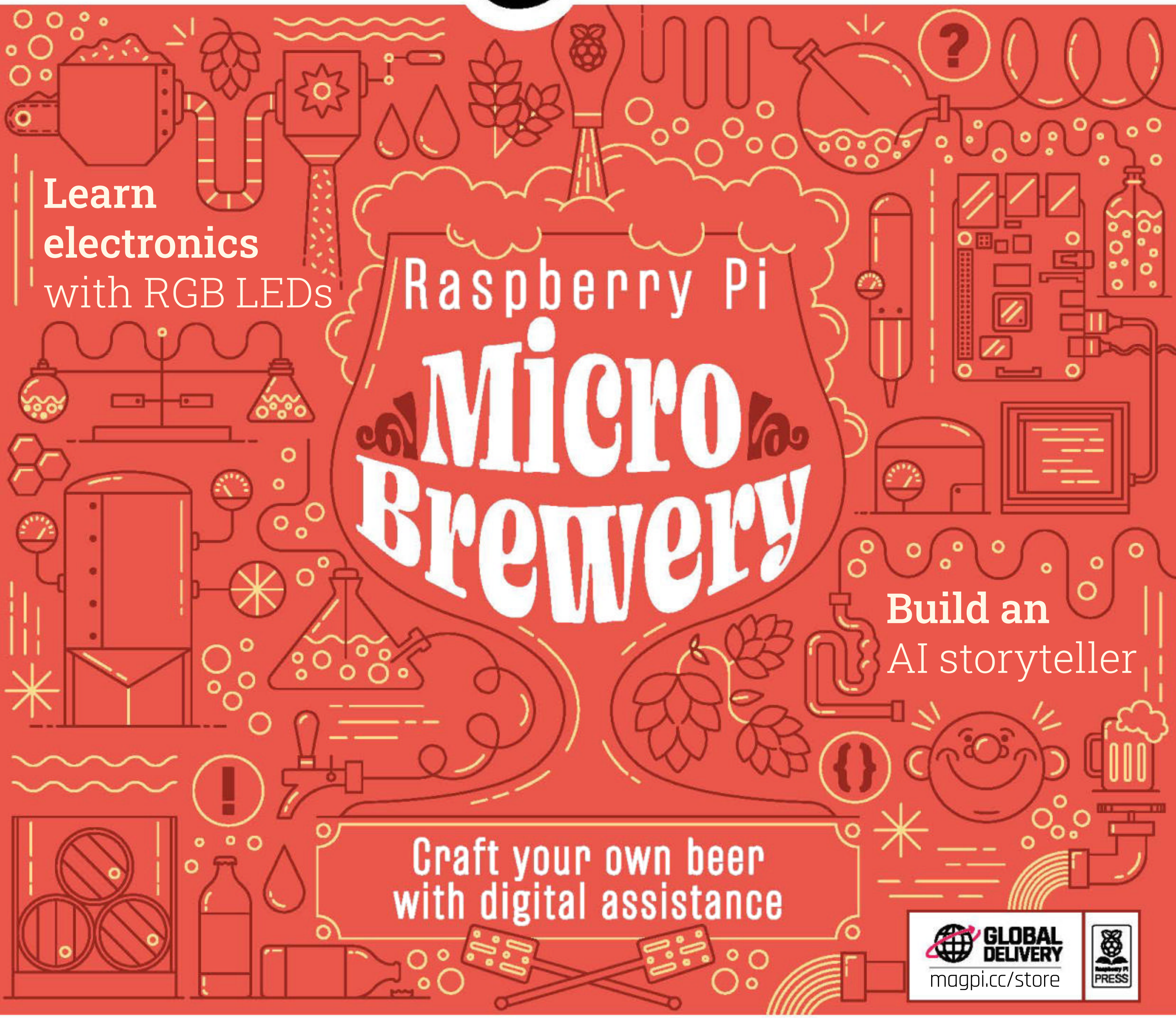
The MagPi

Issue 130 | June 2023 | magpi.cc

The official Raspberry Pi magazine



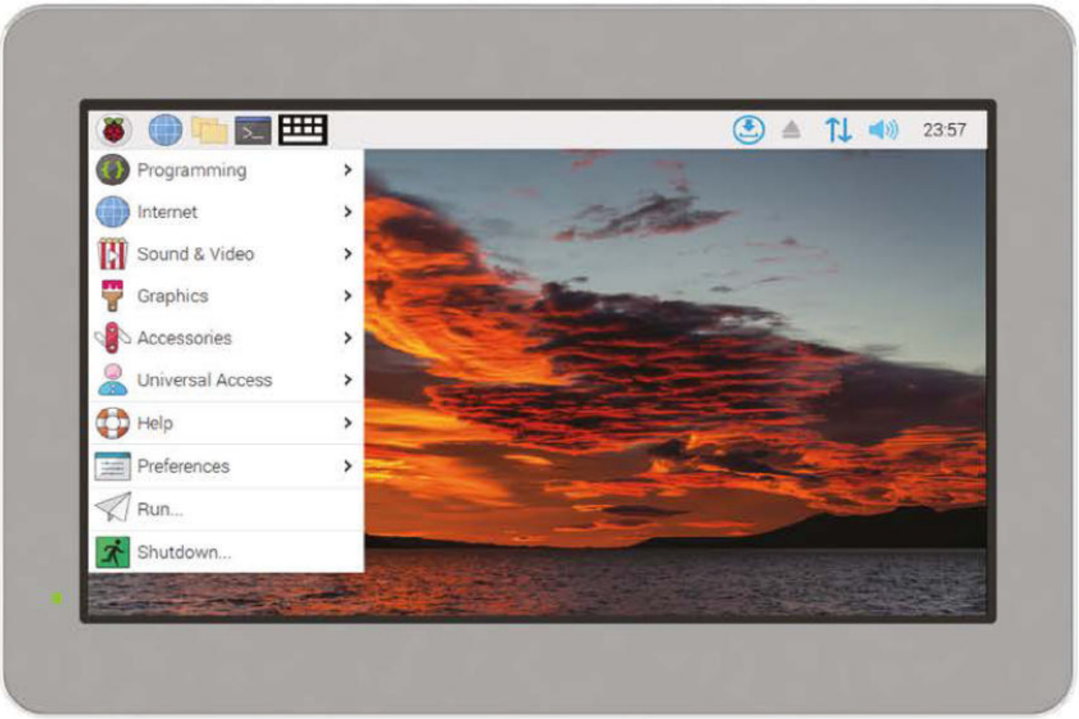
Learn electronics with RGB LEDs

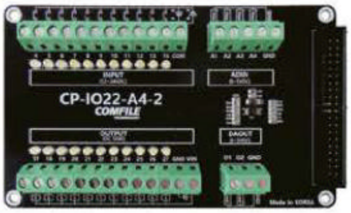
Raspberry Pi Micro Brewery

Craft your own beer with digital assistance

Build an AI storyteller

MASTER THE **LINUX COMMAND LINE INTERFACE**


Programming
Internet
Sound & Video
Graphics
Accessories
Universal Access
Help
Preferences
Run...
Shutdown...
23:57


CP-IO22-A4-2
COMFILE

WELCOME
to The MagPi 130

This month, for those who celebrate, is the Cambridge Beer Festival. The team at Raspberry Pi are all excited and ready to impart their craft-based brewing wisdom. Beer making may be an age-old art, but that doesn't mean it can't be brought into the modern age with a little Raspberry Pi smarts (**page 32**).

We get that not everybody drinks, let alone wants to brew their own beer, so our second feature this month is Master Linux & Command Line (**page 60**). Learning the command line changes how you use computers forever, and I really can't recommend it enough for all technology enthusiasts.

There are some amazing tutorials this month. Sean McManus has created Bedtime Stories (**page 40**) that uses ChatGPT and Stable Diffusion to create surreal dreamlike stories. Stewart Watkins explores electronics with NeoPixel LEDs (**page 46**), and David Plowman takes us on a tour of still image capture with how the Camera Module works **(page 50)**.

The MagPi is a labour of love, even if it's a beer-infused one. Actually, probably moreso this month than usual. Cheers!

Lucy Hattersley Editor

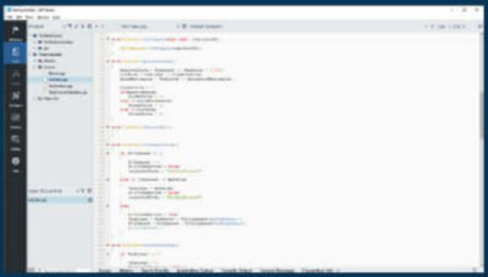
EDITOR

Lucy Hattersley

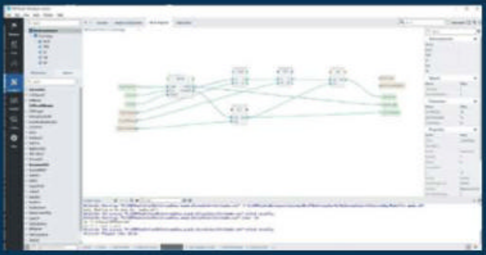
Lucy lives in 'that there' London and is currently dodging the coronation crowds in an art exhibition about milk. Never stop being random, my lovely city.

magpi.cc

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Learn electronics with RGB LEDs
Raspberry Pi Micro Brewery
Build an AI storyteller
Craft your own beer with digital assistance
MASTER THE LINUX COMMAND LINE INTERFACE

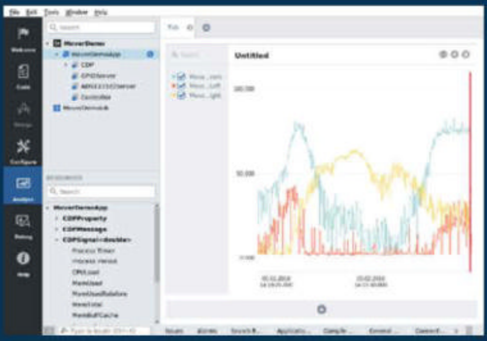


Full-Code



No-Code / Configure

Design



Analyze

Contents

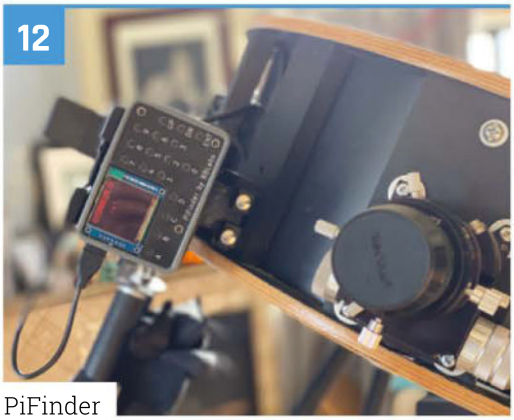
Issue 130 June 2023

Cover Feature

Regulars

Project Showcases

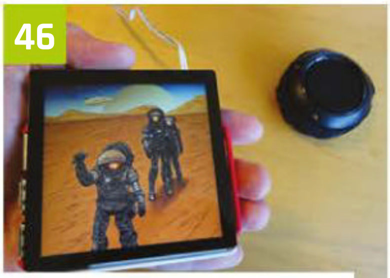
RAD Expansion Unit



PiFinder

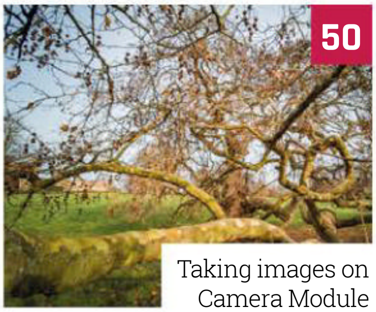
The MagPi magazine is published monthly by Raspberry Pi Ltd. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The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

Tutorials



46

AI-generated bedtime stories



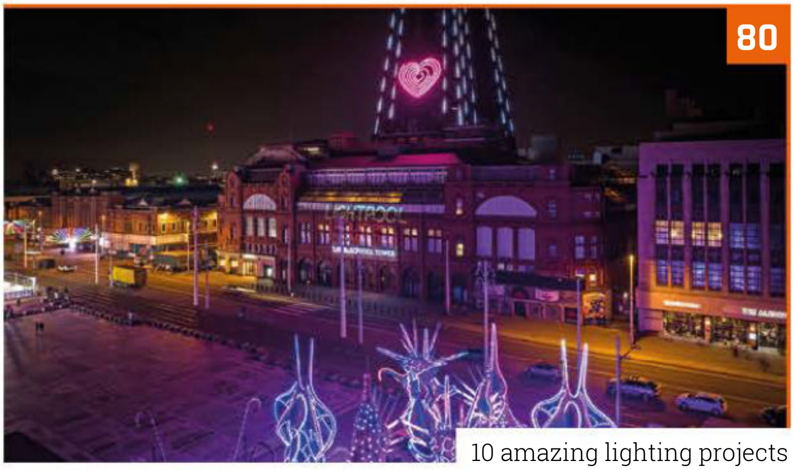
50

Taking images on Camera Module

The Big Feature

60

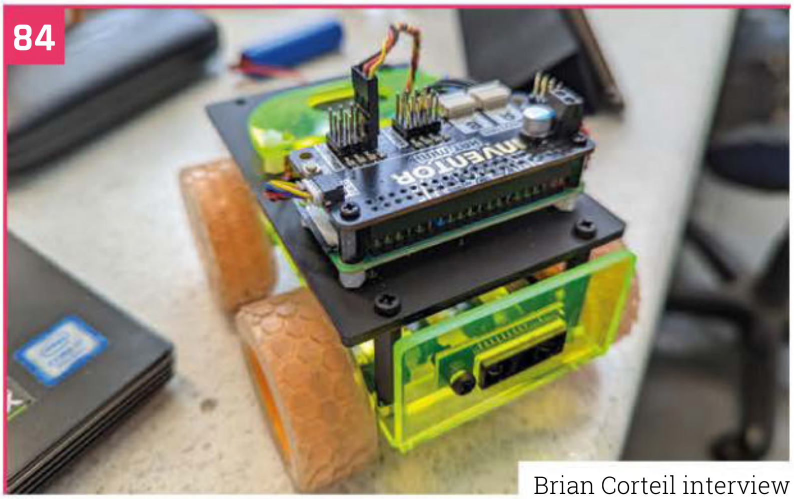
Master the Linux command line



80

10 amazing lighting projects

Reviews

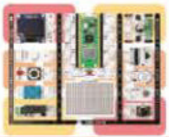


84

Brian Corteil interview

Community

WIN 1 OF 3 PICO BRICKS BASE KIT



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.

Love Machine ChatGPT compliment giver

Is flattery any less welcome when it's computer-generated? Kakapo Labs used Raspberry Pi Pico W and ChatGPT to find out. By **Rosie Hattersley**

MAKER

Will Lahr and India

Will and India work at multidisciplinary design agency Kakapo Labs, which Will set up as "an incubator for exploring new ideas."

magpi.cc/chatgptgumball

A regular fixture in the security software PR calendar is the annual survey that finds 70-plus percent of people quizzed by complete strangers at a train station are more than happy to hand over their essential passwords and memorable information in return for nothing more than a winning smile and perhaps a bar of chocolate.

Despite these alarmist news items gaining media coverage every single year for more than a decade, we punters just don't seem to wise up to such confidence tricks. Tech entrepreneurs Kakapo Labs set about a more benign riff, on the idea that Joe Public loves a bit of flattery, using clever verbal flummery generated by AI darling ChatGPT, creating a Love Machine that dishes out compliments and chocolates in equal measure (**magpi.cc/chatgptgumball**). So far, the gentle love-bombing experiment doesn't seem to have a hidden agenda.

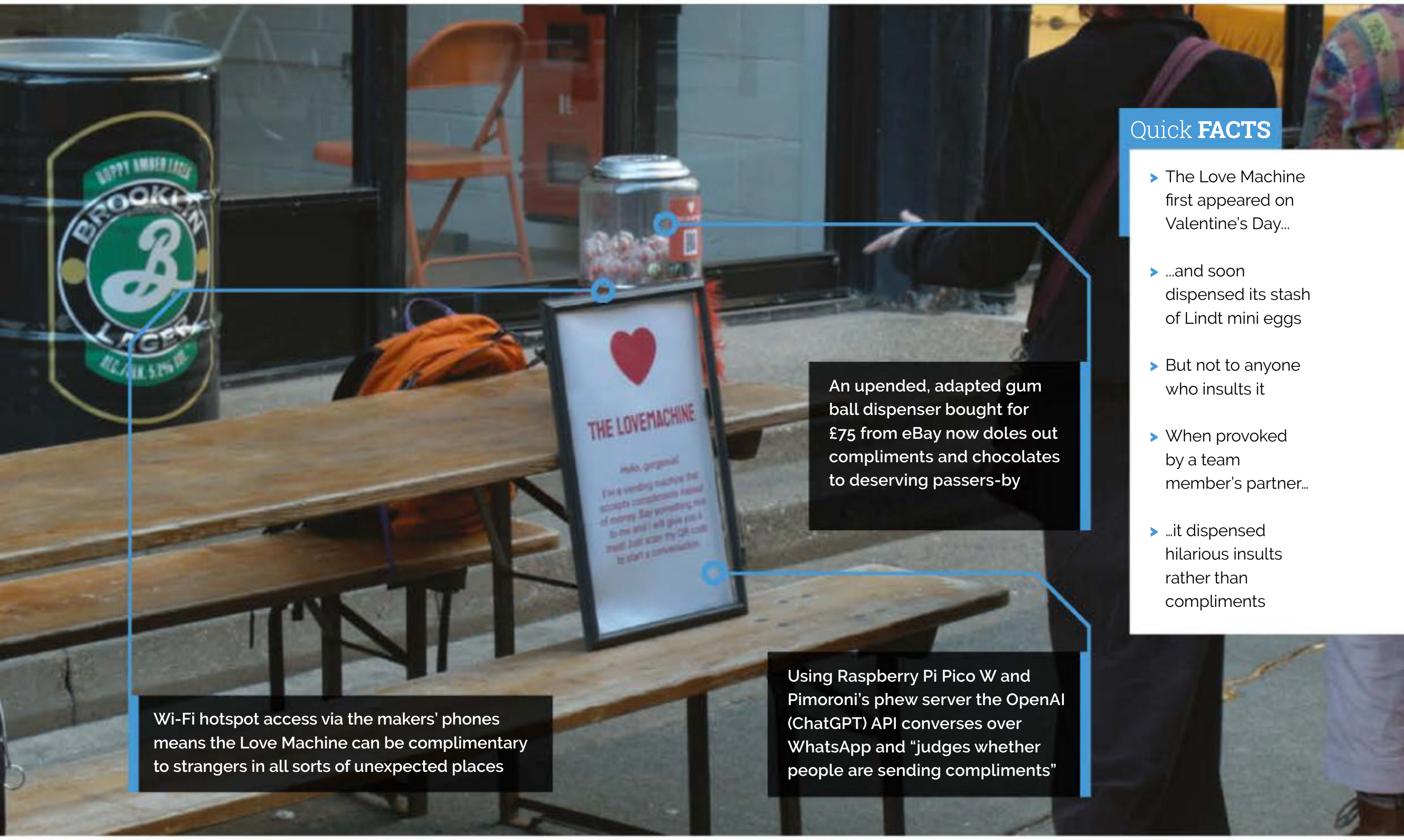
Sweet nothings

Will and India, from London-based Kakapo Labs, have backgrounds in electronics engineering and graphic design, respectively, and are interested in "building fun, positive things that cross the boundary between the internet and real-world objects." As with the password insecurity mentioned above, Will notes that behavioural psychology research shows that it's pretty clear people are highly motivated by small rewards. "We were interested in what people might do to get a small reward and how much they'd enjoy it. ChatGPT was in the news a lot around the time we started this project, but mainly related to its impact on work. "We thought instead we could try to use it to make some fun" – hence the AI cutey sporting a designer fluffy red skirt and matching glowing red LED matrix heart.

They chose Pico W for this ChatGPT Love Machine "because it's powerful but also simple, low-cost, and small but, at the same time, can run a full wireless stack which is easy to use." Kakapo Labs has more than a decade's experience of using wireless microcontrollers, and often found them complicated to use "as the trade-offs needed to squeeze an internet stack into an environment without a full operating system, and limited resources meant it always felt like a bit of an awkward fit." However, Will says that, with MicroPython on Pico W, things work painlessly and the development time is short.

Fiddling about

The Love Machine was originally a gum ball dispenser that Kakapo bought online for less than £100, replacing its 20p coin-operated mechanism with one they designed in Inkscape and laser-cut themselves. This involved several stacked-up pieces sandwiched together, plus a retractable gear wheel attached to a stepper motor. With lots of fiddly parts to connect up, Will and India realised they could simplify access by removing the vending machine's base and upending it. They boosted its power using a voltage converter but tried not to over-egg things and cause jams that could damage the mechanical cogs. A stirrer used to push gum balls towards the dispensing slot was not needed; removing this made things work more reliably. They also used brass inserts



An upended, adapted gum ball dispenser bought for £75 from eBay now doles out compliments and chocolates to deserving passers-by

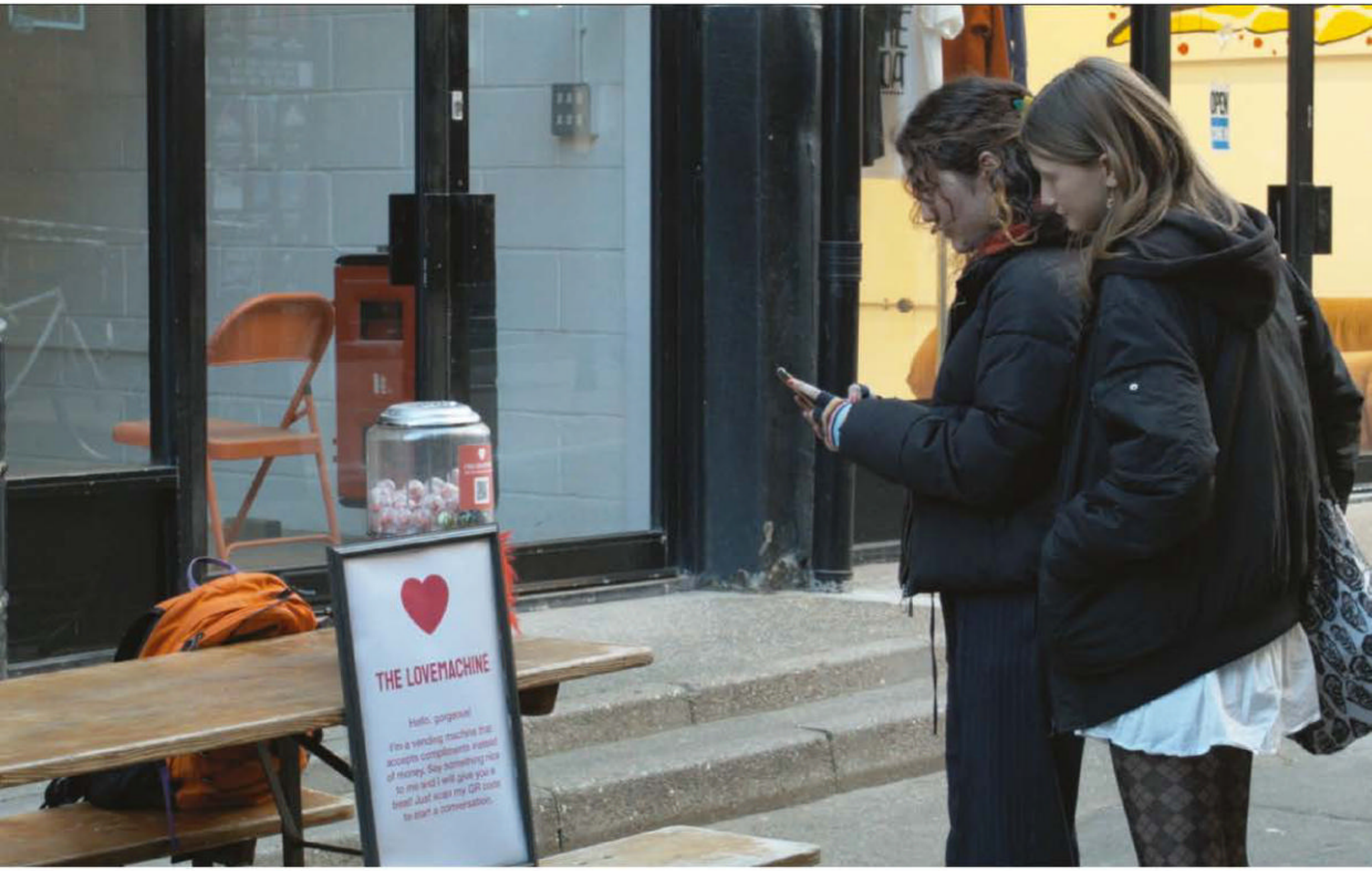
Wi-Fi hotspot access via the makers' phones means the Love Machine can be complimentary to strangers in all sorts of unexpected places

Using Raspberry Pi Pico W and Pimoroni's phew server the OpenAI (ChatGPT) API converses over WhatsApp and "judges whether people are sending compliments"

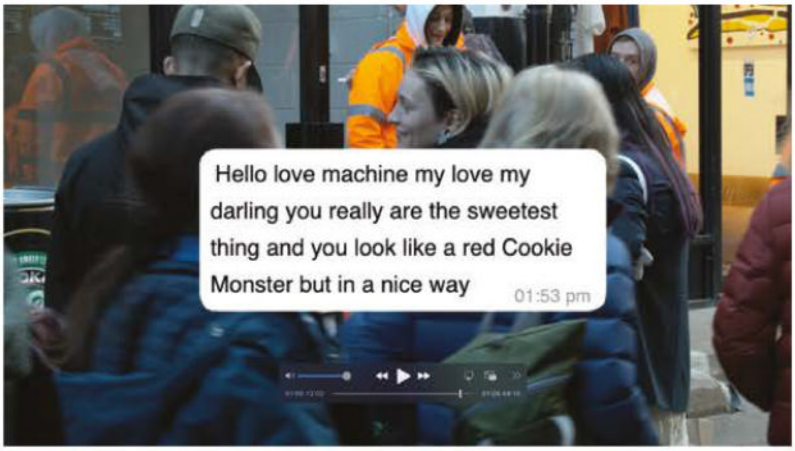
Quick FACTS

- The Love Machine first appeared on Valentine's Day…
- …and soon dispensed its stash of Lindt mini eggs
- But not to anyone who insults it
- When provoked by a team member's partner…
- …it dispensed hilarious insults rather than compliments

Who wouldn't want to be chatted up in an AI-generated chat by a cute fluffy-skirted minx?


THE LOVEMACHINE
Hello, gorgeous!
I'm a vending machine that accepts compliments instead of money. Say something nice to me and I will give you a treat! Just scan my QR code to start a conversation.
OPEN

Cute, approachable, accessible tech is a great way to attract people to a fun and rewarding career or pastime



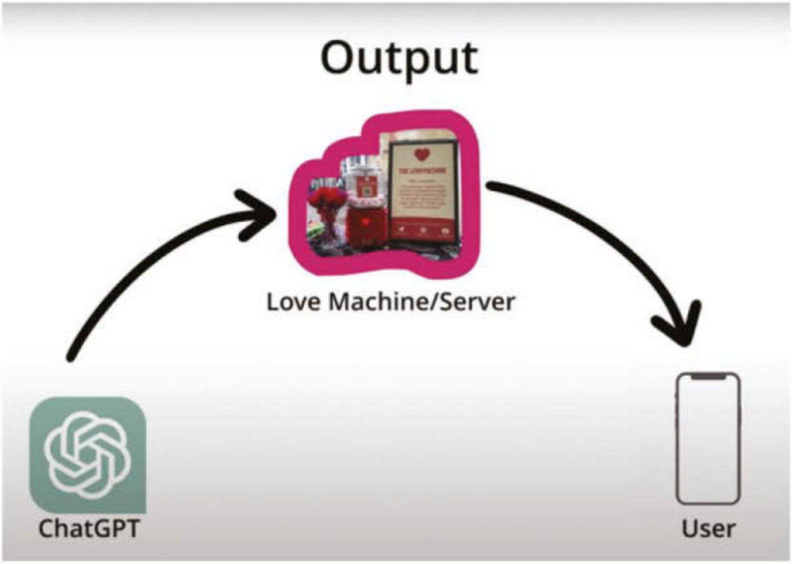
Hello love machine my love my darling you really are the sweetest thing and you look like a red Cookie Monster but in a nice way
01:53 pm

The ChatGPT Love Machine enjoys plenty of WhatsApp flattery

pushed into the acrylic sheet, instead of nuts, reducing "the number of hands/fingers/things to hold simultaneously and [making] assembly and disassembly quicker and easier."

Pimoroni's 'phew' web server provides an access point to the software and allows the team to control access and connect the ChatGPT Love Machine to Wi-Fi via their phones. The Love Machine is controlled using a web socket including, on the client side, the 'micropython_async_websocket_client' library for which WSS (Secure WebSocket) support was so recent that its code hadn't yet been merged into the main library. Using the AWS API gateway keeps running costs down as there's no need to have a server instance constantly handling requests.

This technology is hidden behind the Love Machine's greeting board which tells passers-by



▲ The Pico-powered AI bot connects to a convenient Wi-Fi hotspot to process WhatsApp chats and deliver its responses (and possibly some chocolate)

how to interact. Users send WhatsApp messages to the compliment machine, configured using Twilio, and ChatGPT "provides the conversation and judges whether people are sending compliments," says Will.

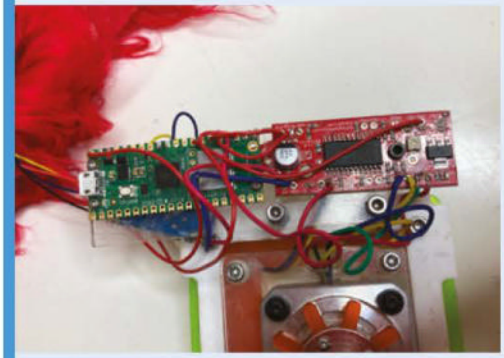
With a company named after David Attenborough's favourite breed of parrot, it was important to the Kakapo team that the build was fast to complete and fun, making it an ideal project that would catch people's imagination and then discover that they could create similar ones.

> " ChatGPT provides the conversation and judges whether people are sending compliments "

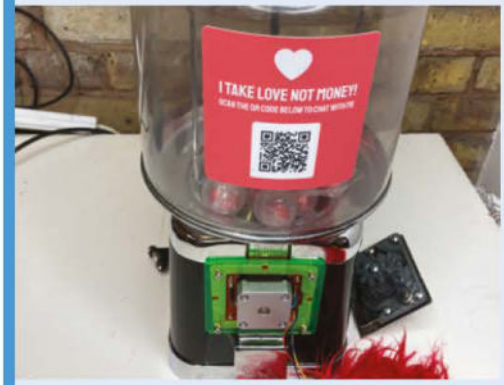
"We think getting people to have a go with tech when they're young is really important! Making things can be very rewarding and is the ideal career for some people. They are also diversity advocates – opening up the chance to have a go can help people who didn't realise 'someone like me can do this'."

Hello, sweetie!

01 A second-hand confectionery dispenser was retrofitted with a new gearing system, stepper motor, and easy stepper motor board. Aligning the dispensing mechanism was a challenge!



02 A voltage converter allows everything to run from a 5V power input. Raspberry Pi Pico W is attached via USB and runs MicroPython and Thonny IDE with data via Amazon Web Services. Messages are shown on an Adafruit I2C LED matrix.



03 ChatGPT is used to communicate with users via WhatsApp and Twilio, and judge whether the conversationalist is worthy of a chocolate-based reward.

PiFinder

Richard Sutherland doesn't need to stab around in the dark any more – his all-in-one astronomy positioning device will point the way, as **David Crookes** explains

MAKER

Richard Sutherland

Richard is a software engineer and hobbyist designer with a passion for astronomy, vintage electronics, and unusual form factor computers.

magpi.cc/pifindergit

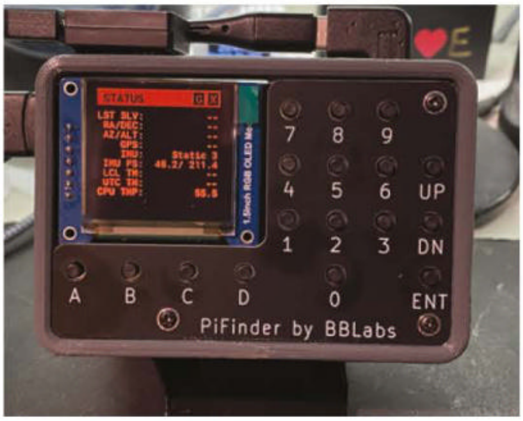
When stargazing, amateur astronomers need to know what their telescope is looking at in the vast darkness of space, yet this is easier said than done. "Finding very small and dim objects in the night sky can be challenging, historically requiring paper charts and a good working knowledge of the constellations," Richard Sutherland explains.

There are, however, newer ways to determine a telescope's position and find objects. "There are computerised pointing systems that rely on rotary encoders mounted securely to a telescope mount," Richard continues. "They monitor a telescope's movement and extrapolate where it is pointing from this information, but they can be difficult to add to a telescope and require accurate alignment information to work well."

As such, Richard decided to create his own system. "I was seeing articles and forum posts describing electronic finders (devices that use a camera and computer to determine absolute telescope pointing)," he says. "So I decided to see if I could build something I'd be happy using and sharing."

Scoping the sky

Richard has based his project – called PiFinder – around a Raspberry Pi 4 computer and a Raspberry Pi High Quality Camera, with the idea being to create a device that would take images of the night sky to determine where a telescope is pointing by analysing the pattern of stars.



PiFinder includes a keypad and a 128×128 RGB OLED screen. Richard decided against linking it to a laptop tablet or phone. LCD screens leak backlight and, under a dark sky, this will prohibit your eyes from becoming dark-adapted

"I wanted PiFinder to be easily added to any type of telescope, requiring no setup aside from a clear view of the sky to indicate where the telescope is pointing," Richard says. "I wanted to combine this ability to know where the telescope is pointed with an extensive catalogue of thousands of object locations to help an observer find objects in the sky with simple guidance on which way to push the telescope." He achieved all of these things and more.

