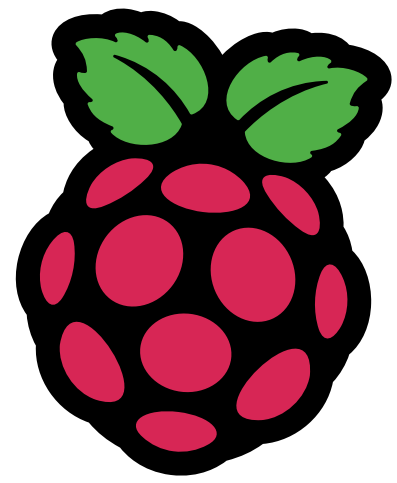


BUY IN PRINT **WORLDWIDE** [MAGPI.CC/STORE](https://magpi.cc/store)

The *MagPi*



Issue 129

May 2023

magpi.cc

The official Raspberry Pi magazine

DIY HOME AUTOMATION WITH RASPBERRY PI

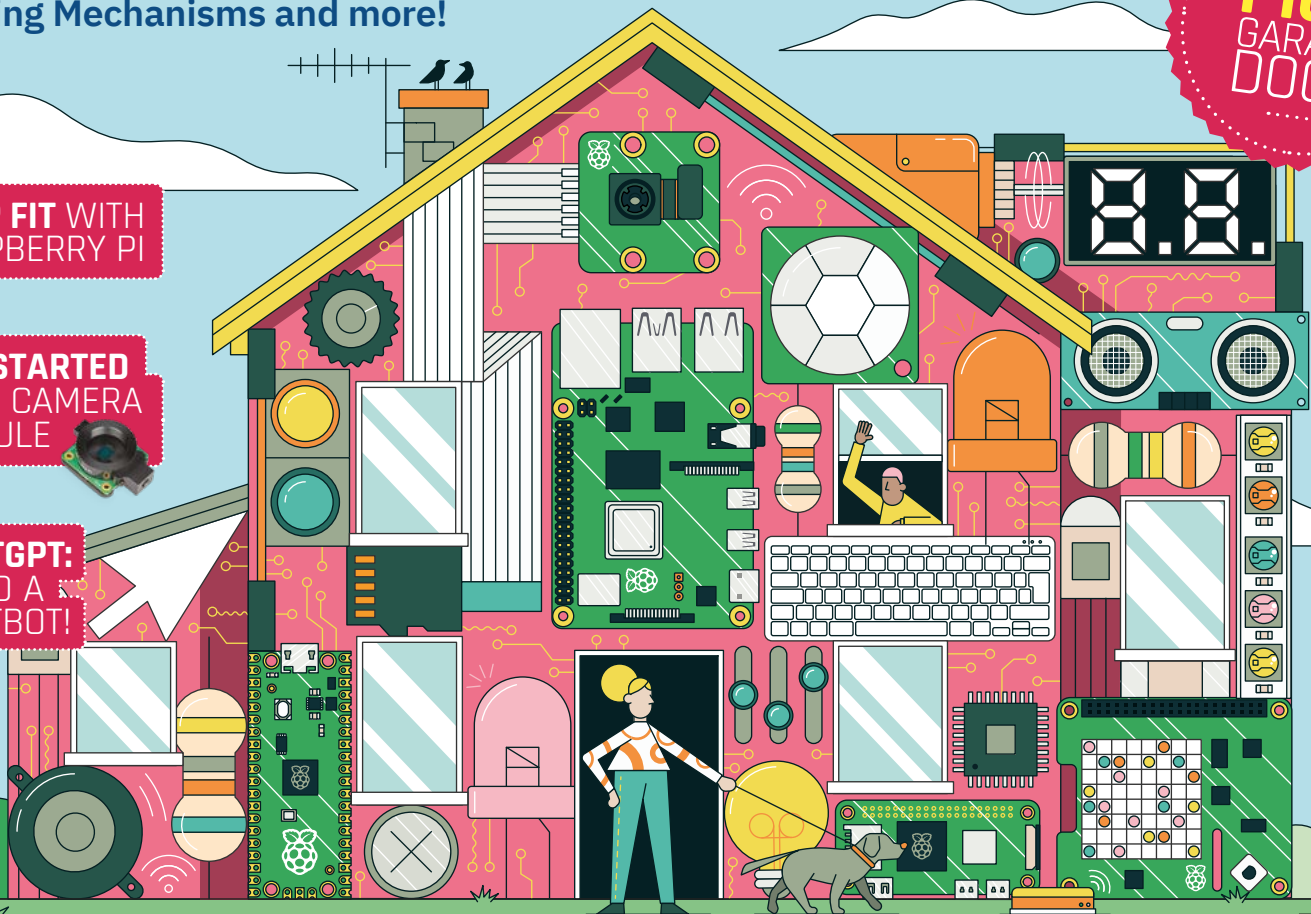
Install Home Assistants, Smart Sensors, Moving Mechanisms and more!

