercury at Greatest Eastern Elongation

Mercury will reach its greatest eastern elongation of 20.6 degrees from the Sun, making it a fantastic time to look for it. It will be low in the western sky after sunset

29 April

Eta Aquarids Meteor Shower

A meteor shower from dust left behind by Halley's Comet, which you'll be able to mostly see around the Aquarius constellation

6-7 May

Total Lunar Eclipse

The Moon will be completely obscured by the Earth tonight, giving it a dark red colour. It will be visible tonight throughout all of North America, Greenland, the Atlantic Ocean, and parts of western Europe and western Africa

16 May

SUMMER

Mercury at Greatest Western Elongation

Mercury will reach its greatest eastern elongation of 23.2 degrees from the Sun, another great time to look at it. On this occasion, it will be visible to the east before sunrise

16 June

Delta Aquarids Meteor Shower

From comets Marsden and Kracht comes this meteor shower, which you'll mostly see around the Aquarius constellation

28-29 July

Perseids Meteor Shower

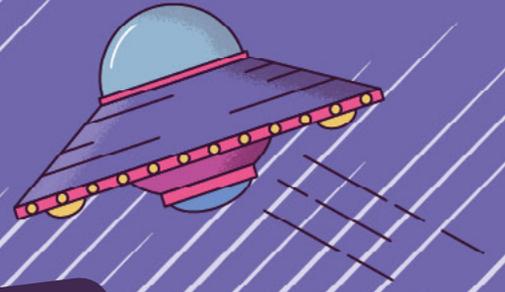
A highly visible meteor shower from Swift-Tuttle, perfect for a nice summer evening. The near-full Moon might disrupt it a little, but look towards the Perseus constellation nonetheless

12-13 August

Saturn at Opposition

The closest Saturn will get to Earth this year, making for the best time to get some photos of probably the prettiest planet in our system, and you can see it all night long

14 August



AUTUMN

Mercury at Greatest Eastern Elongation

Mercury is back, and this time it will 27.3 degrees from the Sun. Look to the west after sunset again for this one

27 August

Neptune at Opposition

The final planet in the Solar System will be at its closest tonight, and visible all night long. It's extremely far away though, so you may only get small blue dots for photos

16 September

Jupiter at Opposition

Our big pal welcomes us to autumn, and will be at its closest point to Earth tonight, and with a medium telescope you'll be able to get some stunning photos all night long. You can even check it out with binoculars!

26 September

Draconids Meteor Shower

A smaller meteor shower which is best viewed in the early evening, in a place with as little light pollution as possible. Look towards Draco for most of the shower

7 October

Uranus at Opposition

Like Neptune, Uranus is extremely distant, but this will be the night when you can get the best view of it. Again, it may just be a small blue-green dot

9 November

Leonids Meteor Shower

Look to the Leo constellation for this particular meteor shower. It peaks every 33 years, although the last one was in 2001, so it may not be as amazing as the Perseids shower earlier in the year

17-18 November

Mars at Opposition

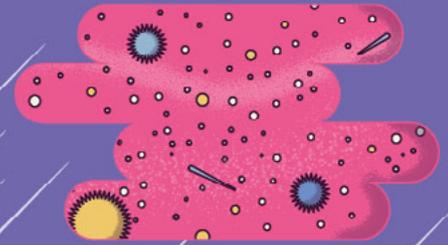
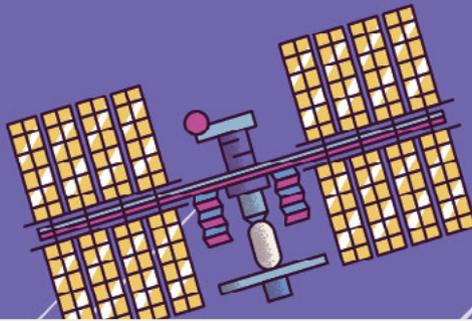
Our neighbour planet will be very close tonight and, with a decent telescope, you'll be able to get some stunning shots of the red planet

8 December

Geminids Meteor Shower

Brave the cold for one of the best shows of the year, with roughly 120 multicoloured meteors expected to fall every hour. Look towards Gemini, but be prepared for them to come from anywhere

13-14 December



Mercury at Greatest Western Elongation

Mercury will be visible in the east just before sunset, with a western elongation of 18 degrees from the Sun

8 October

Orionids Meteor Shower

Another meteor shower originating from Halley's Comet, you'll find most of them around the Orion constellation this time

21-22 October

Taurids Meteor Shower

If you're looking for a bit of a challenge, this small meteor shower will be taking place on a near-full Moon and radiating from the Taurus constellation. With some patience, you may get a nice spot

4-5 November

Total Lunar Eclipse

Another total lunar eclipse will occur, this time in the Eastern Hemisphere. Look out for it in eastern Russia, Japan, Australia, the Pacific Ocean, and parts of western and central North America

8 November



WINTER

2023

Mercury at Greatest Eastern Elongation and Ursids Meteor Shower

On the shortest day of the year, you can get another chance to view Mercury (to the west after sunset) and meteors from the comet Tuttle. It's a small shower, but the Moon will be dark and leave the night sky open for some visible meteors, mostly around Ursa Minor

21-22 December

Quadrantids Meteor Shower

The first meteor shower of the year is larger than some and will tend to be seen from Bootes, but can be seen anywhere in the night sky

3-4 January

Mercury at Greatest Eastern Elongation

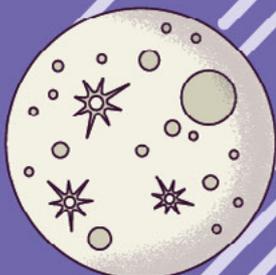
Mercury is back in the west after sunset, with a 19.2 degree elongation from the Sun

7 January

Mercury at Greatest Western Elongation

The end of winter brings one last Mercury viewing time, just before sunrise to the east

16 February



8BitDo Arcade Stick

SPECS

DIMENSIONS:

303 x 203 x 111.5mm

WEIGHT:

2.1kg

CONNECTIVITY:

Bluetooth, RF (through included dongle), wired via USB

MACRO BUTTONS:

2

BATTERY:

1000 mAh Li-ion, 30-40 hour play time

► 8BitDo ► 8bitdo.com ► £78 / \$90

A classic console colour scheme on a very modern arcade stick. Can **Rob Zwetsloot** put it through its paces? Sure he can!

The market for high-quality arcade/fight sticks has been an interesting one over the last decade – while in the West arcades are few and far between, there are people still interested in playing games like they did in the arcade. Whether they’re hobbyists enamoured with the classic style, or hardcore players in the fighting game community (FGC), there will always be people looking for one. Raspberry Pi retro gamers are no different.

8BitDo have answered with their own stick simply called Arcade Stick. Like their other controllers, it’s fairly platform-agnostic, making it playable on a Switch or PC – and that includes

Raspberry Pi. It uses XInput, the Xbox 360 controller API, allowing you to use it with very little hassle in a vast majority of games, whether you’re using RetroPie or Steam Link. The eight face buttons map the major buttons on all modern controllers, and it includes two programmable macro buttons, switches to change how the stick is defined, and a couple of extra function buttons for Start, Select, Home, etc.