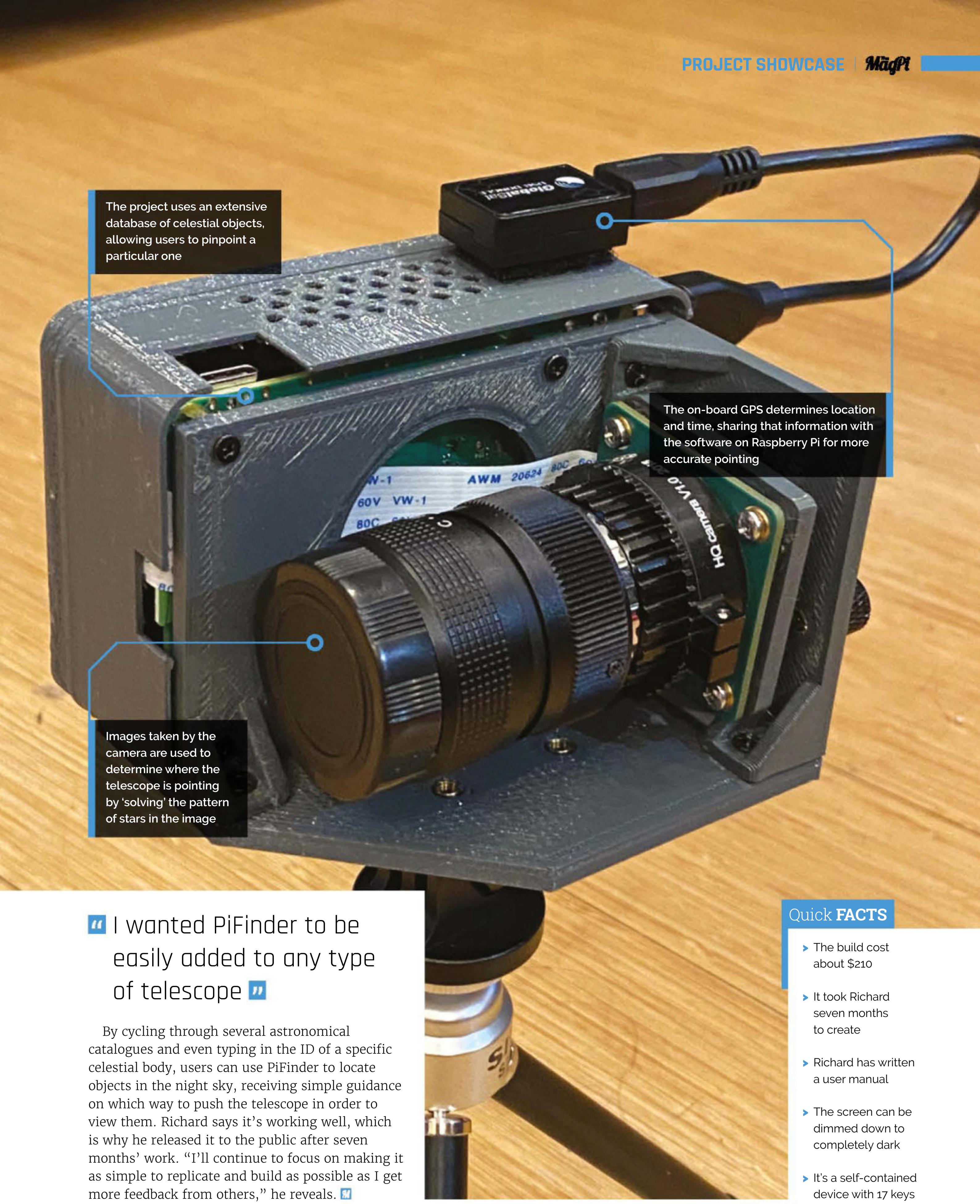
By having the camera constantly taking images, the PiFinder can get to work. "So long as the on-board Inertial Measurement Unit indicates the telescope is static, the camera is snapping images and feeding them to the solving algorithm," Richard explains. "Depending on sky conditions and which camera is being used, it can take anywhere from 0.25 to 1.5 seconds to capture an exposure which contains enough stars to solve."

Stars in his eyes

After an image has been taken, it's processed using code released by the European Space Agency called Tetra3. "It implements a very fast system for extracting stars from an image and using distances between multiple sets of four stars in the image to produce a hash," Richard says. "This hash can be quickly matched against a pre-computed database of star patterns to find candidate positions. Each potential position is then checked using other stars in the same image to verify the correct solution."

Extracting stars from the image on a Raspberry Pi 4B computer takes just 100 milliseconds, while solving the image takes 44ms. "This is all done on a separate thread so the next image can be acquired while the previous image is being solved," Richard says.

"In practice, this means that the PiFinder can usually achieve one capture/solve per second. If using the Raspberry Pi Global Shutter Camera, it can achieve more than two captures/solves per second due to its much larger pixel size and low-light sensitivity."



The project uses an extensive database of celestial objects, allowing users to pinpoint a particular one

The on-board GPS determines location and time, sharing that information with the software on Raspberry Pi for more accurate pointing

Images taken by the camera are used to determine where the telescope is pointing by 'solving' the pattern of stars in the image

> I wanted PiFinder to be easily added to any type of telescope

By cycling through several astronomical catalogues and even typing in the ID of a specific celestial body, users can use PiFinder to locate objects in the night sky, receiving simple guidance on which way to push the telescope in order to view them. Richard says it's working well, which is why he released it to the public after seven months' work. "I'll continue to focus on making it as simple to replicate and build as possible as I get more feedback from others," he reveals.

Quick FACTS

- The build cost about $210
- It took Richard seven months to create
- Richard has written a user manual
- The screen can be dimmed down to completely dark
- It's a self-contained device with 17 keys

MCM/70 Reproduction

Reproducing vintage technology is a labour of love for this maker who is uncompromising in his attention to detail, impressing **Nicola King**

MAKER

Michael Gardi

A retired software developer, living in Waterloo, Ontario, Canada with his wife, who appreciates having the time to make whatever the heck he damn well feels like!

magpi.cc/mgardiyt

Michael Gardi's reproductions of classic computers have featured in *The MagPi*'s pages on several occasions. A proud Canadian, he enjoys spotlighting "Canada's contributions to the personal computer revolution."

Confident that he "had a pretty good handle on all of the machines available in the 1960s and 1970s", he was somewhat flabbergasted when he came across the MCM/70, a "beautiful and innovative" Canadian-built personal computer that he'd never heard of. Even more so because it was conceived and made in Kingston, Ontario, very near to where Michael lives.

Hardware challenges

Keen to build his own repro version, Michael began to source components. While obtaining a keyboard was straightforward, from Dave at **osiweb.org**, he wasn't sure how to reproduce the MCM/70's unique APL keycaps. After considering making his own, he discovered a helpful UK supplier of custom keycaps, **kromekeycaps.com**.

Michael used Inkscape to create SVG versions of the legends that could be scaled to fit the keycaps. Detail is crucial in his reproductions, as he reflects: "With the 'inside' of the machine being emulated, it is very important to me for the outside to look as much as possible like the original. I want my reproductions to be clearly recognised, and operate the same as the originals they replicate."

Michael spent some time looking for a suitable replacement for the original MCM/70's plasma display that's no longer manufactured, and eventually found the high-performance Broadcom HCMS-2972 dot matrix display which offered a close match. "Packaged as eight 5×7 dot matrix arrays, these modules operate at a nice safe 3.3V and can be cascaded together side by side to create the 32×1 character display desired here."

The aesthetically pleasing casing took Michael many hours to 3D-print, and he added two placebo cassette decks for an authentic look: "Virtually all of the online images of the MCM/70 feature the two tape deck model."

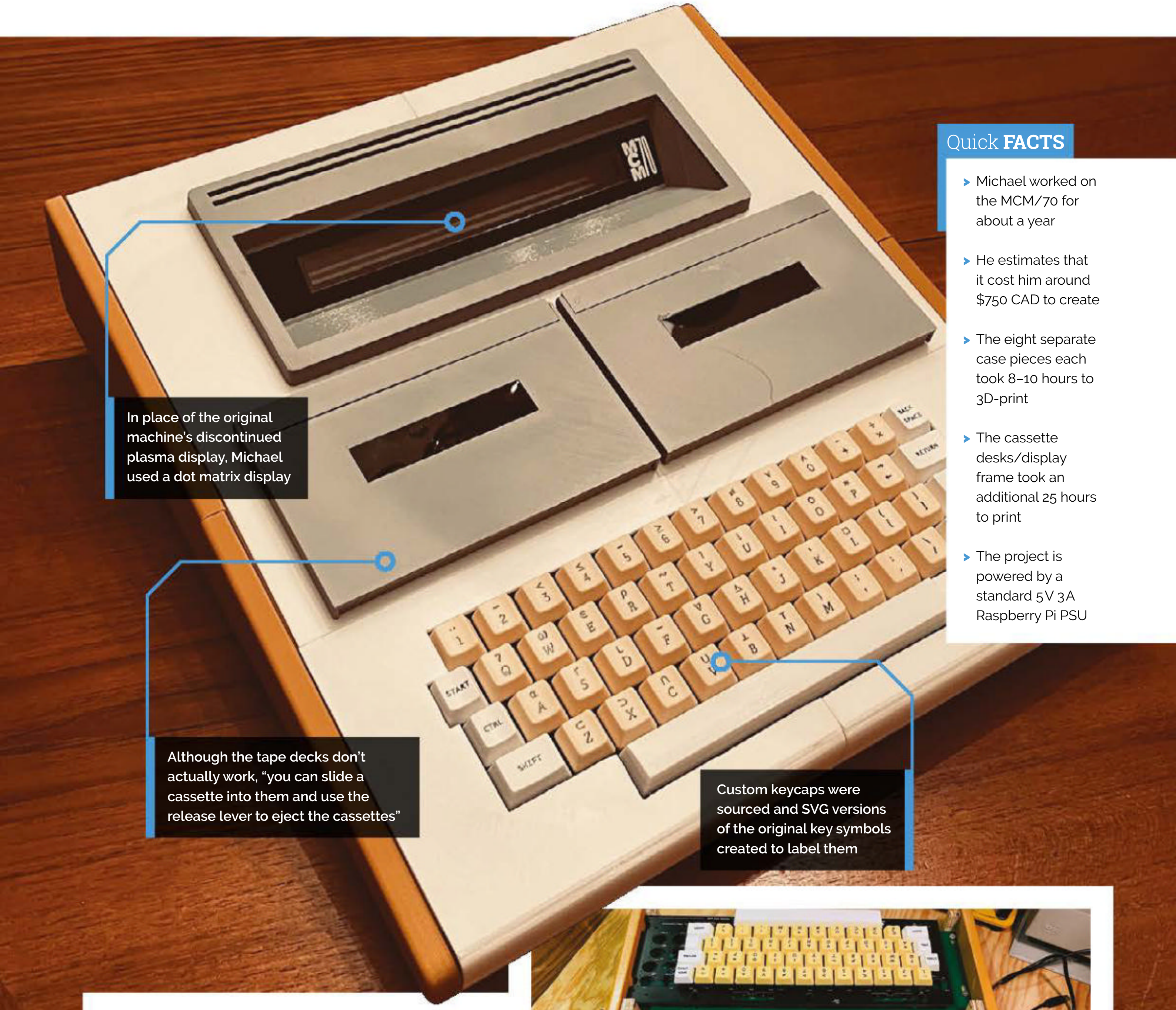
Retro Raspberry Pi

Following a research visit to the York University Computer Museum (YUCoM), Toronto, Michael decided to use the YUCoM working MCM/70 software emulator in his project, which has "high historical accuracy" – important if his version was to work just like the original. "Thanks to the hard working folks at York University, I have a great head start with this project."

> " It is very important to me for the outside to look as much as possible like the original "

To complement that, he needed a solution with a fairly powerful CPU to run the emulator. After first trying it on a Raspberry Pi 2 he had to hand, he found it only executed at about 33% the speed of an original MCM/70. "When I moved the project to a Raspberry Pi 4B, the emulator ran at twice the speed of the original," he reveals.

Raspberry Pi's large number of GPIO pins also allowed Michael to wire the emulator to interact with the display, keyboard, and cassette deck. Plus, he needed "a target system with a Linux-based OS to build and run the emulator (which was written in C). In fact, the emulator built easily on Raspberry Pi OS after a couple of required libraries were loaded."



In place of the original machine's discontinued plasma display, Michael used a dot matrix display

Although the tape decks don't actually work, "you can slide a cassette into them and use the release lever to eject the cassettes"

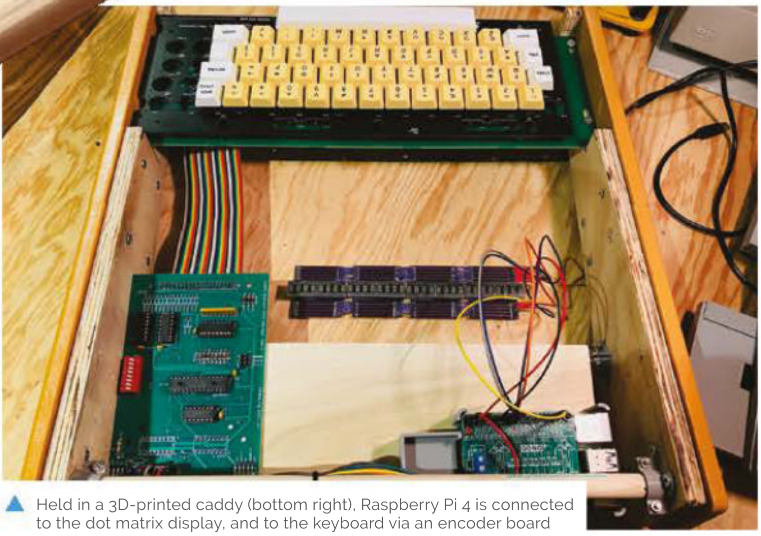
Custom keycaps were sourced and SVG versions of the original key symbols created to label them

Quick FACTS

- Michael worked on the MCM/70 for about a year
- He estimates that it cost him around $750 CAD to create
- The eight separate case pieces each took 8–10 hours to 3D-print
- The cassette desks/display frame took an additional 25 hours to print
- The project is powered by a standard 5V 3A Raspberry Pi PSU

The result of his hard work is impressive and Michael has already put it to good use. "I've been using the MCM/70 Reproduction to learn APL by working my way through the MCM/70 User's Guide. Mind you, I won't be writing any APL programs any time soon, but I would like to get a good feel for the language."

He has also generously shared a detailed description of his project on Instructables (**magpi.cc/mcm70repro**). It's well worth a read, illustrating his enthusiasm for his retro subject, and taking you through how you can create your own version.



Held in a 3D-printed caddy (bottom right), Raspberry Pi 4 is connected to the dot matrix display, and to the keyboard via an encoder board

Refurbished Vending Machine

A tech fan's plan to give something back to his community led to a Raspberry Pi-based dispenser, learns **Rosie Hattersley**

MAKER

Fady Faheem

17-year-old Fady is a "tech enthusiast, programmer and perfectionist" whose love for computing began when he learned Java at school.

magpi.cc/vendingmachinegit

If you can think of it, "you can probably do it on a Raspberry Pi," believes 17-year-old Fady Faheem, who saw a broken vending machine for sale online and decided to give it a tech upgrade. Fady wanted to use his tech chops to benefit his community in Fort Worth, Texas, and the refurbished vending machine seemed an ideal gift to his local church – "the best place I knew" – which could do some micro fundraising too. The project has kick-started his career: "If I didn't do this, I wouldn't be where I am today, working as a software engineer at 17."

Feeling blessed

Task one was the discover whether the vending machine was fixable. Buying it for so little, "knowing it was broken and the motherboard fried, I wasn't sure what I was walking into," reveals Fady.

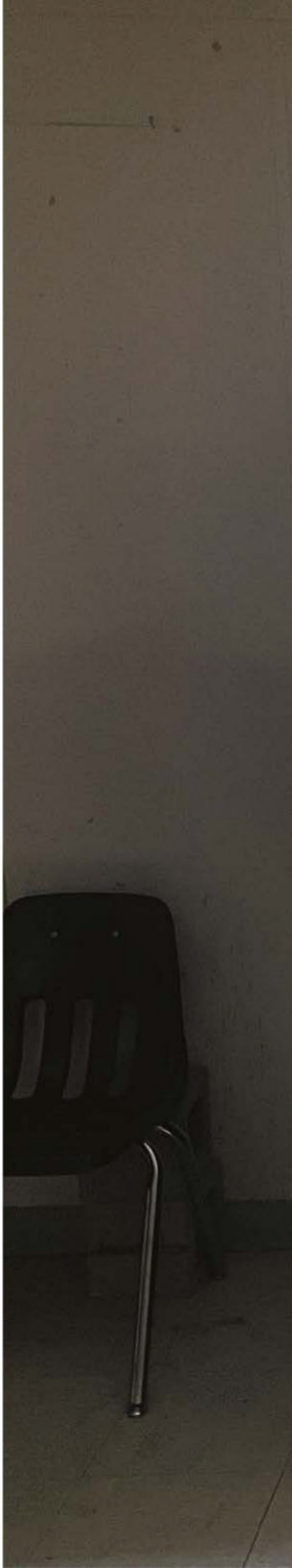
Even so, having learned Java and programming since his freshman year, he felt sure he had the power to accomplish it. He became interested in Raspberry Pi because it can be used for "just about any project", noting the range of different-sized models, "different ways to utilise it, low energy, and overall simplicity."

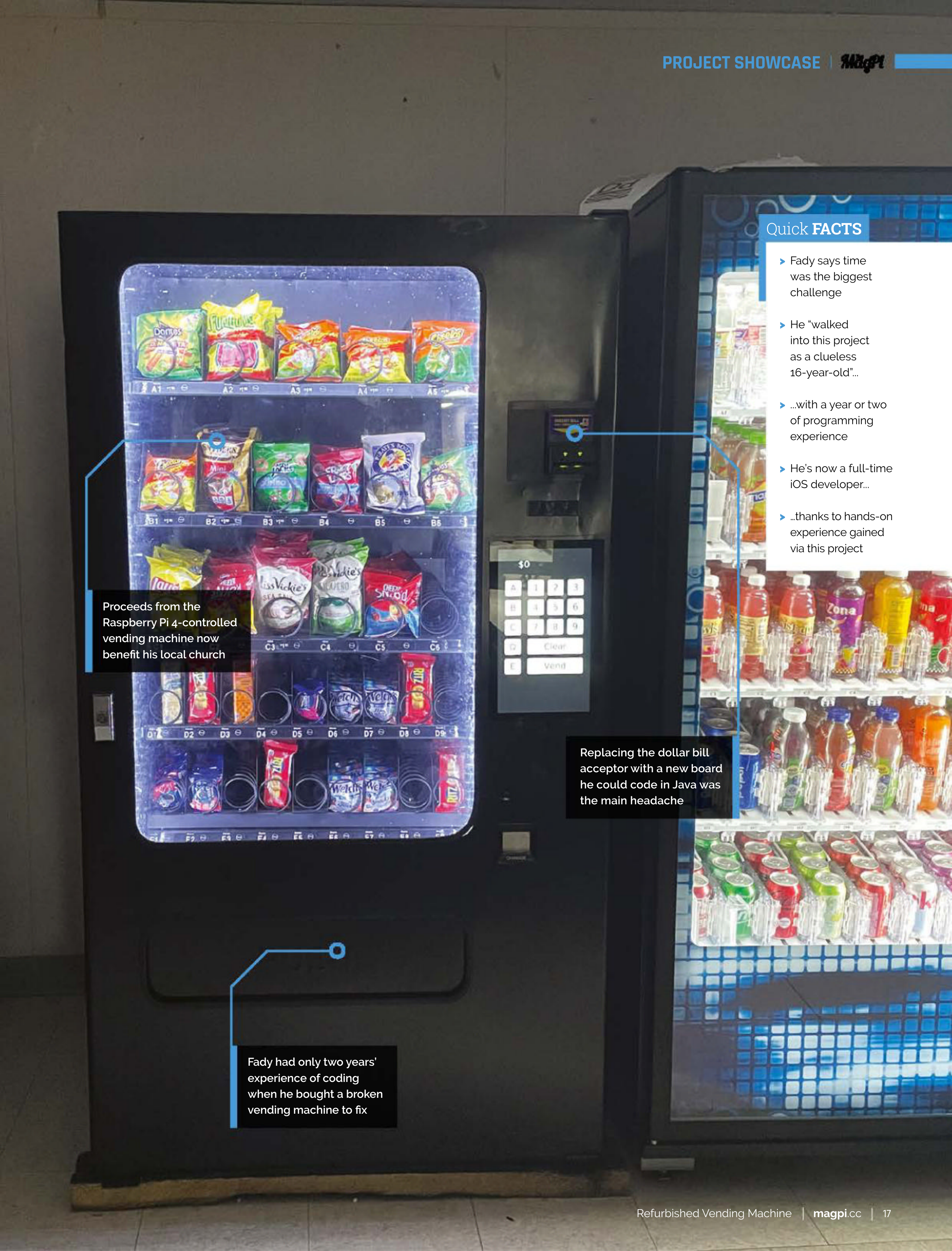
Raspberry Pi would become the heart of his vending machine, connecting everything: communicating with the dollar bill acceptor, sending data to relays to activate them, and handling power between all devices. There was little to no space to put a machine or full-size computer behind the vending machine screen, so the diminutive Raspberry Pi was "the ultimate solution to all my problems," he says. "I had a small corner to run the hardware. If it could fit in my hand, I knew it would fit in the machine." Fady's refurbished vending machine originally used a 3B+, but he recently upgraded

Warning!
Electrical Safety

Please be careful when working with electrical projects around the home. Especially if they involve mains electricity.

magpi.cc/electricalsafety





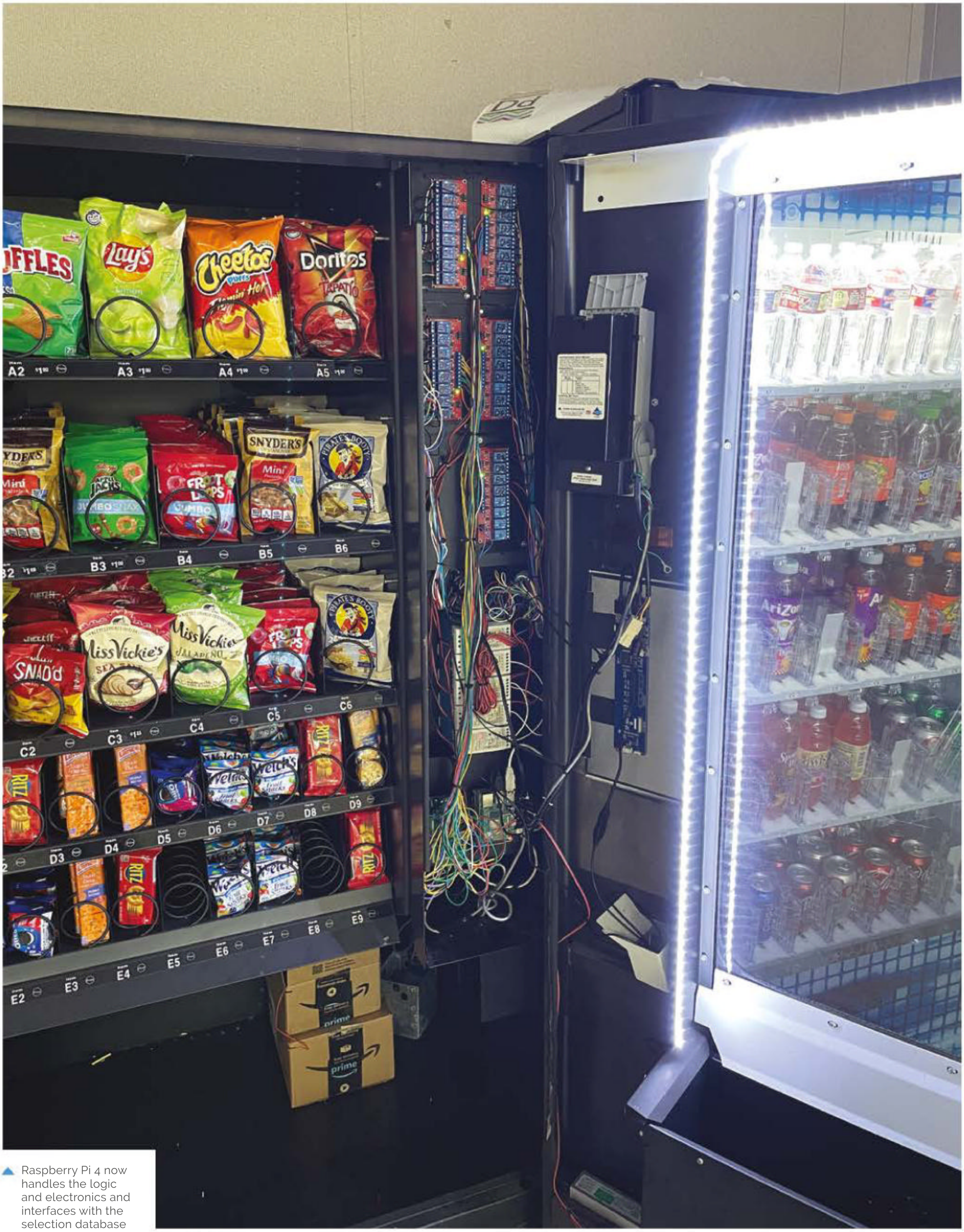
Proceeds from the Raspberry Pi 4-controlled vending machine now benefit his local church

Replacing the dollar bill acceptor with a new board he could code in Java was the main headache

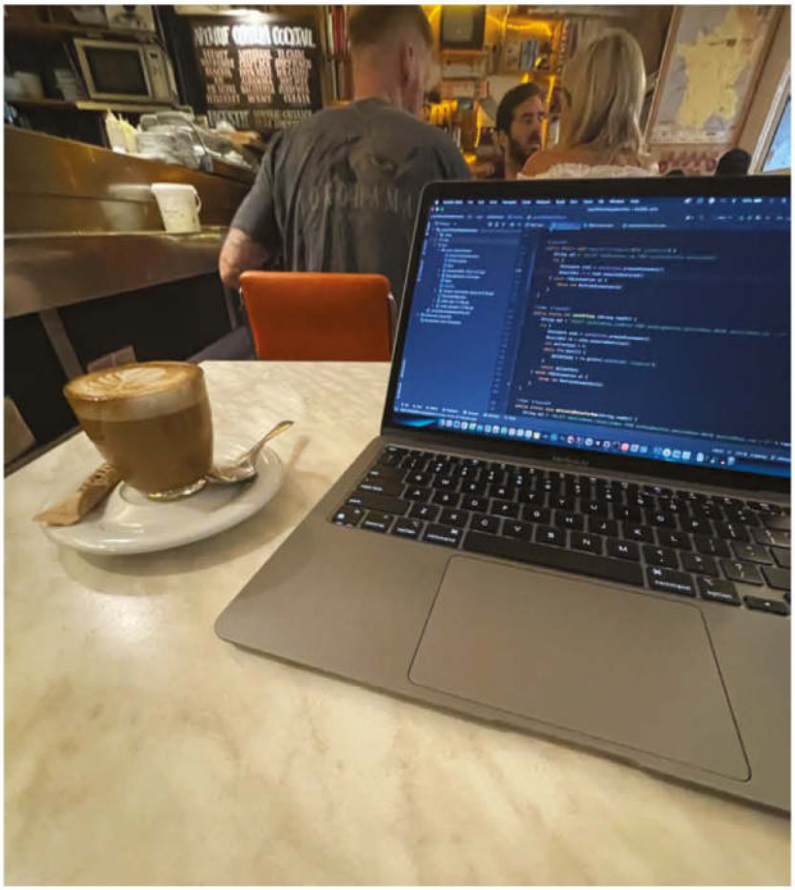
Fady had only two years' experience of coding when he bought a broken vending machine to fix

Quick FACTS

- Fady says time was the biggest challenge
- He "walked into this project as a clueless 16-year-old"...
- ...with a year or two of programming experience
- He's now a full-time iOS developer...
- ...thanks to hands-on experience gained via this project



Raspberry Pi 4 now handles the logic and electronics and interfaces with the selection database



▲ Fady put his love of Java to great use by providing a welcome refreshments machine for fellow members of his church

it to Raspberry Pi 4 for performance on start-up, should he ever need to pull the plug and restart.

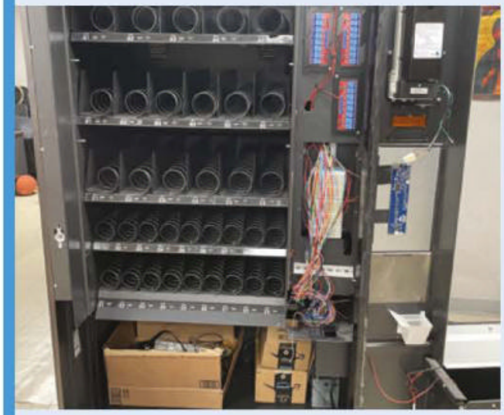
Money makers

Fady began to tackle the project by researching how other people were able to use a specific combination of dollar bill acceptor and a circuit board to run Java. For the first version, Fady used a Pyramid Apex 5400, but later switched to the Apex 7400 bill acceptor, resulting in better support and handling through Java. Vending machine selections are handled by the MariaDB database and Raspberry Pi using a number of libraries Fady found on GitHub. Meanwhile an Arduino connected to Raspberry Pi via the vending machine's printer cable, and accessed using Java, handles the signals to the relays and the relays themselves.

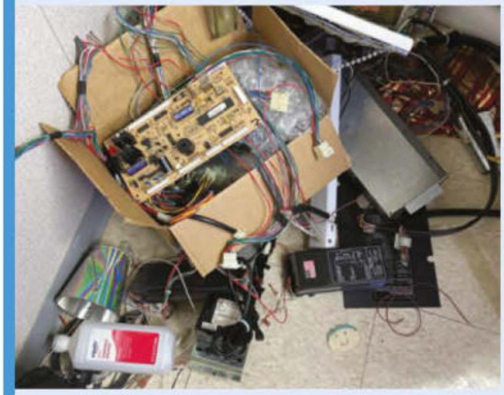
> "If I didn't do this, I wouldn't be where I am today, working as a software engineer at 17"

"At the end of the day, all the work that went into it was designed and 3D-printed by me," he says proudly.

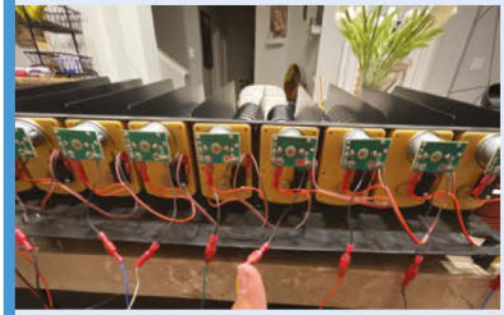
Back in service



01 Having received the machine he bought online, Fady removed the control panels to reveal outdated, underpowered coin and note mechanisms.



02 Fady installed a replacement bill acceptor board that was sufficiently powerful for his needs and that worked smoothly with the Java code, database, and libraries he chose.



03 At the 'heart' of the vending machine a Raspberry Pi 4 controls everything, alongside an Arduino connected to relays to fulfil customer selections.

RAD Expansion Unit

Want to get the most out of old tech? **David Crookes** looks at one device which boosts the Commodore 64 so much, it's capable of playing Doom

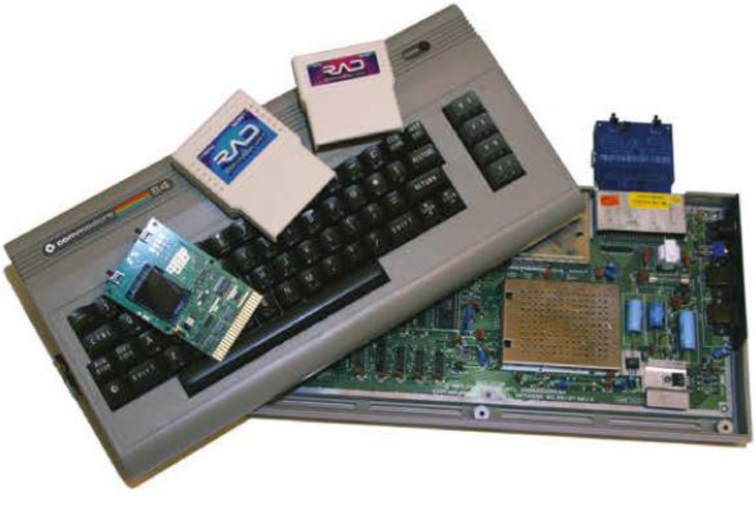
MAKER

Carsten

Carsten is a computer graphics researcher and lecturer. Since rediscovering computers of his childhood, he tries to convince Raspberry Pi boards to communicate with them.

magpi.cc/radeu

▼ Carsten has created a number of different expansion units for the Commodore 64, all of which can plug into the computer's expansion port



Considering Doom was once thought impossible on the Commodore Amiga, running it on an 8-bit Commodore 64 would appear to be sheer fantasy. But, thanks to the efforts of maker Carsten, such an achievement has become reality.

Having previously created a multifunctional cartridge called Sidekick64 for various Commodore 8-bit computers which emulated memory expansions, sound devices, freezer carts and much more, Carsten saw that many people in the retro community still wanted a replacement for the official RAM Expansion Unit (REU) which was released in 1986 and discontinued four years later.

For those unaware, the REU plugged into the C64's expansion port and added extra memory while also allowing for direct memory access (DMA) transfers – something the Sidekick64 couldn't do. This meant data could be transferred to and from the main system memory whole bypassing the MOS Technology 6510/8500 CPU. "My RAD Expansion Unit was designed to do these transfers," Carsten says. It's a major triumph!

Radical thinking

There are many benefits to having the RAD Expansion Unit, which boosts the amount of available memory from the standard 64kB to as much as 16MB. "It helps to run Geos [a C64 OS] and it makes some tasks less annoying, such as copying disks in one go. It also functions as a RAM disk to accelerate working with the system," Carsten says.

In order to create the project, then, two PCBs were produced: one to fit Raspberry Pi 3A+/3B+ and another to fit Raspberry Pi Zero 2 W (the latter creating a less expensive unit). "I didn't want the glue logic that facilitates the bus communication with the available GPIOs to get too extensive and I wanted to avoid Complex Programmable Logic Devices and such which would prevent many people from building their own RAD," Carsten says.

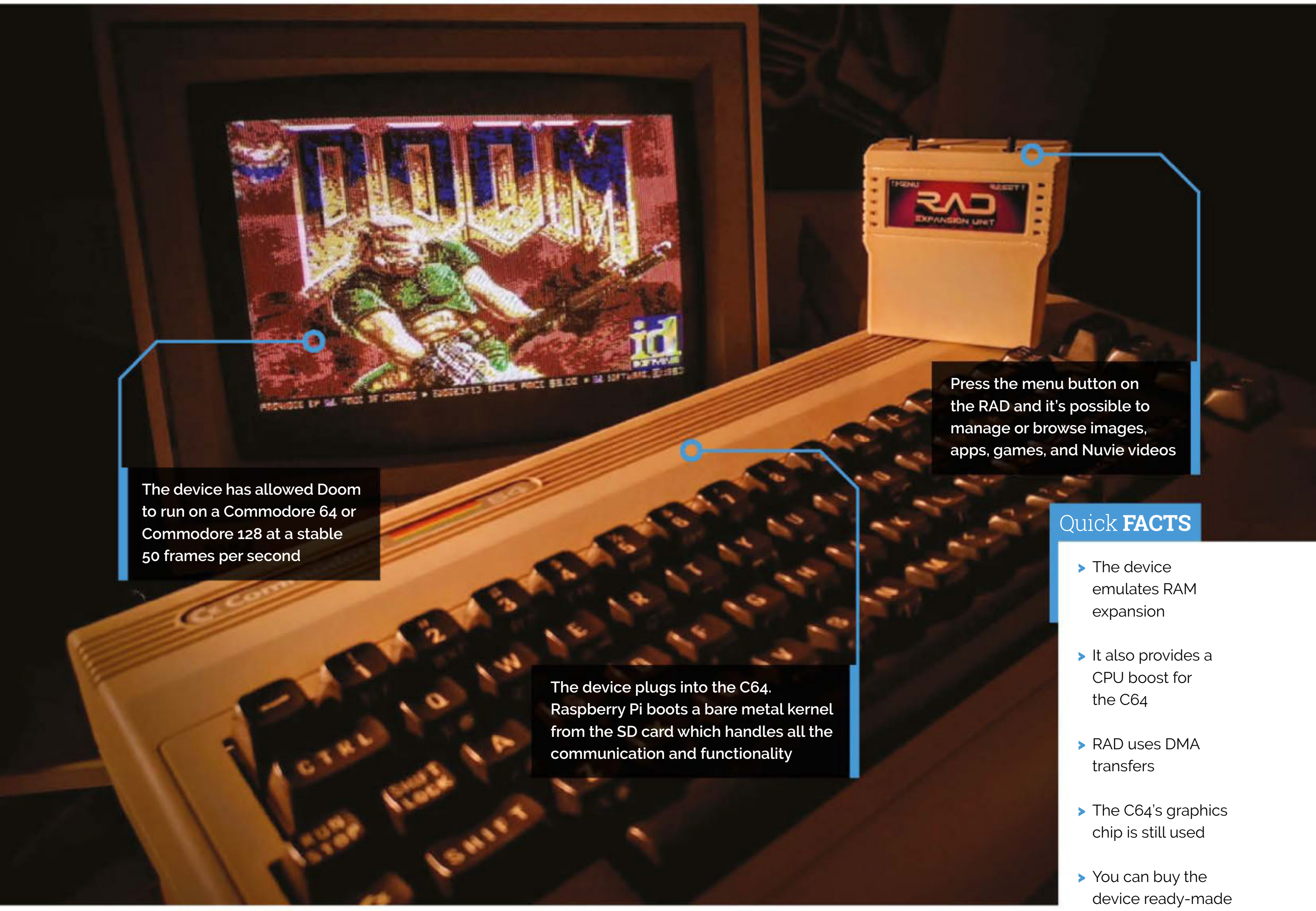
Indeed, the project was designed so that the RAD contains the glue logic to interface Raspberry Pi with the C64 bus. "You simply put it on to Raspberry Pi like a HAT and plug it into the expansion port of the Commodore 64," Carsten continues. "The combination of a fast SoC and a decent number of GPIOs was great."