PICO
GARAGE
DOOR

KEEP FIT WITH
RASPBERRY PI

GET STARTED
WITH CAMERA
MODULE

CHATGPT:
BUILD A
CHATBOT!

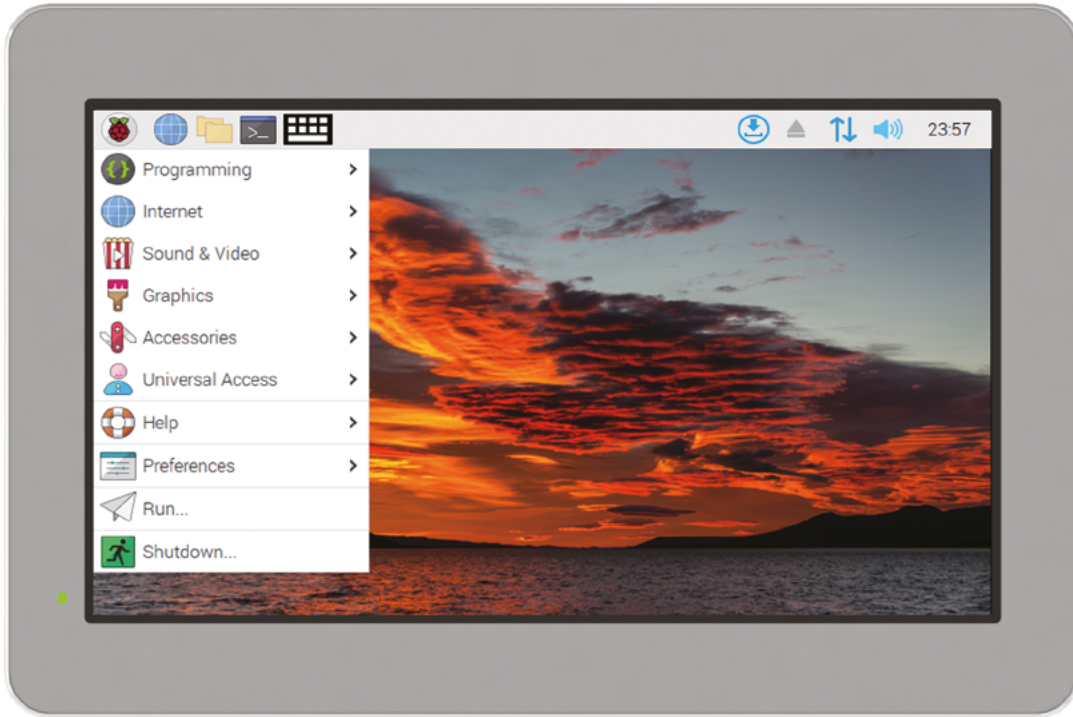


GLOBAL
DELIVERY
magpi.cc/store

40 PAGES OF **TUTORIALS & PROJECTS**



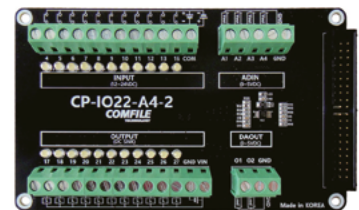
Industrial Raspberry Pi



ComfilePi

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

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



WELCOME

to The MagPi 129

Computers should be put to work, and we think Home Automation is a fun way to get our little computers to do some heavy lifting. Our resident home-hacker, PJ Evans, has put together a great feature about getting Raspberry Pi to open the curtains, manage the garage door, and turn the lights on (**page 34**).

Meanwhile, Rob Zwetsloot has written an excellent article on Getting Fit with Raspberry Pi (**page 66**). It's the perfect way to combine Raspberry Pi's mental stimulation alongside looking out for your physical health. I'll leave you to it.

Meanwhile, we've got some amazing projects this month. KG Orphanides is on-the-ball with a tutorial on how to set up ChatGPT API on Raspberry Pi (**page 42**), there's also a LEGO brick that plays DOOM (**page 16**), and a Clamshell BlackBerry Cyberdeck (**page 18**).

This issue is as packed as ever, and it's been an absolute pleasure to make. I hope you enjoy it as much as we do.