Connect it your way

There are three ways to connect the stick to a Raspberry Pi, each of them good in its own way. While technically the fiddliest way to connect



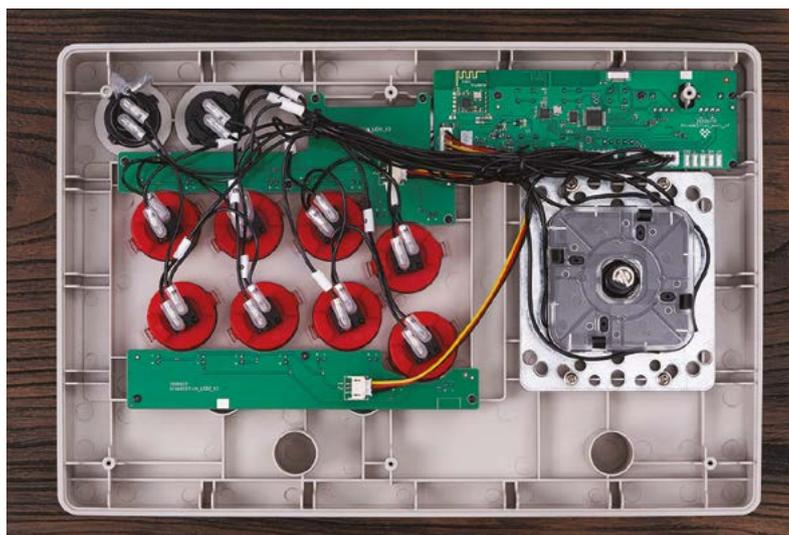
► The buttons use a tried and tested layout, while the switches are unique to the way the stick works

is via Bluetooth, it does mean you get a wireless connection without having to use a dongle. Speaking of dongles, there's a little RF receiver tucked behind a door on the rear of the stick that come pre-paired. Plugging it into a Raspberry Pi

“ It feels weighty and sturdy, and doesn't slip easily if you plan to place it on a table instead of your lap ”

and flicking over to 2.4G allows you to immediately use it wirelessly without any Bluetooth hassle, and it works really well. It also comes a USB-C cable which can be stored before the same door, which also houses the wires connection – just in case you need to perform that frame-perfect parry of Chun-Li's Houyoku Sen without wireless lag.

It feels weighty and sturdy, and doesn't slip easily if you plan to place it on a table instead of your lap. Like a lot of modern arcade sticks, you can also customise and swap out the buttons and joystick. The included square-gated joystick and buttons are fine, but feel a little loose for our liking, and if we were going to start using it in a professional setting, we'd definitely replace the



internals with some high-end Sanwa parts for more satisfying presses.

For the price though, you're getting a very flexible, and still very good, arcade stick. Sadly, like the Pro 2 controller we reviewed last issue, the macro buttons cannot be programmed on Raspberry Pi. However, there is a turbo button that you can use at any point, so there is some minor customisation you can do.

Overall, we love it. It feels like a quality piece of kit, and performs as such too. We're also keen on the use of switches rather than cheat code-esque button combos to change functionality, like in the last generation of 8BitDo products. Now, excuse us as we go take down Shadaloo. **M**

▲ Like all good modern fight or arcade sticks, you can easily swap out and mod the buttons and stick

◀ The included RF dongle makes it very easy to plug and play remotely



Verdict

An exceptionally solid arcade stick let down by some mildly fuzzy buttons, which will admittedly only bother a minority of players.

9/10

Weather HAT + Weather Sensors Kit

SPECS

SENSORS:

Wind speed, wind direction; BME280 rainfall, temperature, pressure, humidity; LTR-559 light sensor

FEATURES:

1.54-inch colour LCD, 4 × push-buttons, 2 × RJ11 sockets, on-board Nuvoton microcontroller with 12-bit ADC

EXPANSION:

4 × analogue inputs, I2C header

► Pimoroni ► magpi.cc/weatherhat ► £83 / \$94

Come rain or shine, you'll know all the details with this fully-fledged weather station. By **Phil King**

We've seen various Raspberry Pi DIY weather station projects and commercial kits before, but the Weather HAT makes setup a whole lot easier.

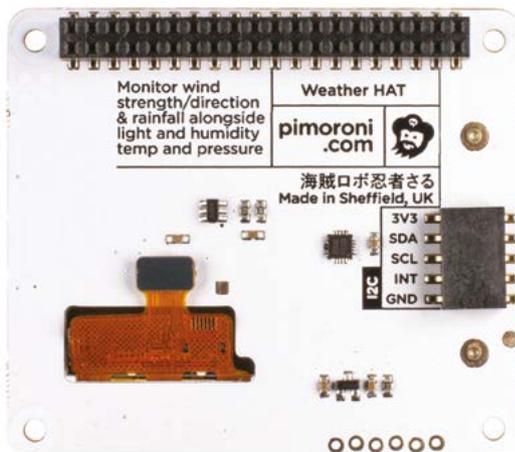
For starters, the board features standard telecoms-style RJ11 ports that enable you to just plug in the connectors from the wind speed, wind direction (vane), and rainfall sensors. Similar to the ones formerly available from Maplin stores, these three plastic sensors are provided in the full Weather HAT kit – or, if you already have some, you can buy the Weather HAT on its own for £36 / \$41.

Secondly, the Weather HAT has an on-board Nuvoton microcontroller with a 12-bit ADC to read analogue signals from the sensors reliably. A nice bonus is that four extra analogue input channels are broken out in an unpopulated header on the bottom edge of the board (along with 3V and GND), so you could add extra sensors. Not only that, but there's an I2C header on the underside of the board.

Whether you'll need extra sensors is a moot point: the HAT already incorporates a standard BME280 temperature, pressure, humidity sensor and an LTR-559 light sensor – as featured on Pimoroni's earlier Enviro boards. A 1.54-inch colour LCD screen completes the package, with four tiny push-buttons around it.

Station setup

Assembling the weather station takes only around 15 to 20 minutes. The wind direction and speed sensors are screwed onto either end of a sturdy plastic arm that fits into the top of a two-section hollow metal mast. A shorter arm is clamped to the mast and holds the rain sensor. We then planted the metal mast into some garden soil, but clamps



► The rear of the Weather HAT. It will fit any 40-pin Raspberry Pi model and also features an I2C header for extra inputs



► It's very easy to hook your weather station up to an Adafruit IO web dashboard to log all the data there

are also supplied in case you want to secure it to a drainpipe or something similar.

As mentioned, the sensors' RJ11 connectors simply plug into the labelled sockets on the Weather HAT. If you're wondering why there are only two sockets, this is because the wind speed sensor's connector fits into a socket on the wind vane, and then wires from both are routed to one RJ11 connector into the HAT.

The cables are approximately 3 m long, so unless you use extenders, your Raspberry Pi needs to be located fairly close by. Naturally, if this is outside, you should place it in a weatherproof box (not supplied). You may also need a mains power extension. Alternatively, you could place

“ The sensors' RJ11 connectors simply plug into the labelled sockets on the Weather HAT ”

your Raspberry Pi indoors, although you then wouldn't get accurate outdoor temperature and humidity readings.

Software library

As usual from Pimoroni, there's a fully-fledged Python library for the Weather HAT, complete with code examples. The main **weather.py** example shows readings from the sensors on the mini



▲ The **weather.py** Python code example shows live data from the sensors on the LCD



◀ The Weather Sensors Kit comprises wind speed, wind direction, and rain sensors, mounted on a metal mast

LCD, with push-buttons used to switch between individual sensor readings and between a digital readout or bar graph. We particularly liked the wind display, with an arrow rotating to show the direction, its size indicating the speed. Note that you may well need to use a compass and rotate the metal mast so that the wind vane is oriented to obtain accurate direction readings.

In the long term, you will want to keep a log of your data and view it on a web dashboard, which is where the Adafruit IO code example comes in. Just sign up for a free account at **adafruit.com** and you can build your own custom dashboard from preset blocks such as line graphs and gauges, linking them to the already created feeds coming from the Weather HAT. Within minutes we had our own weather web dashboard set up. **M**



▲ Along with on-board sensors and a mini LCD to show data, the Weather HAT has inputs for wind and rain sensors, plus a breakout header

Verdict

Packed with all the features and expandability you could need, this comprehensive kit is very good value for money and makes it a breeze to set up your own weather station.

9/10

Tiny 2040

SPECS

CONNECTORS:

16 × pins including 12 × GPIO, 3-pin debug header, USB-C port

RAM / STORAGE:

264kB SRAM, 2MB or 8MB QSPI flash

FEATURES:

Reset and boot select buttons, RGB LED, 4 × 12-bit ADC channels, PIO, I2C, SPI, UART

DIMENSIONS:

22.9 × 18.2 × 6mm

► Pimoroni ► magpi.cc/tiny2040 ► From £7 / \$7

A truly tiny RP2040-powered microcontroller board. By **Phil King**

Raspberry Pi Pico is only the size of a stick of gum, but if you need an even smaller microcontroller, the Tiny 2040 could fit the bill. Roughly the dimensions of a regular UK postage stamp, this little board is about 40% the length of a Pico, but packs the same powerful RP2040 system-on-chip.