An explosive result

There were challenges, of course. "The biggest was getting the bit banging right," Carsten says. "Most of the communication has to happen within a time window of less than 500 nanoseconds – most often there's significantly less time between all signals being read and putting data on the bus."

To make the device run smoothly, Carsten had to use multiplexers. "More signals on the expansion port need to be read/written to than Raspberry Pi has GPIOs," he explains. The correct data had to be put on the bus at the right time to prevent memory corruption and, in the worst case, random instructions being executed by the CPU, causing a crash. "In general, to get the timing right, I needed to hit intervals at a spacing of approximately tens of nanoseconds, which I did using CPU cycle counters."

With all that in place, it was time to chill and RAD-Doom proved a great way to do so. Most of the processing is being done by Raspberry Pi ("it's essentially a CPU replacement where the new CPU



The device has allowed Doom to run on a Commodore 64 or Commodore 128 at a stable 50 frames per second

Press the menu button on the RAD and it's possible to manage or browse images, apps, games, and Nuvie videos

The device plugs into the C64. Raspberry Pi boots a bare metal kernel from the SD card which handles all the communication and functionality

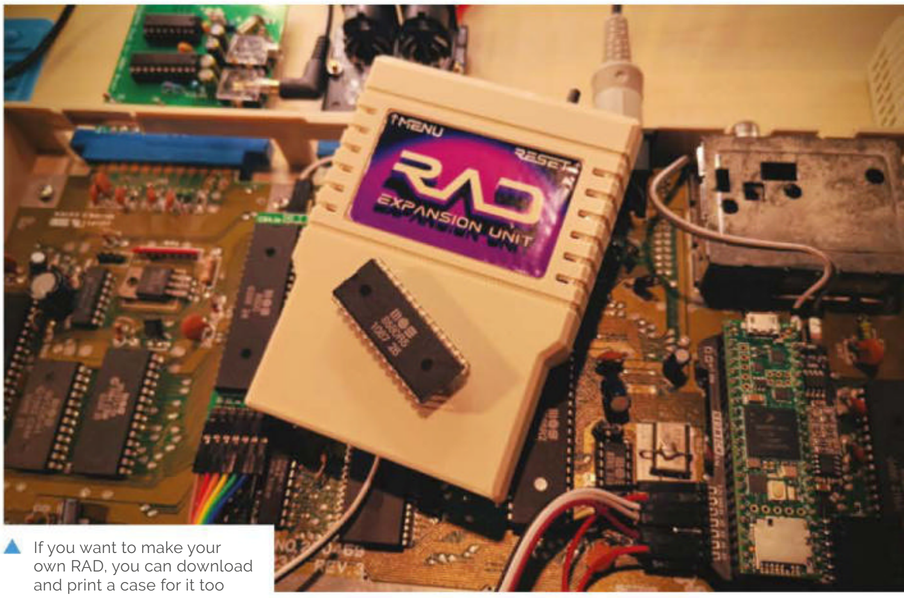
Quick FACTS

- The device emulates RAM expansion
- It also provides a CPU boost for the C64
- RAD uses DMA transfers
- The C64's graphics chip is still used
- You can buy the device ready-made

is a one-core ARM running at 1.4GHz with its own 512MB RAM," Carsten says). But the important thing is that the tech demo uses the C64's VIC-II graphics chip – and works! Sound is also streamed to the iconic SID chip.

> It makes some tasks less annoying, such as copying disks in one go

"I wanted to see how the C64/C128's VIC-II and SID performed if CPU power and memory was not an issue," Carsten says. "I also wanted to experiment with real-time colour dithering for the VIC-II." By making use of Doom to output graphics and sound, he's certainly managed to achieve that.



If you want to make your own RAD, you can download and print a case for it too

Antarctic Picam

A simple waterproof camera by a school student is being used to look under the cracks in the Antarctic. **Rob Zwetsloot** takes a peek

MAKER

Oliver & Autun Purser

Oliver is a student who knows Autun and his work from a stay in Germany. He was inspired by an article in *The MagPi*. Autun is co-ordinator for the scientific use of the deep sea robots and camera systems for the Alfred Wegener Institute of Polar and Marine Research, Bremerhaven, Germany.

awi.de

Way back in issue 74 (magpi.cc/74), we wrote about PiCam Marine, little monitoring devices for cold water coral deployed by submarine. We like to think the projects we publish in the magazine can act as inspiration, and this one did…

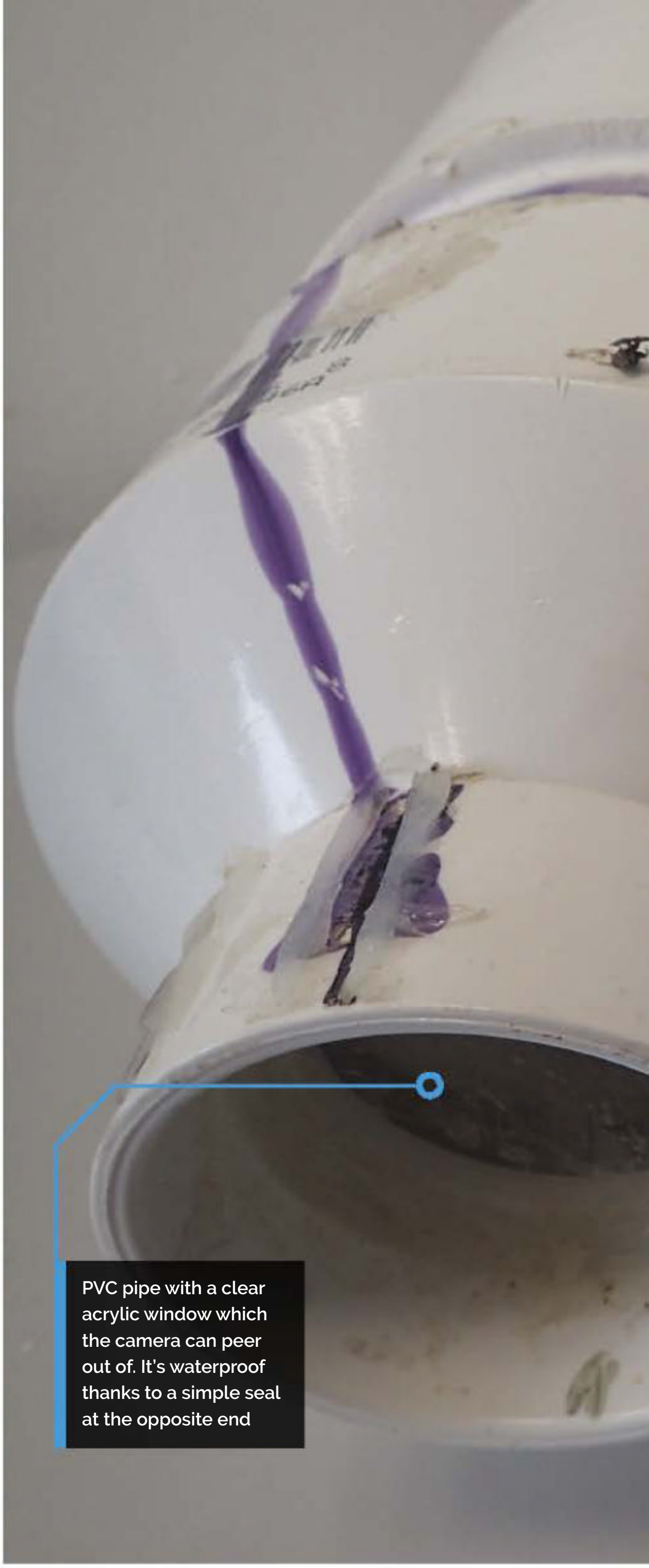
"A couple of years ago, we started designing and using Raspberry Pi-based camera systems for under-sea research work," Dr Autun Parser tells us. He's a research scientist for the Alfred Wegener Institute of Polar and Marine Research. "These cheap, programmable platforms allow us to place many cameras on the sea-floor, for the same costs as one 'off the shelf' marine camera. We had an article on these cameras in *The MagPi* magazine and Oliver contacted us, a student then at the International School Bremen, Germany, who wanted to design and program his own camera."

"[I really] wanted to see what laid in the river outside of my house," Oliver explains. "After I first deployed the camera, I told Autun, who said that he could take it with him to Antarctica on his next mission." With the camera tested, Autun stood by his word, took it on the RV Polarstern, a research icebreaker ship, and deployed it during two helicopter missions.

Plumbing the depths

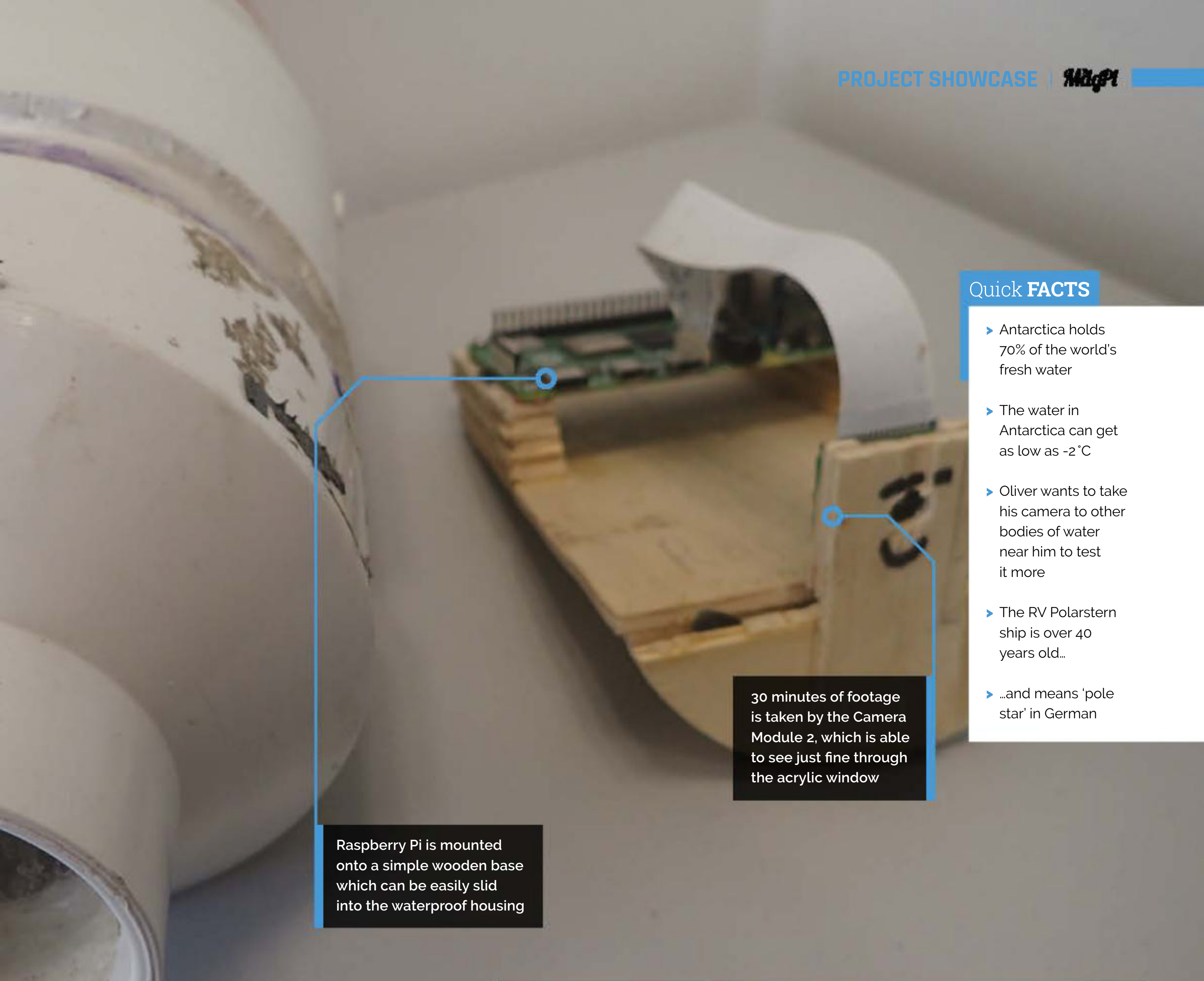
The solution for the water-tight camera ended up being nice and simple.

"Oliver made the camera housing out of robust plastic, and used an interesting plumbing seal to close one end," Autun says. "You put a Raspberry Pi in, turn on the power, and turn a nut to close the housing… the nut presses two plastic pieces together to expand a rubber seal – very effective, quick, and robust. The front of the housing is a simple acrylic sheet, against which a standard Raspberry Pi Camera Module 2 was placed. This was connected to a Raspberry Pi 3, attached to a wood mounting block. Below the block, a simple USB power bank was placed to power the system."



PVC pipe with a clear acrylic window which the camera can peer out of. It's waterproof thanks to a simple seal at the opposite end

Oliver's original version also had a little bait container to try and lure fish closer. "The program is written entirely in Python, simply takes a 30-minute recording, and after the 30 minutes, it automatically shuts off," Oliver elaborates. According to Autun, this 30-minute length of the



Raspberry Pi is mounted onto a simple wooden base which can be easily slid into the waterproof housing

30 minutes of footage is taken by the Camera Module 2, which is able to see just fine through the acrylic window

Quick FACTS

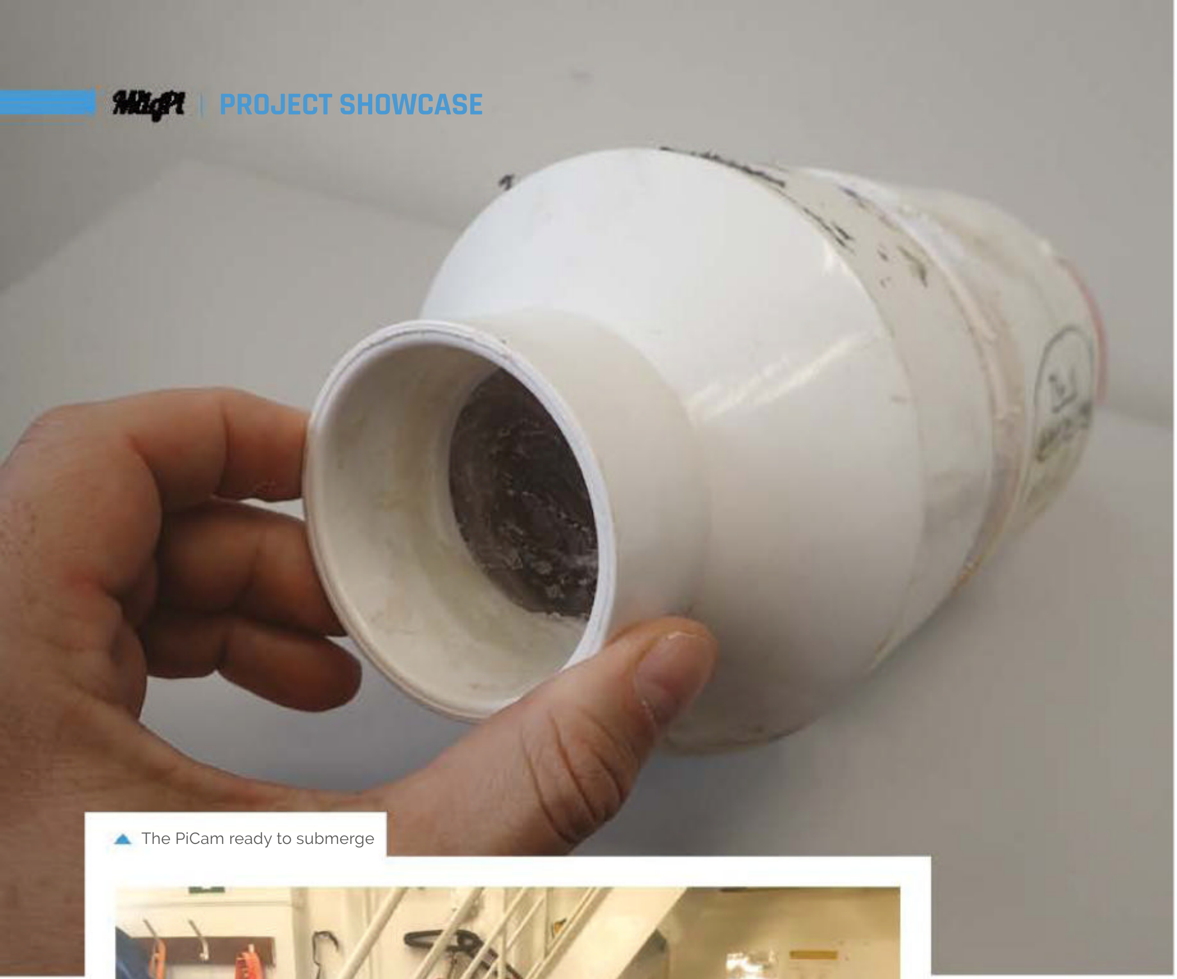
- Antarctica holds 70% of the world's fresh water
- The water in Antarctica can get as low as -2˚C
- Oliver wants to take his camera to other bodies of water near him to test it more
- The RV Polarstern ship is over 40 years old…
- …and means 'pole star' in German

script was enough for what they had planned: lower from a helicopter, let the video record, and then retrieve it. It was deployed five times during the mission, mainly through ice cracks. "We even slid it from the helicopter down the flank of a large iceberg and down onto a shelf of ice 10m or so below the waterline," Autun mentions.

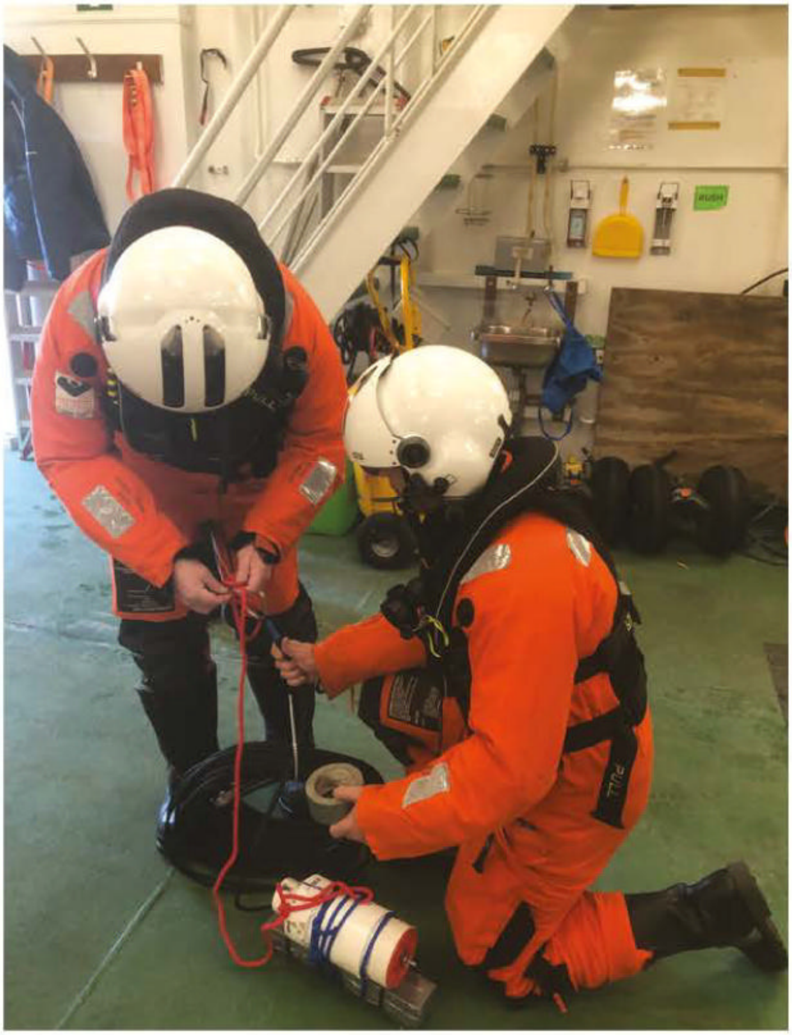
Icy depths

"The results from the camera have been really good," Olive tells us. "It's able to capture decent

▲ The seal uses a very simple plumbing system



The PiCam ready to submerge



The PiCam was used in helicopter missions

Algae grows on ice, apparently

footage, and I can clearly make out everything that it captures. Raspberry Pi is relatively low on power usage, and the battery has enough capacity that, theoretically, the camera could run for a couple of hours."

Autun sent us some amazing images from under the ice to prove it, and is planning to use similar cameras for a future trip on the RV Polarstern, although this time up to the Arctic and the North Pole.

"Within our working group, we are very happy to support the development of environmental

▲ View from under the ice

monitoring devices and sensors for use in polar regions, both above and below the ice, with a Raspberry Pi base," Autun reveals. "We travel every year to the Arctic and/or Antarctic and we can gladly discuss taking your devices with us for testing and scientific use – the scientific community can definitely benefit from the development of systems and coding by enthusiastic makers. You can email [me] at **autun.purser@awi.de** if you have an idea of something you would like to see above or below the polar ice!"

> The nut presses two plastic pieces together to expand a rubber seal - very effective, quick, and robust

Getting cold snaps

01 Getting to the Antarctic takes a special kind of ship – in this case, an icebreaker. Load it up with people and helicopters and you're ready to go.

02 The camera needs to be started and then deployed from the helicopter into ice cracks. Due to the length of the video capture being 30 minutes, there's plenty of time to set up and go.

03 Scrubbing through the video once the camera has been retrieved helps with scientific study of the icy waters near the poles.

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
LEARN TO CODE WITH PYTHON
DIY HOME WITH RASPBERRY PI
Raspberry Pi Micro Brewery
MASTER THE LINUX COMMAND LINE INTERFACE
NEW! RASPBERRY
40 PAGES OF TU
+
RaspberryPi PicoW © 2022
BOOTSEL

Subscriber Benefits

- FREE Delivery
  Get it fast and for FREE
- Exclusive Offers
  Great gifts, offers, and discounts
- Great Savings
  Save up to 23% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- Free delivery to your door
- £10 (UK only)

Subscribe for 6 Months

- Free Pico W
- 6 issues of The MagPi
- Free delivery to your door

£30 (UK)	$43 (USA)
€43 (EU)	£45 (Rest of World)

Subscribe by phone: **01293 312193**

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

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

Subscribe for £10 is a UK only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a 6-month subscription in USA, Europe and Rest of World.

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print today and get a FREE development board

WORTH $6

- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

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

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Learn electronics with RGB LEDs
Raspberry Pi Micro Brewery
Build an AI storyteller
Craft your own beer with digital assistance
MASTER THE LINUX COMMAND LINE INTERFACE
DIY HOME AUTOMATION WITH RASPBERRY PI
60 PAGES OF TUTORIALS & PROJECTS

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

SUCCESS STORY magpi.cc/success

Kenya and Malawi nursing schools

Students needed equipment for remote learning. Raspberry Pi 400 gave them access to online opportunities. By **Rosie Hattersley**

When the pandemic hit, nursing and midwifery students training for vital front-line roles had to switch to remote learning for some of their studies, but many lacked equipment at home to access online material. Wisenet (**wisenet.co**) stepped in to help by donating computers: the compact desktop Raspberry Pi 400 was the ideal choice.

People around the world had to adapt to working and studying from home when measures to tackle the COVID-19 pandemic meant that educational organisations and workplaces needed to shut their doors. For a remote model to work, organisations need the right equipment, software, and practices; and so do individuals. However, access to computing equipment at home is far from universal, and for many people the lack of a computer suitable for working presented a huge obstacle.

THE CHALLENGE

Nursing schools across Kenya and Malawi worked hard to provide remote learning opportunities for their students when the COVID-19 pandemic hit. With a number of offices across Africa, education management software provider Wisenet (a division of Adapt IT, **adaptit.com**) was well-placed to help with the response to the challenge. They

To brain
1 Extensor muscle stretched
2 Muscle spindle stimulated
3 Primary afferent neuron excited
4 Primary afferent neuron stimulates alpha motor neuron to extensor muscle
5 Alpha motor neuron stimulates extensor muscle to contract
6 Primary afferent neuron stimulates inhibitory interneuron
7 Interneuron inhibits alpha motor neuron to flexor muscle
8 Flexor muscle (antagonist) relaxes
DELL

> Raspberry Pi 400 is these students' only computer, so a key feature is that it offers a complete desktop experience

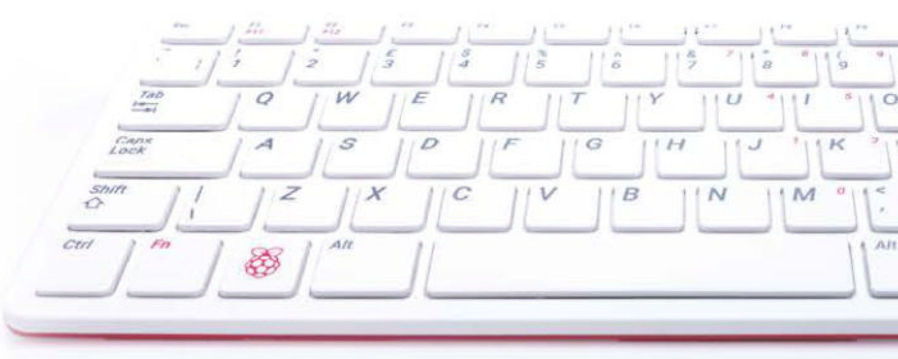
partnered with a charity, the Medical Benevolence Foundation (MBF), to provide online courses across a group of schools that host 2500 nursing and midwifery students across the region, using the open-source learning platform Moodle. However, while faculty reported a high level of satisfaction with the way the new platform allowed them to create course content and customise it for different student groups, they discovered that some learners didn't have the equipment they needed to access this content at home.

THE SOLUTION

Wisenet began donating Raspberry Pi 400 compact desktop computers to MBF-supported nursing colleges, so that their students can use the devices for online learning. Students can use Raspberry Pi 400 to access remote learning resources, as well as for general computing. To extend the reach of the programme as far as possible, Wisenet also committed to matching donations of Raspberry Pi 400 by their customers and by other organisations.

WHY RASPBERRY PI?

Raspberry Pi 400 is these students' only computer, so a key feature is that it offers a complete desktop experience while being straightforward to set up and use. A fast processor, USB 3.0 ports, and 4GB of RAM are all built into a compact, high-quality keyboard, keeping setup simple for students so they can focus on learning. Its low cost and



portability, notes Wisenet, make it "ideal for course delivery and access to online resources."

Meanwhile, Raspberry Pi OS, Raspberry Pi's free and fully supported operating system, provides an intuitive and easy-to-use environment for study. The optimised version of the Chromium web browser means students can use web-based resources easily and quickly, including essential everyday applications like Gmail and Zoom.

> Learners are able to use their Raspberry Pi 400s to attend virtual classes

THE RESULTS

Learners are able to use their Raspberry Pi 400s to attend virtual classes and access study materials, as well as to interact with tutors and other students in online forums, giving them easy access to the help they need. The impacts extend beyond home study: in addition to enabling students to learn at home, some of the computers are used within institutions to increase access to in-service training.

"Wisenet is incredibly proud to be associated with MBF and the schools," says Ben Hamilton, its CEO and co-founder: "We know these devices will help learners reach new heights. Having access to this technology provides the schools with the opportunity to do incredible things."

Lisa Alianiello, director of the Centre for Global Nursing Development, is heading up the initiative for the MBF. "Our partnership with Wisenet assists in putting the most current nursing education information directly in the hands of faculty and students," Alianiello reports. "As a result, new nursing graduates are ready to help provide much-needed medical care in their countries."

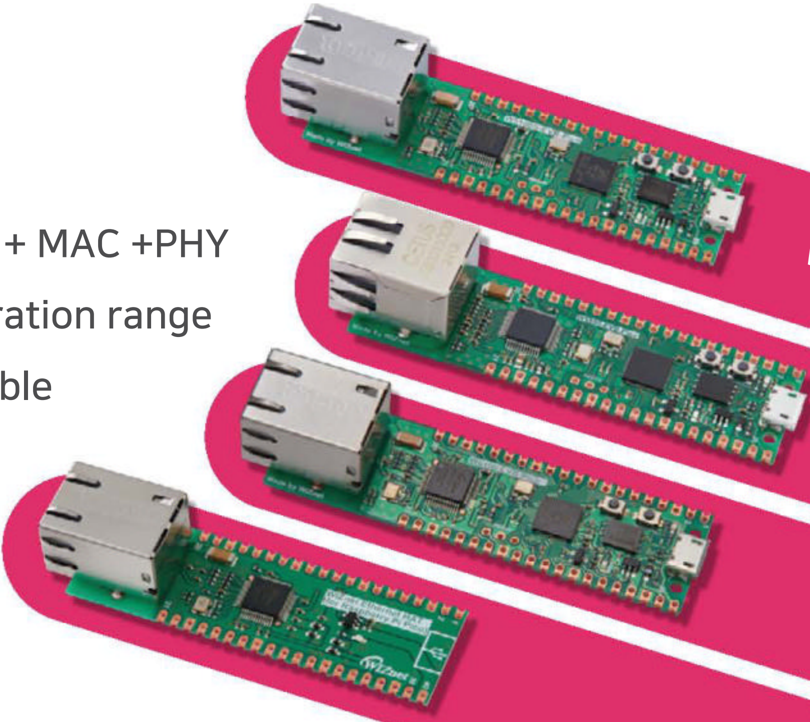
WIZnet

Meet the WIZnet Pico family

Experience the Power of Connectivity with WIZnet

Powered by Raspberry Pi

- 3-in-1: TCP/IP Stack + MAC +PHY
- Industrial grade operation range
- RPi Pico pin-compatible



W5100S-EVB-PICO

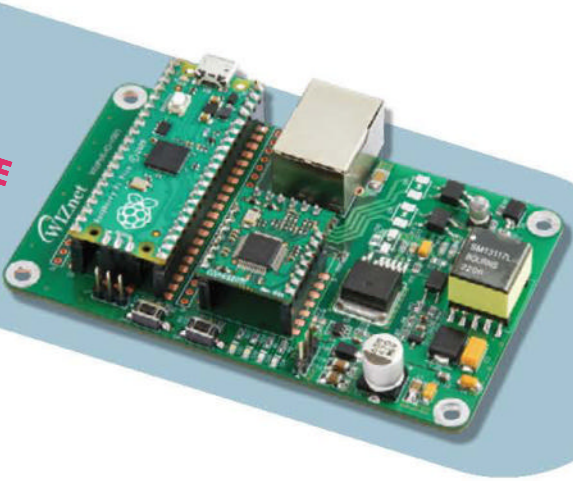
W5500-EVB-PICO

W6100-EVB-PICO

WIZnet Ethernet HAT

COMING SOON

WIZnet Pico PoE



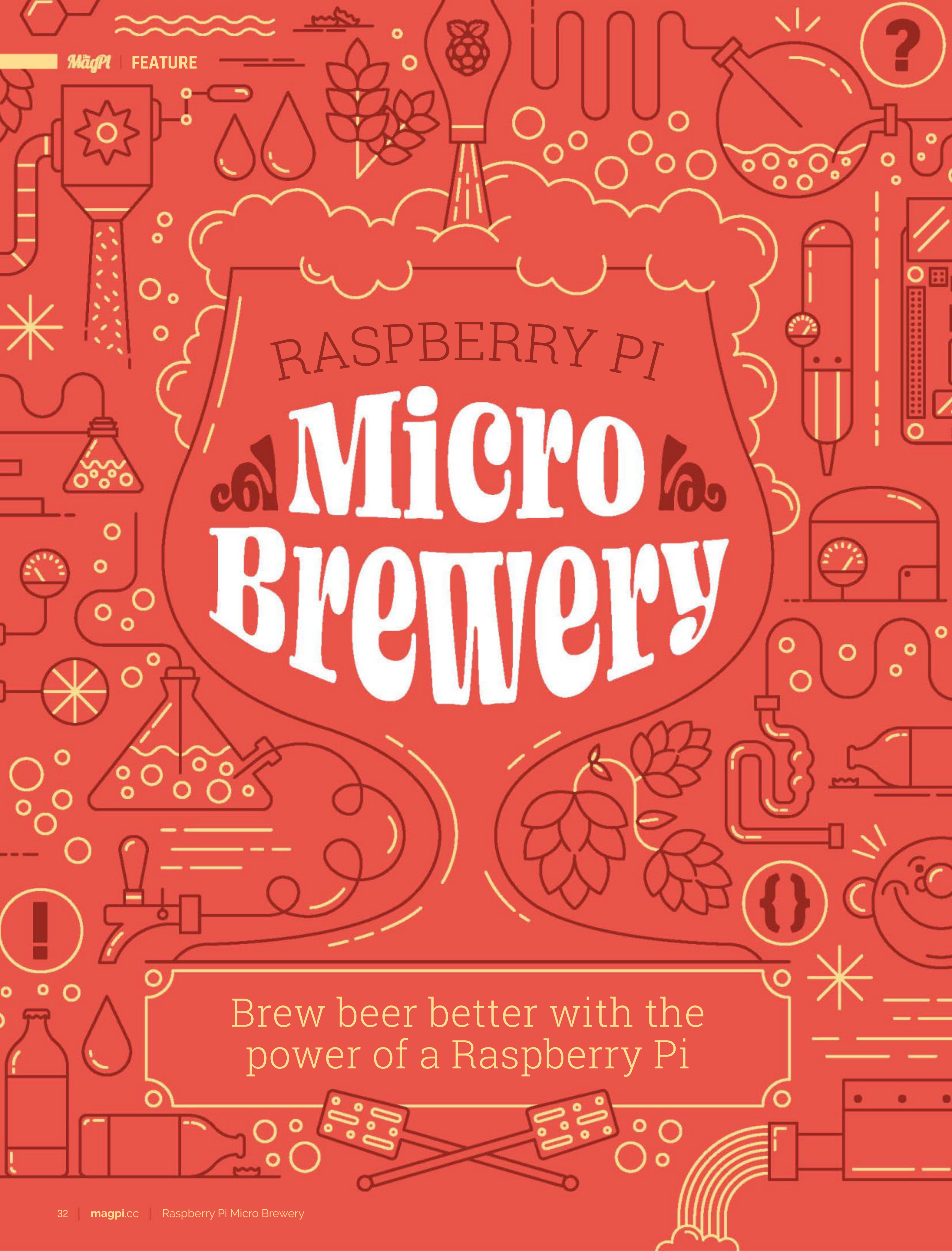
- IEEE802.3af compliant
- Mode A(Endspan), Mode B(Midspan)
- Circuit Protection (OV,OC)

Reseller suggestions
and inquiries

direct@wiznet.io

Available at
WIZnet online store

eshop.wiznet.io



RASPBERRY PI Micro Brewery

Brew beer better with the power of a Raspberry Pi

Brewing beer is an ancient practice and not a huge amount has changed with it in thousands of years. Ingredients may have changed a bit, technology may have made it easier, but it's essentially the same process humans have been performing for most of recorded history.

Thanks to this, we know a lot about what makes beer good, and part of that is sensing temperature and liquid density – things we can do with Raspberry Pi! This makes controlling your brew better, cheaper, and easier; just in time for those summer BBQs.

Tools and ingredients

Everything you need to start brewing

Raspberry Pi

You'll need a Raspberry Pi model with built-in Wi-Fi and Bluetooth – this includes Raspberry Pi 3, 3B+, 4, Zero W, and Zero 2 W. You'll be using it headless (without a monitor), so you just need a power supply and an appropriate case.

Tilt hydrometer | tilthydrometer.com

Hydrometers were one of the early technological advancements brought to beer brewing. This device measures density and specific gravity of the liquid it's in. Tilt is a very fancy wireless version that works with Raspberry Pi for more accurate beer brewing, and also senses the temperature of the brew as well.