Lucy Hattersley Editor



EDITOR Lucy Hattersley

Lucy is editor of *The MagPi* and is currently arguing the merits of good wine over beer to a massively unreceptive bunch of techies. She's almost been run out of the company. Anyway, beer next month!

magpi.cc

GET A
**RASPBERRY PI
PICO W**
WITH A SUBSCRIPTION!
PAGE 28

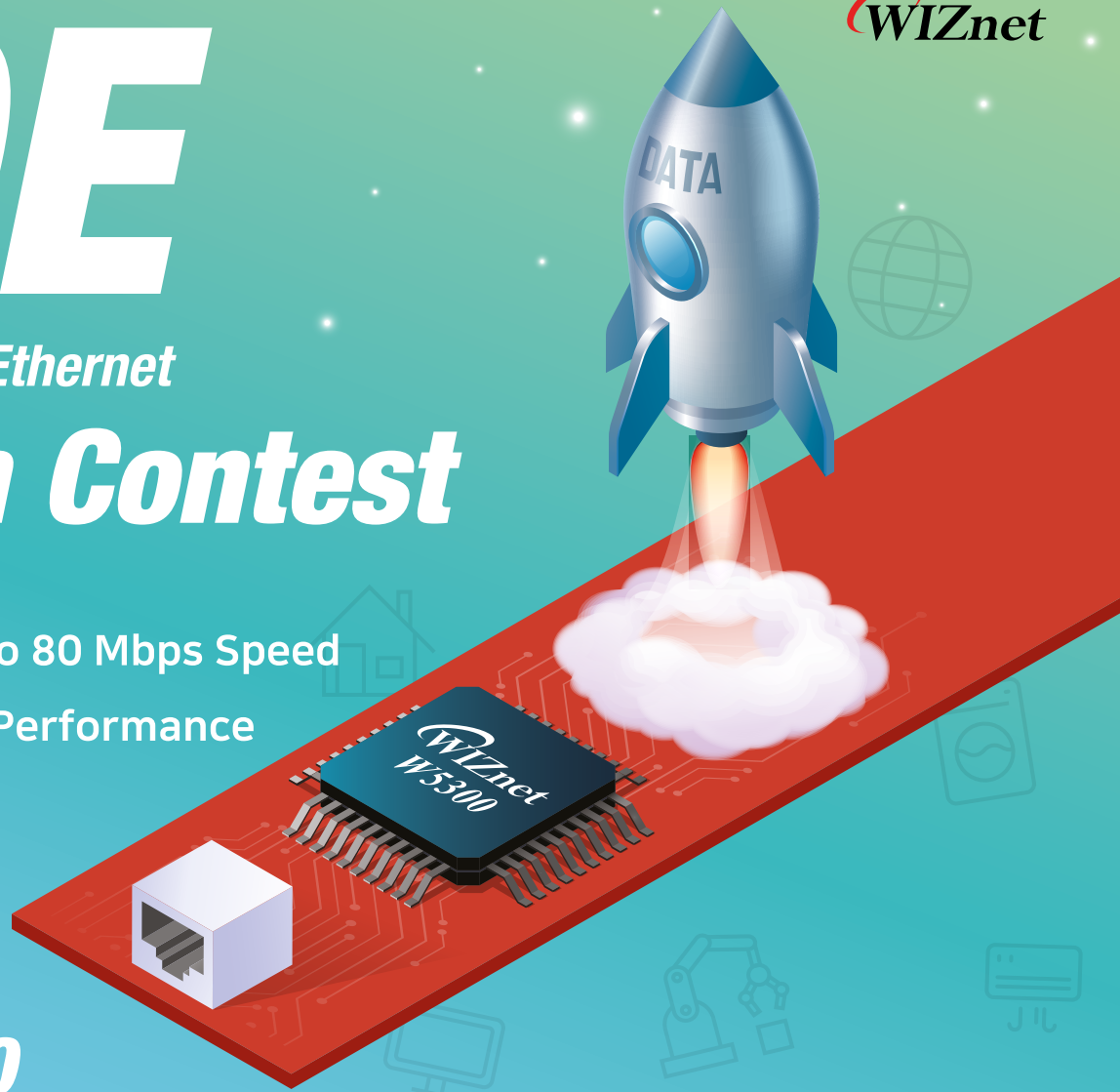


TOE

TCP/IP Offload Ethernet

Design Contest

- Supporting up to 80 Mbps Speed
- Targeting High Performance Applications



USE W5300

- 8/16 bit Data Bus Interface (Direct & Indirect Address Mode)
- High Network Performance : Max 80Mbps (by DMA)
- 8 Independent hardware sockets