“ The Tiny 2040 adds a very welcome on-board reset button, which saves repeatedly unplugging it ”

You can flash it with the Pico firmware and program it with MicroPython, C/C++, or CircuitPython. So it's very versatile and easy to get started with.

To achieve a smaller footprint, a few compromises have been made. Most notably, the Tiny 2040 only has a total of 16 pins versus Pico's

40, plus a debug header. Of these, there are 12 GPIO pins, compared to 26 on Pico. So it won't be compatible with most Pico add-on modules. As with Pico, there's no wireless connectivity.

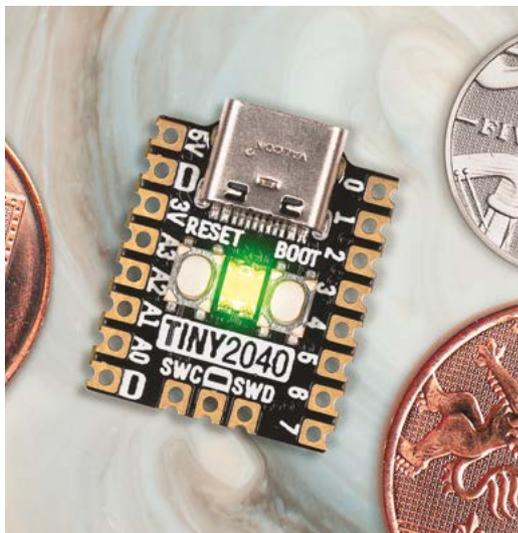
On the plus side, the Tiny 2040 does break out a fourth ADC input – which is connected to an on-board temperature sensor on Pico.

Tiny choices

You can buy the Tiny 2040 with or without (2.54 mm pitch) header pins attached. There's also a choice between 2MB of on-board QSPI flash storage (as on Pico) or 8MB. The Tiny 2040 is a fair bit pricier than Pico itself, which is only £3.60 / \$4.

On the plus side, the Tiny 2040 adds a very welcome on-board reset button, which saves repeatedly unplugging and reconnecting the board. There's also a programmable RGB LED (linked to three internal GPIO channels), rather than the single-colour green one on Pico.

Other than that, its advantages are mainly down to its diminutive size, making it suitable for wearable projects or even a tiny robot (e.g. magpi.cc/tiny2040robot). While not quite as minuscule as a Nionics Atto or Seeeduno XIAO, its RP2040 chip is a lot more powerful. ”



► “I can do anything you can do smaller!” The Tiny 2040 is a fraction of the size of a Pico



▲ The RP2040 SoC is located on the underside, which may make the board a little trickier to surface-mount

Verdict

Perfect for wearables and other portable projects, it packs the same RP2040 SoC as Pico but on a much smaller board, plus a few bonus features.

8/10

HackSpace

TECHNOLOGY IN YOUR HANDS

THE **MAGAZINE**
FOR THE **MODERN MAKER**



ISSUE #53
OUT NOW

hsmag.cc



Badger 2040

SPECS

SCREEN:

2.9-inch black and white e-ink display (296 × 128 pixels)

PROCESSOR:

RP2040 (dual Arm Cortex-M0+ running at up to 133MHz with 264kB of SRAM); 2MB of QSPI flash supporting XiP

I/O:

Five front user buttons; Reset and boot buttons; White LED USB-C connector for power and programming; JST-PH connector for attaching a battery (input range 2.7V – 6V)

Verdict

Badger 2040 is a fun accessory that integrates nicely with Raspberry Pi thanks to its RP2040 base. Simple to set up, but a lot of potential for integration with your projects.

8/10

► Pimoroni ► magpi.cc/badger2040 ► £12 / \$16

Raspberry Pi 2040-based e-ink badge and buttons make for an interesting device. By **Lucy Hattersley**

Raspberry Pi 2040 is the chip at the heart of Raspberry Pi Pico, and it's making its way across the tech ecosystem, powering a range of unique devices.

Badger 2040 is an RP2040 mounted on a 2.9-inch black and white e-ink display with five buttons.

The device comes fully assembled, so all you need to do is turn it on and load up the latest version of the software from Pimoroni's GitHub page (magpi.cc/pimoronipicoreleases). Installation is a case of holding down the BOOT button to mount RP2040 onto the desktop, and copying the UF2 file to the mounted storage.

There's an app

There is a clock app, ebook app (pre-loaded with *Wind in the Willows*), an image app, interactive list app, badge app, along with a QR code, info, and help information displays.

As with many projects, the fun begins when you start exploring what you can do with Badger 2040 in a coding environment. Pimoroni's documentation is, as typical, excellent, including a Getting Started with Badger 2040 guide (magpi.cc/getstartedbadger).



▲ Badger 2040 features an e-ink display and five navigation buttons



▲ On the rear of Badger 2040 sits the RP2040 chip along with BOOT and RST buttons and a white LED

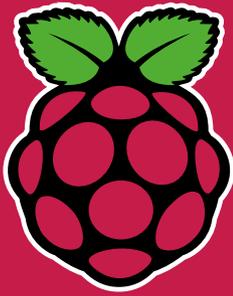
“ An optional accessory kit includes a AAA battery holder ”

The tutorials take you through writing ‘Hello Badger’ to the screen and customising the default apps. This is done by exploring a range of text files in Thonny (using View > Files). Images can be converted using Pillow and the `convert.py` file

An optional accessory kit includes a AAA battery holder and batteries, Velcro square, lanyard, and cable. You can also power Badger 2040 by connecting a lithium battery to the JST-PH connector.

Moving on from the default Badger OS and test projects, Badger 2040 has a Qwiic/STEMMA QT port for connecting breakouts with a JST-SH cable and STEMMA QT adapter (magpi.cc/stemmaqt). With this, you can explore integration with a variety of sensors, breakout boards, and accessories. ”

THE *Official*

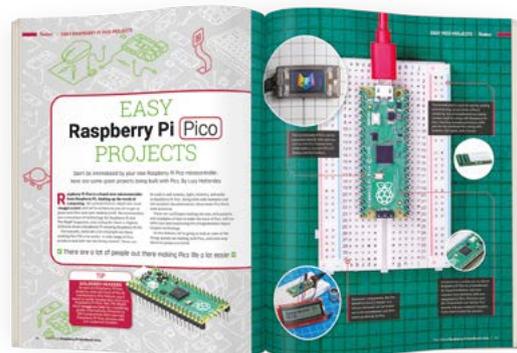
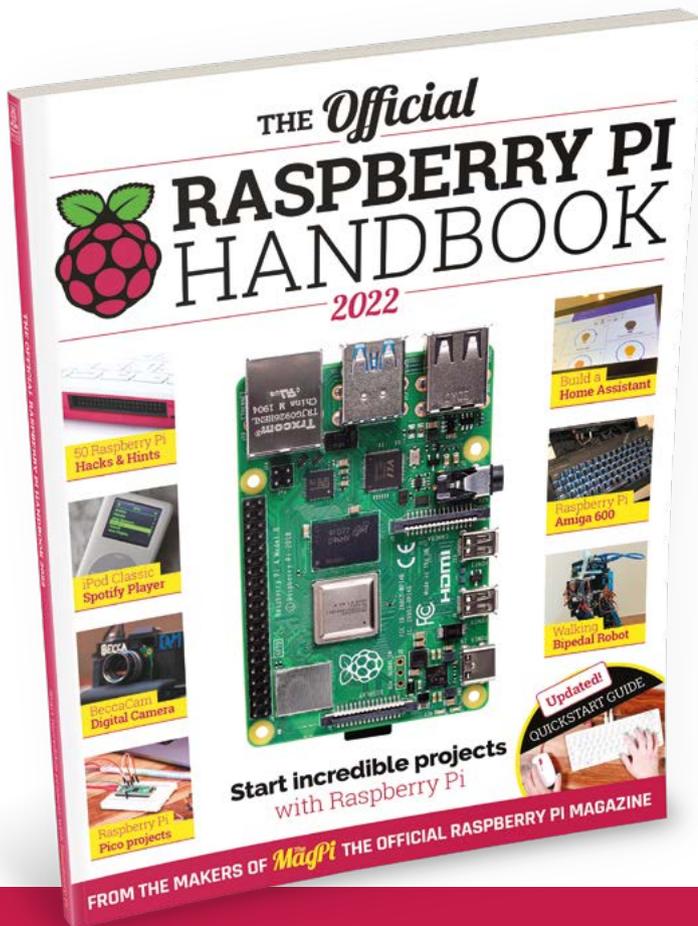