30-litre beer brewing bucket | magpi.cc/brewbucket

You need to make sure it comes with a lid, grommet, and airlock. You might need to buy the airlock separately if you get a different one.

Siphon

magpi.cc/siphon

Bottles, or other form of storage

Home brew steriliser

magpi.cc/steriliser

Mixing paddle

magpi.cc/paddle

Beer brewing guide

Time to make some Raspberry Pi-powered beer

Get your Tilt Pi ready

01 Installing Tilt Pi

You'll need to install Tilt Pi from another computer with an SD card slot (or USB adapter). You can grab the SD card image for Tilt Pi from **magpi.cc/tiltpi** and, with Raspberry Pi Imager (**magpi.cc/imager**) you can burn it to the SD card.

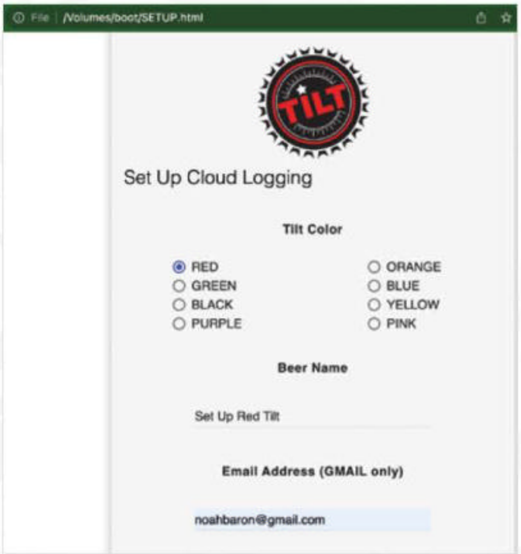
Once you've burnt the image to the card, navigate to the SD card on your system (it should be named 'boot') and open the **SETUP.html** file in your browser. From here, you can enter Wi-Fi settings and an email address for the system to send your cloud login to monitor your Tilt, along with a link to the Tilt Pi dashboard.

02 Connecting Tilt

Power-on your Tilt, put it in a glass of water, and then put the SD card into your Raspberry Pi and boot it up. After a couple of minutes, it will detect your Tilt and then send you the email link mentioned before. From here, you can set names and change some of the other settings that show up on the cloud dashboard. You can also maintain a local log file on Raspberry Pi which you can download any time.

Top Tip

You can also connect Tilt to a smartphone using the Tilt app: **magpi.cc/tiltapp**



Set Up Cloud Logging

Tilt Color

● RED ○ ORANGE
○ GREEN ○ BLUE
○ BLACK ○ YELLOW
○ PURPLE ○ PINK

Beer Name

Set Up Red Tilt

Email Address (GMAIL only)

noahbaron@gmail.com

Step 01: Editing the SETUP file

Step 02: Submerge Tilt while powering it up

Get brewing

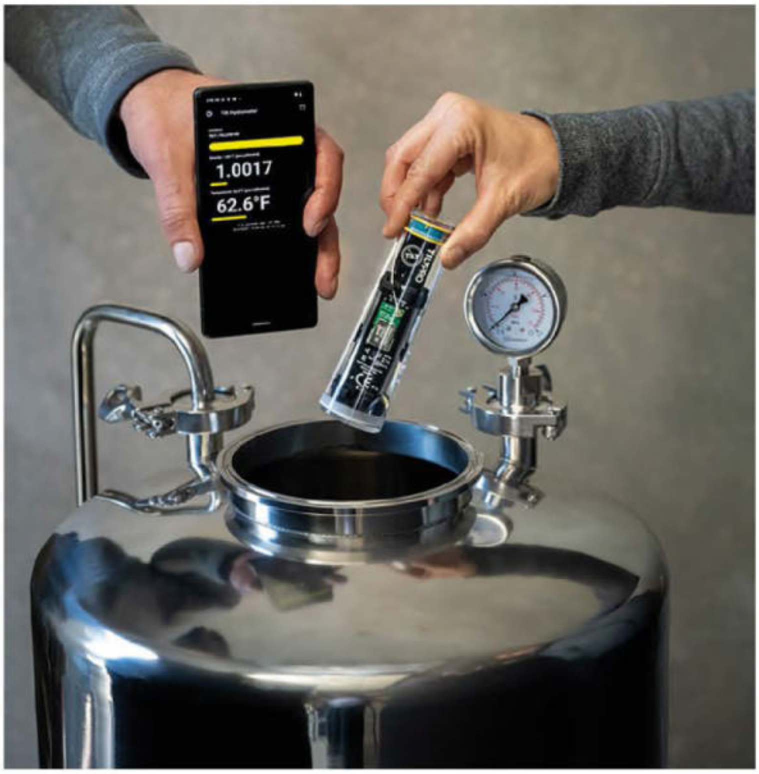
01 Sterilise, sterilise, sterilise

Everything you're going to use in the brewing process needs to be sterilised thoroughly, apart from the pan you're going to use to warm up the malt can in the next step. Follow the recommended instructions on the sterilising agent before wiping down all of the bucket and its lid, grommet, and airlock, and even the outside of your Tilt hydrometer.

You don't need to be NASA clean-room thorough, but it does help with the home brewing kits – if you plan to go from scratch, you can get a little more leeway (see page 39 for more on that.)

02 Prepare the ingredients

When using a brew kit, you'll get a can of malt to add to the mix. Before emptying it into the bucket, you'll need to warm it up first so that it softens. Do this by placing it into a container of hot water, such as a pan, for about 20 minutes. It shouldn't be boiling – about 65–70°C should do the trick.



Step 04: Add Tilt to your brew bucket

Once it's warmed up, pour it into the bucket. If some malt remains, rinse it with hot water to get the rest out – you'll be adding more water to it soon anyway, so it's perfectly fine to do this.

> You'll get a can of malt to add to the mix

03 Fill up the bucket

Depending on your brewing kit, you may need to add some sugar at this step for the yeast – check the instructions on your specific kit. Once that's done, top up the rest of the bucket with water, with the final temperature of the current mix being about 25°C.

If you need to maintain the 25°C temperature with a heating element, refer to the 'Maintain Temperature' box (overleaf) for how to do this with Raspberry Pi or a separate Raspberry Pi Pico.

Stir the mixture well, and then add the yeast to the bucket before stirring again.

04 Add Tilt

At this point, you can immerse your Tilt into the mix. You can keep an eye on its readings using Tilt Pi that was set up earlier – they should

BEER BREWING BASICS

The easiest way to get brewing is to make use of a beer brewing kit – it comes with all the necessary ingredients to get started pretty quick, and you can choose the kind of beer you like as well, without having to get into the deep beer science.

You can find these online or in stores that specialise in gifts. They'll include instructions that might slightly differ from this article, so make sure to keep that in mind when brewing.

Step 05: You'll need to wait a few weeks for the beer to ferment

change now it's out of the glass of water it was sitting in. You can use Tilt to keep an eye on the temperature of the water as well as the density.

Once that's done, you can replace the lid and make sure the airlock is filled halfway with clean water.

05 Good things...

Beer brews by means of fermentation by the yeast in the mix. Fermentation is not a quick process, so at this point you're going to need to leave your bucket of soon-to-be beer in a place where the ambient temperature is around 21–27°C (or keep the mix heated to that temperature, see the 'Maintain Temperature' box for more) for about five to seven days.

A cooler temperature means the fermentation will take longer; warmer, and it goes faster. If the airlock is bubbling a lot, then it may be too warm – you don't want it to try and ferment too fast.

Now put your feet up and wait...

Get bottling

01 ...come to those who wait

Your beer will be brewed once Tilt reads around 1.010 and the bubbles in the airlock have stopped. With analogue hydrometers, you would need to keep a reminder of the initial reading along with regular checks of the brew in a separate container. However, Tilt puts all this data online for you in a cloud database.

With these readings, you can also calculate the ABV of the beer using the simple formula:

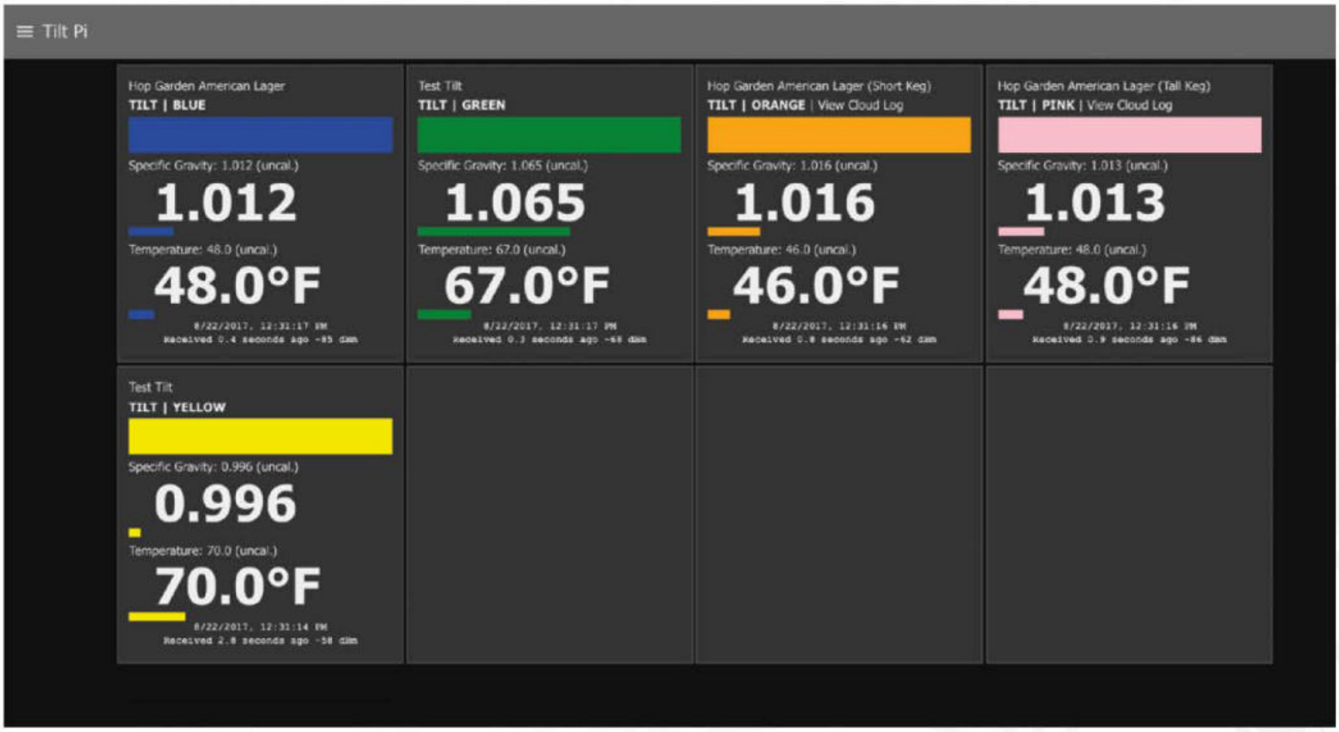
```
(Initial Gravity – Final Gravity) * 131.25
```

You can implement this formula in the spreadsheet with the data quite easily.

MAINTAIN TEMPERATURE

With Tilt, or another kind of temperature sensor, you are able to keep track of the temperature of the mix during the brewing process. With this data, you can easily have Raspberry Pi or Raspberry Pi Pico turn on a heating element to maintain a constant temperature. One such beer-based project for this can be found at **magpi.cc/craftbeerpi**.

This does involve the use of relays attached to heating elements, though, as they require a lot of power to run. Only build these if you're confident in your electronics skills, and be extraordinarily careful at the same time – or create your own script that uses an Energenie.



≡ Tilt Pi

Hop Garden American Lager
TILT | BLUE
Specific Gravity: 1.012 (uncal.)
1.012
Temperature: 48.0 (uncal.)
48.0°F
8/22/2017, 12:31:17 PM
Received 0.4 seconds ago -85 dBm

Test Tilt
TILT | GREEN
Specific Gravity: 1.065 (uncal.)
1.065
Temperature: 67.0 (uncal.)
67.0°F
8/22/2017, 12:31:17 PM
Received 0.3 seconds ago -68 dBm

Hop Garden American Lager (Short Keg)
TILT | ORANGE | View Cloud Log
Specific Gravity: 1.016 (uncal.)
1.016
Temperature: 46.0 (uncal.)
46.0°F
8/22/2017, 12:31:16 PM
Received 0.8 seconds ago -62 dBm

Hop Garden American Lager (Tall Keg)
TILT | PINK | View Cloud Log
Specific Gravity: 1.013 (uncal.)
1.013
Temperature: 48.0 (uncal.)
48.0°F
8/22/2017, 12:31:16 PM
Received 0.9 seconds ago -86 dBm

Test Tilt
TILT | YELLOW
Specific Gravity: 0.996 (uncal.)
0.996
Temperature: 70.0 (uncal.)
70.0°F
8/22/2017, 12:31:14 PM
Received 2.8 seconds ago -58 dBm

▲ Step 01: Monitor your brew on the Tilt Pi dashboard

02 Prepare storage

Whether you plan to put your brew in a barrel or into some bottles, you'll need to add some priming sugar, as per the instructions on your brew kit – it can be something like half a teaspoon of granulated sugar per pint.

Use the siphon to get the beer from the brew bucket to your storage vessels – you can use the one we listed on page 33 or do it the old-fashioned way with a small hose and lung power. Seal the barrel/bottles and keep them in a warm place for two to three days.

03 Optional additions

Before sealing, you can add extra bits at this stage to change the flavour and final product. This includes hop extract to enhance the hoppy smell and taste of the beer, heading agents so the head of the beer lasts longer, or a beer fining agent to make the clearing process faster – which we discuss in the next step.

> “ A layer of sediment will have formed at the bottom of the bottles and barrel ”

04 Clearing and sampling

The beer will need to go through a process of clearing and maturing before you can drink it – this is referred to as the secondary. Move the beer to a cool place and let it sit for two to four weeks – bottles will clear faster than a barrel.

Now you can sample the fruits of your labour. A layer of sediment will have formed at the bottom of the bottles and barrel. To pour a bottle, do so gently and in one go, making sure no sediment makes it into the glass.

For the barrel, sediment can build up near the tap, and the first pour can be quite high pressure, so be prepared to fill up about half a pint of a glass quickly before disposing. Then you can pour a full glass and see how you did. Enjoy!

Top Tip

Freshly brewed beer like this can keep for six months!

Advanced brewing

Tweak your ingredients for your preferred beer

CHANGE THE FLAVOUR

Different hops

There are many different varieties of hops you can use for brewing, all with different tasting notes and aromas that can really change the beer.

Keep an eye on the alpha acid percent (AA%) of any different hops you buy, though – a higher percentage means it can make the beer more bitter. A more bitter hop means you'll need less to create your wort – the bitterness scales linearly, though, so if you're using 50 g of 5% hops normally, but get 10% hops for a new brew, you'll need 25 g.

Different yeast

The strain of yeast can change the flavour and is used in different styles of beer brewing – some yeast is better for ale and others for lager. It's a good idea to look at how your favourite commercial beers are made to get an idea of how yeast might change up the flavour.

Filter your beer

The idea of sediment in your beer might not be quite to your liking, and we understand. After the clearing process of your beer in a barrel, you can filter it to another barrel, and this results in a cleaner and clearer beer. You will need to force-carbonate any beer that's been filtered though, which is an extra process of adding CO_2 to your barrel.

Add fruits, spices, and more

You can really jazz up your beer by adding any number of weird and wonderful ingredients while boiling your wort, or as you're bottling them. Berries, figs, cherries, cinnamon, ginger, vanilla, coconut; you can go wild so long as you think the flavours will go together.

You get a mellower taste by adding these extras to the last five minutes of the boil (which will also helpfully sterilise them), or during the secondary phase while it's clearing, after soaking them in little vodka or other spirit.

MALT EXTRACT BREWING

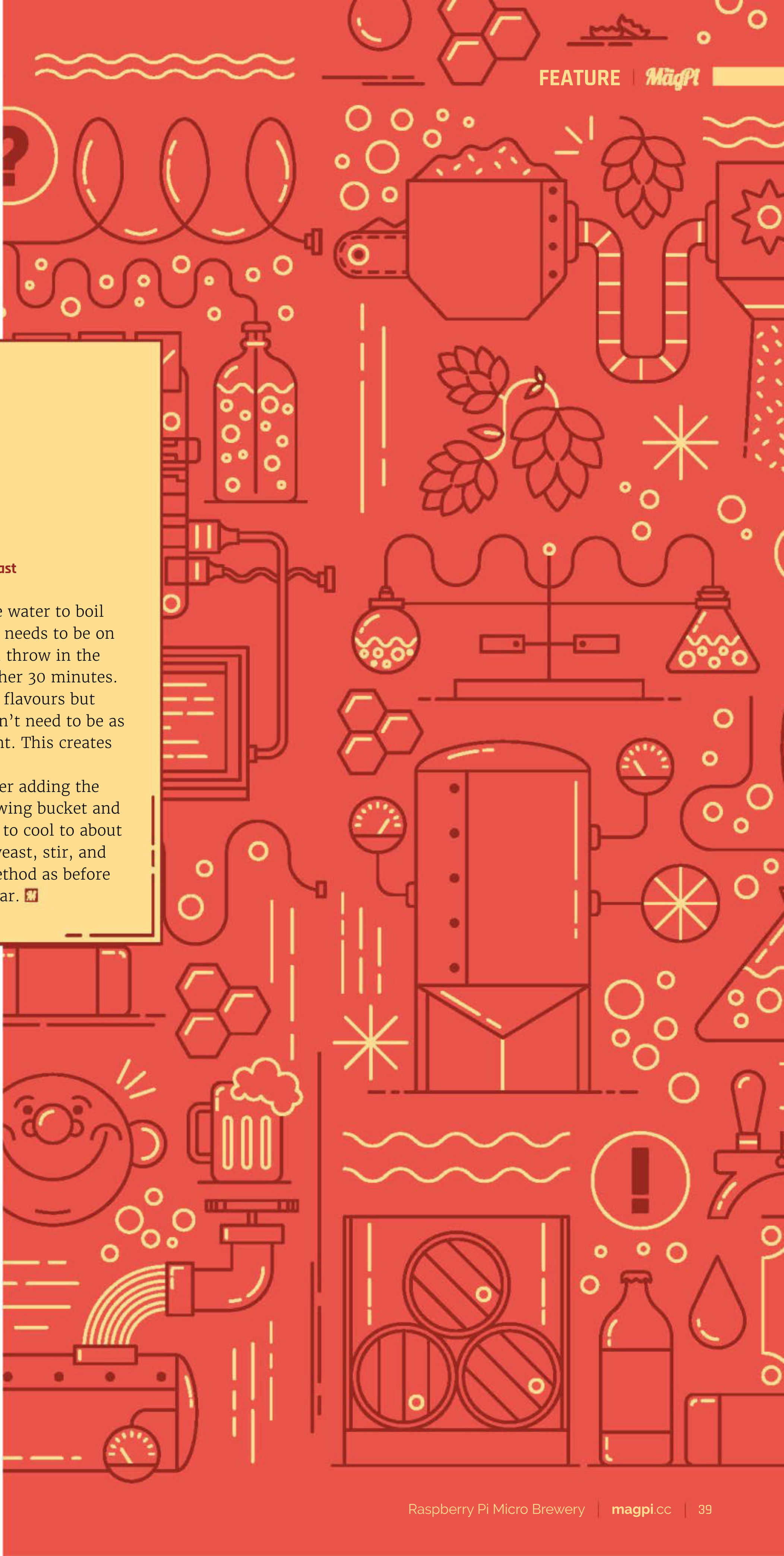
Ingredients

- **13 l water**
- **1 kg malt extract,** magpi.cc/malt
- **55 g dried hops,** magpi.cc/hops
- **750 g sugar**
- **Brewer's yeast,** magpi.cc/brewersyeast

In a large stock pot, bring half the water to boil and pour in the malt extract. This needs to be on the boil for 30 minutes before you throw in the dried hops, and then boil for another 30 minutes. This process not only extracts the flavours but also sterilises the brew, so you don't need to be as thorough with any sterilising agent. This creates what is called a wort.

After the 30 minutes are up, after adding the hops, strain the wort into the brewing bucket and add the rest of the water. Allow it to cool to about 25°C as before, and then add the yeast, stir, and add your Tilt. Follow the same method as before to allow it to brew, bottle, and clear.

> “ There are many different varieties of hops you can use for brewing, all with different tasting notes ”



Build an AI bedtime storyteller with Stable Diffusion

Tell Bedtime Stories who's the hero and where it happens, and the program creates an illustrated tale for you, using the magic of artificial intelligence

MAKER

Sean McManus

Author of *Mission Python*, *Web Design in Easy Steps*, and *Raspberry Pi For Dummies* (with Mike Cook). Get free chapters at Sean's website.

sean.co.uk

You'll Need

- ChatGPT account and API key **platform.openai.com**
- Pimoroni HyperPixel Square (optional) **magpi.cc/hyperpixel4**
- External speakers (optional)

Once upon a time there were two artificial intelligences, who were famed throughout the land. ChatGPT was an inventive storyteller, while Stable Diffusion was an ace illustrator. Together, they made up children's stories, which Raspberry Pi read aloud while showing the pictures. Listeners could put themselves, or their friends, in the stories, and decide where they should take place. Sometimes the stories were strange, and the pictures even more so. But as an art experiment, Bedtime Stories was a revelation. Anyone spying on the code could discover how to use ChatGPT and Stable Diffusion to create text and pictures from Python.

01 Get your DeepAI API key

DeepAI aims to make artificial intelligence (AI) accessible to new developers. The company provides an online text-to-image tool, accessed through an application programming interface (API). You send it a prompt (a request) and it sends you back an image. The API has been around since 2016 and its latest version uses the Stable Diffusion image generation software. Visit **deepai.org** to create your account. You can buy credit for 100 API calls for $5. On your Profile page, change the settings to allow unfiltered content. The filtered API rejected requests that included terms such as 'hand in hand' or 'creepy', resulting in incomplete stories.

02 Install Python modules

We'll need three Python modules for this project. OpenAI's ChatGPT is used to create the story text and the image prompts for DeepAI. An image prompt is a description of the image we'd like the AI software to make for us. The `requests` module is used to access Stable Diffusion through the DeepAI API. The `pyttsx3` module reads the stories aloud. Using speech in this project makes it feel like a friendly robot is reading you a bedtime story, but also means we can dedicate the screen to the story images. Open a terminal window and enter these instructions to install the Python modules and text-to-speech software. You may see some warnings about OpenAI not being on PATH, but you can safely ignore them.

```
pip install pyttsx3 requests openai
sudo apt install espeak mopidy
```

03 Create the ChatGPT prompt

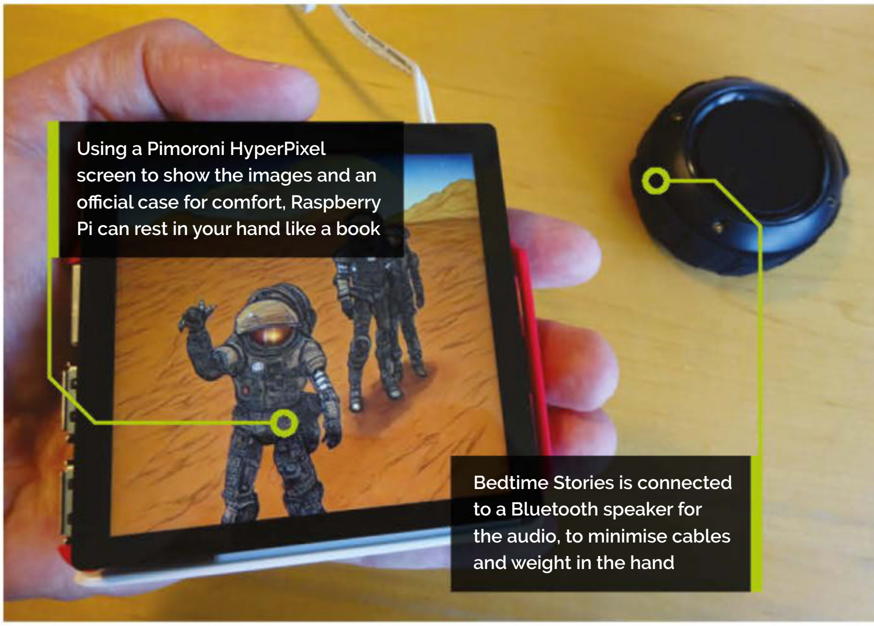
This project has two parts: Bedtime Story Maker (**story_maker.py**), and Bedtime Story Reader (**story_reader.py**). Stories are created and saved by the Maker and loaded and read by the Reader. Take a look at **story_maker.py**. When you run the program, it asks you to enter the character names and types, location, and genre for the story. It then builds them into a string called `story_prompt` that is used to ask ChatGPT to make the story. The story prompt also asks for an image prompt for every paragraph, starting with 'Image Prompt:'. The story prompt tells ChatGPT to suggest a book illustrator for each picture.

04 Using the ChatGPT API

Lines 33 to 39 send ChatGPT the story prompt. Add your own ChatGPT API key. We've told ChatGPT it is a children's author. Hints like this help to guide it to the best responses, based on the vast amount of online content it's been exposed to. The response comes in as a single piece of text, which should alternate between story paragraphs and image prompts. Lines 42 to 46 split that response into a list of image prompts and story paragraphs. Sometimes ChatGPT fails to return any image prompts, so the `while` loop keeps asking ChatGPT for stories until it also sends some image prompts.

> "Saving the stories means you can play them on demand, without waiting for them to generate"

ChatGPT came up with the idea for this female ghost with long, flowing hair and ethereal robes. Stable Diffusion made it real



Using a Pimoroni HyperPixel screen to show the images and an official case for comfort, Raspberry Pi can rest in your hand like a book

Bedtime Stories is connected to a Bluetooth speaker for the audio, to minimise cables and weight in the hand

05 Saving the story and image prompts

The story paragraphs and image prompts are saved to text files using the `write_list()` function. It accepts a list and a file name and saves each list item using the file name with a number on the end (e.g. **story-0.txt**, **story-1.txt**). They're saved in a folder named with the time, from the year to the second, such as 'Story 2023-04-10-09-34-52'. Saving the stories means you can play them on demand, without waiting for them to generate, as often as you like. It also enables you to study what worked well, and what didn't. ChatGPT wrote the first version of this function for us, by the way. It's good at basic Python programming, as long as you know how to guide it. We used the online front-end to ChatGPT for that at **chat.openai.com**.

06 Generate Stable Diffusion images

Next, the program goes through the list of image prompts and sends each one to DeepAI's API. You'll need to add your own DeepAI API key in line 59. To make the images more consistent, the program adds something to each prompt to clarify who the characters are and where the story

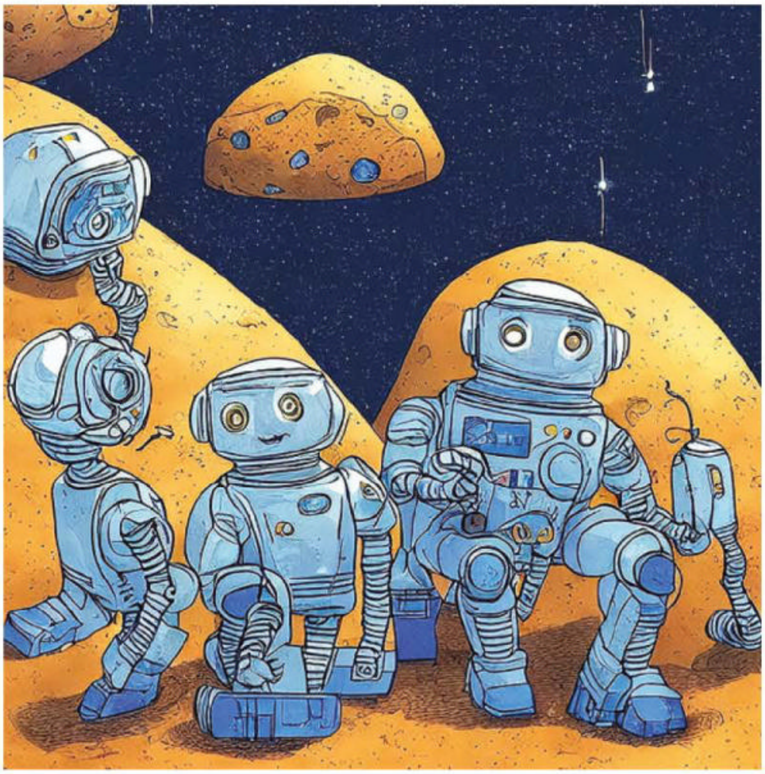
Top Tip

Watch your wallet

The DeepAI API requests are inexpensive, but if you make a lot of stories, it adds up fast.

Alert! Strange images

Stable Diffusion may produce images that are unsettling for young children, such as distorted faces.



The AI image generation software does particularly well with sci-fi images

takes place. We've also added a negative prompt to provide some hints that can help improve image quality. We asked for images that are 720×720, to match the size of the HyperPixel display you are using. Valid values are between 128 and 1536, but you might get odd results below 256 and above about 700. DeepAI returns your image as a URL, so the program downloads the file and saves it for you. It also displays the URL, so you can download the image (you've paid for!) manually if something goes wrong. You can find a simple version of the code to generate and download an image at **magpi.cc/seandeepai**.

Top Tip

Flip the images

If your display is upside down, you can flip your images by inserting this before line 29 in Story Reader:

```
image_to_show
= pygame.
transform.
flip(image_
to_show, True,
True)
```

07 Do your quality control

Use the Run button in Thonny to run **story_maker.py** a few times to make some stories. When it finishes, the story folder should contain story files (e.g. **story-0.txt**, **story-1.txt**); image prompt files (**prompt-0.txt**, **prompt-1.txt**); and an image file for each prompt (**1.jpg**, **2.jpg**, etc.). The Reader reads you the story paragraph aloud while it shows you the related image, so there should be the same number of image and story files. Sometimes there's an extra story file because ChatGPT adds a title, and the Reader can handle that. Occasionally, ChatGPT doesn't send enough image prompts, so there aren't enough pictures. In that case, you can still read the story and enjoy the illustrations yourself, but the Reader won't work. Find a valid story folder you want to play and copy its folder name.

08 Set up Bedtime Story Reader

Now you've made some stories, let's use the Bedtime Story Reader (**story_reader.py**) to enjoy them. Add the name of the folder where your story is stored in line 3 of the program.

09 Discovering the story files

After it's set up the Pygame window for displaying images, and the voice for reading them, the program indexes the story and image files. Their file names go into the `story_files` and `image_files` lists, which are then sorted into the correct order. If there are more story files than image files, the `story_has_title` variable is set to True.

10 Displaying the picture

The program goes through the `story_files` list and displays the associated image for each story paragraph. If the story has a title, the folder will be one image file short because there is no image prompt for the title. In that case, the first image needs to be shown for both the title and the first paragraph. Line 24 ensures this happens.

It's common for characters in AI-generated images to have odd-shaped heads, distorted faces, and an unconventional number of limbs. It all adds to the surreal effect

story_maker.py

Language: **Python**

DOWNLOAD THE FULL CODE:
magpi.cc/bedtimestories

```python
# Story Maker - by Sean McManus - www.sean.co.uk
import os, openai, requests, datetime, random, time

def write_list(list_to_write, filename):
    for number, paragraph in enumerate(list_to_write):
        filepath = os.path.join(directory, f"{filename}-{number}.txt")
        with open(filepath, "w") as file:
            file.write(paragraph)

date_time_str = datetime.datetime.now().strftime("%Y-%m-%d-%H-%M-%S")
directory = f"Story {date_time_str}"
os.mkdir(directory)
image_prompts, story_paragraphs = [], []
character1_name = input("What is your main character's name? ")
character1_type = input("What kind of a character is that? ")
character2_name = input("What is your second character's name? ")
character2_type = input("And what kind of a character is that? ")
venue = input("Where does the story take place? (e.g. in a castle, on Mars) ")
genre = input("What is your story genre? ")
story_prompt = f"Please write me a short {genre}\
story. In this story, {character1_name} is a\
{character1_type} and {character2_name} is a\
{character2_type}. The story takes place {venue}.\
For each paragraph, write me an image prompt for\
an AI image generator. Each image prompt must\
start in a new paragraph and have the words\
'Image Prompt:' at the start. Choose a book\
illustrator and put something in the image\
prompts to say the images should be made in the\
style of that artist."

while len(image_prompts) == 0:
    print("Making ChatGPT request")
    openai.api_key = "YOUR API KEY" ### PUT YOUR API KEY HERE
    ChatGPT_output = openai.ChatCompletion.
create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "You are a children's author."},
            {"role": "user", "content": story_prompt}
          ] )
     new_story = ChatGPT_output.choices[0].message["content"]
    print(new_story)
    for paragraph in new_story.split("\n\n"):
        if paragraph.startswith("Image Prompt"):
            image_prompts.append(paragraph)
        else:
            story_paragraphs.append(paragraph)
write_list(story_paragraphs, "story")
write_list(image_prompts, "prompt")

for number, image_prompt in enumerate(image_prompts):
    image_prompt += f"{character1_name} is {character1_type} and {character2_name} is {character2_type}. They are {venue}."
    print(f"Generating image {number}")
    r = requests.post(
    "https://api.deepai.org/api/text2img",
    data={'text': image_prompt,
          'negative_prompt': "poorly drawn face, mutation, deformed, distorted face, extra limbs, bad anatomy",
          'width': "720", 'height': "720",
          'grid_size': "1"},
    headers={'api-key': 'YOUR API KEY'} ### PUT YOUR API KEY HERE
    )
    image_url = (r.json()["output_url"])
    print(f"Image {number} is at {image_url}. Saving now...\n\n")
    filename = f"{number}.jpg"
    filepath = os.path.join(directory, filename)
    img_data = requests.get(image_url).content
    with open(filepath, 'wb') as handler:
        handler.write(img_data)
print(f"Your files are in {directory}.")
```

Here, Stable Diffusion has created a bright, colourful image that matches the look of a preschool book

It turns the list indices 0,1,2,3... into 0,0,1,2... and ensures we don't run out of images. If we had four story paragraphs (numbered 0 to 3) and three images (0 to 2), the last image and the last paragraph would coincide as they should. Images are scaled to fit the display. Although images are requested at 720×720, they come through at 768×768 because image sizes are accurate to the nearest multiple of 64.

11 Read out the story paragraph

Lines 32 to 35 open the text file for the story paragraph, read it in to the variable `story` and then use the text-to-speech module to read it aloud. For tips on customising the voice, see Raspberry Radio in issue 122 (**magpi.cc/122**).

12 Run the code

You can run Bedtime Story Reader using any display and in the desktop. Here we are using the HyperPixel Square display. To make the images fill the display, we boot to the command line and run the program using:

```
python story_reader.py
```

There is lots you can do to build on this project. You could add a menu for selecting stories, transitions between images or background music. The code shows you how to use AI to make text and images for any project, opening up a world of adventures.

story_reader.py

Language: **Python**

DOWNLOAD THE FULL CODE:

magpi.cc/bedtimestories

```
001. # Story Reader - by Sean McManus - www.sean.co.uk
002. # Put your story folder below
003. path = "Story 2023-04-10-12-35-04"
004. import os, pygame, pyttsx3
005. win_width, win_height = 720, 720
006. pygame.init()
007. windowSurface = pygame.display.set_mode((
     win_width, win_height))
008. pygame.mouse.set_visible(False)
009. voice = pyttsx3.init()
010. voice.setProperty('rate', 170)
011.
012. story_files, image_files = [], []
013. for file in os.listdir(path):
014.     if file.lower().endswith('.jpg'):
015.         image_files.append(file)
016.     elif file.lower().startswith('story'):
017.         story_files.append(file)
018. story_has_title = len(story_files) > len (
     image_files)
019. story_files = sorted(story_files)
020. image_files = sorted(image_files)
021.
022. for number, story in enumerate(story_files):
023.     if story_has_title:
024.         image_path = os.path.join(
     path, image_files[max(0, number - 1)])
025.     else:
026.         image_path = os.path.join(
     path, image_files[number])
027.     image_to_show = pygame.image.load(image_path)
028.     image_to_show = pygame.transform.scale(
     image_to_show, (win_width, win_height))
029.     windowSurface.blit(image_to_show, (0,0))
030.     pygame.display.update()
031.     story_path = os.path.join(path, story)
032.     with open(story_path, "r") as file:
033.         story = file.readlines()
034.     voice.say(story)
035.     voice.runAndWait()
036. pygame.quit()
```

English not your mother tongue?