Buy

W5300(\$5) or W5300 TOE Shield(\$10)
at WIZnet e-shop



W5300



W5300 TOE Shield



Contest begins

Mar 16



Special price at WIZnet eshop

Mar 16 ~ May 20



Submissions close

Jul 20



Winners announced by

Aug 8

WIN A CHANCE FOR **\$ 30,000** IN TOTAL

1st PRIZE
\$ 5,000

x 2

2nd PRIZE
\$ 2,500

x 4

3rd PRIZE
\$ 1,000

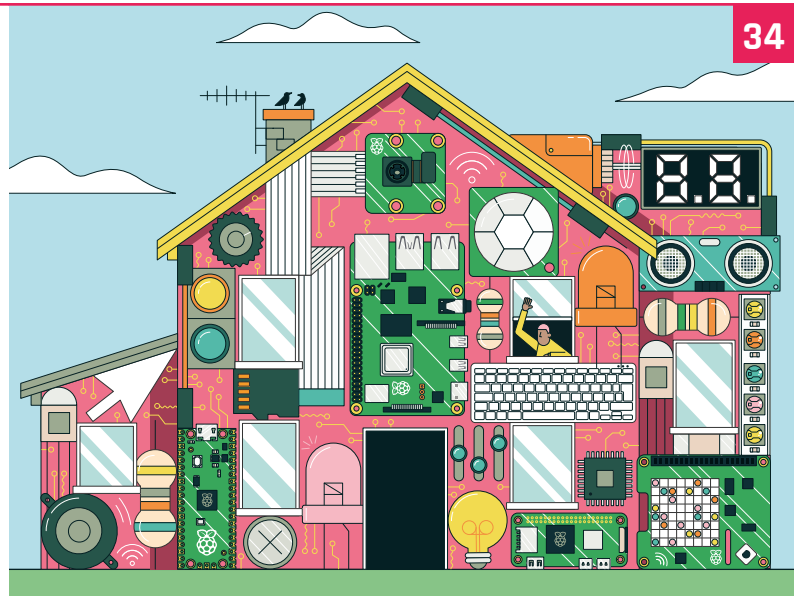
x 10

Contents

► Issue 129 ► May 2023

Cover Feature

34 DIY Home Automation



34

Regulars

- 30** Case Study: Freight Farms
- 90** Your Letters
- 92** Community Events Calendar
- 97** Next Month
- 98** The Final Word

Project Showcases

- 08** Shiloh Studios
- 12** ClippyGPT
- 16** Doom on a LEGO brick
- 18** Clamshell BlackBerry cyberdeck
- 22** Greening the Spark
- 24** Bugg.xyz acoustic monitoring

16



Doom on a LEGO brick

18

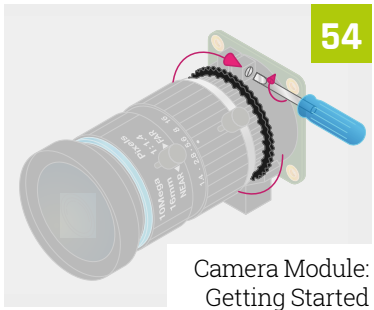


Clamshell BlackBerry cyberdeck

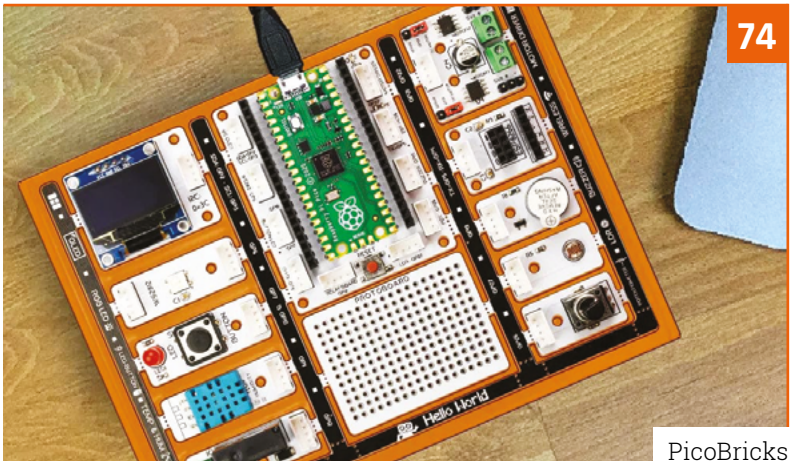
The MagPi is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The Magpi, c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

Tutorials

- 42 Build your own GPT Chatbot
- 46 Introduction to inputs and outputs
- 50 Bluetooth music visualiser
- 54 Camera Module: Getting Started
- 60 RFID spells fun