RASPBERRY PI HANDBOOK

2022

200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects

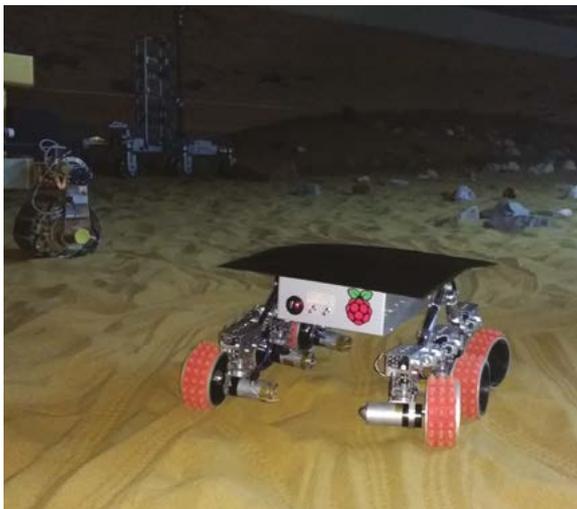


Buy online: magpi.cc/store

10 Amazing: Robot projects

Create an automaton
with Raspberry Pi

Robotics and Raspberry Pi have gone hand in hand for years, with projects, kits, racing tournaments, and even battles coming from the marriage of microcomputers and mechanics. There's never been a better time to try and build one yourself, so here are ten amazing projects you can try! [\[1\]](#)



▲ Yuri 3

Rover doppelganger

This six-wheeled robot takes inspiration from the ExoMars rover, a future rover planned to be sent to Mars at the end of 2022. This version stays on Earth though.

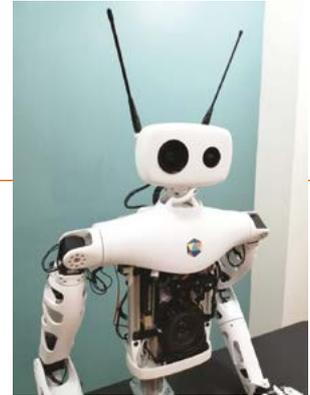
magpi.cc/82

► Reachy

Torso robot

This amazing-looking robot is very interactive and can play games and talk with its owners. It's also all powered by a Raspberry Pi 4.

magpi.cc/reachy



▲ DoodleBorg

Giant towing robot

From the incredible folk at PiBorg, this monstrous robot is powerful enough to tow caravans – as you may have guessed from the trailer hitch on top.

magpi.cc/doodleborg



▲ Build a robot buggy

Your first robot

A great project for building your very first robot, created by the Raspberry Pi Foundation. It's simple and easy to follow.

magpi.cc/robobuggy

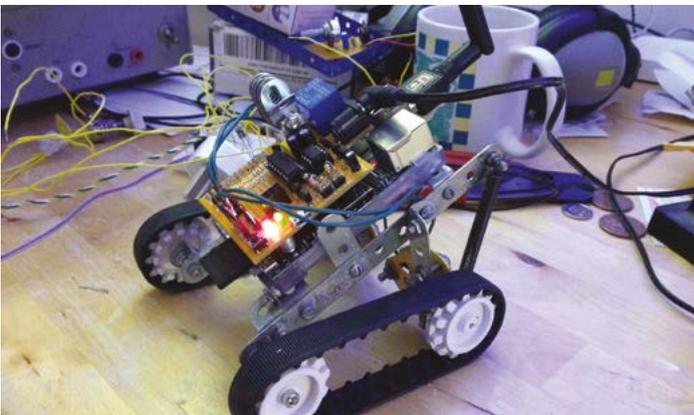


▲ C-Turtle

Expendable mine sweeper

This cardboard-based robot is cheap, quick to make, and easily replaceable if it has to take a trip to circuit heaven after completing its explosive job.

magpi.cc/cturtle



▲ Ping Pong Pursuit

Tidy up time

This robot uses a lot of 3D printed parts and a camera to clean up ping pong balls in the 37signals office in Chicago.

magpi.cc/pingpongbot

▶ PARSLER

Earth rover

This happy robot was made by Dr Jamie Molaro, a NASA scientist, and it includes a seismometer to record earthquakes and other seismic activity.

magpi.cc/83



▲ PiMowBot

Automated green keeping

Take the concept of a Roomba robot vacuum cleaner and add some powerful spinning blades to its underside, and you get PiMowBot – although there's a lot more to it than that.

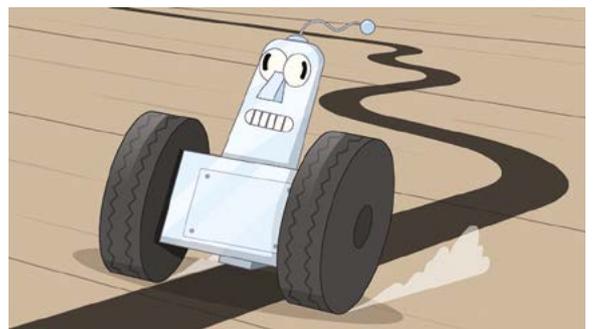
magpi.cc/pimowbot

▶ Project Zed

Intimidating citizen encouragement

This upcycled robotic wonder was built to help aid people when human contact might not be a good idea. It uses machine learning to give it a bit of a personality.

magpi.cc/projectzed



▶ Build a line-following robot

Simple robot project

This project from the Raspberry Pi Foundation is a great follow-up to the 'build a robot buggy' project, which lets you add some automation to the buggy.

magpi.cc/linefollow

Learn Java with Raspberry Pi

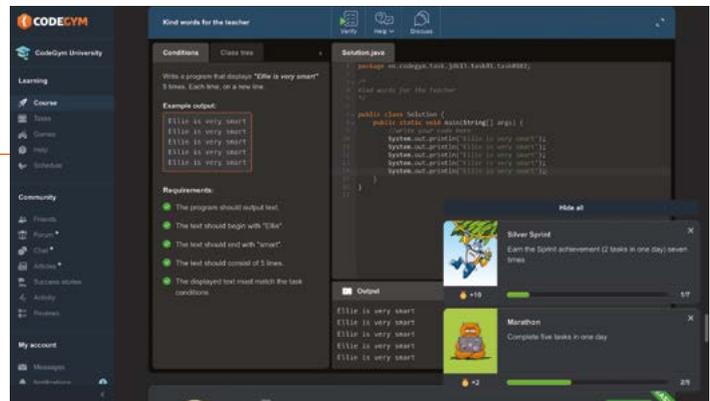
Start coding in Java on Raspberry Pi with these resources. By **Phil King**

CodeGym

CREATOR CodeGym
 Price: Free trial / \$50 pcm
codegym.cc

Created by Sun Microsystems in 1995, Java is now one of the world's most popular programming languages. Object-oriented, it's highly versatile and fairly easy to learn. You can code with it on Raspberry Pi using Visual Studio Code (or another suitable IDE) to create applets that will run on any machine with Java Runtime Environment installed.

One of the best ways to start learning Java is to get coding.



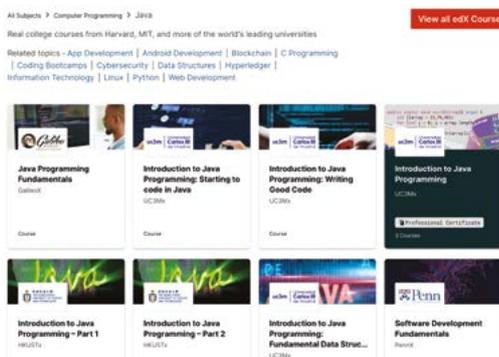
CodeGym is a practical, gamified course that involves solving 1200+ hands-on tasks of increasing complexity in a web IDE. At the start, you're asked which learning style you prefer and whether you have any previous Java experience – a mini quiz determines your current level. 'Dark matter'

earned from completing lessons and tasks is used to unlock the next ones.