The MagPi is also available in German!

elektor
MagPi
DAS MAGAZIN FÜR DIE RASPBERRY PI COMMUNITY
Das offizielle Raspberry Pi Magazin
März/April 2022 Nr. 2
magpi.de
BETRIEBS-SYSTEME
TÜFTELN, LÖTEN & PROGRAMMIEREN MIT DEM ZERO 2
RASPBERRY PI SMART GADGETS
+ ÜBERTAKTUNG BEIM RASPBERRY PI ZERO 2 W
57 SEITEN PROJEKTE UND PRAXIS
48 SEITEN PROJEKTE UND PRAXIS
#29 Juli/August 2022
Raspberry Pi Pico Grundlagen
LoRa mit Raspberry Pi Pico

Subscribe to the German edition of The 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

Part 04

Build an NeoPixel LED colour changer with dials

Learn about analogue inputs on the Raspberry Pi Pico using potentiometers. Use the value of the dials to change the colour of RGB LED NeoPixels

MAKER

Stewart Watkiss

Also known as Penguin Tutor. Maker and YouTuber that loves all things Raspberry Pi and Pico. Author of *Learn Electronics with Raspberry Pi*.

penguintutor.com

twitter.com/stewartwatkiss

In the world of computers, data is stored as digital values with electronic switches that turn on and off. The real world is analogue, with different levels of brightness throughout the day or different temperatures. This tutorial will shows one of the ways of taking analogue values from the real world and converting them to a digital signal that a Pico microcontroller can understand.

Knowing the value of the analogue inputs, these can then be converted into colour values which are used to light up NeoPixel LEDs.

01 Potentiometers

Potentiometers are a type of variable resistor, a resistor that changes in value. They have three terminals and often have a shaft that can be turned to change the resistance. You can also get miniature potentiometers which can be adjusted using a screwdriver.

Inside the potentiometer there is a resistive track with a slider that moves along it. As the slider moves towards terminal 1, the resistance between terminal 1 and terminal 2 decreases, and the resistance between terminal 2 and terminal 3 increases. A photo and schematic symbol are shown in **Figure 3** (overleaf).

02 Voltage divider

A common way of using a potentiometer is as a voltage divider. Connecting the voltage divider across the power supply, the output voltage will change as you turn the shaft. To understand how this works, we can imagine the potentiometer as two separate resistors, as shown in **Figure 2**. One resistor goes from terminal 1 to terminal 2 (R1), the other from terminal 2 to terminal 3 (R2). Connect terminal 1 to ground and terminal 3 to the supply voltage. The voltage output is proportional to the ratio between R1 and R2. For a 3.3V power supply, with the wiper towards terminal 1, the output will be 0V; with the wiper towards terminal 2, the output will be 3.3V; and halfway between, the output will be 1.65V. This provides an analogue output voltage.

03 Analogue to digital converter

If we want to use the analogue voltage as an input to a microcontroller, it needs to go through an analogue-to-digital converter (ADC). Fortunately, there are three inputs on the Pico which have built-in analogue-to-digital converters. These are physical pin 31 (ADC0), pin 32 (ADC1), and pin 34 (ADC2). These pins sample the voltage and provide a digital value for it. In

You'll Need

- 2 × half-size breadboards **magpi.cc/breadboardhalf**
- 3 × 10 kΩ potentiometers **magpi.cc/10kpot**
- 2 × NeoPixels **magpi.cc/smartneo**

MicroPython this is represented as a value between 0 and 65,536 (although the resolution of the ADC is 12 bits and not 16 bits as the range suggests).

04 Sampling the analogue value

To see this in action, connect the potentiometer to the Pico between pin 33 (GND) and pin 36 (3.3V output). Then connect the centre wiper pin to pin 31 of the Pico (ADC0). With just three lines of MicroPython code, you can query the value of the ADC port.

```
from machine import ADC, Pin
adc = ADC(Pin(26))
print (str(adc))
```

Running this will show a number representing the value at the pin. It's unlikely it will cover the full range, but you should be able to get a reasonable range starting at about 300 upwards.

If you divide the value received by 19,859 (roughly 65,535÷3.3), you should get a fairly accurate representation of the voltage.

05 Adding more potentiometers

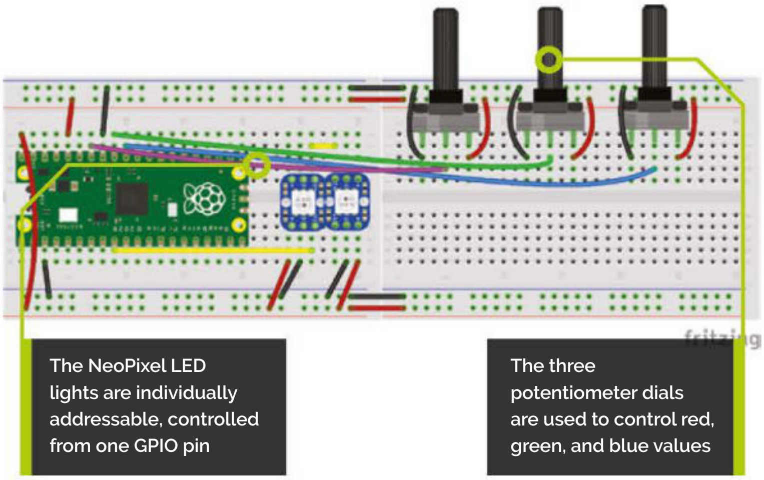
The colours we see on a computer screen are created using three primary colours: red, green, and blue. These are referred to as 'RGB'. To be able to represent different colours, we use three potentiometers, one for each of these colours. The potentiometers used for this can be plugged into a breadboard, but there isn't enough space on a single breadboard. Breadboards are designed to be joined, so you can use two breadboards connected (as in the **Figure 1** wiring diagram) or one full-size breadboard.

Alternatively, you can use potentiometers with wires connected to them, in which case they can be connected to the appropriate pin using just one breadboard.

06 Addressable RGB LEDs

To represent the colour output, we will be using addressable RGB LEDs. These are also known by their model number, such as WS2182B, or Adafruit refers to them as NeoPixels. These are often found on long strips, or made into rings, but you can also get them as individual pixels which can be connected to a breadboard. You can individually set the colour of each of the individual LEDs, but this example uses just two LEDs which will be set to the same colour.

To control the NeoPixels, you need to set the RGB colours, providing each value in the range 0 to 255. This can be achieved by dividing the ADC output by 257.



The NeoPixel LED lights are individually addressable, controlled from one GPIO pin

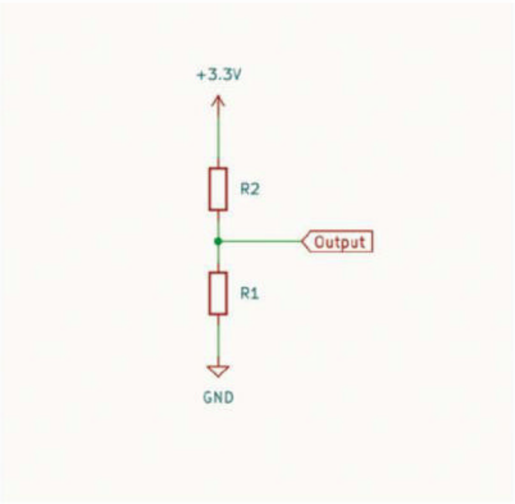
The three potentiometer dials are used to control red, green, and blue values

▲ **Figure 1** The breadboard wiring diagram for the circuit

> "We will be using addressable RGB LEDs"

07 How addressable RGB LEDs work

Addressable RGB LEDs are the combination of multicolour LED (with elements for red, green, and blue) and an integrated circuit used to determine the colour. The LED is often mounted directly on top of the integrated circuit (IC), making these appear as a single unit. The microcontroller sends a long string of values



◀ **Figure 2** The potentiometer is connected as a voltage divider, shown here as two resistors. The values of R1 and R2 will depend on the position of the wiper

Top Tip

From 3.3 V to 5 V

To increase the data voltage for the NeoPixels, you can use a voltage level shifter, such as the Adafruit bi-directional logic level converter.

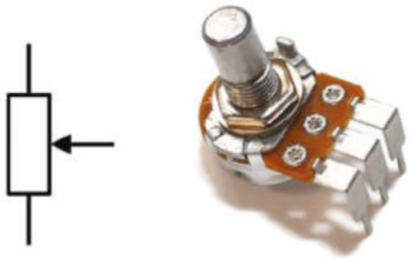


Figure 3 Schematic diagram and photo of a potentiometer. This type has right-angle pins that can be inserted into a breadboard

representing the desired colour for each LED. The IC reads the first value for setting its own LED and then passes the rest of the colour values along to the next LED. The microcontroller needs to generate precise timing for sending the values.

Top Tip

Other variable resistors

Swap the potentiometers for other variable resistors to detect the environment. A light-dependent resistor (LDR) with a 470 Ω fixed resistor is a good starting point.

08 Wiring up the NeoPixels

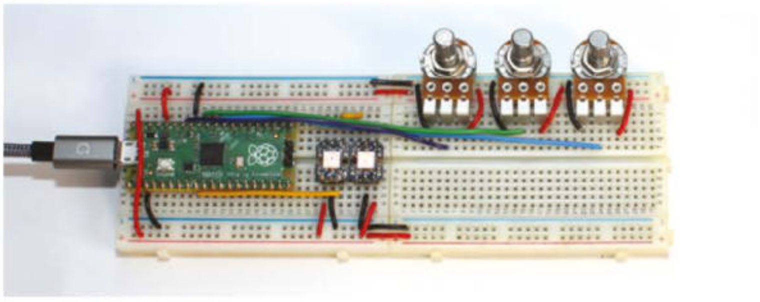
The LEDs shown in **Figure 4** are Adafruit breadboard NeoPixels. The input of the first NeoPixel (labelled 'In') needs to be connected to an output pin of the microcontroller – in this case, Pico's GPIO 6 (physical pin 9). The output from the first LED (labelled O) then needs to go the input of the second LED. To make the wiring easier, you can alternate the direction in which the NeoPixels are mounted so that the output of the first is on the same side of the breadboard as the input to the second NeoPixel.

09 Powering the NeoPixels

The potentiometers are connected to the 3.3V output of Pico (pin 36). This is important, as exceeding that could damage the input pins on Pico. To provide normal brightness levels the NeoPixels needs to be powered with 5V, so they need to be connected to VSYS (pin 40) on Pico.

The data input for the NeoPixels should ideally be 5V but, using Pico's GPIO pins, it will be 3.3V. When the NeoPixels are physically close to Pico, that works fine, but if they were positioned further away then it may be necessary to buffer the signal first.

Two breadboards have been joined to provide space for the three potentiometers to be mounted. The top power rail is 3.3 V for the potentiometers, the bottom power rail is 5 V for the NeoPixels



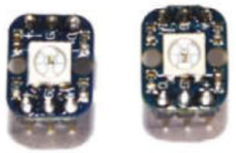


Figure 4 These NeoPixels can be inserted in a breadboard. There are three different coloured LEDs in the centre. The integrated circuit is hidden underneath the white square

10 Controlling the NeoPixels

To control the NeoPixels from your Pico, the signal needs precise timings. We don't need to get involved in the details, as the hard work of generating the correct signals is already included in a Raspberry Pi example program that can be found at **magpi.cc/neopixelringgit**.

The supplied code uses the Programmable Input Output (PIO) feature included in Pico. To use it, you just need to copy the appropriate functions into your own code, which is what has been done for the supplied code in **neopixel_rgb.py**.

11 How the code works

The first part of the code imports the required libraries and sets the values for the pin and number of LEDs. It then defines the three ADC inputs from the potentiometers.

The next section, marked with the comment block, is taken from the example code mentioned previously.

The code to read the values from the potentiometers and to then set the colours of the NeoPixels is contained within the `while True` loop. The three values are read separately and divided by 257 to convert them to 8-bit values. This is then combined into a tuple which is used to fill the LEDs with that colour. The `pixels_show` function is used to send the new values to the NeoPixels.

12 Other uses

The potentiometers could be changed for other devices that can measure the environment. This includes light-dependent resistors whose value changes based on the amount of light, or thermistors whose value changes based on the temperature (**Figure 5**). These would be wired as a voltage divider, replacing R1 with the variable resistor and R2 with a fixed resistor.

You could add more NeoPixels and set their values individually using the `pixels_set` function. You will need an external power supply when adding more NeoPixels.

neopixel_rgb.py

Language: **MicroPython**

DOWNLOAD THE FULL CODE:

magpi.cc/neopixelrgbpy

```
# Set RGB values based on analogue inputs
import array, time
from machine import ADC, Pin
import rp2

# Configure the number of WS2812 LEDs.
NUM_LEDS = 2
PIN_NUM = 6
brightness = 0.2

red_adc = ADC(Pin(28))
green_adc = ADC(Pin(27))
blue_adc = ADC(Pin(26))

@rp2.asm_pio(sideset_init=rp2.PIO.OUT_LOW, out_shiftdir=rp2.PIO.SHIFT_LEFT, autopull=True, pull_thresh=24)
def ws2812():
    T1 = 2
    T2 = 5
    T3 = 3
    wrap_target()
    label("bitloop")
    out(x, 1)               .side(0)    [T3 - 1]
    jmp(not_x, "do_zero")   .side(1)    [T1 - 1]
    jmp("bitloop")          .side(1)    [T2 - 1]
    label("do_zero")
    nop()                   .side(0)    [T2 - 1]
    wrap()

# Create the StateMachine with the ws2812 program, outputting on pin
sm = rp2.StateMachine(0, ws2812, freq=8_000_000,
sideset_base=Pin(PIN_NUM))
# Start the StateMachine, it will wait for data on its FIFO.
sm.active(1)
# Display a pattern on the LEDs via an array of LED RGB values.
ar = array.array("I", [0 for _ in range(NUM_LEDS)])

##########################################################################
def pixels_show():
    dimmer_ar = array.array("I", [0 for _ in range(NUM_LEDS)])
    for i,c in enumerate(ar):
        r = int(((c >> 8) & 0xFF) * brightness)
        g = int(((c >> 16) & 0xFF) * brightness)
        b = int((c & 0xFF) * brightness)
        dimmer_ar[i] = (g<<16) + (r<<8) + b
    sm.put(dimmer_ar, 8)
    time.sleep_ms(10)

def pixels_set(i, color):
    ar[i] = (color[1]<<16) + (color[0]<<8) + color[2]

def pixels_fill(color):
    for i in range(len(ar)):
        pixels_set(i, color)

while True:
    red = red_adc.read_u16() / 257
    green = green_adc.read_u16() / 257
    blue = blue_adc.read_u16() / 257

    # Get a single value
    color_value = (int(red), int(green),
int(blue))

    pixels_fill(color_value)
    pixels_show()
    time.sleep(0.25)
```

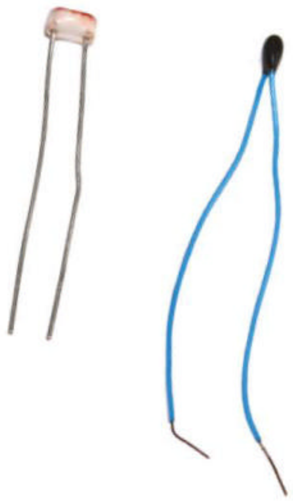


Figure 5 The light-dependant resistor and thermistor can be used to measure analogue light and temperature levels

Part 02

Camera Module: Still image capture

Your in-depth guide to the Raspberry Pi Camera Module's still image capture modes and settings

MAKER

David Plowman

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

raspberrypi.com

In *The MagPi* #129 we introduced libcamera-still which allows us to capture still images. But it also has many more options controlling how it captures images, and the file formats and available image encoders. We'll discover some of those in this tutorial. For now, we shall continue to assume that you have your keyboard and a monitor plugged directly into your Raspberry Pi.

Capturing images at different resolutions

By default, libcamera-still will capture images at the maximum available resolution supported by the camera. For the HQ Camera, for example, this means images that are 4056×3040 pixels in size. But it's easy to change this with the `--width` and `--height` parameters. For example, if you want to capture an image that is 1536×1024 pixels instead, you should use:

```
libcamera-still --width 1536 --height 1024
-o smaller.jpg
```

Autofocus is available on Raspberry Pi Camera Module 3, enabling you to focus in on specific parts of an image

Image encoders and file formats

Images are normally encoded in some way (often they are compressed so that they aren't so large) and then saved to a standard file format on the disk.

JPEG files

The default file format used by libcamera-still is JPEG because of the very wide support that it has, combined with a useful level of compression. Usually, JPEG files are given the extension .jpg, or sometimes .jpeg, as we have already seen.

JPEG is a lossy compression format, meaning that the result of loading your JPEG file is no longer identical to the image that you started from, though the differences are arranged to be ones that you will least notice. In return, you get to specify a quality factor indicating how much compression you want. At very low values your image will indeed deteriorate, but at very high values the JPEG process will be visually lossless; that is, it really will look as good as the original camera image.

The quality factor ranges from 1 to 99, and if you don't give one it will choose the value 93, which equates informally to 'quite high quality'. To save a JPEG at a different quality level, use the `-q` (or equivalently `--quality`) parameter like this:

```
libcamera-still -q 80 -o test.jpg
```

Experiment with quality values like 10 and 50 to see what you get!

PNG files

PNG (or 'Portable Network Graphic') is another file and compression format which enjoys wide support. PNG is a lossless compression format,

You'll Need

- Raspberry Pi
- Camera Module **magpi.cc/cameramodule**
- Raspberry Pi OS **magpi.cc/imager**

meaning that you are guaranteed to be able to recover exactly the same image that you started from (unlike JPEG). On the downside, PNG files are normally larger than JPEG files, and they are harder work to create, which takes longer.

To save a PNG file, you'll need to tell libcamera-still that you want to use a different encoder using the `-e` or `--encoder` parameter, like this:

```
libcamera-still -e png -o test.png
```

Note that it's necessary to specify the `-e` option to get a PNG file – changing the file name on its own is not enough. PNG does not support a quality parameter.

DNG files

DNG, or 'Digital NeGative', files are quite different from PNG files. As the full name suggests, they're somewhat analogous to the 'negatives' we had from film cameras before developing them into photos. In our case, the DNG file stores the raw numbers received from the image sensor before the hardware on Raspberry Pi 'develops' it into a viewable (JPEG or PNG) image. DNG files are saved alongside the 'developed' JPEG or PNG version of the same image. To save a DNG file, use the `-r` (or `--raw`) parameter:

```
libcamera-still -r test.jpg
```

libcamera-still will automatically replace .jpg by .dng in the DNG file name, giving both **test.dng** and **test.jpg** in this case.

Many third-party software programs exist to 'develop' these DNG files interactively – a rather complex procedure beyond the scope of this guide.

> “ By default, libcamera-still will capture images at the maximum available resolution ”

The High Quality Camera Module is focused manually; the new Camera Module 3 has autofocus that can be controlled in software

By default, each camera will capture images at the maximum resolution. It is possible to adjust the capture resolution in software

Time-lapse captures

A time-lapse capture is where we capture an image at regular intervals, perhaps every minute, hour or day, and then reassemble them into a video where we play them back at a much faster rate. libcamera-still is all set up to capture the images we need out of the box.

We need to introduce the `--timelapse` option and we'll review a couple that we've seen before.

- **`-t`** or **`--timeout`**. The length of time in milliseconds for which to perform a capture. In the case of timelapse, libcamera-still will run capturing images in total for this duration. You can pass the value 0 which means 'run indefinitely' (you will have to stop libcamera-still manually, for example by pressing **CTRL+C** on the keyboard).
- **`--timelapse`**. The length of time, again in milliseconds, between each of the time-lapse captures.
- **`-o`** or **`--output`**. The name of the output file or files. For time-lapse captures, we can't give all the images the same name so we use a special syntax that includes an image counter. For example, `-o capture_%04d.jpg` means that all the files are named 'capture_', followed by a counter, and then .jpg. `%04d` specifies how the counter is formatted, in this case the 0 means to add leading zeroes to the number and the 4 means 'so that every number has at least 4 digits'. This is useful so that listing your image files will return them to you in chronological order.

Raspberry Pi Camera Modules are capable of capturing images in a variety of different formats including JPEG, PNG, and DNG

Let's try an example.

```
 libcamera-still -o capture_%04d.jpg
--timelapse 5000 -t 30000 --width 1024
--height 768
```

This will run for 30 seconds, capturing an image every 5 seconds, and they'll be called **capture_0000.jpg** through to **capture_0004.jpg**. Note how we've reduced the resolution of the images to something that is more appropriate to the final video that we want to create.

Assembling your Images into a Video

There's a handy tool called FFmpeg which is capable of turning your sequence of still images into a video. We can use it like this:

```
ffmpeg -r 2 -i capture_%04d.jpg video.mp4
```

Note how we format the name of the input files with the special `%` syntax in the same way as we did for libcamera-still. The `-r` parameter gives the frame rate of the output video, 2 frames per second in this case. We've chosen the output file to have the MP4 format and called it **video.mp4**.

FFmpeg is a highly versatile tool that we recommend learning about.

Capturing when a key is pressed

Rather than doing regular captures, we can also do them in response to an event, or a key press. There are a couple of new option parameters to learn.

- **`--datetime`**. Use this instead of `-o` to name the output file after the current date and time. The format will be **MMDDhhmmss.jpg**, where MM and DD are the month and date number, and hh, mm and ss are hours, minutes and seconds.
- **`-k`** or **`--keypress`**. Capture an image when **ENTER** is pressed on the keyboard. Type **X** and press **ENTER** to quit.

HDR (high dynamic range) can be activated on Camera Module 3 to enable higher levels of contrast in your captured images

So the command to use is this:

```
libcamera-still -t 0 --keypress --datetime
```

Here we're running the capture indefinitely, so we'll have to type **X** followed by **ENTER** to quit (or press **CTRL+C**). Files would have names like 0405102742.jpg, meaning '10:27am, and 42 seconds, on 5 April'.

Capturing in response to a signal

For those familiar with Linux signals, an alternative to pressing a key is to send a signal instead. To do this, simply use `-s` or `--signal` instead of `-k` (or `--keypress`).

To send a capture signal to libcamera-still, first start it and then type the following into another terminal window:

```
kill -SIGUSR1 `pidof libcamera-still`
```

And you can force libcamera-still to quit with:

```
kill -SIGUSR2 `pidof libcamera-still`
```

Autofocus and High Dynamic Range

Autofocus and High Dynamic Range imaging are supported at the time of writing only on Raspberry Pi Camera Module 3.

Autofocus

When using the Camera Module 3, autofocus is enabled automatically in continuous mode. This means that the camera lens will move whenever necessary to maintain optimal focus on the centre part of the image, and this is probably what most users will want most of the time.

> It's also possible to turn off autofocus and set the focus position of the lens

It's also possible to turn off autofocus and set the focus position of the lens manually. To do this, use the `--lens-position` parameter, and pass it a value measured in dioptres, meaning the reciprocal of the focus distance. Thus, to focus at a very close distance of about 0.1m, pass in the value 10 (which is 1 / 0.1). To focus at infinity, pass in 0 (informally, the value of 1 / infinity). You can pass in non-integer values too. For example:

```
libcamera-still --lens-position 0 -o
infinity.jpg
```

...will set the focus position to infinity and not move the lens again.

High Dynamic Range imaging

The Camera Module 3 supports High Dynamic Range (HDR) imaging. To use it, specify the `--hdr` option on the command line, for example:

```
libcamera-still --hdr -o hdr.jpg
```

Note that non-HDR captures can be performed at a maximum resolution of 4608×2592 pixels, but HDR captures, because of the special nature of the sensor required to support HDR, are limited to 2304×1296 pixels (exactly half the width and height of the non-HDR mode).

Part 02

Run an open-source conversational AI

Host your own local conversational AI using an 8GB Raspberry Pi and Stanford University's Alpaca model

MAKER

K.G. Orphanides

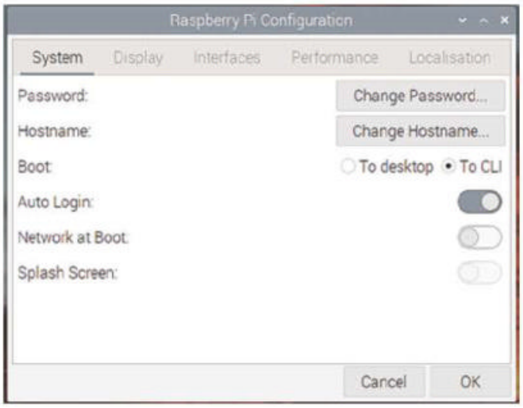
KG is a writer, developer, and software preservationist.

magpi.cc/owlbear

You'll Need

- llama.cpp **magpi.cc/llamacpp**
- Alpaca LLM model **magpi.cc/alpaca**

Last month, we built a chatbot interface using OpenAI's gpt-3.5-turbo model. This is effective, but there are numerous problems in using OpenAI's models. Despite the 'OpenAI' moniker, the company is highly proprietary, and most users aren't allowed to download a copy of the model to use on their own hardware, or even to examine the source code or training data. It's a black box, and this is an intentional decision on OpenAI's part. So now we've got to grips with using and prompting the most popular AI chatbot, it's time to go open source.



Password:

Hostname:

Boot: ○ To desktop ● To CLI

Auto Login: on

Network at Boot: off

Splash Screen: off

We don't need a GUI where we're going. Open Preferences > Raspberry Pi Configuration and set Raspberry Pi to boot to CLI to free some resources

01 Open-source LLM resources

Once you escape OpenAI's walled garden, you'll have a wealth of choices when it comes to LLMs (Large Language Models – the text-token predictors used for conversational chatbots and text generation engines). You'll find a number of these on open-source AI platform, Hugging Face (**huggingface.co**), along with plenty of documentation, community forums, courses, and both free and paid-for hosting for your machine learning applications. Of course, you'll also find open-source machine learning models, tools, and applications hosted on the usual platforms such as GitHub, GitLab, and Codeberg.

02 What we're looking for in a bot

For a chatbot, there are a few obvious candidates when it comes to open-source LLMs. Ideally, both source code and tensor weights should be publicly available. Finally, although it's going to be slow, we want to be able to actually run it on Raspberry Pi hardware rather than using Raspberry Pi as an interface, as we did in last month's GPT-3.5 chatbot tutorial.

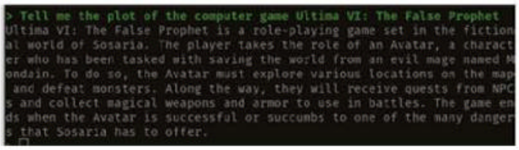
03 Choose your fighter

EleutherAI's GPT-J-6B is a fantastic conversational AI, but unless you happen to have a workstation or server packing some of the most powerful GPU hardware on the market, you'll need to host it elsewhere. KoboldAI, designed to help you create fantasy scenarios, is open, great fun, and a little easier on resources than GPT-J, but requires a GPU and x86 CPU architecture. Meta AI's LLaMA (Large Language Model Meta AI) is significantly more efficient, but although it is licensed under GPL v3, you currently have to apply (**magpi.cc/metaairequest**) to gain access to the model, and Meta is in no hurry to answer its emails. Fortunately, Stanford university has released Alpaca (**magpi.cc/alpacagit**), a fine-tuned version of the model, optimised for conversation, available for anyone to download.

> "For a chatbot, there are a few obvious candidates when it comes to open-source LLMs"

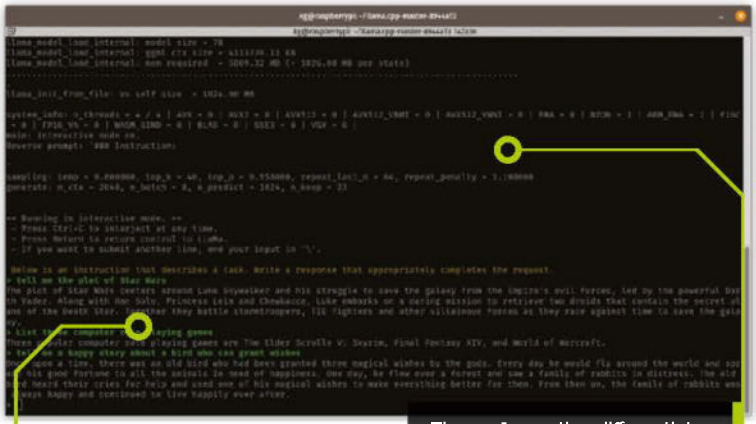
04 Run Alpaca on your CPU

Georgi Gerganov, developer of the whisper.cpp port we used for the transcriber we built in issue 128, has created llama.cpp, an inference system that allows users to interact with a tuned LLM. It's written in C++ and, critically for our purposes, uses the CPU, rather than requiring a GPU to run. All you have to do is drop in the models you want to use, appropriately converted to work with llama.cpp. Gerganov provides Python-based conversion tools models that you can use to make Meta's Llama models work correctly once you've got them. You can also download ready-to-use models for various other models, including Alpaca (**magpi.cc/alpacaggml**).



```
> Tell me the plot of the computer game Ultima VI: The False Prophet
Ultima VI: The False Prophet is a role-playing game set in the fiction
al world of Sosaria. The player takes the role of an Avatar, a charact
er who has been tasked with saving the world from an evil mage named M
ondain. To do so, the Avatar must explore various locations on the map
 and defeat monsters. Along the way, they will receive quests from NPC
s and collect magical weapons and armor to use in battles. The game en
ds when the Avatar is successful or succumbs to one of the many danger
s that Sosaria has to offer.
```

Alpaca can summarise even rather obscure information from its training corpus for you, but model 7B can be an extremely unreliable narrator



```
llama_model_load_internal: model size = 7B
llama_model_load_internal: ggml ctx size = 4113739.11 KB
llama_model_load_internal: mem required = 5809.32 MB (+ 1026.00 MB per state)
...................................................................................................
llama_init_from_file: kv self size = 1024.00 MB

system_info: n_threads = 4 / 4 | AVX = 0 | AVX2 = 0 | AVX512 = 0 | AVX512_VBMI = 0 | AVX512_VNNI = 0 | FMA = 0 | NEON = 1 | ARM_FMA = 1 | F16C = 0 | FP16_VA = 0 | WASM_SIMD = 0 | BLAS = 0 | SSE3 = 0 | VSX = 0 |
main: interactive mode on.
Reverse prompt: '### Instruction:

sampling: temp = 0.800000, top_k = 40, top_p = 0.950000, repeat_last_n = 64, repeat_penalty = 1.100000
generate: n_ctx = 2048, n_batch = 8, n_predict = 1024, n_keep = 23


== Running in interactive mode. ==
 - Press Ctrl+C to interject at any time.
 - Press Return to return control to LLaMa.
 - If you want to submit another line, end your input in '\'.

 Below is an instruction that describes a task. Write a response that appropriately completes the request.
> Tell me the plot of Star Wars
The plot of Star Wars centers around Luke Skywalker and his struggle to save the galaxy from the Empire's evil forces, led by the powerful Darth Vader. Along with Han Solo, Princess Leia and Chewbacca, Luke embarks on a daring mission to retrieve two droids that contain the secret plans of the Death Star. Together they battle stormtroopers, TIE fighters and other villainous forces as they race against time to save the galaxy.
> List three computer role playing games
Three popular computer role playing games are The Elder Scrolls V: Skyrim, Final Fantasy XIV, and World of Warcraft.
> Tell me a happy story about a bird who can grant wishes
Once upon a time, there was an old bird who had been granted three magical wishes by the gods. Every day he would fly around the world and spread his good fortune to all the animals in need of happiness. One day, he flew over a forest and saw a family of rabbits in distress. The old bird heard their cries for help and used one of his magical wishes to make everything better for them. From then on, the family of rabbits was always happy and continued to live happily ever after.
>
```

The -color option differentiates between your queries, in green, a pre-loaded prompt, in yellow, and our conversational bot's responses, in white

llama.cpp summarises the parameters you feed as it loads. Here, n_predict defines the maximum number of tokens our conversational bot can use in a response – and more than that get cut off

05 System requirements and expectation management

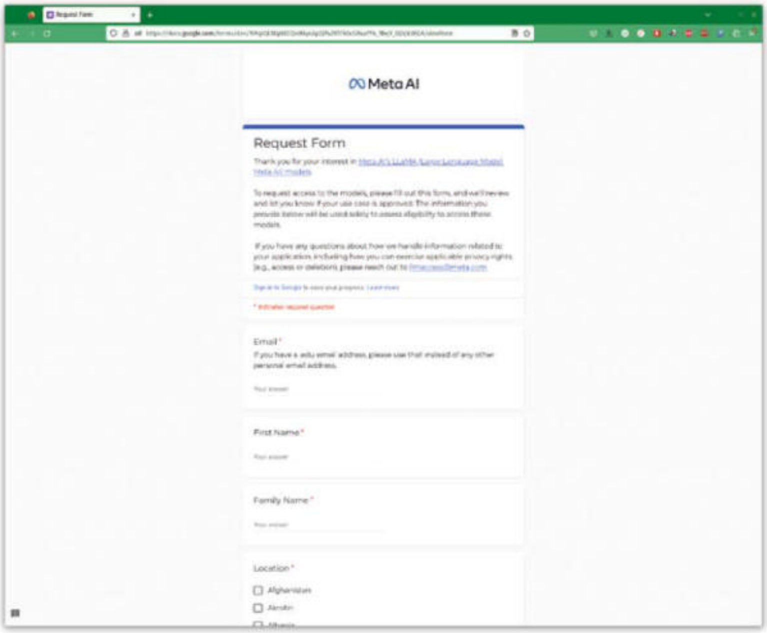
llama.cpp and Alpaca are efficient, but running a conversational LLM is still a big ask of a single-board computer – we got about one token per second. Ensure that you're running a 64-bit OS. For an easy life, you'll want a Raspberry Pi 4 with 8GB RAM. If you're using a 4GB model, you'll have to increase your swap size, but be aware that this can dramatically reduce the lifespan of your microSD card. If this is your situation, increase swap by opening a terminal, turning off the swapfile, editing its configuration to give yourself both a SWAPSIZE and MAXSWAP of 4096, and re-enabling it at the terminal, thus:

```
sudo dphys-swapfile swapoff
sudo nano /etc/dphys-swapfile
sudo dphys-swapfile setup
```

Top Tip

How was it trained?

LLaMA (and by extension Alpaca)'s training corpus only includes open-source material. Read more in Meta's paper at **magpi.cc/llamapaper**.



Meta AI

Request Form

Thank you for your interest in Meta AI's LLaMA (Large Language Model Meta AI) models.

To request access to the models, please fill out this form, and we'll review and let you know if your use case is approved. The information you provide below will be used solely to assess eligibility to access these models.

If you have any questions about how we handle information related to your application, including how you can exercise applicable privacy rights (e.g., access or deletion), please reach out to [illegible]

Sign in to Google to save your progress. Learn more

* Indicates required question

Email *
If you have a .edu email address, please use that instead of any other personal email address.

Your answer

First Name *

Your answer

Family Name *

Your answer

Location *

☐ Afghanistan
☐ Akrotiri
☐ Albania

If you want to use Meta's LLaMA models, you'll have to fill out a form requesting access and justifying your need to do so

Top Tip

What's a token?

Tokens are sections of words, usually around four characters each, that LLMs divide their text inputs and outputs into for processing.

06 Install llama.cpp

We've used the latest release at the time of writing, master-8944a13 (**magpi.cc/llama8944a13**), which we've confirmed functional on Raspberry Pi 4 8GB hardware running Raspberry Pi OS 64-bit. We can also confirm that version master-0f07cac is functional on Raspberry Pi 4 and 400 hardware. However, llama.cpp is seeing new releases several times a day, so we recommend choosing the latest from its GitHub release page (**magpi.cc/llamareleases**). Open a terminal and run the following to download and make llama.cpp, optionally substituting the latest release where we've used master-8944a13:

Stanford Alpaca

Stanford Alpaca: An Instruction-following LLaMA Model

This is the repo for the Stanford Alpaca project, which aims to build and share an instruction-following LLaMA model. The repo contains:

- The 52K data used for fine-tuning the model.
- The code for generating the data.
- The code for fine-tuning the model.
- The code for recovering Alpaca-7B weights from our released weight diff.