The Big Feature

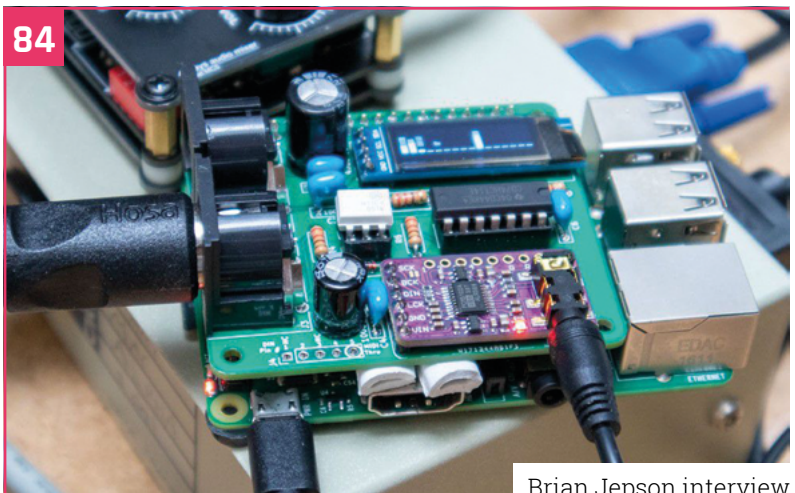


Reviews

- 72 Compute Blade
- 74 PicoBricks
- 77 10 amazing browsers
- 81 Learn Sense HAT

Community

- 84 Brian Jepson interview
- 86 This Month in Raspberry Pi

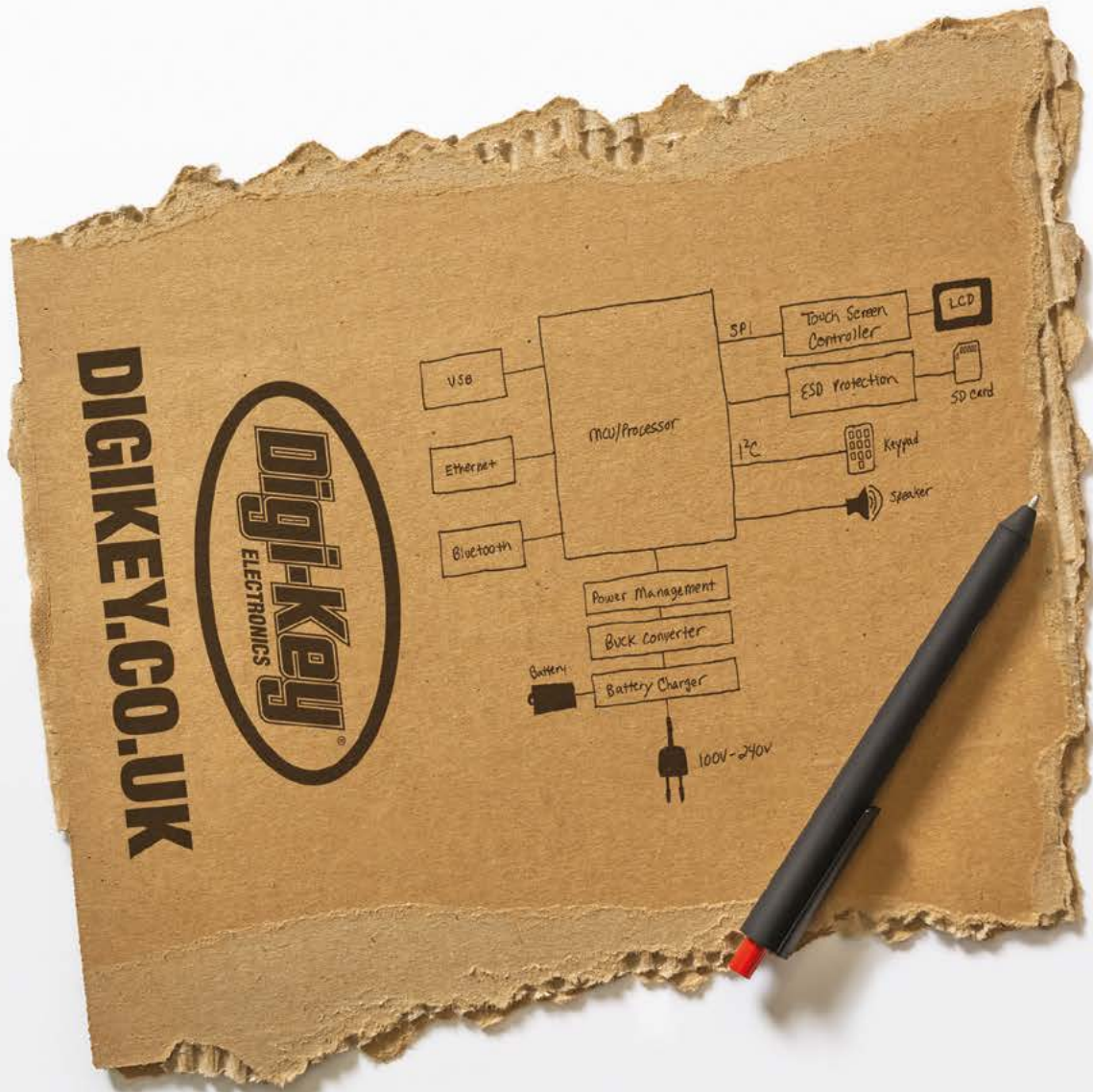


WIN 1 OF 5 GLOBAL SHUTTER CAMERA MODULES  **95**

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.