While the first level of lessons is free, you'll need a paid subscription (or access via your educational establishment) to access the rest, but it's ideal for beginners. **M**

Free online courses

Study Java with these free courses



EDX JAVA

On the edX site, you'll find a wide range of Java-based courses from top educational institutions – from introductory guides and fundamentals to more specific topics, such as data structures and algorithms.

► magpi.cc/edxjava

INTRODUCTION TO JAVA

Created by Duke University, this beginner's course is available for free on Codecademy. As usual,

a web IDE lets you write and run code, and the course covers all the fundamentals of Java.

► magpi.cc/codecademyjava

JAVA MULTITHREADING

Uncover the mysteries of multithreading – executing two or more threads simultaneously on the CPU – with this three-hour video course aimed at coders with existing Java experience.

► magpi.cc/javamulti

Java: A Beginner's Guide

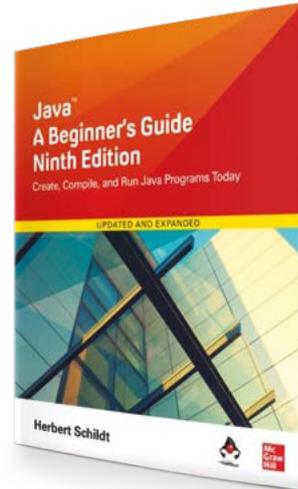
CREATOR

Herbert Schildt

Price:
£31 / \$40
[magpi.cc/
javabeginners](http://magpi.cc/javabeginners)

For those who prefer a textbook to help get started with Java, this is one of the best. Now in its ninth edition, it guides you step-by-step through the learning process. Throughout, there are self-tests and projects to try.

Starting with the fundamentals of Java and object-oriented programming, the book introduces the reader to data types and operators, program control statements (such as loops and conditionals), and then classes, objects, and methods. After elaborating further on these, it moves on to cover inheritance, packages and interfaces, exception handling, and I/O. Advanced topics include multithreaded programming,



generics, lambda expressions, and modules. The final chapter introduces the Swing GUI toolkit for Java.

It should certainly give you a good grounding, after which you may want to check out the author's *Java: The Complete Reference*. 

Reading material

Books for learning about Java



HEAD FIRST JAVA

Written in an entertaining style, this beginner's book offers a seriously good introduction to object-oriented programming and Java. Short stories and puzzles help to keep you engaged.

► magpi.cc/headfirstjava

EFFECTIVE JAVA

Aimed at experienced Java developers seeking to improve their code, it explains 90 best practices to follow. These are organised into sections such as methods, generics, concurrency, and serialisation.

► magpi.cc/effectivejava

JAVA PUZZLERS

Even expert Java coders can get tripped up by oddities of the language. This book addresses common traps, pitfalls, and corner cases via a series of 95 puzzles to educate and entertain.

► javapuzzlers.com

Coderanch

CREATOR

Coderanch

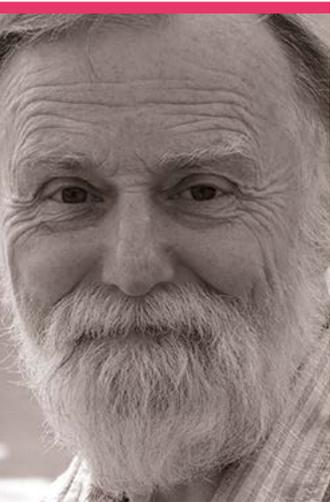
Price:
Free
coderanch.com

Whenever you have a problem with your Java code, getting some advice from more experienced programmers and developers is invaluable. As with other programming languages, there are numerous online forums to help you out,

including the trusty Stack Overflow and the official Oracle developers community. One of the most beginner-friendly forums can be found at the Coderanch. Despite the Wild West theme, it's a hospitable place for newcomers, moderated by volunteer marshals and sheriffs. There's even a Beginning Java subforum to help you get established.

In addition, among the wide variety of topics covered on the site are book reviews, career advice, mobile, databases, caching, build tools, and frameworks. Hit the FAQs button at the top left to view a range of useful wiki articles. 





Mike Cook

Veteran maker and previous writer of our Raspberry Pi Bakery series, here's the story of Mike Cook

- > Name **Mike Cook** | > Occupation **Retired journalist**
- > Community role **Maker** | > URL **magpi.cc/mikecook**

For a very long time, you could find a wonderful tutorial from Mike Cook and his Pi Bakery right here in the pages of *The MagPi*. From dancing skeletons to handmade MIDI guitars, he showed readers how to create an incredible variety of projects. He's well-known to older readers from his regular columns in *The Micro User* and *Acorn User* in the 1980s.

"I am addicted to making stuff," Mike tells us. "Now retired, but I was a university lecturer for 21 years and an electronic engineer in a mass-production consumer electronics company for the rest of my career."

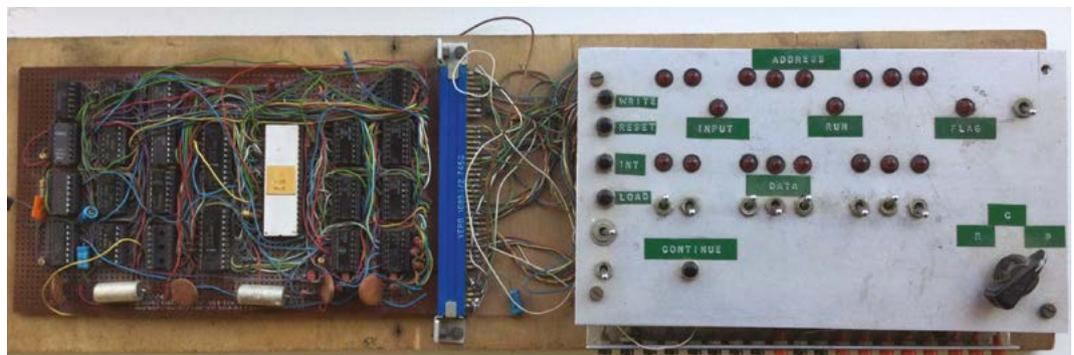
What is your history with electronics and programming?

My first electronic kit was called Trans Tronic. I got it when I was nine, and was making and selling crystal sets to my classmates when I was 13. Left my secondary modern school at 16, although 14 was the minimum school leaving age then. I went to work for a local industrial electronics company. There I did day release and night school for an ONC qualification. This allowed me to attend Newcastle Polytechnic, where I took a joint degree in Physics/Electronics. This is where I had my only formal education about computers. I had three one-hour lectures on programming

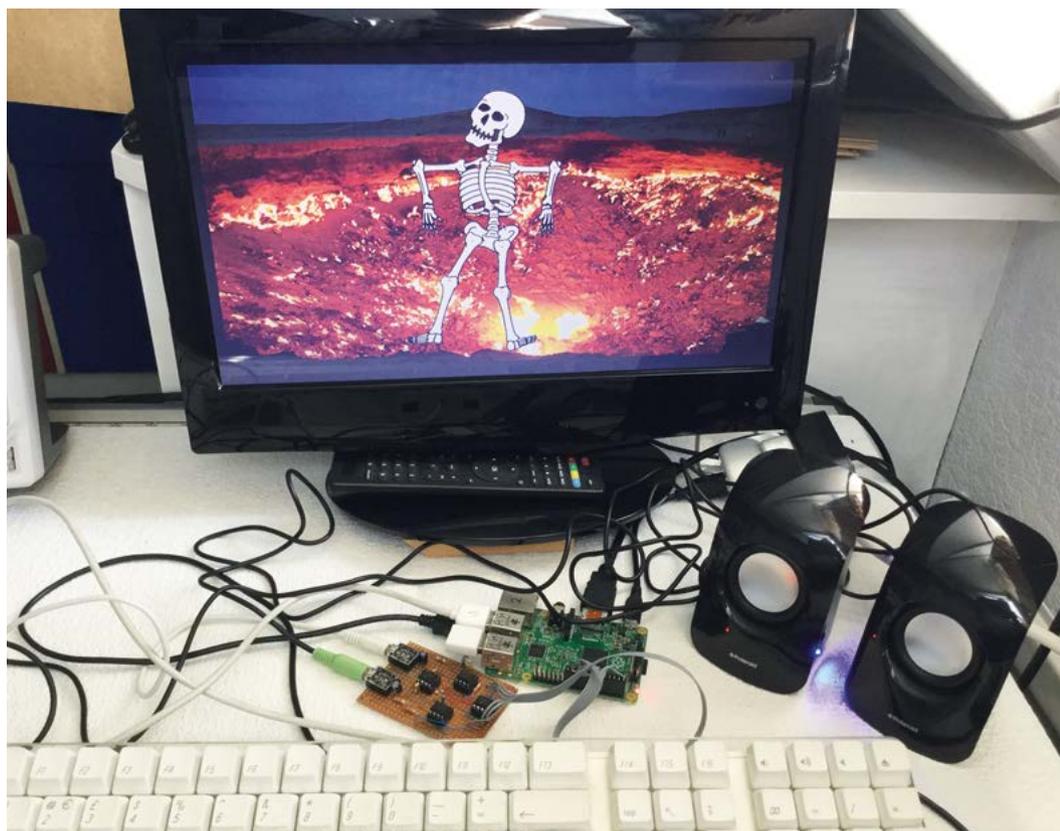
with Fortran IV, but that was enough to get me hooked.