Note: We thank the community for feedback on Stanford-Alpaca and supporting our research. Our live demo is suspended until further notice.

Usage and License Notices: Alpaca is intended and licensed for research use only. The dataset is CC BY NC 4.0 (allowing only non-commercial use) and models trained using the dataset should not be used outside of research purposes. The weight diff is also CC BY NC 4.0 (allowing only non-commercial use).

We'll be using Stanford University's Alpaca model, a fine-tuned LLaMA variant that's available for anyone to use

```
 sudo apt install g++ build-essential
 wget https://github.com/ggerganov/llama.cpp/
archive/refs/tags/master-8944a13.tar.gz
 tar -xzf master-8944a13.tar.gz
 cd llama.cpp-master-8944a13/
 make -j4
```

07 Download Alpaca

We're going to use Stanford's Alpaca model. A version of the 4GB 7B model has already been converted into a format appropriate for llama.cpp. If you need to convert your own model, for example because you've been granted access to Meta's LLaMA models, follow the conversion instructions at **magpi.cc/llamarun**. For speed reasons, you can carry this out on a faster computer than Raspberry Pi and then copy over to your models directory. For the moment, return to your terminal emulator and:

> It responds particularly well to prompts

```
 cd models
 wget https://huggingface.co/Sosaka/Alpaca-
native-4bit-ggml/resolve/main/ggml-alpaca-
7b-q4.bin
 cd ..
 ./main -m models/ggml-alpaca-7b-q4.bin
--color -f prompts/alpaca.txt -ins --n_parts
1 --ctx_size 2048 --n_predict 1024
```

08 Options and parameters

Notice that we're using the `-m` option to select the model we wish to use; `-color` to colour the text to make it easier to differentiate between user and LLM text. `--ins` uses 'instruct mode', which Alpaca favours, prompting the user to provide instructions for the chatbot to follow. `-f prompts/alpaca.txt` tells it to load a supplied prompt file optimised for Alpaca's manner of interaction. `--n_predict` determines the maximum allowed response length, and `--ctx_size` is the session’s token cap. You can type `./main --help` to see a list of parameters that you can feed llama.cpp.

09 How to address Alpaca

Alpaca can answer questions, give you lists of information, and follow directions. It responds particularly well to prompts asking it to list items and tell you about things. We’ve had particular luck with prompts asking it to ‘tell me a story’ about something, and it’ll even have a decent stab if you ask it to ‘write me a poem’. The more detail you provide in your prompt, the more you’ll generally get back.

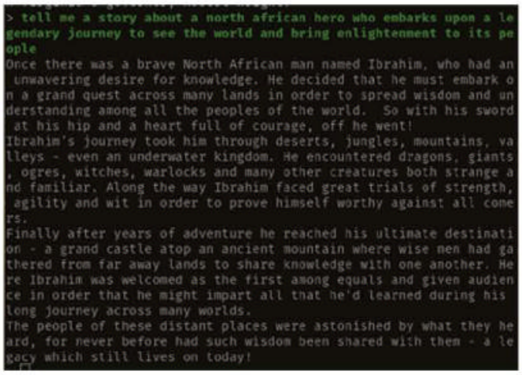
llama.cpp doesn’t give your chatbot a memory by default. Like gpt-3.5-turbo last month, you’d have to implement a method of returning previous questions and responses to it as part of every prompt, but that adds to its processing load.

10 Performance

On our Raspberry Pi 4 8GB, after initial processing time for the prompt, llama.cpp with Alpaca 7B was returning about one token per second or our initial at default clock speeds, with `arm_boost=1` enabled in **boot/config.txt** to set its clock speed to 1.8GHz. This slightly improved when we overclocked it, editing **config.txt** to add:

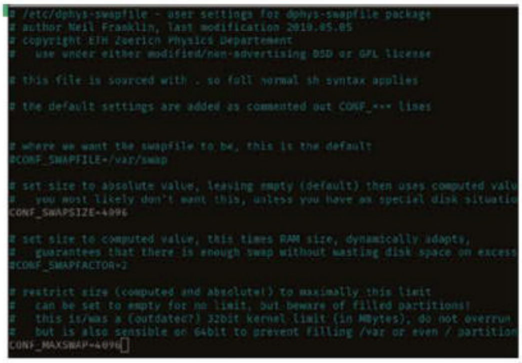
```
arm_freq=2000
over_voltage=6
```

You may be able to further overclock your Raspberry Pi. See our guide at **magpi.cc/overclock** to learn more.



```
> tell me a story about a north african hero who embarks upon a le
gendary journey to see the world and bring enlightenment to its pe
ople
Once there was a brave North African man named Ibrahim, who had an
 unwavering desire for knowledge. He decided that he must embark o
n a grand quest across many lands in order to spread wisdom and un
derstanding among all the peoples of the world.  So with his sword
 at his hip and a heart full of courage, off he went!
Ibrahim's journey took him through deserts, jungles, mountains, va
lleys - even an underwater kingdom. He encountered dragons, giants
, ogres, witches, warlocks and many other creatures both strange a
nd familiar. Along the way Ibrahim faced great trials of strength,
 agility and wit in order to prove himself worthy against all cone
rs.
Finally after years of adventure he reached his ultimate destinati
on - a grand castle atop an ancient mountain where wise men had ga
thered from far away lands to share knowledge with one another. He
re Ibrahim was welcomed as the first among equals and given audien
ce in order that he might impart all that he'd learned during his
long journey across many worlds.
The people of these distant places were astonished by what they he
ard, for never before had such wisdom been shared with them - a le
gacy which still lives on today!
```

Given a sufficient token threshold and a bit of time, Alpaca can generate surprisingly coherent mythic stories



```
# /etc/dphys-swapfile - user settings for dphys-swapfile package
# author Neil Franklin, last modification 2010.05.05
# copyright ETH Zuerich Physics Departement
#   use under either modified/non-advertising BSD or GPL license

# this file is sourced with . so full normal sh syntax applies

# the default settings are added as commented out CONF_*=* lines


# where we want the swapfile to be, this is the default
#CONF_SWAPFILE=/var/swap

# set size to absolute value, leaving empty (default) then uses computed valu
#   you most likely don't want this, unless you have an special disk situatio
CONF_SWAPSIZE=4096

# set size to computed value, this times RAM size, dynamically adapts,
#   guarantees that there is enough swap without wasting disk space on excess
#CONF_SWAPFACTOR=2

# restrict size (computed and absolute!) to maximally this limit
#   can be set to empty for no limit, but beware of filled partitions!
#   this is/was a (outdated?) 32bit kernel limit (in MBytes), do not overrun
#   but is also sensible on 64bit to prevent filling /var or even / partition
CONF_MAXSWAP=4096
```

If using a 4GB model, you’ll need to increase the size of the swapfile size

11 Licensing limitations

Both LLaMA and Alpaca are designed for academic research purposes only: LLaMA (and thus Alpaca) is distributed under a non-commercial licence. A non-commercial licence is fine for the purposes of this tutorial and personal experimentation, but if you’re hoping to develop your own customer service chatbot, look elsewhere. Alpaca’s instruction data is based on OpenAI’s text-davinci-003 – the T&Cs for that ban using it for the development of models that compete with OpenAI. Finally, Stanford warns that “we have not designed adequate safety measures, so Alpaca is not ready to be deployed for general use”, which means that Alpaca may generate disturbing or harmful responses.

12 Alpacas have personality

You’ll notice that Alpaca has a rather different tone to its expression than the gpt-3.5-turbo chatbot model we used last month. It’s less hyper-enthusiastic, less insistent of its own correctness, and generally feels less like it’s writing a LinkedIn post about the startup grindset hustle life. It’s better at telling stories than ChatGPT, with an intimate tone that may reflect the quantity of public domain literature from Project Gutenberg in LLaMA’s training data. It can answer direct, fact-based questions, sometimes correctly, but – especially if you’re using the smallest Alpaca 7B model – usually not. In any given answer, especially lengthy ones, there’s likely to be at least one error. Larger models may produce better results, but also require both more disk space and more RAM than we have readily to hand.

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
NEW 2022 UPDATE
RETRO GAMING WITH RASPBERRY PI
2ND EDITION
164 PAGES OF VIDEO GAME PROJECTS
+ BUILD AN ARCADE MACHINE
PLAY & CODE GAMES!

Experimental Pi
PiBoy DMG
SELECT
START

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

Master the **Linux command line interface**

Discover the adventure of learning Linux and the command line with Raspberry Pi OS. By **Lucy Hattersley**

Linux and Raspberry Pi form a powerful combination and learning to use Linux, and especially the command-line interface (CLI), is a vital computing skill.

In this feature, we're going to delve into the CLI. This is the text-based input and output system that sits underneath the more commonly used graphical user interface (GUI).

Mastering the command line gives you increased control over your computer, allowing you to automate tasks, manipulate files with ease, and troubleshoot issues. Linux skills are always in high demand because they make such a valuable asset in the tech industry.

Raspberry Pi serves as the perfect tool for navigating Linux and the command line. It is a low-cost computer and Raspberry Pi OS is built on top of a recent version of Debian Linux. Furthermore, the Raspberry Pi community is incredibly supportive, offering a wealth of resources and tutorials online to help you throughout your journey.

Let's get ready to unlock your potential as a Linux master.

Top Tip!

Directories

Folders are called 'directories' in Linux, although they mean the same thing.

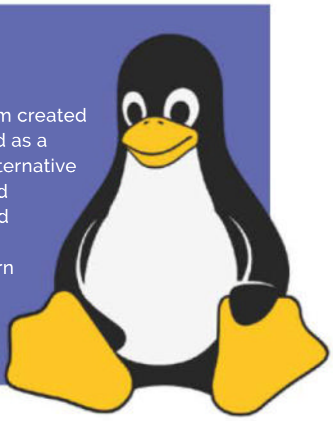
The Linux command line, also known as the command-line interface (CLI), is a powerful and efficient tool for interacting with your computer. The command line enables users to perform tasks quickly and precisely using text-based commands that can be chained together and scripted. This feature will cover the basics of using the CLI, including basic Linux commands, file management, piping commands, and accessing the command line remotely. To get started, you'll need to access the command-line interface. Typically you will use the Terminal application, but you can also use a full-screen teletype (tty) interface. Here are the ways to access the command line:

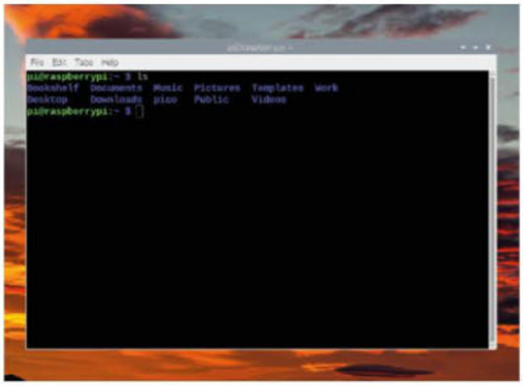
- Click on the black Terminal icon in the menu bar
- Choose Accessories > Terminal from the menu icon
- Press **CTRL+ALT+T** on the keyboard
- Press **CTRL+ALT+F1-F6** on the keyboard (full screen tty interface. Press **CTRL+ALT+F7** to return to Raspberry Pi OS interface)
- Access your Raspberry Pi remotely using SSH

Typically, when you start out, you'll use the Terminal app. This makes it easier to combine the Terminal with the more user-friendly Raspberry Pi OS graphical user interface (GUI). Click on the black Terminal icon in the menu bar to get started.

What is Linux?

Linux, the open-source operating system created by Linus Torvalds in 1991, was conceived as a free, robust, and highly customisable alternative to proprietary systems like Windows and macOS. Over the years, Linux has earned a solid reputation among developers, sysadmins, and tech enthusiasts. To learn more about Linux's inspiring legacy, visit the Linux Documentation Project (**magpi.cc/linuxhistory**).





```
pi@raspberrypi:~ $ ls
Bookshelf  Documents  Music  Pictures  Templates  Work
Desktop    Downloads  pico   Public    Videos
pi@raspberrypi:~ $
```

Using the **ls** command to list the contest of the home directory

Basic Linux commands

When starting out with the CLI, it's essential to become familiar with a few basic commands. First, you should see something similar to this:

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

The 'pi' is your username; after the @ symbol is the hostname of your computer. The '~' symbol is your current location ('~' is a short cut for your home directory, more on that later). Finally the '$' symbol signifies the part where you can enter commands.

These commands help you navigate and manipulate your files and directories. Some common commands are:

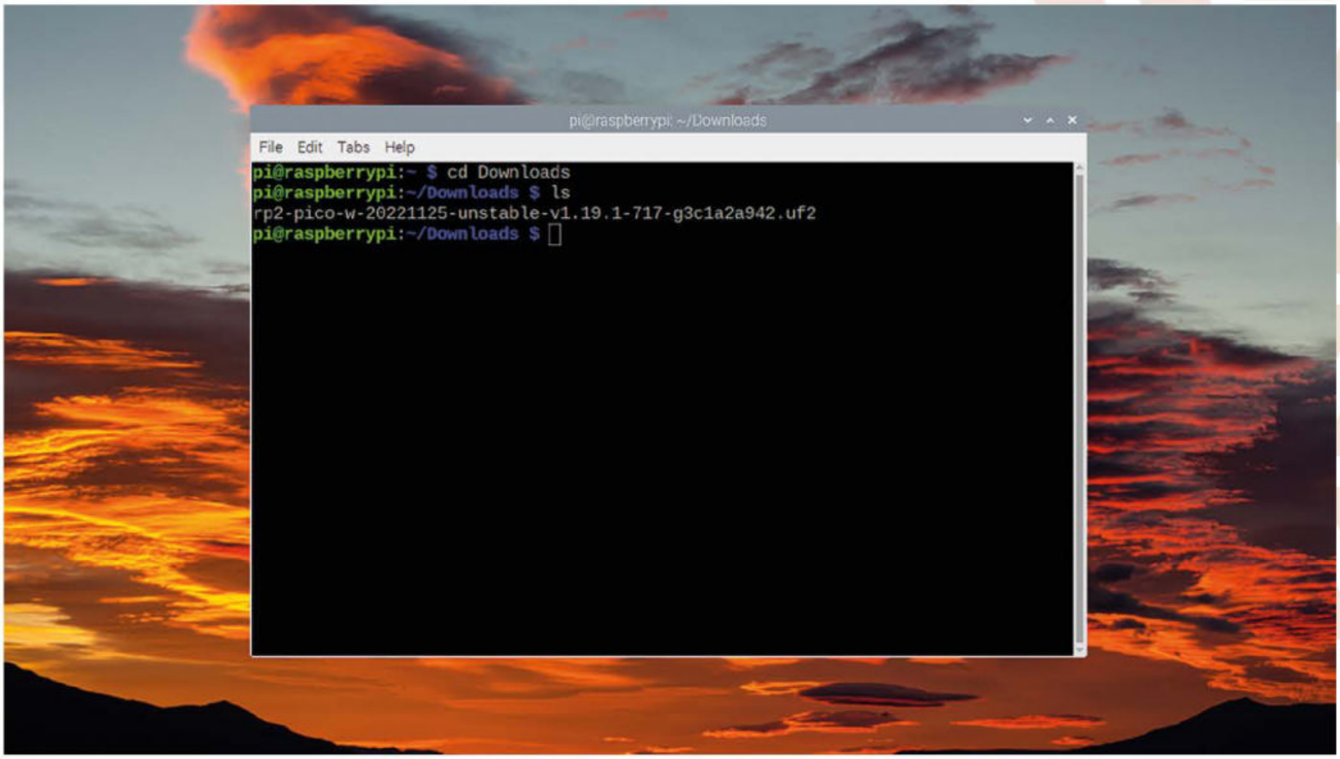
- `pwd` **(print working directory):** This command displays the current directory you are in
- `ls` **(list):** This command lists the contents of a directory
- `cd` **(change directory):** This command helps you navigate between directories
- `cp` **(copy):** This command copies files or directories
- `mv` **(move):** This command moves files or directories to a new location

Understanding your location

Understanding file paths is essential for navigating and manipulating files and directories in the CLI.

There are two types of path that can lead to our report in the command line: 'absolute' and 'relative'.

Here is an absolute path to a file called **report.txt**:



```
pi@raspberrypi:~ $ cd Downloads
pi@raspberrypi:~/Downloads $ ls
rp2-pico-w-20221125-unstable-v1.19.1-717-g3c1a2a942.uf2
pi@raspberrypi:~/Downloads $ 
```

▲ Using **cd** and **ls** to navigate into the **Downloads** directory and list the contents

```
/home/pi/Documents/report.txt
```

Notice the slash '/' character at the start. The slashes are similar to folders in the File Manager, and the very first slash is the 'root directory' that sits at the very base of your storage. The path then leads through the **home** directory into **pi** and into **Documents** before identifying the file (each '/' indicating a new folder).

A relative path indicates the location of a file or directory relative to the current working directory. For example, if your current working directory is **/home/pi** (the home directory), the relative path to **report.txt** would be:

```
Documents/report.txt
```

Notice the lack of a '/' at the start of the relative path.

Changing the file path

To locate this **report.txt** file in the File Manager, we would typically start in the **pi** folder and double click on **Documents** to open that folder.

To change the file path in CLI we us the `cd` (change directory) command. You can navigate using absolute or relative paths, depending on your preference. For example:

> " The Linux command-line interface is a powerful and efficient tool for interacting with your computer "

- **Navigate to an absolute path:** `cd /home/pi/Documents`
- **Navigate to a relative path:** `cd Documents`
- When navigating with relative paths, you can use the following shortcuts:
- **'.' (single dot):** Represents the current directory. For example, `cd ./Documents` would navigate to the **Documents** subdirectory in the current directory
- **'..' (double dots):** Represents the parent directory. For example, `cd ..` would move you up one level from the current directory

You will use `cd`, `ls`, and `cd ..` frequently to navigate around the file system

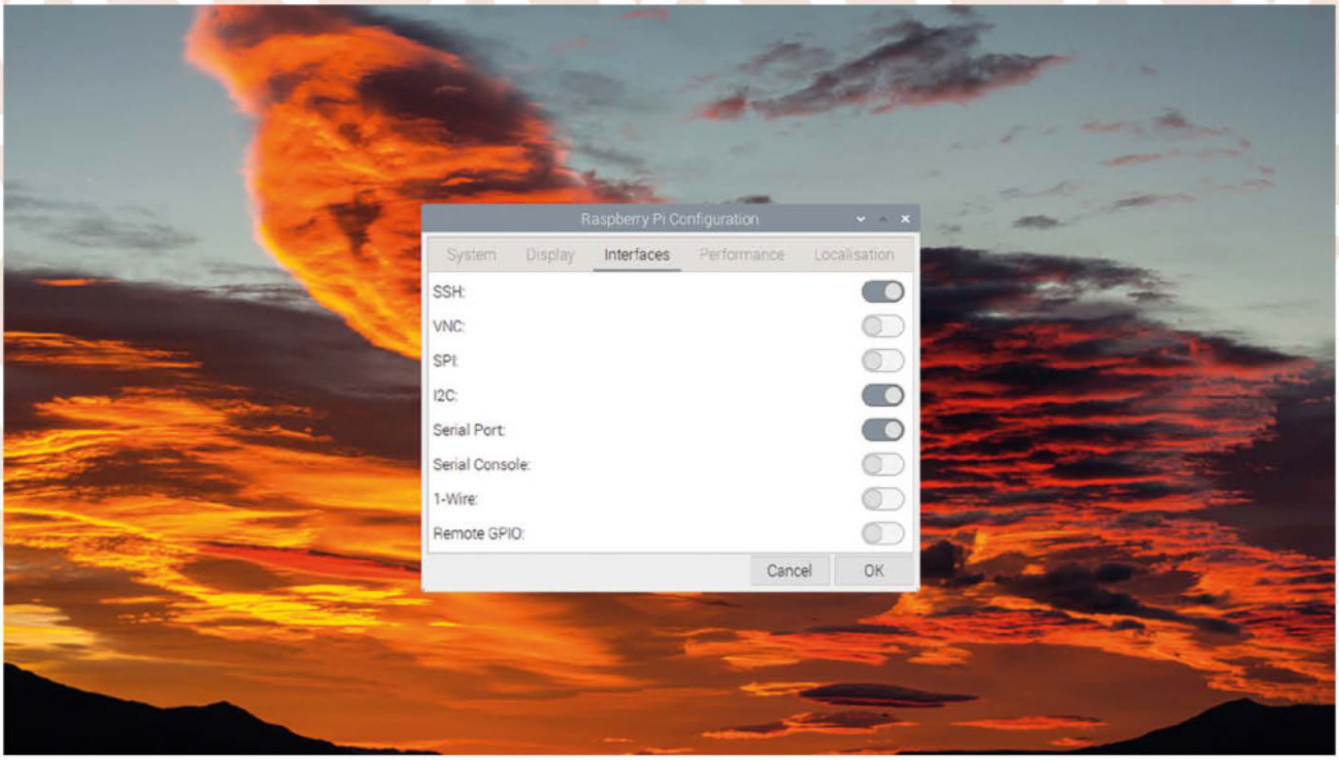
Moving back to the home directory

The home directory is the default directory for a user. To quickly navigate back to your home directory, you can use one of these commands:

Top Tip!

Press TAB!

Press the **TAB** key when entering commands or file and directory names. CLI will auto-complete to matching commands and save you typing out the whole thing.



SSH: ●
VNC: ○
SPI: ○
I2C: ●
Serial Port: ●
Serial Console: ○
1-Wire: ○
Remote GPIO: ○

▲ Use Raspberry Pi Configuration to enable the SSH interface

- `cd` **(without any arguments):** This command will take you back to your home directory
- `cd ~` **(the tilde ('~') symbol represents the home directory):** This command is equivalent to the previous one

Understanding file paths is crucial for efficient navigation and file management in the Linux environment.

File management

Creating, deleting, and moving files and directories are also fundamental tasks you'll perform on the command line. Some important commands for file management are:

- `touch`**:** This command creates an empty file
- `mkdir` **(make directory):** This command creates a new directory
- `rm` **(remove):** This command deletes a file or directory. Use `rm -r` for directories (the `-r` option stands for 'recursive')
- `cp` **(copy) and** `mv` **(move):** These commands copy and move files and directories, respectively

You'll use these commands with paths and file names. For example, to create a **reports.txt** file in the **Documents** folder, you would enter:

```
touch Documents/reports.txt
```

If you wanted to move it to a new directory in your home location, called 'Work' you would first create the directory:

```
mkdir Work
```

Then move the file using `mv` and both a source and destination paths.

```
mv Documents/reports.txt Work/reports.txt
```

You also use `mv` to rename a file as you are moving it from one file name to another.

Piping and redirection

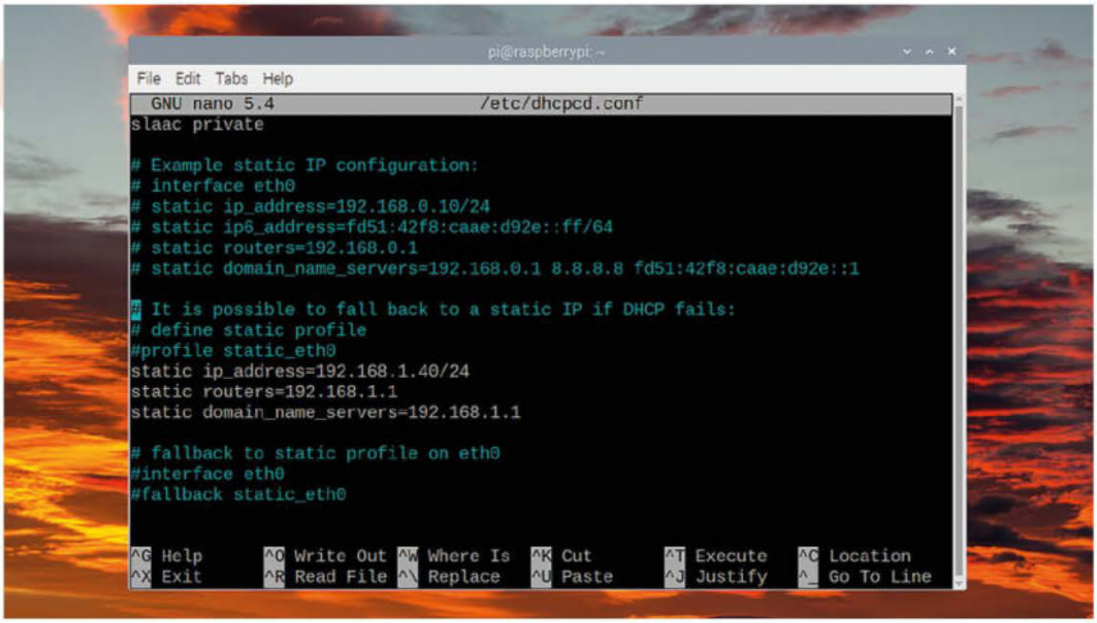
Piping and redirection are techniques used to control the flow of data between commands and files. Some common operators include:

- `|` **(pipe):** This operator sends the output of one command as input to another command and is used to chain commands together. In the home directory, enter `ls | grep D`. This lists list all files and directories starting with 'D': **Desktop**, **Documents**, **Downloads**.
- `>` **(redirect output):** This operator sends the output of a command to a file, overwriting its

Top Tip!

Hey, man!

Use `man` (manual) to find out more about any commands. To learn more about `rm` and its options, use `man rm`. You can also use `man man` to learn about the manual.



```
  GNU nano 5.4                    /etc/dhcpcd.conf
slaac private

# Example static IP configuration:
# interface eth0
# static ip_address=192.168.0.10/24
# static ip6_address=fd51:42f8:caae:d92e::ff/64
# static routers=192.168.0.1
# static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
static ip_address=192.168.1.40/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0



^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Editing the **dhcpcd.conf** file to set a static IP address

contents. For example, enter `ls > filelist.txt`. This saves the list of files outputted by `ls` to the file **filelist.txt**. You can use `cat filelist.txt` to view the contents of the text file.

- `>>` **(append output):** This operator appends the output of a command to a file. For example, `ls >> filelist.txt`. Adds the list of files to the existing **filelist.txt** without overwriting its contents.
- `<` **(redirect input):** This operator sends the contents of a file as input to a command. For example, `sort < filelist.txt` would sort the contents of **filelist.txt**.

Command-line text editors

Editing files directly on the command line can be efficient with the right tools. Two popular text editors are:

- **nano:** A beginner-friendly, easy-to-use text editor. Open a file with `nano filename.txt`, edit the contents, and save with **CTRL+O**, followed by **CTRL+X** to exit.
- **vi:** An advanced text editor with powerful features and an extremely strange interface. Open a file with `vi filename.txt`, switch to insert mode by pressing **I**, edit the contents, and save by pressing **ESC** followed by ':wq' and **ENTER**.

> "Understanding networking and remote access is essential when working with Linux systems"

Networking and remote access

Understanding networking and remote access is essential when working with Linux systems, especially when using devices like the Raspberry Pi which are often deployed around the network and accessed remotely.

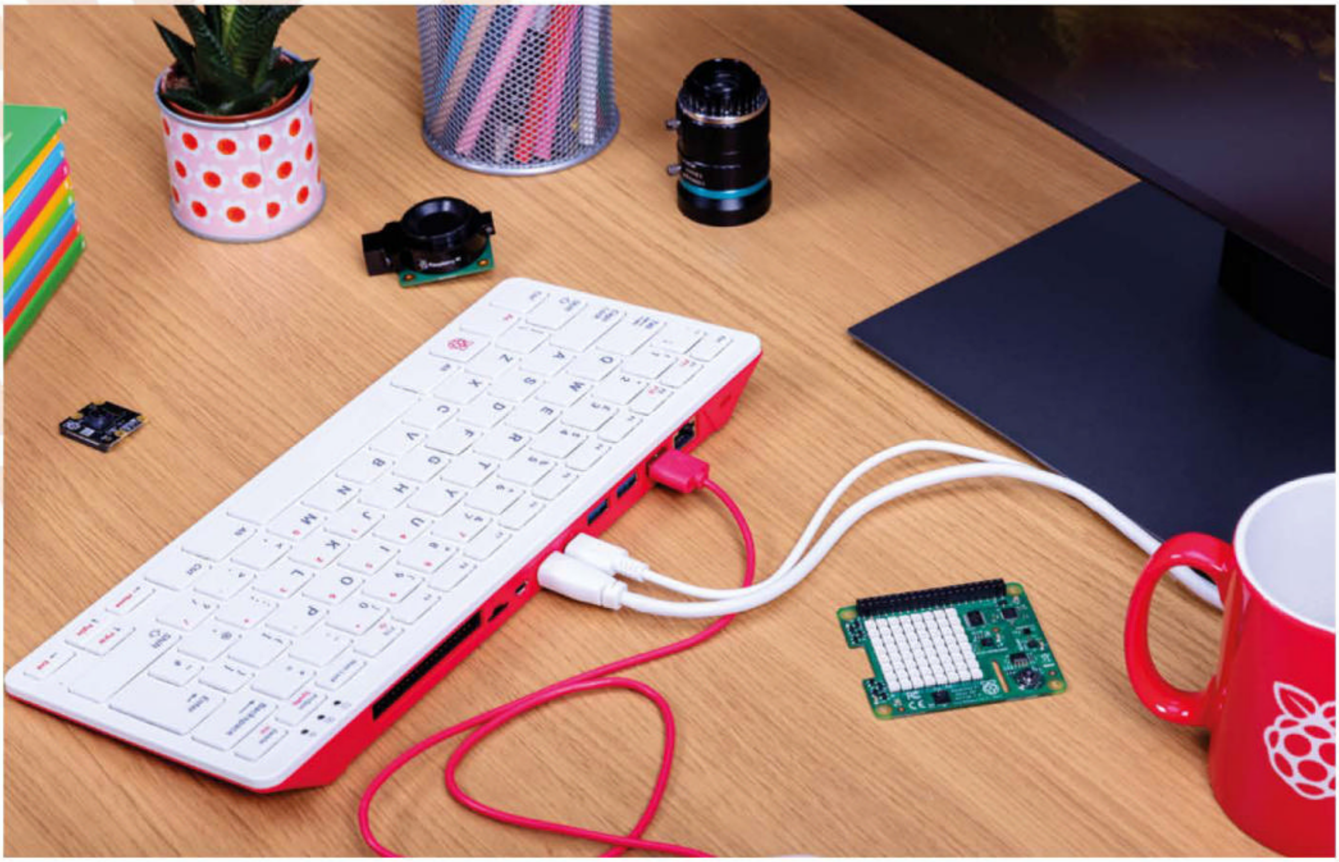
Raspberry Pi OS uses the 'dhcpcd' (Dynamic Host Configuration Protocol Client Daemon) to manage network configurations. Find out your Raspberry Pi's current IP (internet protocol) address using either the hostname or your IP address.

```
hostname -I
```

It should return two addresses: a short IPv4 containing four digits separated by dots, and a much longer IPv6. Like this:

192.168.1.63 2a02:6b61:9897:0:6fb:19d3:1ee3:1281

Normally the shorter IPv4 (in this case, 192.168.1.63) is the one we will use. Your



Raspberry Pi is an ideal environment for learning Linux and the command-line environment

IP address will be assigned by your router and different.

To access your Raspberry Pi remotely, choose Preferences > Raspberry Pi Configuration from the menu and click on Interface. Switch on SSH and click OK.

You can now connect to your Raspberry Pi from another computer using the `ssh` command on another Linux or macOS machine, or PuTTY on Windows (**putty.org**). Use the following command, replacing 'your_username' with your Raspberry Pi username and 'your_ip_address' with your Raspberry Pi's IP address:

```
ssh your_username@your_ip_address
```

You can also use the hostname and '.local' to access the Raspberry Pi, which is useful if the IP address has dynamically changed

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

The first time you log in via ssh will produce a warning, enter 'yes' to connect. You can now enter CLI commands from your remote computer. To return to the CLI on your machine use `exit`.

Set a static IP

You can set a static IP address for your Raspberry Pi by editing the **dhcpcd.conf** file located in the **/etc** directory. This ensures that your router does not dynamically change the IP address, causing you to lose access to your Raspberry Pi. Enter:

```
nano /etc/dhcpcd.conf
```

Edit the following lines to the file, removing the '#' at the start and replacing 'X' and 'Y' with your desired values:

```
static ip_address=192.168.X.Y/24
static routers=192.168.X.1
static domain_name_servers=192.168.X.1
```

Press **CTRL+O** to save the file and **CTRL+X** to exit nano. Now restart the dhcpcd service:

```
sudo systemctl restart dhcpcd
```

Top Tip!

File transfer

Secure File Transfer Protocol (SFTP) enables you to transfer files securely between your computer and Raspberry Pi. Many FTP clients, like FileZilla (**filezilla-project.org**), support SFTP. To connect, use the same credentials as your SSH connection.

```
pi@raspberrypi:~ $ sudo apt update
Hit:1 http://security.debian.org/debian-security bullseye-security InRelease
Hit:2 http://deb.debian.org/debian bullseye InRelease
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Hit:4 http://archive.raspberrypi.org/debian bullseye InRelease
Fetched 44.1 kB in 1s (37.9 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
pi@raspberrypi:~ $
```

> Discover the limitless potential of Raspberry Pi and the Linux environment

Package management and software installation

Package management involves the process of installing, updating, and removing software packages on a Linux system.

Packages are pre-compiled software applications, along with their dependencies, configuration files, and documentation. Use the Advanced Package Tool (APT) system for package management. Some common apt commands include:

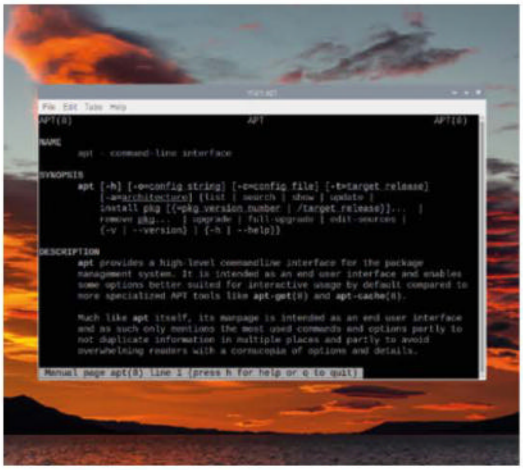
- `sudo apt update`: Updates the package list from repositories
- `sudo apt upgrade`: Upgrades installed packages to their latest versions
- `sudo apt install package_name`: Installs a package by its name
- `sudo apt remove package_name`: Removes an installed package

To ensure your system and installed software are up-to-date and secure, regularly run the following commands:

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

These commands update the package list and upgrade all installed packages to their latest versions.

There is more, much more, to the CLI than this and as you venture into new projects and expand your skills, you'll discover the limitless potential of Raspberry Pi and the Linux environment. From scripting, to automating tasks and troubleshooting projects. Keep exploring, learning, and creating!



```
APT(8)                              APT                              APT(8)

NAME
       apt - command-line interface

SYNOPSIS
       apt [-h] [-o=config_string] [-c=config_file] [-t=target_release]
           [-a=architecture] {list | search | show | update |
           install pkg [{=pkg_version_number | /target_release}]...   |
           remove pkg...   | upgrade | full-upgrade | edit-sources |
           {-v | --version} | {-h | --help}}

DESCRIPTION
       apt provides a high-level commandline interface for the package
       management system. It is intended as an end user interface and enables
       some options better suited for interactive usage by default compared to
       more specialized APT tools like apt-get(8) and apt-cache(8).

       Much like apt itself, its manpage is intended as an end user interface
       and as such only mentions the most used commands and options partly to
       not duplicate information in multiple places and partly to avoid
       overwhelming readers with a cornucopia of options and details.