IDEAS

START HERE



From millions of in-stock parts to the latest new product inventory, we've got everything you need to turn breakthrough ideas into reality.

Get inspired at [digikey.co.uk](https://www.digikey.co.uk) or call 0800 587 0991.



Shiloh Studios



Michael Peters

Michael has been an IT executive, police officer, animal cop, musician, and music producer



Darrell Lehman

Audio recording engineer, AVL systems integrator, and musician. He also serves in roles as a photographer, videographer, and network administrator.

shilohstudios.us

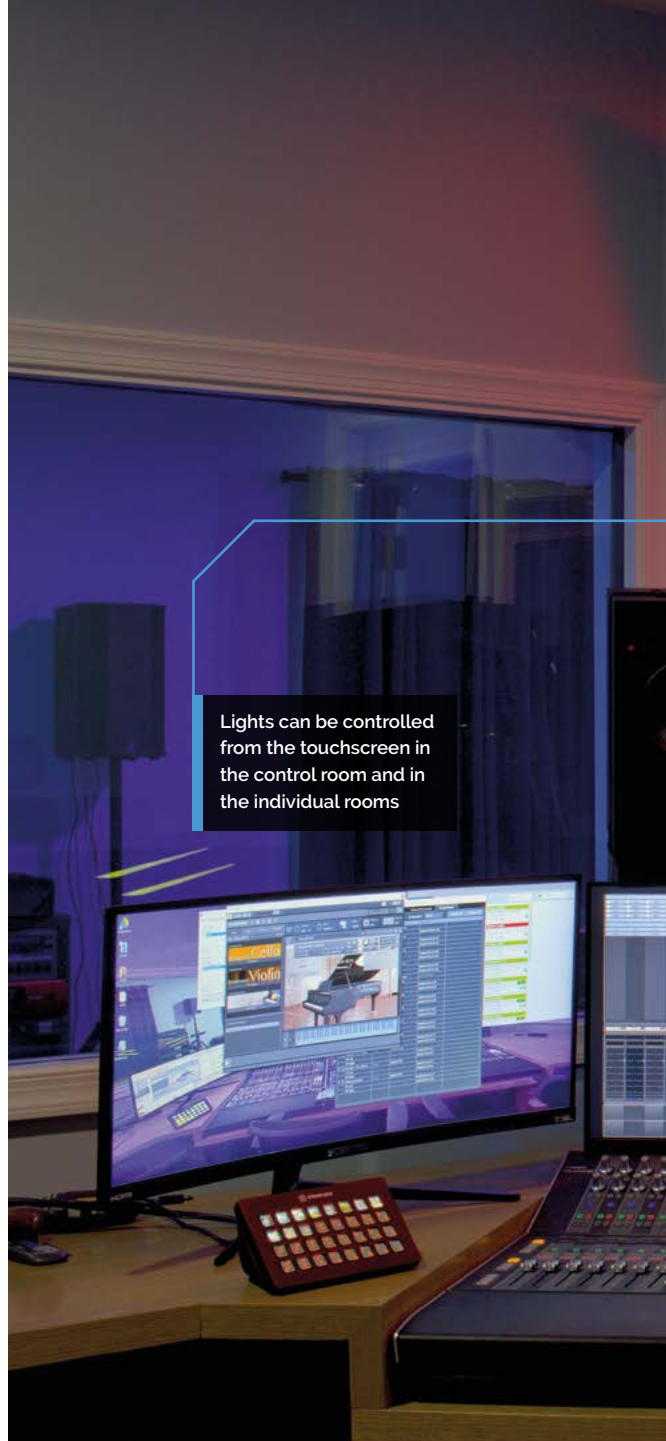
Upgrading a professional music studio can be as easy as pie with Raspberry Pi. **Rob Zwetsloot** books a session

When you need to add remote control and automation that is easily customisable, as readers of *The MagPi*, you'll understand the desire to do it with Raspberry Pi, which is what Michael Peters of Shiloh Studios figured out too.

"Shiloh Studios is a Dante-based audio production facility which leverages the latest technology best practices and solutions to ensure the highest-quality experience and outcomes for its clients," Michael says. "As the studio project design was unfolding, several challenges were encountered. After extensive research and assessment of numerous potential solutions, Raspberry Pi hit all the marks as the answer."