When did you start writing about it?

After I made my own computer in 1976, from just a datasheet of the processor, and no idea what software did in them. My first published article on computing was in *73 Magazine*, a ham radio magazine from the USA. It was about making music with a microcomputer. Later, I went on to write about the BBC computer in *The Micro User*, and after that in *Acorn User*. My first publication was in *The MagPi* #5, but I didn't start writing my Pi Bakery series until *The MagPi* #33 (magpi.cc/33).



> The first computer Mike ever made in 1976, just from a datasheet



◀ Dancing Skeletons is very much a favourite at *The MagPi*

When did you learn about Raspberry Pi?

From a blog post online about six months before it first came out. I wasn't quick enough to get one of the first batch, although I did apply for, and got, a second-batch machine.

What was the first thing you made with a Raspberry Pi?

I adapted a project I had done in *The Micro User* for a computer-controlled glockenspiel.

What is your favourite thing you've made?

What, just one? Too hard, you get eight on *Desert Island Discs*. Well, I will give you my top three.

Dancing Skeletons – *The MagPi* #50 (magpi.cc/50), Tap-n-LED, – *The MagPi* #97 (magpi.cc/97), and Pico Voice – *The MagPi* #106 (magpi.cc/106).

What other hobbies have inspired you?

Astronomy. I have a 10"

Cassegrain telescope. I even lectured in it as part of the physics courses at the university. Speleology, although it is some years since my wetsuit actually fitted me. Photography, analogue stuff with trays of chemicals. I was a professional wedding photographer for three years in the late 1960s. Also, amateur radio, live music gigs,

stained glass work, and hill walking.

Anything else you'd like to add?

I just love helping people with their projects. There was no 'online' when I started, and so I had to work it all out myself. It is good to give people a helping hand. **M**

▼ Mike's first Raspberry Pi build was a recreation of an older glockenspiel project



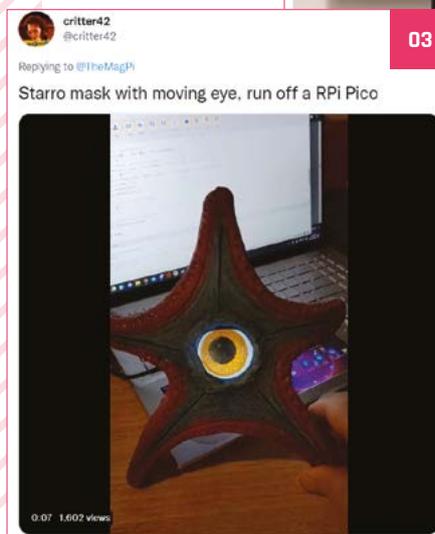
MagPi Monday

Amazing projects direct from our Twitter!

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.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!!

01. Pico and micro:bit working together? Not as strange as you think.
02. This is one way to appreciate every shot from *Raiders*.
03. We think it's a bad idea to create the villain that helped form the Justice League, but it is cool.
04. What a cool and impressive use of computer vision!
05. This is an impressive first project! A nice robot to test with.
06. A very cute musical project with cool lights.
07. Kevin's cluster computer has been getting bigger and better.
08. This wall control is very smart and full of cool functions.
09. Cute cat robot that helps you learn to code? Incredible.
10. We wonder if this can also power a Raspberry Pi.
11. Pi Wars continues in Puerto Rico.



04

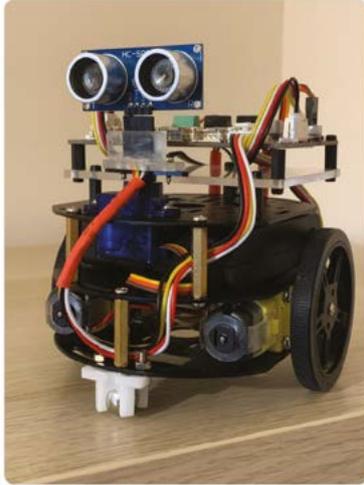
Brian Starkey @usedbytes
 Replying to @TheMagPi
 I got my Pico-powered robot following a coloured blob using an OV7670 camera and PIO 📷🤖

Brian Starkey @usedbytes - Jan 23
 After an extremely frustrating week of debugging electrical issues, this is intensely satisfying 🥳 Everything being done by a @Raspberry_Pi Pico (which is a beast, "find the red blob" takes ~200us) #PiWars



05

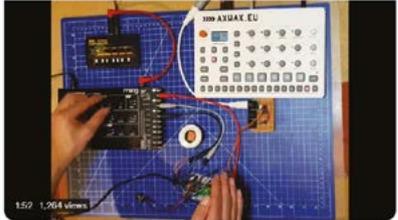
Spike Tennyson @spiketennyson
 Replying to @TheMagPi
 First pi project, based on Maker RP2040 board.



06

AsMax @AsWax
 Replying to @TheMagPi
 I worked on improving last week's Pico MIDI to CV interface by adding a calibration pot, a second CV output, an LDR to control it and some NeoPixels to show the currently playing note. And I made a tune to celebrate my favourite microcontroller's birthday.

AsMax @AsWax - Jan 21
 Replying to @Raspberry_Pi
 Happy First Birthday @Raspberry_Pi Pico! 🎉 To celebrate I added a second DAC, an LDR and some blinkenlights to my Pico MIDI to CV converter and jammed a little birthday ditty to show them off. Hope you like it! 🎵🎸 #RaspberryPiPico #RP2040



07

Kevin McAleer @kevsmac
 Replying to @TheMagPi
 This weekend I was at the @raspberrypi exhibit launch, but I did manage to take this photo of the Clustered-Pi (12 raspberry pi zero 2w's) on the Cray-1 that inspired it! #raspberrypi #MagPiMonday #retro #retrocomputing clustered-pi.com



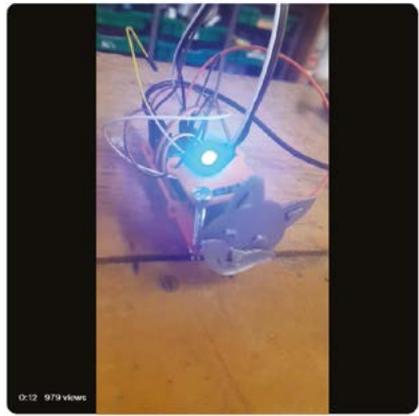
08

MrTony007 @MrTony007
 Replying to @TheMagPi
 Wall terminal built with RPi, official touchscreen, pir motion sensor, camera, RFID sensor, fingerprint scanner and speaker. Placed in custom 3D printed enclosure. Using it to manage the alarm in particular.



09

Tom Faux (Vulpestruments) @Vulpestruments
 Replying to @TheMagPi
 Mini Pi Pico companion bots for my year 8 classes. Teaching them how to code + connect a servo and neopixel then 3D design and print a face for it.