 Manual page apt(8) line 1 (press h for help or q to quit)
```

`man` is used to access the manual pages. Here we can see the listing for `man apt`

Argon POD System for Raspberry Pi Zero 2 W

▶ Argon 40 ▶ **argon40.com** ▶ From £10 / $10

Turn your Raspberry Pi Zero 2 W into a custom hardware powerhouse with this modular system. **PJ Evans** plugs in

Over the past few years, Argon 40's range of Raspberry Pi cases has raised the game but, until now, the humble yet powerful Zero 2 W has been left out in the cold. No more! Not only has Argon released an aluminium case for the Zero, but also a range of modular add-ons that can be used to create your ideal solution. So, are they worth the investment?

Firstly, there is the case itself (£10/$10). Those familiar with the Flirc case will notice more than a passing resemblance. A solid metal case with plastic covers on the top and bottom provides an industrial-capable enclosure on its own. Thoughtfully, Argon provides a colour-coded, 40-pin header that you can solder on to your Zero. Although it is compatible with all Zero models, for the Zero 2 W Argon includes a thermal paste pad that provides heat dissipation between the large SoC and the case. Four screws ensure a tight connection between the Zero body and the case; everything else 'snaps' together. Clear instructions are provided.

The first add-on module, USB/HDMI (£12/$12), connects directly to mini-HDMI and micro-USB ports, breaking them out into full-sized equivalents and separating the power, freeing up another USB port. Installation is as simple as lining up the connectors and pushing together, with no drivers required. Underneath, a header and power connector are exposed that are used to connect other modules, so you may require this module for compatibility.

But wait, there's more!

Need some more USB ports? How about Ethernet? The USB/LAN module (£20/$20) does both. A 'full-size' module, it sits below our current setup, connecting via an exposed header on the USB/HDMI module. Easily snapping together, you now get an additional four USB connectors (two on the front, two on the back) and an Ethernet connector for hard-wired networking. Again, a one-minute install and Raspberry Pi OS recognises all peripherals without any additional software.

SPECS

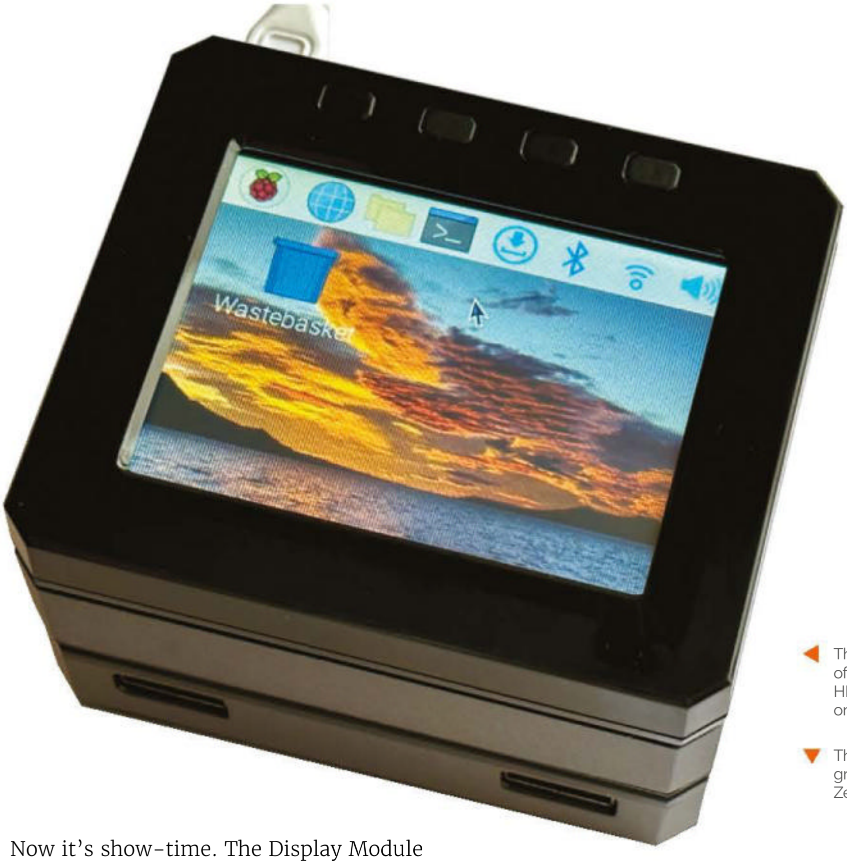
CASE:
Dimensions: 35 × 80 × 20 mm; SD card access; Removable cover

USB/HDMI MODULE:
Dimensions: 35 × 80 × 20 mm; 2 × USB; 1 × full-size HDMI; 1 x micro-USB power

LAN MODULE:
Dimensions: 70 × 80 × 20 mm; 4 × USB; 1 × Ethernet

LAN MODULE:
Dimensions: 70 × 80 × 16 mm; 2.8˝ capacitive touchscreen; 4 × buttons; IR sensor

▶ The module system can have many combinations, such as this headless Ethernet setup



The complete tower of Zero. Ethernet, HDMI, and display in one neat package

The case alone is a great choice for your Zero projects

Now it's show-time. The Display Module (£30/$30) is a 2.8-inch capacitive touchscreen that plugs into Zero's GPIO. Framed in an attractive black bezel, with four programmable buttons, it transforms Zero into a stand-alone unit opening up the potential for makes. Drivers are required to get the screen running, which we found easy to install. A simple configuration system allows the buttons to be preprogrammed with common functions such as power off or reboot. Provided documentation explains how to take control of them yourself using GPIO. The screen is perfectly adequate, although you'll need a stylus for meaningful use of the touch capabilities.

> "A great collection of add-on gadgets for your Zero"

All in all, this is a great collection of add-on gadgets for your Zero. The modular nature means you can keep costs (and size) down. Argon has also teased future modules for further expansion. One slight niggle was a lack of access to GPIO and camera connectors, although this can be overcome with USB devices. It would also be interesting if documentation was provided on how you could make your own modules using the through-module header.

Verdict

Easier GPIO and camera access would be nice, but the exquisite design, tough materials, and sensible pricing more than make up for that. The Argon POD has seriously impressed.

9/10

EXT3-1 Kit and 3.70˝ E-ink display

▶ Pervasive Displays ▶ **pervasivedisplays.com** ▶ £29 / $37

An updated e-ink display and driver board for Raspberry Pi Pico that saves on space. **Rob Zwetsloot** takes a look

SPECS

DISPLAY: 3.7˝, 240×416 resolution, 130 dpi

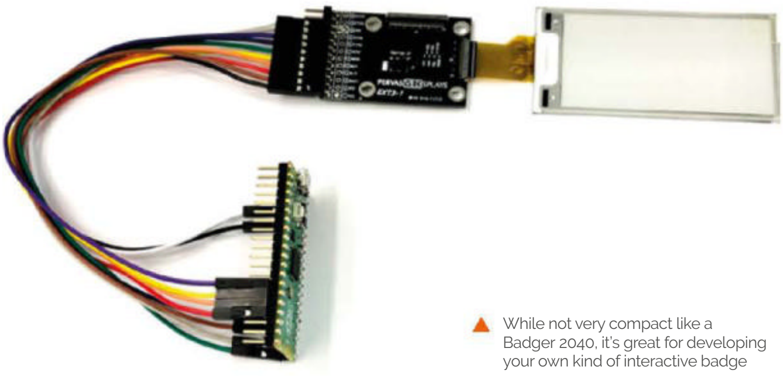
DRIVER: Drive displays between 1.54˝ and 12˝, 8MB flash memory, open-source

INTERFACE: Capacitive touch with customisable UI

When we last looked at the display and extension board that makes up Pervasive Displays' line of Pico development kits, we quite liked the customisability of the display and its code (magpi.cc/121). However, we were concerned with the amount of space it took up. As a development kit for making product with a Pico, it's not too much of an issue as you're likely going to have everything be a little more compact in a final product. Hobbyists not as much, though.

The newer EXT3-1 expansion board, along with the 3.70˝ touchscreen display, have shrunk down in the right areas, making them far more appealing to hobbyists. The display itself is very cool and almost wafer-thin, while still having great contrast, and even capacitive touch capabilities. It's limited to black and white but that's not too unusual still for e-ink displays.

Hooking it all up is quite simple – if you've ever added a Raspberry Pi Camera Module to a Raspberry Pi, the ribbon and connector will be familiar to you. These cannot be directly plugged into Pico, so once you've installed the screen to the extension board, you'll need to connect to Pico using GPIO pins and wires. You can prototype easily on a breadboard, but eventually you'll need to break out the soldering iron.

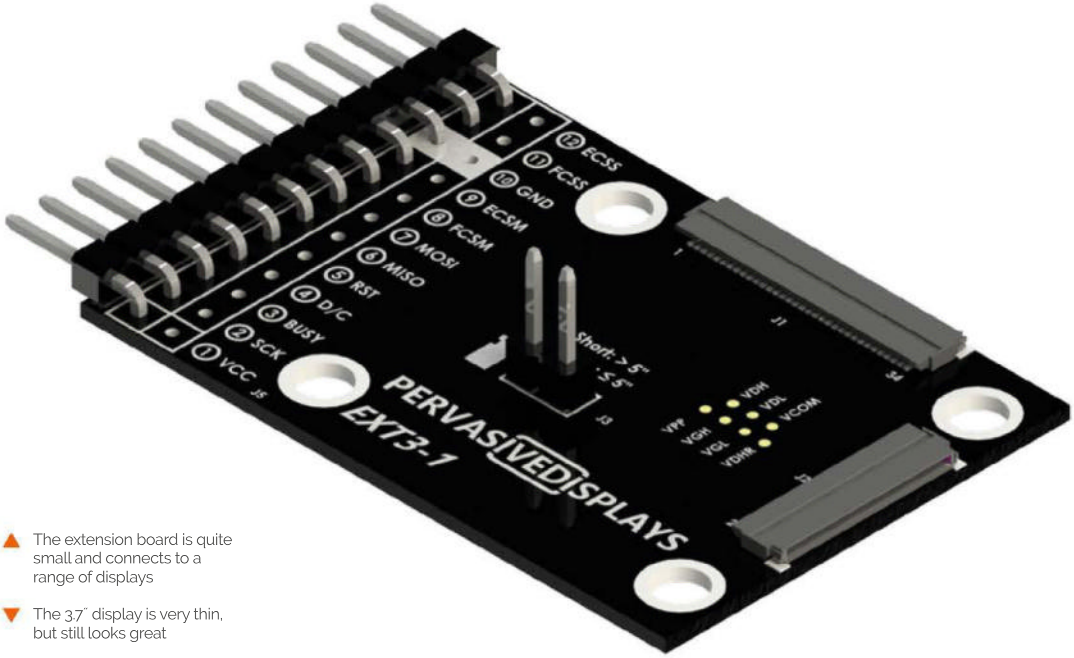


▲ While not very compact like a Badger 2040, it's great for developing your own kind of interactive badge

Code differently

While many Pico kits and products use MicroPython or variations on it like CircuitPython, here Pico uses Arduino code, which is much closer to C++. While this may be different to some, the way the display is programmed is easy to understand thanks to the example code provided – and the very readable way most of the PDLS library is set out. The documentation is pretty good too, especially if you plan to use the whole system for development. Wireless capabilities are not currently included in the Pico build, though, so you'll need it to interact with something else if you want to do something networked.

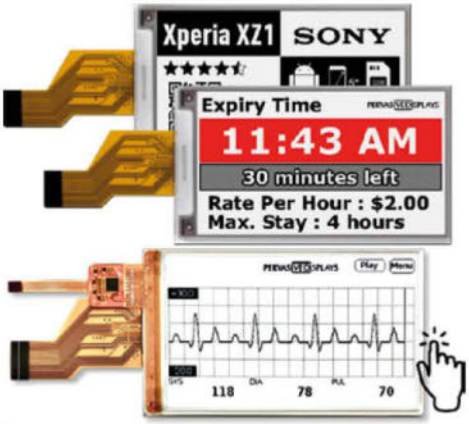
The whole thing runs very smoothly once all set up and coded, with the sort of response time



▲ The extension board is quite small and connects to a range of displays

▼ The 3.7˝ display is very thin, but still looks great

> The whole thing runs very smoothly once all set up and coded, with the sort of response time you might expect from an e-ink display



you might expect from an e-ink display – so not as quick to refresh as an LCD or LED display, but good enough for the kind of functions you'd want with e-ink. One thing we will add: we mentioned it's quite a thin display and, while the contrast is good, putting it right in front of a strong light source may see some of that light come through. In such use cases, you can always reverse the black and white pixels on the display. However, that still isn't a perfect solution.

We do like the Pervasive Displays kits, and the expanding range of display sizes is making them more and more attractive for both commercial and hobby projects. We need to find a way to permanently use this screen now.

Verdict

A great display that has some quite powerful code backing it, even if it is more unique code than other Pico products.

8/10

Iono RP / RP D16

▶ Sfera Labs ▶ **magpi.cc/iono** ▶ From €198 / €276 (+ tax)

Control industrial equipment with the help of RP2040. By **Phil King**

Warning!
Electrical Safety

Please be careful when working with electrical projects around the home. Especially if they involve mains electricity.

magpi.cc/electricalsafety

The Iono RP and RP D16 are industrial-grade programmable logic controllers (PLCs) based around the RP2040 SoC, as used in Raspberry Pi Pico. A key advantage of this is the ability to program them with familiar languages such as MicroPython and C++, rather than the esoteric ones – such as Ladder Logic – typically used for PLCs.

A programmable logic controller is an I/O module used to operate industrial equipment such as motors, pumps, lights, and other machinery. As such, it needs to be rugged enough to deal with an industrial setting. The Iono RP PLC models come in a tough plastic, DIN rail-mountable case. They can also handle operating temperatures between -20 and +60°C.

Inputs and outputs

Along the top and bottom of each Iono unit are rows of screw terminal block connectors for inputs/outputs, along with power, ground, and an RS-485 interface.

The standard Iono RP model features four digital relay outputs, set to normally open by default. While we've covered relay boards for Pico and Raspberry Pi before, most have a fairly limited maximum voltage and current. The Iono RP's relays can handle loads of up to 6 A at 250 V AC, or 3 A at 30 V DC. So it should be able to drive a wide range of machinery.

In addition, the Iono RP has four multi-mode inputs for analogue (voltage or current) or digital signals – higher levels are converted down to safe

Remove the front panel of the Iono RP to access the micro-USB port and a tiny BOOTSEL button

0–3.3V for the RP2040 – along with two dedicated digital inputs which can also be used as general-purpose TTL-level (0–5V) inputs/outputs. There's also a single 0–10V analogue output.

The Iono RP D16 model lacks relays and instead offers a larger array of 16 digital inputs/outputs. The inputs are current-sinking 24V, while the 640mA 24V outputs can be configured as high-side switches or push-pull drivers for high-speed switching.

Familiar coding

Both models require a 12–24V DC power supply, although they can be programmed without it connected. As on Raspberry Pi Pico, coding can be done via a USB or SWD (Serial Wire Debug) connection to a computer.

On the Iono RP, removing the front panel gives easy access to the unpopulated SWD and micro-USB port. On the RP D16, however, you need to dismantle the case to access them, which is simple enough but a little annoying.

To start programming in MicroPython, you'll need to install the Pico UF2 firmware. This entails holding down the BOOTSEL button. You can then start programming in Thonny. For this you'll also need to download the Iono RP GitHub repo (**magpi.cc/ionorpgh**) containing libraries and a couple of code examples. So it's fairly easy to get started for anyone who has already done any Pico programming.

This familiarity should prove a key benefit for some industrial customers, along with makers looking to control higher-voltage/current equipment from a more robust, Pico-style microcontroller. While the Iono RP and D16 lack the immediate control of a button or touchscreen interface found in some PLCs, and there's no wireless connectivity, they're reasonably priced for their level of functionality.

> The Iono RP PLC models come in a tough plastic, DIN rail-mountable case

The Iono RP D16 model has 16 digital inputs/outputs connected via screw terminals

SPECS

FEATURES: Raspberry Pi RP2040 SoC with dual ARM Cortex-M0+ @ 133MHz, 264kB on-chip SRAM, 2MB flash storage; 3-module DIN rail case

INPUTS/OUTPUTS (IONO RP): 4 × digital relay outputs (up to 6A 250V AC / 3A 30V DC), 4 × multi-mode (analogue/digital) inputs, 2 × digital inputs/outputs, 1 × analogue output, RS-485 serial port

INPUTS/OUTPUTS (IONO RP D16): 16 × digital inputs/outputs, RS-485 serial port

OPTIONAL MODULES: RTC, Secure Element

DIMENSIONS: 53.5 × 90.5 × 62.0mm

Verdict

They may seem expensive to hobbyists, but these are industrial-grade PLCs with higher-capacity I/Os while offering a familiar coding environment.

8/10

ONLY THE BEST

Yes, you can buy these products - they are in stock…for now!

Products that, at the time of writing, were in stock

By Marc de Vinck @devinck

Stock levels. It's been an issue for all of us for a while now. Cars, appliances, and anything that has some kind of electronics, especially microcontrollers, seem to have experienced a supply chain issue at some point over the last few years. But, then again, even paper towels or general commodities have been hard to come by at times. It's been a rough few years trying to acquire parts for your projects, but that seems to be getting a bit better.

All is not lost, especially if you are looking for microcontrollers, rather than a single-board computer. Raspberry Pi started selling the RP2040 integrated circuits to manufacturers so they can continue to support, develop, and innovate in that ecosystem, and it's been great! And the most important spec: they are in stock!

In this roundup, we'll be looking at a few new boards and products, but also featuring a few that we have seen before. The key theme being that, at least at the time of these reviews being written, they are all in stock, and they all look like they should continue to be in stock. So, if you have plans for making a cool new IoT device, robot, or so many other DIY electronic projects, don't worry about stock levels: there are some options out there.

And yes, you may not be able to do all the things you imagined, but you can still do a lot, and you can still have fun!

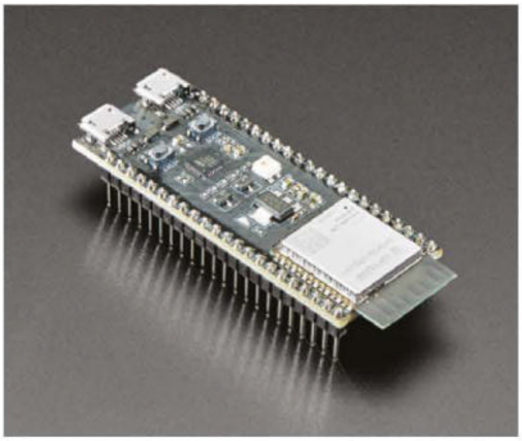
ESP32-S3-DevKitC-1-N8R8 vs Raspberry Pi Pico W

ADAFRUIT $19.95 | adafruit.com

PIMORONI $6.46 | pimoroni.com

The ESP32-S3-DevKitC-1, designed by Espressif Systems and available from Adafruit, is an entry-level development board for the ESP32-S3-WROOM-1. It features an ESP32-S3, an Xtensa 32-bit LX7 CPU that operates at up to 240MHz, Wi-Fi, and Bluetooth LE with an on-board PCB antenna, and 8MB flash memory and 8MB PSRAM.

That's a lot of power and memory! There are a few variations of the board that are available, so be sure to check out the link to the product page. Just keep in mind, programming these types of integrated circuits can be more difficult than something like an Arduino, but you also typically get a lot more features.



How can you go wrong with the Raspberry Pi Pico? It's the original microcontroller based on the RP2040 and made by Raspberry Pi Ltd itself. Not only is it very affordable, but it also features a dual-core Arm Cortex-M0+ processor with 264kB internal RAM. You can do a lot with that combo! Since its launch, we have seen a lot of boards featuring the RP2040, but the classic Pico W with built-in Wi-Fi is still an excellent choice, especially if you are just getting started. And if you don't want to solder anything, be sure to check out the Pico WH variation with its header pins already soldered on, for just about a dollar more. It's great value! →

VERDICT

ESP32-S3-DevKitC-1-N8R8

Powerful and in stock!

8/10

Raspberry Pi Pico W

The OG RP2040.

10/10

Badger 2040 W

PIMORONI $27.68 | pimoroni.com

A programmable badge with fast updating e-ink display! What's not to love? The Badger 2040 W features a Raspberry Pi Pico W and couples it with a beautiful and simple e-ink display. Don't think this is just a conference badge: it's an RP2040 with e-ink. You can use it in so many projects. Ignore the optional lanyard! But sure, it makes for a great badge too!

On the front of the Badger 2040 is a black and white 2.9˝ e-paper screen, five buttons, and a slot to clip it onto a lanyard. On the back of the board there is a battery breakout connector, a reset button, and a convenient Qwiic/STEMMA QT connector, making the board very versatile for other projects. Conference or not, this is a great Pico W-powered board.

VERDICT

Badger 2040 W

Another fun 2040 board.

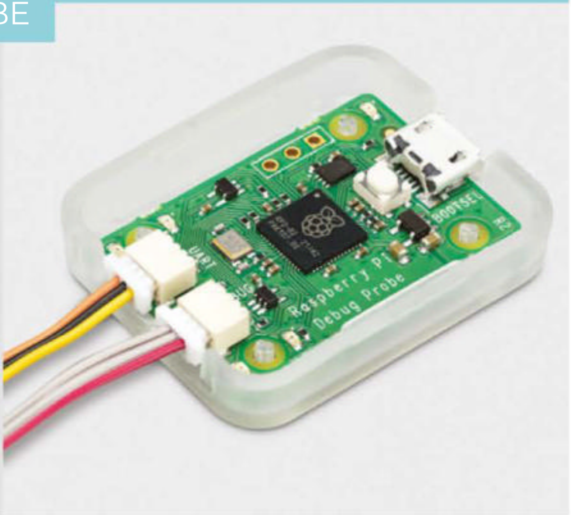
9/10

RASPBERRY PI DEBUG PROBE

PIMORONI $12.61

pimoroni.com

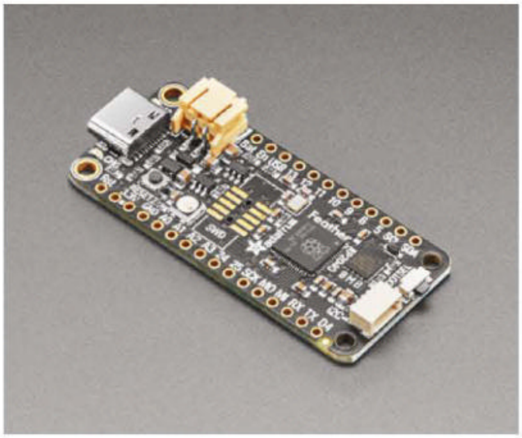
Got a Raspberry Pi Pico? Check out the Raspberry Pi Debug Probe by the creators at Raspberry Pi, and available at Pimoroni. The Debug Probe allows you to easily hook Pico's debug pins up to any computer via USB for debugging. It's also very handy to have for troubleshooting other microcontrollers and microprocessors. You can use it with either the processor serial debug interface or industry-standard UART. Both interfaces use a 3-pin debug connector and make debugging affordable and accessible. Check out the website for more information on this powerful little device.



Adafruit Feather RP2040

ADAFRUIT $11.95 | adafruit.com

The Feather ecosystem continues to grow, and the Adafruit Feather RP2040 still stands strong amongst all the variations of Feather boards. The board features an RP2040 32-bit Cortex-M0+ dual core running at ~125MHz, perfectly packaged into the Feather form factor. This allows you to add dozens of different shields and accessories with ease. Or, you can just design your own circuits and use up the 21 GPIO pins that feature four 12-bit ADCs, two I2C, two SPI, and two UART peripherals. It also has 16 PWM outputs that you can use for servos or LEDs. It's an impressive board, and a great choice for most projects.



VERDICT

Adafruit Feather RP2040

A great addition to the Feather ecosystem.

10/10

Raspberry Pi 3A+

RASPBERRY PI £28.50 | raspberrypi.com

So, how about any single-board computers that are in stock and ready to ship? Not all variants of Raspberry Pi are easy to get at the moment (though hopefully the situation will be improving by the time you read this), but the 3A+ is usually the most available. It's smaller than the model B, and it doesn't have the latest processor. However, for many embedded options, it's a great choice. The smaller size and lower power requirements mean it's more portable, and it's still got enough processing grunt for a huge range of tasks than microcontrollers struggle with. While it seems a bit unusual to recommend a board that's not the latest model, this really is a capable little computer. →

VERDICT

Raspberry Pi 3A+

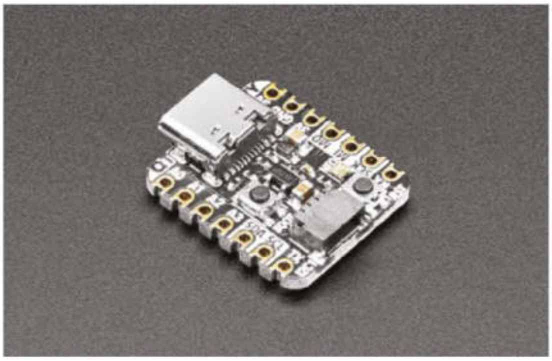
Powerful and readily available single-board computer.

10/10

Adafruit QT Py RP2040

ADAFRUIT ◆ $9.95 | adafruit.com

After something a little smaller? The Adafruit QT Py RP2040 is ridiculously small and amazingly powerful too! The board features the popular RP2040 microcontroller and couples it with the growing ecosystem of STEMMA QT-compatible accessories. You get a reset and bootloader button, an RGB NeoPixel LED, and 13 GPIO pins. You can do a lot with this board. If your project needs to be small, yet powerful, check out the Adafruit QT Py RP2040.



VERDICT

Adafruit QT Py RP2040

You want smol, we got smol.

9 /10

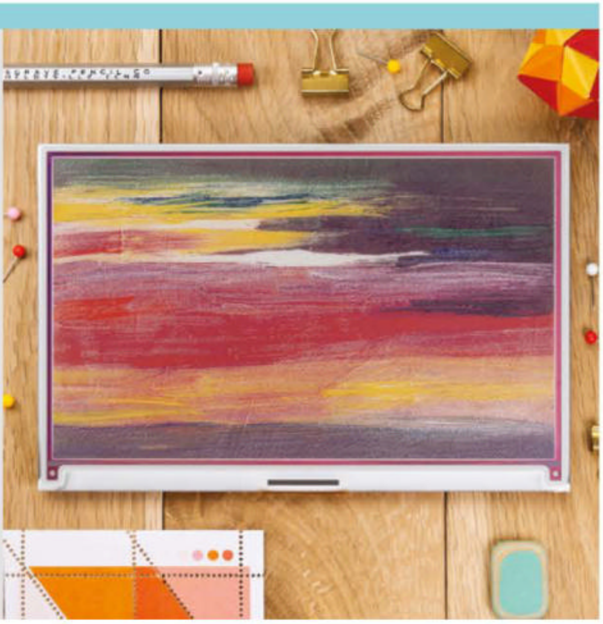
INKY IMPRESSION 7.3″

PIMORONI ◆ $71

pimoroni.com

Admittedly I took a little liberty when including this in the 'it's available', as the Inky Impression from Pimoroni is available for pre-order and not in stock just yet. Although that will change once this is published, and I hope it stays in stock. I had to include this new display because of just how amazing it looks. No, it's not super-high definition, but it does have high pixel density, especially for a low power e-ink display. It's also fairly large at 7.3″, which is great for e-ink. But the real reason I had to include it in the roundup is because it's also a seven-colour display. Most e-ink displays are two or three colours – this one really shines at seven.

If you have a Raspberry Pi, this should be on your list of accessories.



HackSpace

These reviews are from HackSpace magazine. Each issue includes a huge variety of maker projects inside and outside of the sphere of Raspberry Pi, and also has tutorials. Find out more at **hsmag.cc**.

10 Amazing: Lighting projects

Let there be light with these incredibly illuminating projects

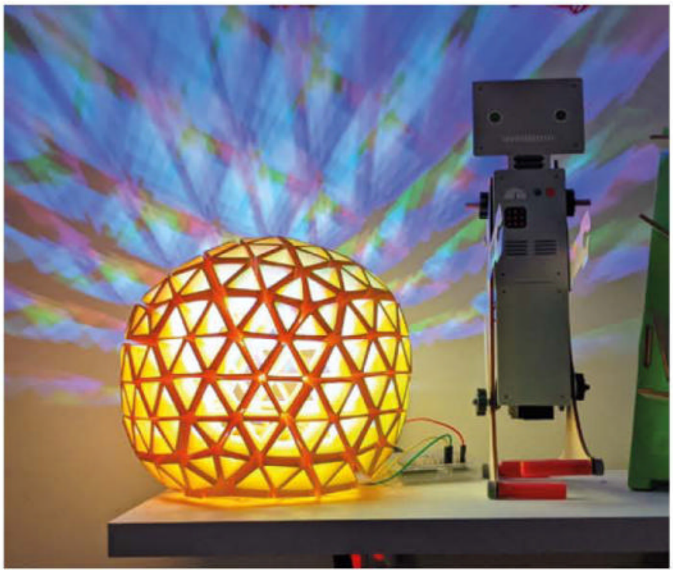
Lighting an LED may be a simple beginner project for Raspberry Pi and Raspberry Pi Pico coders, but that doesn't make it any less cool. There's so much you can do with an LED when it's connected to a computer. Here are just some of the ways a Raspberry Pi project focussed on lights has wowed us.

▲ PiHue

Automated lights

Control Phillips Hue-brand lights with this little Raspberry Pi hack, and take full control of the living room lighting.

magpi.cc/pihue



▲ Sunrise Lamp

SAD no more

Waking up to natural sunlight is apparently a good thing for the body and, towards the end of the year, these kind of sunlamps become very welcome.

magpi.cc/sunlamp

▲ Raspberry Pi Beret

Stand clear, vehicle reversing

This fancy-looking beret has some very clever code that lets creator Sean control it from his smartphone, so people know when he's turning left or right.

magpi.cc/picoberet

▼ Dinner klaxon

Teenage alert

Want to make sure the kids know to come down for dinner? This traffic light indicator and an alarm, controlled by a phone, will let them know.

magpi.cc/73

NeoPixel LED Mirror

You, simplified

Magic mirror? Nah. A mirror made up of giant pixels so that you get an abstract representation of yourself is where it's at.

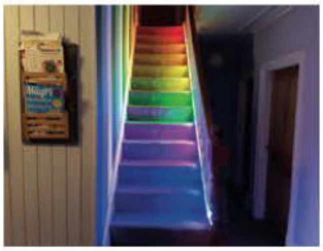
magpi.cc/ledmirror

Eye of Sauron

Spooky, scary

Keep an eye out for hobbits, elves, dwarves, and other fantasy creatures on Halloween with this roving ocular display.

magpi.cc/sauroneye



Temperature-controlled stairlights

Visual indicator

Lorraine Underwood's smart stairs let you know how hot or cold it is outside before you even get to the front door.

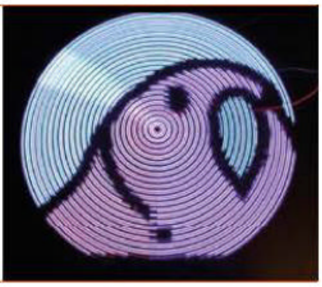
magpi.cc/tempstairs

POV Display

Not a trick

This illusion uses a very similar process to film, showing you images quickly enough that it fools your eyes into thinking there's movement. In this case, it's an LED stick spinning really fast.

magpi.cc/povdisplay



Odyssey Lights

Blackpool illuminated

Lorraine is back again, this time trading in the stairs for giant interactive displays at the famous Blackpool Illuminations.

magpi.cc/odyssey

Cosplay vacuum

Ghostbusting with Luigi

Using a 3D-printed vacuum and some tactically inserted NeoPixels, you too can be Luigi this Halloween (or Comic Con).

magpi.cc/poltertutorial

Learn MicroPython

It may be straightforward by programming standards, but it's mighty powerful, says **Rosie Hattersley**