The different Raspberry Pi solutions were pulled off by Darrell Lehman, also of Shiloh Studios, who has been making Raspberry Pi-based projects for over five years, and has been working with electronics for over 25 years.

"I've completed a lot of prior projects on Raspberry Pi with touchscreens, [Power over Ethernet], TouchOSC, and Node-RED, so Raspberry Pi was a familiar and easy choice," Darrell explains. "I gravitate toward Raspberry



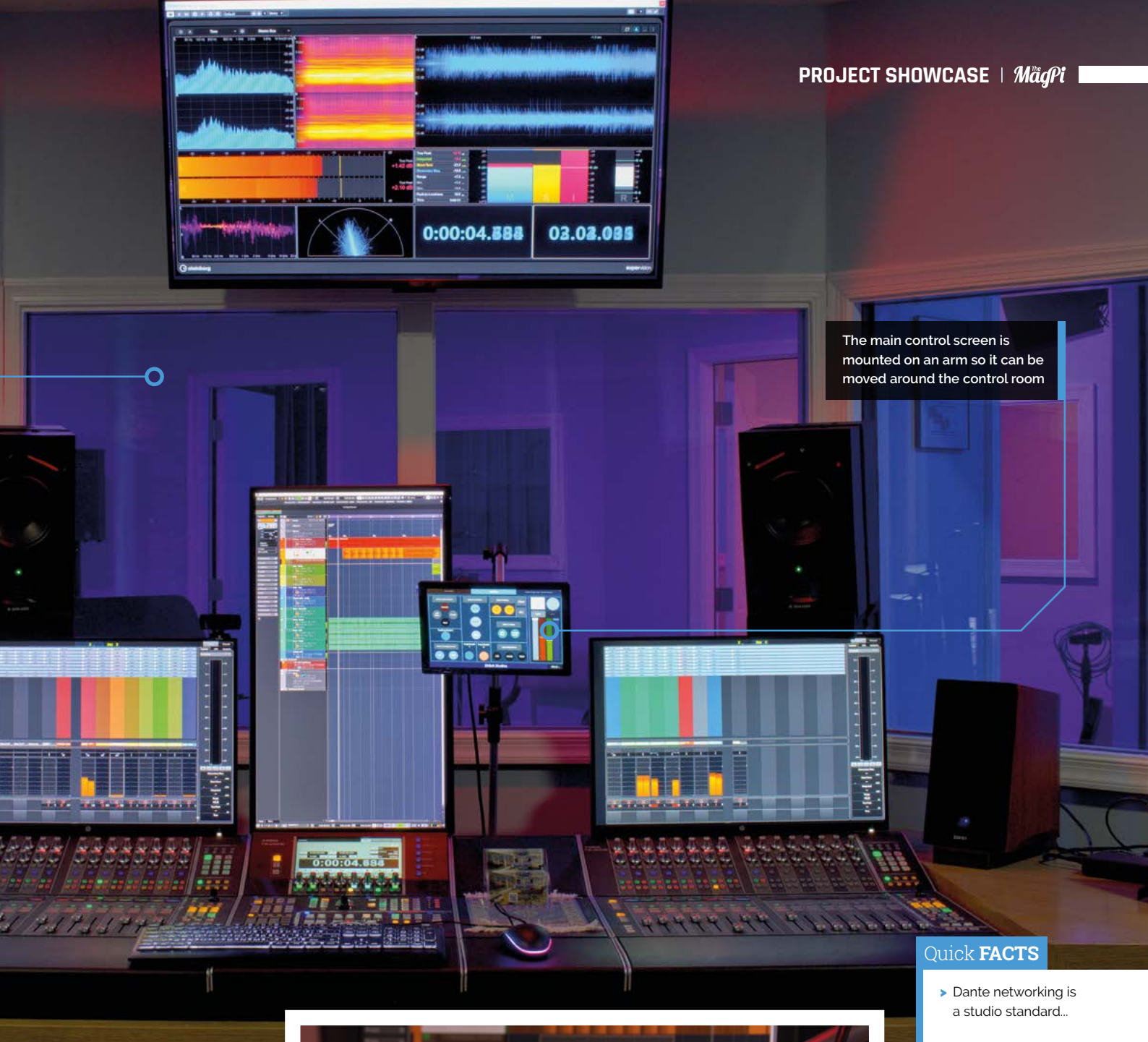
Lights can be controlled from the touchscreen in the control room and in the individual rooms

Pi's ecosystem when I need to solve problems that aren't readily addressed by an off-the-shelf hardware solution, or it doesn't make sense to implement costly proprietary control systems."