10

Cam Davidson-Pilon @Cmnn_DP
 Replying to @TheMagPi
 Building a bioreactor on top of a Raspberry Pi!



Pioreactor

11

Alex @geomarstar
 Replying to @TheMagPi
 Yes! Getting ready for PiWars



Dr Footleg - Roboteer
@drfootleg

12

Replying to @TheMagPi

I got the first track running for my next Raspberry Pi robot. Printed on my dual material 3D printer in PLA with TPU treads. #MagPiMonday

Dr Footleg - Roboteer @drfootleg · Jan 29

This is going to be an absolute beast with these motors! So much torque I can't stop them with my hands. Fingers will need to be kept clear when this is running.

Show this thread



0:06 1,222 views

Raul Portales
@sla_shalafi

13

Replying to @TheMagPi

Just a simple retro gaming system with a carry bag. Loving the Nes3Pi case.



Zero Retries Newsletter
@ZeroRetries

15

Replying to @TheMagPi

Built from wb7fnc.com/intro.html



Andy Piper @andypiper · Jan 31

14

Replying to @TheMagPi

Spent some time with my GPI Case 2 (Pi Zero 2 W inside), does that count?

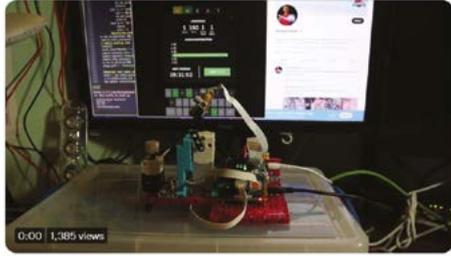


Richard Hayler ✨
@rghayler

16

Replying to @TheMagPi

I made a silly machine that uses a Sense HAT and Build HAT to tweet a photo of my daily #wordle result as displayed on the LED matrix.



0:00 1,385 views

Rincewind
@rincewind5013

17

Replying to @TheMagPi

Completed my Raspberry Pi server. Now using it to learn some simple electronic projects. #MagPiMonday



- 12. This robot could probably go up a near vertical slope.
- 13. Portable retro gaming at its finest.
- 14. A great way to upgrade your GPI.
- 15. This is a macro key, which we think is used for ham radio stuff.
- 16. A genius use of Sense HAT, Build HAT, and Wordle.
- 17. This 3D-printed case is a very cool idea.

Crowdfund **this!** Raspberry Pi projects you can crowdfund this month



Pico 87 Mechanical Keyboard

Customisable keyboards are pretty cool, and in this case you can even customise what every single key does, not just change the caps. There's also a detachable number pad and some USB ports connected to it.

► kck.st/3LT38tF



UHF HAT

Created by SB Components, the UHF HAT for Raspberry Pi is "an advanced and compact 'Ultra High Frequency' RFID reader that consists of powerful RFID technology [designed] for a broad range of applications..."

► kck.st/3AVq7zc

COMING SOON

pod ZERO 2 W POD CASE

MODULAR RASPBERRY PI ZERO 2 W CASE

POD MODULES

- ✓ USB-HDMI
- ✓ 2.8" INCH DISPLAY
- ✓ USB-LAN
- ✓ UPS + RTC
- ✓ ACCESS TO ALL PORTS AND GPIO



✓ PASSIVE COOLING

COOLER THAN EVER

ARGON ONE V3 NVMe

- ✓ NVMe BOARD
- ✓ 40mm FAN



ARGON ONE V3

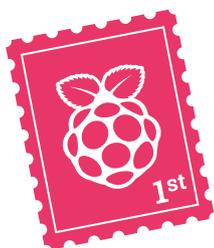
- ✓ 40mm FAN



COMPATIBLE WITH ARGON RTC AND UNINTERRUPTED POWER SUPPLY MODULE



Your Letters



First tutorials

I have been looking for some kind of guide or book or website for absolute beginners to Raspberry Pi as my teenagers have recently got into programming and got a Raspberry Pi for Christmas. Do you have any suggestions on issues they could look at, or other places?

Nes via email

We have a few books that may be of interest. The *Raspberry Pi Beginner's Guide 4th Edition* (magpi.cc/bgguide4) will guide you through getting started with a Raspberry Pi, and even has some basic projects to get on with.

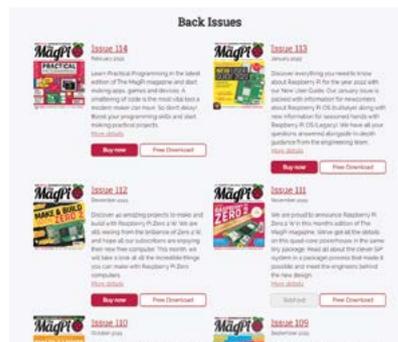
If they pick up things very quickly, there's always *The Official Raspberry Pi Handbook 2022* (magpi.cc/handbook2022), which has a quick getting started guide, as well as a load of tutorials to be inspired by and learn how to do.

There's also the For Home section at raspberrypi.com, which can help you out with loads of online resources.

▶ *The Official Raspberry Pi Handbook 2022* is a great way for those who pick up technical things quickly to get started with Raspberry Pi



▼ All our magazines are available for free as a PDF from our website



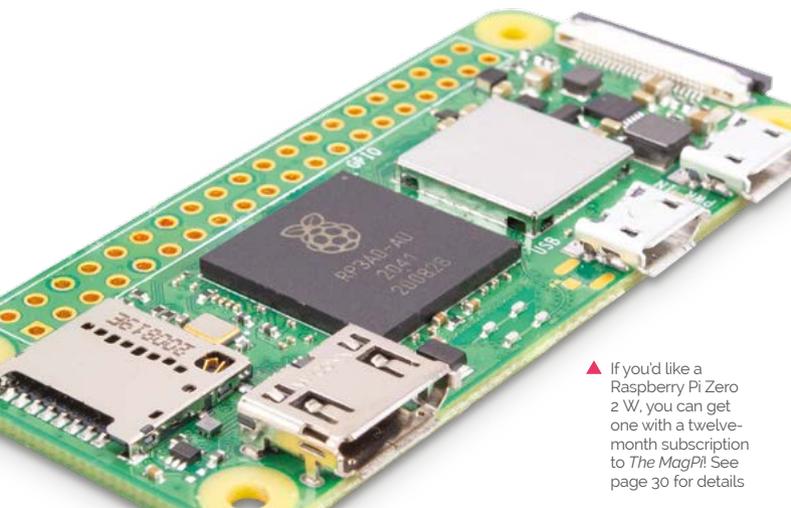
Read for free

Are you still doing free PDFs for your magazine? Do I need to sign up anywhere to get them?

Miles via email

All our books and issues are released as a free PDF on the same day as the physical copy of the magazine! You can always find them on the page for each issue at magpi.cc/issues, but we also provide a link in the newsletter on the same day so that you kinda get it delivered to you through there. You can sign up to our newsletter using our website: magpi.cc.

3 ISSUES FOR £5



▲ If you'd like a Raspberry Pi Zero 2 W, you can get one with a twelve-month subscription to *The MagPi*! See page 30 for details

Pi Day Pi

Is there a new Raspberry Pi coming out on Pi Day? I've read a few rumours that there may be a Raspberry Pi 5 that day!

Tina via Facebook

If you're reading this, no, it's long past 14 March, aka Pi Day due to it being 3/14 in the American date format, and as you may have seen, nothing was released. Raspberry Pi doesn't really announce items ahead of time so, even if it were true, we wouldn't be able to confirm it!

Contact us!

- Twitter [@TheMagPi](#)
- Facebook [magpi.cc/facebook](#)
- Email magpi@raspberrypi.com
- Online [forums.raspberrypi.com](#)



📞 Subscribe by phone:
01293 312193

📧 Subscribe online:
magpi.cc/subscribe

✉ Email: magpi@subscriptionhelpline.co.uk

Wireframe

Join us as we lift the lid
on video games



Visit wfmag.cc to learn more

WIN
1 OF 5

8BITDO PRO 2 CONTROLLERS

IN ASSOCIATION
WITH 8BITDO

We reviewed the excellent 8BitDo Pro 2 controller last issue and gave it a 9/10. It's one of the best retro game controllers out there, with a comfy D-pad and many advanced settings.