Getting Started With MicroPython on Raspberry Pi Pico

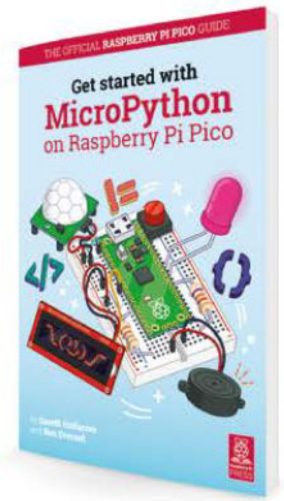
AUTHOR

Raspberry Pi Press

Price:
£10

magpi.cc/picobook

MicroPython is a version of the Python computer language optimised to run on microcontrollers such as Raspberry Pi Pico and other devices that use the RP2040. As the title implies, Raspberry Pi's very own guide to exploring MicroPython provides an ideal introduction to the programming language you are likely to use if you have Raspberry Pi Pico, Pico W, or any sort of RP2040 microcontroller. Rather than using a GUI, you program the microcontroller directly from another computer, like Raspberry Pi. So MicroPython is a case of typing in the code and letting it run. Specify input/output pins and MicroPython will control lights, send messages to an LCD matrix, produce on-screen images, or generate sounds. This detailed guide walks you through the essentials of coding with MicroPython for Pico, along with some interactive projects to try.



Videos

MicroPython and Pico in action

ELECTRONIC HOURGLASS

A Pico, MicroPython code, and the rotary encoder from Elecrow's Raspberry Pi Pico Starter Kit (**magpi.cc/elecrowpicokit**) are used to replicate a sand and glass hourglass. This satisfying project involves setting the time using the encoder and then measuring subsequent intervals. Results are displayed on a TM1637 4-digit LED.

▶ **magpi.cc/picohourglass**

TRUNDLEBOT EDUCATIONAL ROBOT

Deservedly winning Electromaker's PCB project design contest, this super robot provides an excellent introduction to programming with Pico and MicroPython and has both parts lists for DIY makers and alternatives if you want to focus on the code. Step-by-step guidance and ready-to-use code means a really rewarding build becomes achievable, even if you are new to coding.

▶ **magpi.cc/trundlebot**

DRAWING FILLED CIRCLES AND TRIANGLES WITH MICROPYTHON

Add a Waveshare screen to Raspberry Pi Pico, enter some colour and location variables in MicroPython, and you can get some wonderful visual effects, as this remarkable tutorial demonstrates. Once you've got the hang of things, you can even make merry with photos you import.

▶ **magpi.cc/drawingpico**

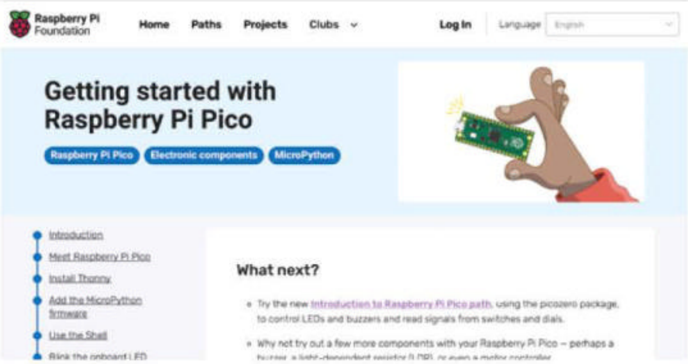
Getting started with Pico

AUTHOR

Raspberry Pi Foundation

Price:
Free

magpi.cc/startedwithpico



Raspberry Pi Foundation | Home | Paths | Projects | Clubs | Log In | Language English

Getting started with Raspberry Pi Pico

Raspberry Pi Pico | Electronic components | MicroPython

- Introduction
- Meet Raspberry Pi Pico
- Install Thonny
- Add the MicroPython firmware
- Use the Shell
- Blink the onboard LED

What next?

- Try the new Introduction to Raspberry Pi Pico path, using the picozero package, to control LEDs and buzzers and read signals from switches and dials.
- Why not try out a few more components with your Raspberry Pi Pico – perhaps a buzzer, a light-dependent resistor (LDR), or even a motor controller

We are focusing on MicroPython here, and RP2040 on Pico is the microcontroller you'll almost certainly learn to program. As this tutorial explains, there's some MicroPython firmware you may need to install the first time you use your Pico to code with it. Once set up, you'll learn how to make LEDs blink and change their brightness levels, plus how to make use of Pico's GPIO input/output pins to add sensors and other components. The companion 'Introduction to Raspberry Pi Pico' guide (**magpi.cc/picointro**) will then show you how to use potentiometers, switches, dials, buzzers, and more.

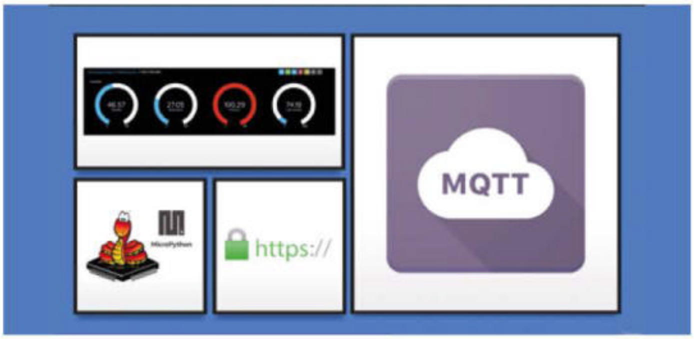
Learn IoT using MicroPython and Raspberry Pi Pico W

AUTHOR

Udemy

Price:
£20/$20

magpi.cc/udemyiotpico



Udemy does some great online courses, and this Internet of Things-focused series, featuring eight hours of video instructions, is ideal for getting to grips with both your Pico board and coding to add smarts to your home and life.

This course covers soldering headers to Pico, connecting sensors, and interfacing with LDR, LM35, DHT22, and DS18B20 sensors. Then you move on to cloud services with ThingSpeak (**thingspeak.com**) to send and receive temperature and humidity data. Finally, it explains MQTT (**mqtt.org**), which is a standard messaging protocol for IoT devices. This course is a great way to move on from the basics.

Projects

Show off your MicroPython prowess

RASPBERRY PI PICO WITH MICROPYTHON

Kevin McAleer has put together 16 free YouTube lessons focused on how to get the most from MicroPython and use Pico's GPIO pin connections to work with servos, temperature sensors, lights, and I2C. Dive straight into MP projects using his easy-to-follow instructions, friendly delivery, and clear component lists.

▶ magpi.cc/gpiomastery

PICOINVADERS

James, from Print 'N Play, takes you through setting up your Pico with MicroPython to code a Space Invaders-type video game. Adding an SSD, battery pack and OLED screen, plus a potentiometer, even makes the whole thing portable.

▶ magpi.cc/picoinvaders

RASPBERRY PI PICO – GETTING STARTED WITH MICROPYTHON REPL (ON WINDOWS)

This Core Electronics video guide is admirably clear and shows you all you need to know to start using MicroPython on Pico, from initial setup and Hello World print statement, to learning how to code in MicroPython in a Windows environment.

▶ magpi.cc/picowindows

Brian Corteil

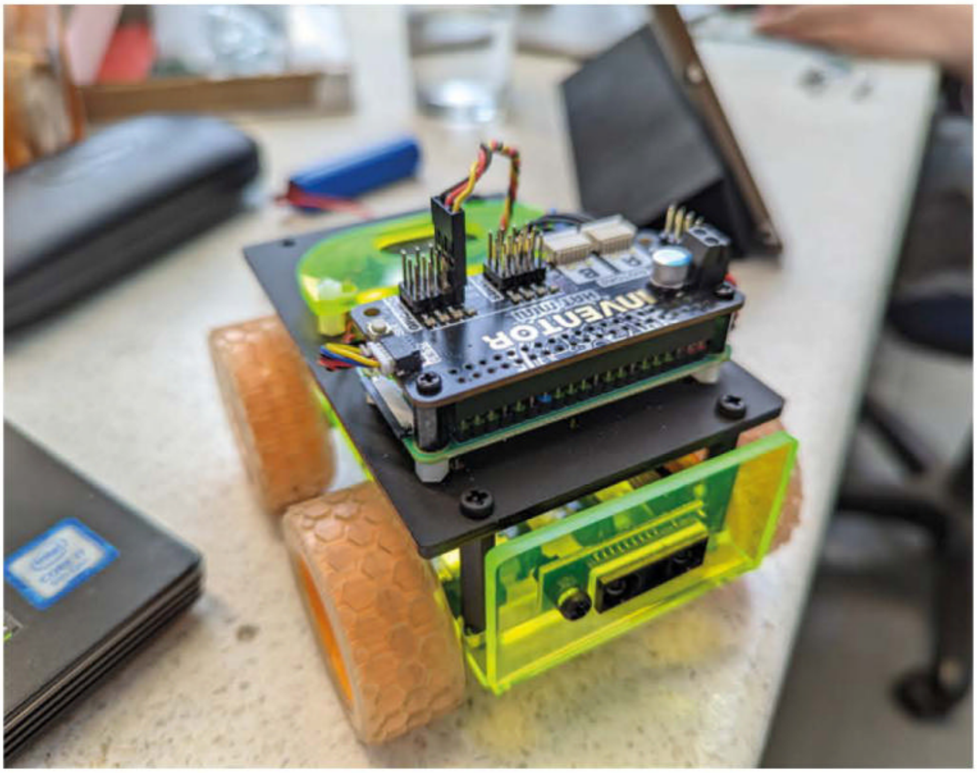
The new event runner for CamJam is an old face in the community

- Name **Brian Corteil** | Occupation **Electronic Security Systems Engineer**
- Community role **Event organiser** | URL **camjam.me**

Brian Corteil has been in *The MagPi* many times – over the years we've both shared his projects as showcases or screenshots in #MagPiMonday, and he's even written a few articles. He's a long-term member of the community who has taken on a new role recently.

"I am part of the team that has taking over organising CamJam from Mike Horne and Tim Richardson," Brian tells us. "[They] organised the last couple of CamJams before 2020. Early this year, I had realized that it had been over three years since the last CamJam and I decided to it was time for CamJam to return, and the only way it would happen is if I organised it. So, I pitched the idea to hold the jam at Cambridge's Makespace to both Mike, Tim, and Makespace. Luckily, everyone thought it was a great idea."

You can see some pictures from the success of the new CamJam on page 88.

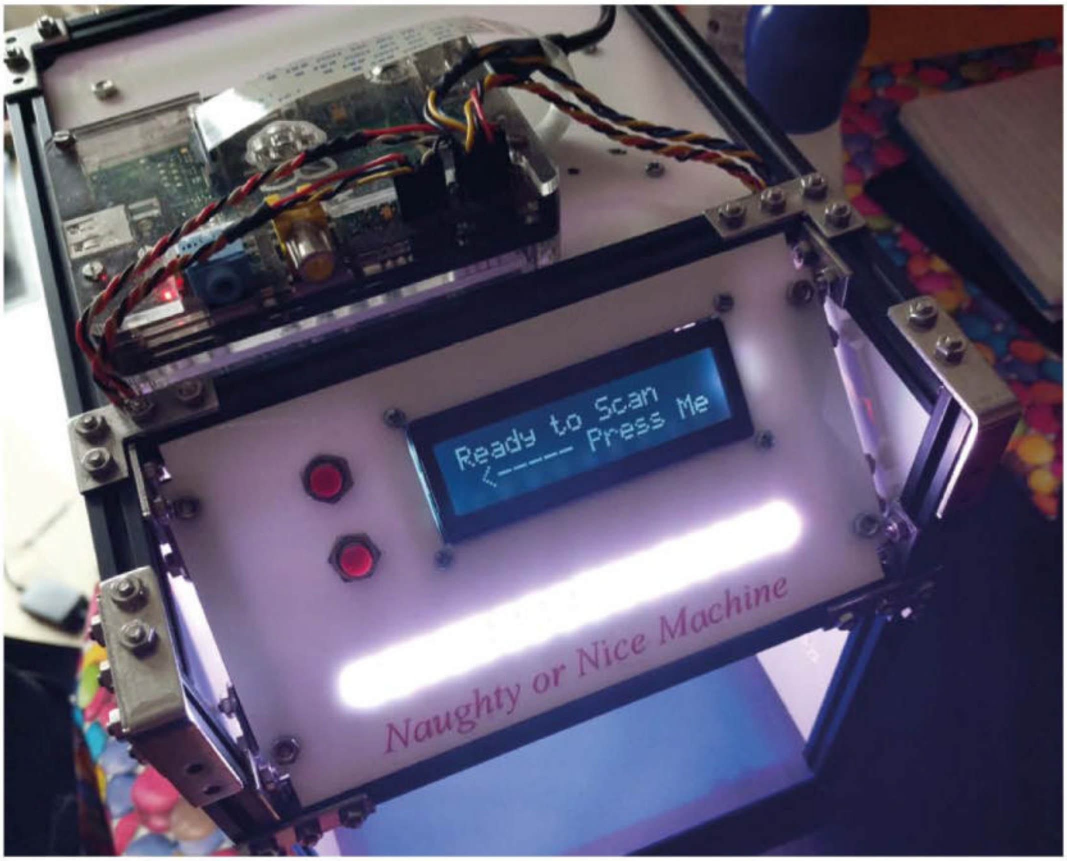


Originally designed and built for issue 51, Tiny 4WD has seen many upgrades over the years

What is your history with making?

I've really been making all my life; I have always enjoyed the journey of making something. I have been making since I was a child, starting off with wooden building blocks, then Meccano and LEGO. One of the earliest things I remember making was a zip wire for my Action Man, and I made a retrieval system from garden wire to rescue

One of the most visually compelling projects is this digital zoetrope which replicates very early animation from yester-century



This Naughty or Nice machine tends to say that adults are much naughtier

The Mood Giraffe was Brian's first Raspberry Pi project, and could be controlled over the internet

> "I have always enjoyed the journey of making something"

him. *Star Wars: A New Hope* came out when I was about 8-years-old. It jump-started a love of robots and computers. I started programming when my dad brought me a ZX80, then various other 1980s computers including an Acorn Electron. I returned to programming in the early 2000s when I started an OU Computer studies degree. The last 15 years, I have been making projects with Arduinos and Raspberry Pi. I got into digital making via the video console hacking scene where I first heard about Arduinos.

What was your first Raspberry Pi project?

The first project I can remember making was an internet-connected Giraffe Mood Lamp. It was built on top of a Raspberry Pi model A, way back in 2012. I removed the base of a LED night light, mounted it on top of an array of RGB LEDs, and it could be controlled over the internet.

What are some of your favourite projects you've made?

My 'Naughty or Nice' Machine – it scans your hand and gives you a naughtiness rating. It tips the scales so little kids are less naughty than their parents. Tiny 4WD robot, which I designed for a robotics feature I wrote for *The MagPi* (**magpi.cc/51**) and have taken to events like Micro Pi Noon (I nicked the idea for Micro Pi Noon from the Pi Wars Pi Noon challenge). Face-plant, my Pi Wars balance bot, and Zoey, a zoetrope made from 15 Pimoroni Badgers, two Raspberry Pi, six USB hubs, a flat bed scanner to scan in cell sheets, and a Pico.

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 have 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. We're concerned that this seems to be a robotic Dullahan

02. As great poet once said: magnets, how do they work?

03. The idea of Pico super-computers warms our papery hearts

04. A great and fun project making good use of online APIs

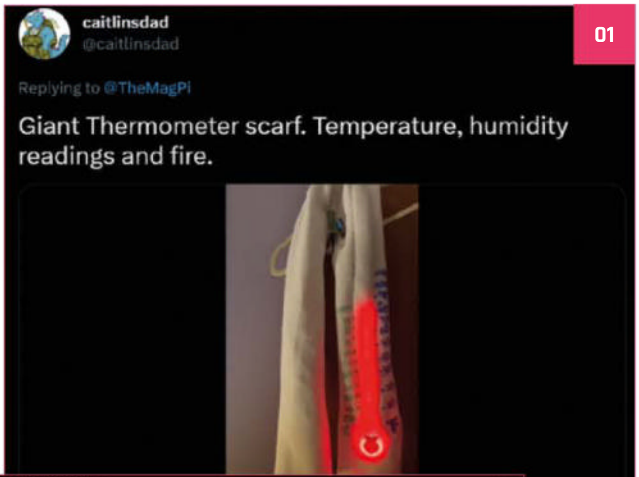
05. We do love a good, over-engineered birthday card

06. We really like this! There's plenty of game add-ons you could probably integrate like this

07. This could be a great little office robot, delivering your stuff

08. An excellent start on this big arcade cab

09. This watering system just gets better and better

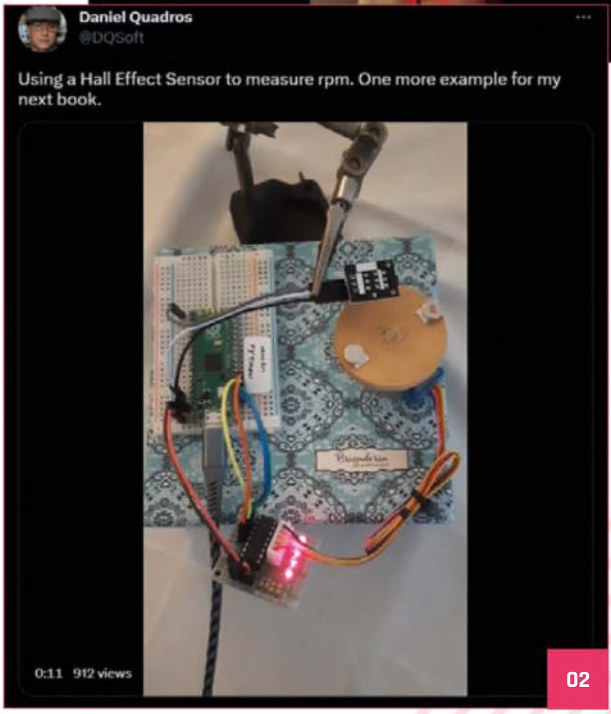


01

caitlinsdad
@caitlinsdad

Replying to @TheMagPi

Giant Thermometer scarf. Temperature, humidity readings and fire.

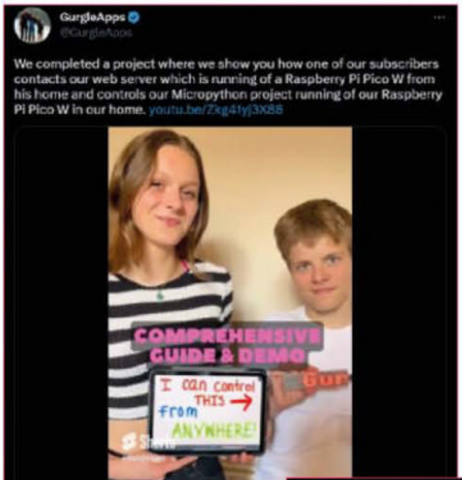


02

Daniel Quadros
@DQSoft

Using a Hall Effect Sensor to measure rpm. One more example for my next book.

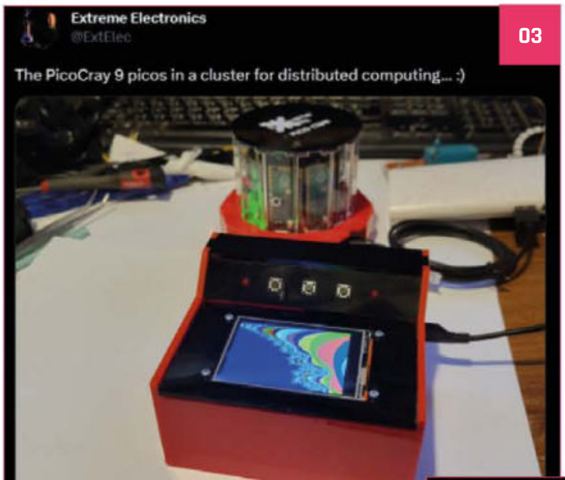
0:11 912 views



GurgleApps
@GurgleApps

We completed a project where we show you how one of our subscribers contacts our web server which is running of a Raspberry Pi Pico W from his home and controls our Micropython project running of our Raspberry Pi Pico W in our home. youtu.be/Zkg41yj3X88

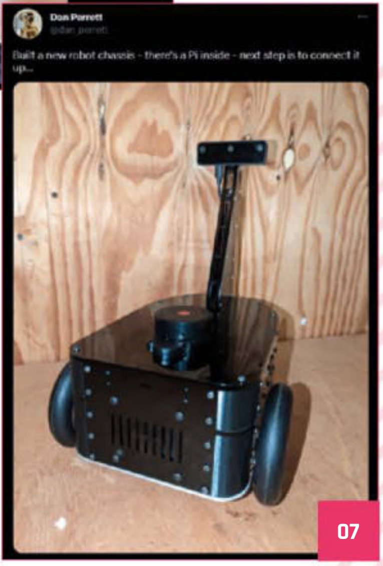
04



03

Extreme Electronics
@ExtElec

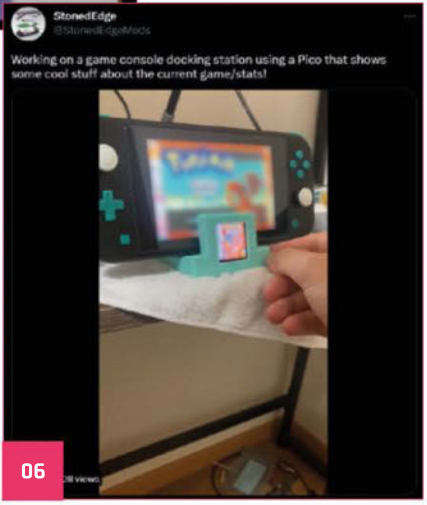
The PicoCray 9 picos in a cluster for distributed computing... :)



Dan Parrett
@dan_parrett

Built a new robot chassis - there's a Pi inside - next step is to connect it up...

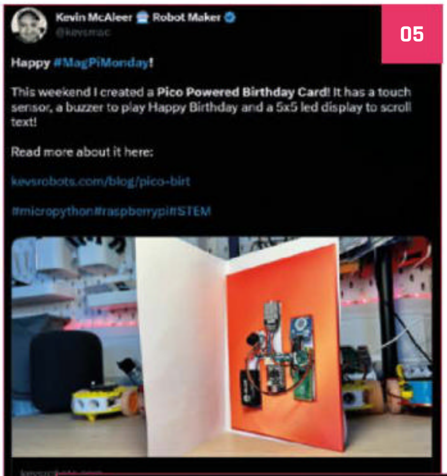
07



StonedEdge
@StonedEdgeMods

Working on a game console docking station using a Pico that shows some cool stuff about the current game/stats!

06



Kevin McAleer 🤖 Robot Maker
@kevsmac

05

Happy #MagPiMonday!

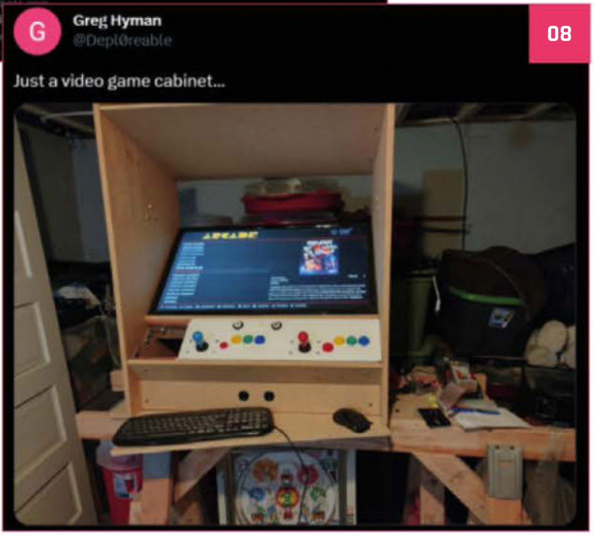
This weekend I created a **Pico Powered Birthday Card!** It has a touch sensor, a buzzer to play Happy Birthday and a 5x5 led display to scroll text!

Read more about it here:

kevsrobots.com/blog/pico-birt

#micropython#raspberrypi#STEM

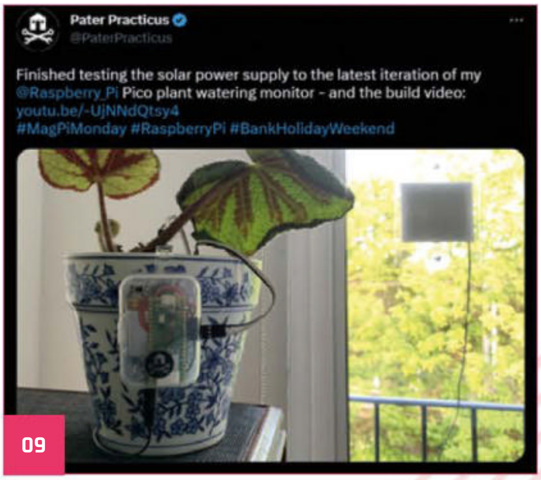
kevsrc
Pico Bi
Raspbe



Greg Hyman
@Depl0reable

08

Just a video game cabinet...



Pater Practicus
@PaterPracticus

Finished testing the solar power supply to the latest iteration of my @Raspberry_Pi Pico plant watering monitor - and the build video: youtu.be/-UjNNdQtsy4
#MagPiMonday #RaspberryPi #BankHolidayWeekend

09

Events in pictures: CamJam returns!

Community and official events in the wild

CamJam is one of the oldest Raspberry Jams still running, and the veteran event came back after three years at Makespace Cambridge to huge success. Check out some pictures from the day…

01. The Makespace was full of people checking out projects and workshops

02. Talks on various subjects were given throughout the event

03. Games are a great way to inspire makers of all ages

04. CamJam started Pi Wars, so of course there had to be a robot royal rumble

01

03

02

FIND OUT ABOUT NEXT MONTH'S EVENTS AND POP-UP ON PAGE 92

04

Best of the rest!

Other great things we saw this month

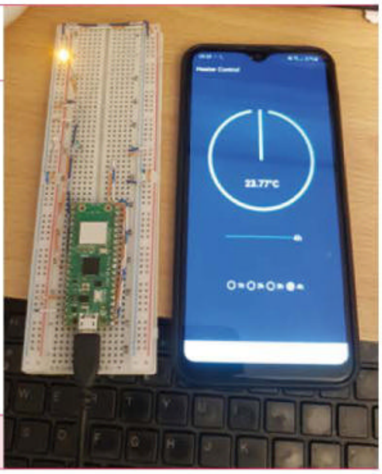
Pico W MQTT automation

Camilo Sanchez showed us his project of controlling a relay using MQTT and a Raspberry Pi Pico W, and it seemed quite neat.

"The idea of making this project started when arriving home from work, or elsewhere, in winter and waiting for the house to warm up," Camilo tells us. "Although setting a timer on the thermostat control could be the solution, I wanted something fancy and efficient. So I decided to make my own boiler control. After researching on alternatives to port forwarding, MQTT met the requirements. So after this, it was time to test the MQTT API for MicroPython on Raspberry Pico W, where I was able to communicate with a cloud MQTT server system from Adafruit (Adafruit IO) using feeders(topics)."

He also created an Android app that communicates with it and displays data. Be exceptionally careful if you plan to recreate this project though – boilers and relays shouldn't be messed with so lightly.

▶ **magpi.cc/picowiot**



Your Letters

Monday emails

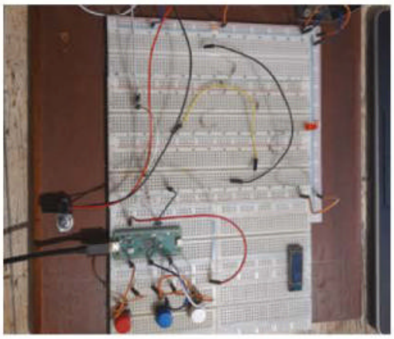
This is my entry for #MagPiMonday. As I don't have Twitter, or the like, I need to email it to you.

It is my Pico experiment board that I have just made bigger. It is based on my small board that I made for my 12 Projects of Codemas advent calendar I received last year.

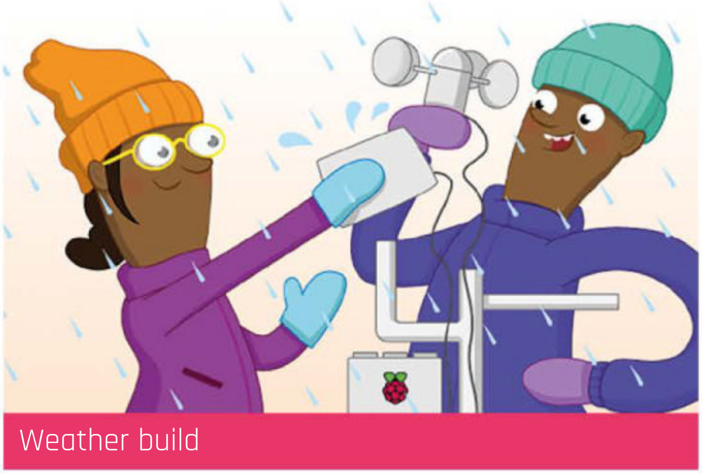
If it is good enough to make it in, it would make my year.

Moss via email

We welcome any kind of entry for #MagPiMonday, especially if it's as cool as this. You can email us at **magpi@raspberrypi.com** or, Rob directly, at **rob@raspberrypi.com**. We love to see them!



The layout of this breadboard is very neat and we feel like we should learn from Moss



Weather build

I'm interested in building a Raspberry Pi-based weather station that can collect and transmit data to a central server. I'd like to be able to monitor temperature, humidity, barometric pressure, and anything else that would be suitable. Are there any tutorials or guides available for this kind of project?

Irma via Facebook

The Raspberry Pi Foundation's project site (**magpi.cc/projects**) has a good tutorial on building a weather station based around a special kit that they produced around 2016. It's quite an involved build, so if you're looking to measure just about everything you can, it's a great place to start. Check it out at **magpi.cc/weatherstation**.

Error in printing

I had issue 129 of *The MagPi* magazine delivered to me this week and there is a misprint in it. Page 35 is not correct – it is a duplicate of page 37. I don't have the 'Getting started with Home Assistant' page.

I have verified this by downloading the pdf version.

Simon via email

Unfortunately there was an error during printing we were unable to catch before the issues were sent out, and some readers like yourself may have noticed how page 37 was printed twice. The correct page 35 can be found in the free PDF, which you can download at **magpi.cc/129**.

Apologies for this inconvenience.

Contact us!

Twitter	**@TheMagPi**
Facebook	**magpi.cc/facebook**
Email	**magpi@raspberrypi.com**
Online	**forums.raspberrypi.com**

Community Events Calendar

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

01. Hybrid Raspberry Pint

- Tuesday 30 May
- **Microsoft Reactor (London), London, UK**
- **magpi.cc/pint129**

Raspberry Pint is a meetup for sharing digital making experiences. Most presentations are about building personal or professional projects with Raspberry Pi and other maker technology. You can also learn skills and techniques such as website design, PCB design, software development, 3D printing, and soldering.

02. Melbourne Raspberry Pi Meetup

- Sunday 4 June
- **Docklands Makerspace and Library, Melbourne, AUS**
- **magpi.cc/mrpm130**

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

03. Cornwall Tech Jam

- Saturday 10 June
- **Fraddon Village Hall, St Columb Major, UK**
- **magpi.cc/ctm130**

Cornwall Tech Jams are run by volunteers working in IT and education throughout Cornwall. They are supported by Software Cornwall, its members, and other local businesses. Our volunteers give their own time and expertise to plan each Cornwall Tech Jam and to the regular maintenance of all our equipment.

04. Riverside Raspberry

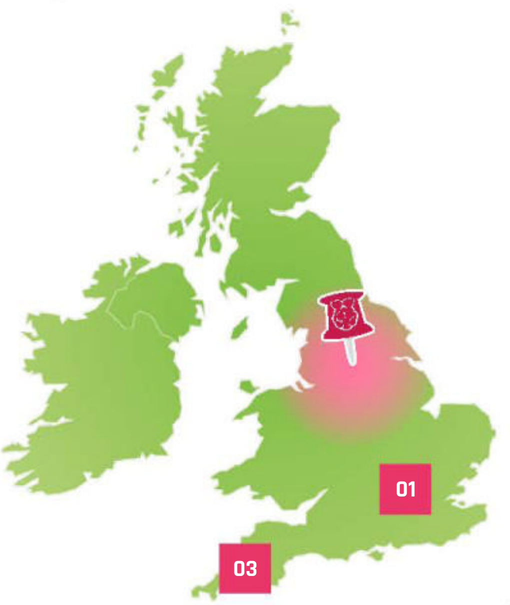
- Monday 12 June
- **3600 Lime St, Riverside, CA, USA**
- **magpi.cc/riverside**

Riverside Raspberry is a California-based enthusiast group that's been meeting up for almost a decade. This group will appeal to thinkers and tinkerers, the scientifically curious, software developers looking to get into embedded development, or anyone young or old that has a knack for tech in general.

FULL CALENDAR

Get a full list of upcoming community events here:
magpi.cc/events

RASPBERRY PI STORE POP-UP



Where can you find Raspberry Pi next?

- Next location Raspberry Pi Store pop-up Leeds
- Where Victoria Gate, Leeds, UK
- When Friday 2 June and Saturday 3 June

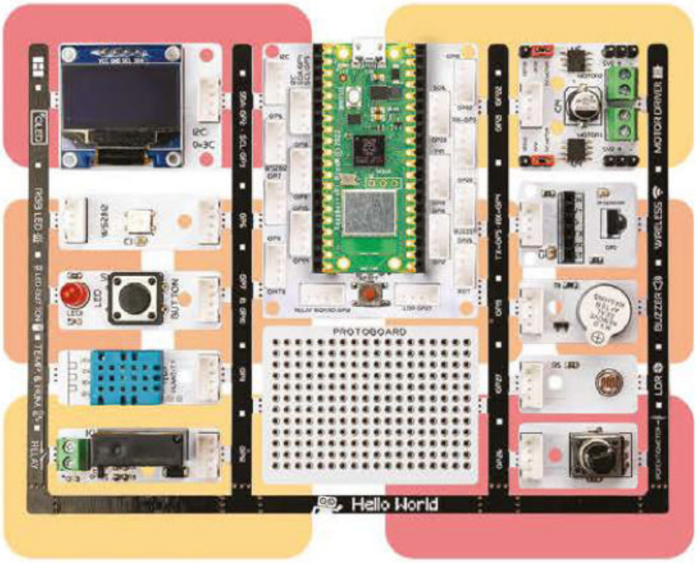
At this Raspberry Pi Store pop-up in Leeds, you can experience and buy Raspberry Pi products. Explore some of the things you can do with a Raspberry Pi, discover our accessories and books, and get your hands on limited edition exclusives.

magpi.cc/popup2023leeds

MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
3D Evolution
NFC MAKE PERSONALISED GREETINGS CARDS
ULTRASONIC LIGHTS
MOTION-DETECTING LIGHTS FOR YOUR WORKSPACE
PRINTING A LARGE FORMAT CAMERA
PRUSA'S NEWEST PRINTER ON TEST
PICO CHATGPT CATAPULT SCOPE

WIN ONE OF THREE PICO BRICKS BASE KITS IN ASSOCIATION WITH The PiHut

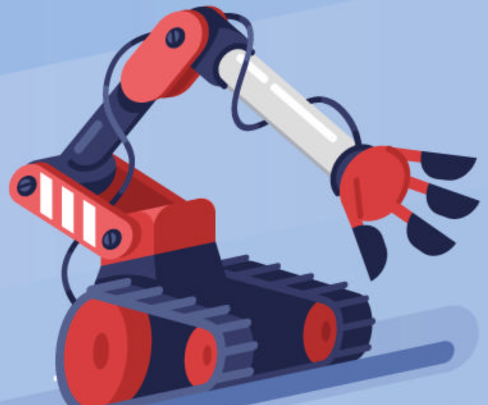
We reviewed Pico Bricks last issue, and thought it was a great way to experiment with electronics and other projects that are powered by Raspberry Pi Pico. It includes loads of different components, including a tiny screen, and we have three to give away!



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

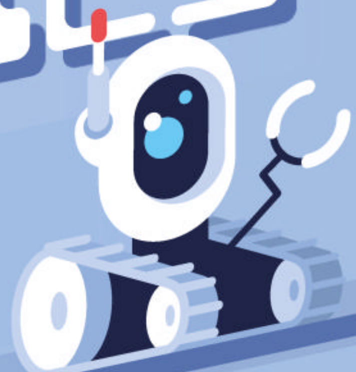
Terms & Conditions

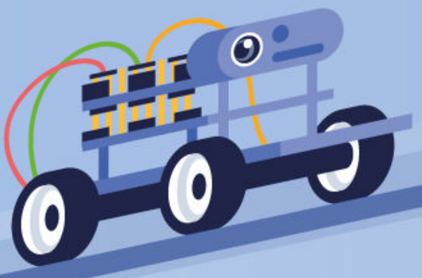
Competition opens on **24 May 2023** and closes on **29 June 2023**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By 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, Facebook, Twitter or any other companies used to promote the service.



RASPBERRY PI ROBOTICS

ROLL YOUR OWN ROBOT





THE MAGPI **#131**
ON SALE **29 JUNE**

Plus!

Wildlife photography

Vintage computing

Monitoring volcanoes

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

Illustrator
Sam Alder

CONTRIBUTORS

David Crookes, Marc de Vinck, PJ Evans, Rosemary Hattersley, Nicola King, Phil King, Sean McManus, David Plowman, KG Orphanides, Stewart Watkiss

PUBLISHING

Publishing Director
Brian Jepson
brian.jepson@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

A retro world tour

Back in the day, local makers built a range of diverse computers with fresh new ideas. We could learn from the past. **By Lucy Hattersley**

Ah, the good old days of computing. The 1980s were a vibrant time, and the computer revolution was just starting to spin.

This month, thanks to the MCM/70 Reproduction (page 14), I've been reminded of the astonishing amount of variety in retro computing. Before this month, I'd never even heard of the MCM/70 and it's wonderful to encounter it, thanks to the efforts of Canadian Raspberry Pi maker Michael Gardi.

It reminds me that each country, and often each region within a country, had its own computing startups, and their products reflected the unique cultural and technological idiosyncrasies of the time and place.

Around the world

The MCM/70, produced by the Canadian company Micro Computer Machines, holds a special place in the annals of computing history as one of the first, if not the first, personal computers. Introduced in 1973, it boasted an Intel 8008 processor, 4KB of RAM, and an APL (A Programming Language) operating system: another thing I've never heard of before.

The MCM/70's unique design, including a twin cassette deck and a built-in segmented LED display, reminds me of a modern cyberdeck. It's a pleasure to see a reproduction of this iconic machine today.

In the US, the Apple II was a revelation when it was launched in 1977. It was the first computer many Americans ever saw (here in the UK, mine was a Dragon 32). The Apple II's role in popularizing the concept of the personal computer cannot be overstated.

I preferred the Commodore 64, with its powerful multimedia capabilities. It became a beloved home computer, and also a favourite platform for game developers. It's still my favourite.

> “Each country, and often each region within a country, had its own computing startups”

Moving to the UK, the ZX Spectrum and the BBC Micro were pivotal in the country's computing history. I had a ZX Spectrum, and despite its rubbery keyboard and modest specifications, it was a revolution in the home computing market due to its affordability. The BBC Micro, on the other hand, was a staple in schools, providing many British children with their first taste of computing. Over in Japan, the NEC PC-8801 and Sharp X1 series were leading the pack. I don't think I ever saw one until visiting computing museums as an adult. Down under, the Microbee, an Australian-built computer, had a cult following. In the Soviet Union, locally produced machines like the Agat and the Electronika BK series were reverse-engineered Western designs.

Today, the world of computing can seem homogenous, dominated by a few large corporations. A MacBook is a MacBook all over the world. We use devices with eerily similar designs and identical operating systems. But in the 1980s, computers were wonderfully diverse. They were local, they were personal.

One of the most fascinating aspects of the Raspberry Pi is its ability to emulate retro computers. This enables enthusiasts to experience the computing past first hand, and younger generations to discover the rich history of personal computing.

AUTHOR

Lucy Hattersley

Lucy has been shipped a copy of Zelda: Tears of the Kingdom a day early on press deadline day and it's like she touched a monkey's paw.

magpi.cc/lucyhattersley

HiPi.io

PiKVM

Manage your servers or workstations remotely

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

PiKVM HAT
for DIY and custom projects

Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz Váš dodavatel Raspberry Pi · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io