Head here to enter: magpi.cc/win | Learn more: magpi.cc/pro2

Terms & Conditions

Competition opens on **30 March 2022** and closes on **28 April 2022**. 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 The MagPi 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 or Facebook.



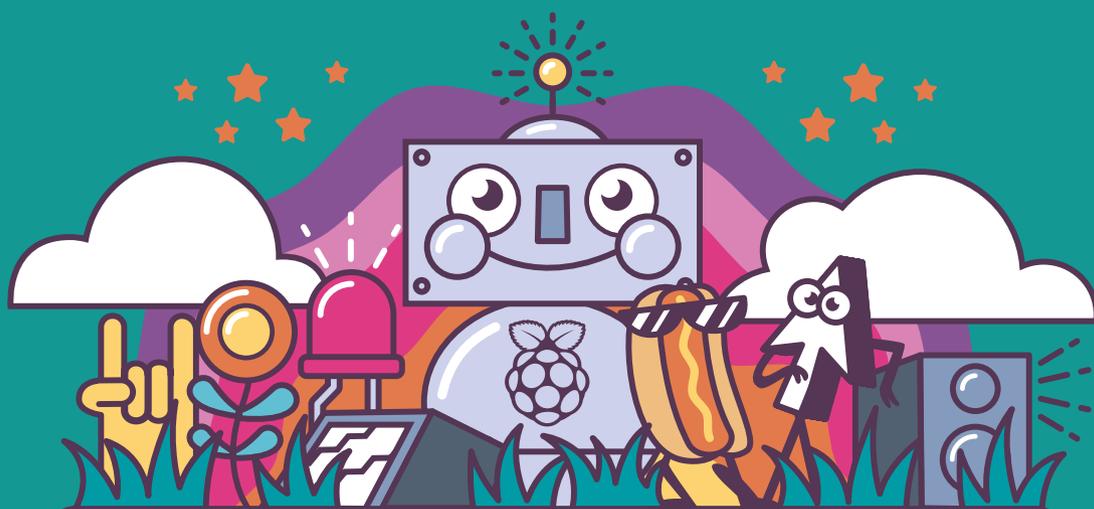
BUY ONE GET TWO FREE!



GO TO

magpi.cc/store

Offer applies to The MagPi magazine only. Latest issue not included in the promotion. Must add 3 copies of The MagPi magazine to the basket for the offer to be applied. Subject to availability, while stocks last.



FIELD WORK

TAKE RASPBERRY PI OUTSIDE



THE MAGPI #117
ON SALE 28 APRIL

Plus!

A digital TV transmitter

Create art with computers

Raspberry Pi audio special

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER @TheMagPi

FACEBOOK fb.com/MagPiMagazine

EMAIL magpi@raspberrypi.com

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Nicola King

ADVERTISING

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

DESIGN

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Olivia Mitchell, Sam Ribbits,
Ty Logan

Illustrator

Sam Alder

CONTRIBUTORS

David Crookes, Robert Hamilton-MacLaren, Rosemary Hattersley, Nicola King, Phil King, KG Orphanides, Stephen Smith

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Director of Communications

Liz Upton

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
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk



This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under

a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2051-9982.





Rubber Soul

Looking after the old and the new

We have an interesting relationship with nostalgia here at *The MagPi* magazine. We like to stay close friends with it, but don't move in with it and make ourselves at home. In case it gets the wrong idea.

Which is to say: while we keep one hand firmly on our retro joysticks, enjoying and loving the past, we have another hand pointing directly at the future and are saying, "Look! That's where we belong."

This makes sense. Raspberry Pi is ideal for rediscovering retro classics; its modern processor is capable of faithfully emulating just about every beloved computer of our generation. And like many computer geeks of our generation, we loved many different 8- and 16-bit computers.

My beloved computer of this month is the ZX Spectrum. Which is – and I can barely believe I'm saying this – 40 years old this April.

I mean, I know this must be true. It had '© 1982 Sinclair Research' on the screen every time I looked at it. But it can't be 40 years, can it!?

I first got a ZX Spectrum around 1984. A wonderful year (despite its infamous book namesake). I fell in love with prodding the rubber buttons and making things move around my small black and white TV screen. The BASIC programming language was – in retrospect –

absolutely basic. And the gulf between BASIC and Z80 machine code was depressingly huge.

And there's no getting away from the fact that computers were sold as toys, which meant they became heavily gendered. The Spectrum was pitched at boys and let's not forget the cliquy club with "no smelly girls in our treehouse." Thank goodness those days are over.

Still, I stuck with my beloved ZX

“ My beloved computer of this month is the ZX Spectrum ”

Spectrum. Even though its keys were squidgy and each button had four or five confusing commands printed on it. And the Spectrum broke so often that the factory, at one point, was rumoured to have stopped taking returns (because there physically wasn't enough space to store them). Mine broke five times before my Dad bought me a Commodore 64 out of frustration.

Honour the past

We've honoured the ZX Spectrum many, many times in *The MagPi* (and will continue to do so in the future). Two years ago, the ZX Spectrum Next project (magpi.cc/specnext) produced a modern

version of the classic computer. I bought one and I love it. It uses a Raspberry Pi Zero inside to provide sound support, enabling users to load tapes with sound, screeches, and all. Meanwhile, a modern FPGA (field-programmable gate array) reproduces the Z80 chip exactly.

If you've ever loved the ZX Spectrum, I wholeheartedly recommend you read (Wireframe editor) Ryan Lambie's

wonderful homage to Sinclair computing and the ZX Spectrum (magpi.cc/wireframespectrum).

Nostalgia has its place, and I defy any Spectrum fan to hear the loading screech and not have their heart melt. But Raspberry Pi and Raspberry Pi Pico are ultra-modern computer platforms remaking the modern computing landscape. As Eben says, "the future of computers is only \$4 away". That's the important thing! 📺

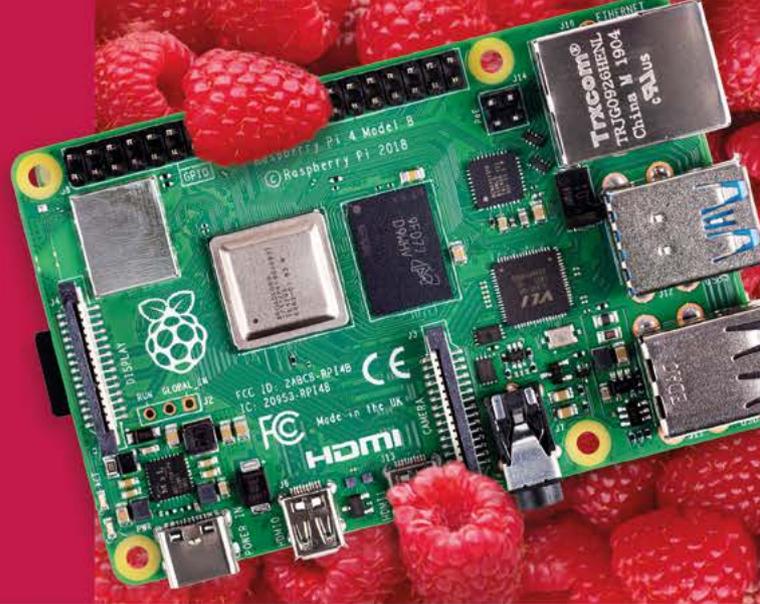
AUTHOR

Lucy Hattersley

Lucy is Editor of *The MagPi* and loves all her retro computers equally, even that Dragon 32 in junior school.

magpi.cc

American Raspberry Pi Shop



- Displays
- HATs
- Sensors
- Cases
- Arcade
- Swag
- Project Kits
- Cameras
- Power Options
- Add-on Boards
- Cables and Connectors
- GPIO and Prototyping

Partner and official reseller for top Pi brands:



and many
others!

Price, service, design,
and logistics support for
VOLUME PROJECTS

USA

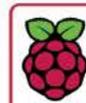


PiShop.us

Canada



PiShop.ca



Raspberry Pi

APPROVED RESELLER

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:



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