“ I gravitate toward Raspberry Pi's ecosystem when I need to solve problems ”

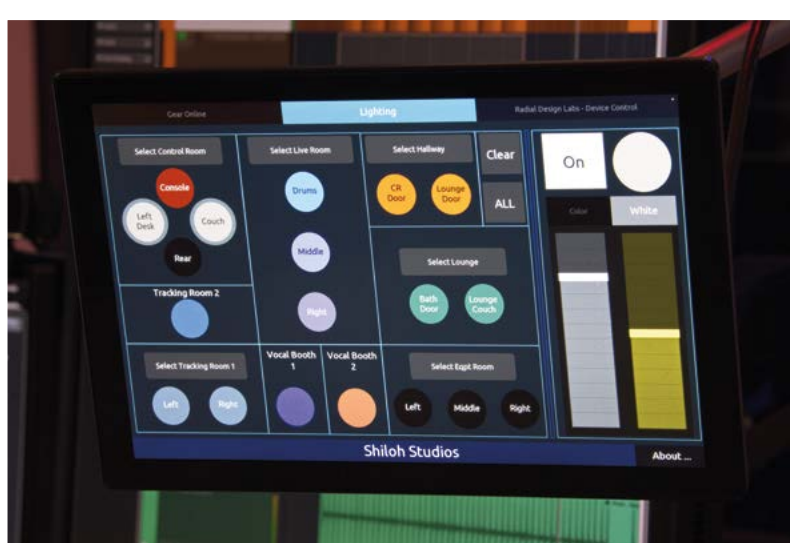
Musical fruit

There's four main ways that Raspberry Pi is used at Shiloh Studios. First of all, as a lot of things are connected via Ethernet, a Raspberry Pi helps monitor those connections.



The main control screen is mounted on an arm so it can be moved around the control room

▶ A custom interface is used for the light controls, with the studio version being the master controller



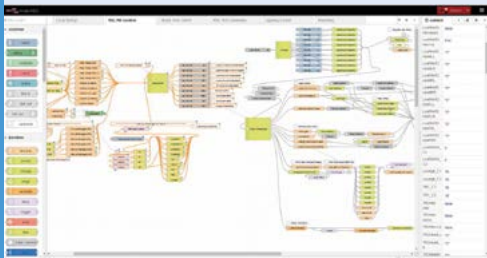
Quick FACTS

- ▶ Dante networking is a studio standard...
 - ▶ ...that can have zero latency over Ethernet
- ▶ Ethernet is used throughout the studio as a standard connector
- ▶ Lights in the studio use RGB LEDs, so can change colour
- ▶ The remote screens are independent, but there is a master remote

Building remote controllers



01 Installing a Raspberry Pi and a PoE HAT into the back of a touchscreen allows for minimal cabling needed to get everything running.



02 Once each is set up, custom Node-RED code is created to work with TouchOSC, so everything communicates properly between remotes and the lights.



03 Apparently, the hardest step was wall brackets, although they ended up quite simple in this case anyway.



▲ The screens installed in the different rooms can act independently of the main screen

There's also a system for full remote control over the lighting systems, with remote screens in multiple rooms. Similarly, there's remote control of RDL devices in various studio rooms so performers can listen to audio live.

The last main way is a USB to Ethernet connected keyboard, as Michael explains: "to facilitate the composer client's needs, a Native Instruments S88 keyboard was placed in the control room, but it also needed to be flexibly deployable to allow relocation to other studio spaces as needed. The S88 requires a USB connection to the DAW [Digital Audio Workstation] PC. Twisted pair to USB converters were not reliably sufficient to the cause. Long USB cables

▶ Everything is visible but does not block the view of other interfaces





▲ The keyboard connection makes use of better Ethernet connections over a long USB cable

posed a similar liability. Raspberry Pi 4 with a PoE HAT provided the right solution by enabling a rock-solid USB to Ethernet connection.”

“I would say construction was probably the easiest part of the project because Raspberry Pi has such a great ecosystem of add-ons available online,” Darrell tells us. “And for each accessory that I needed, I had options to choose from. The most challenging part of the construction was figuring out how I wanted to build the wall brackets.”

“ I would say construction was probably the easiest part of the project ”

On tour

On a digital tour of the studio, the various Raspberry Pi implementations had a great professional finish, and the UI created for the light remote looked and worked just like we’d expect from tablets with apps, or touchscreen products specifically made for that use.



The latency over Ethernet is very impressive – three milliseconds for the keyboard connected over Ethernet even – and Raspberry Pi is able to help control over 256 interfaces around the studio.

Perhaps the next step will be an entire DAW made with Raspberry Pi. **M**

▲ The monitors and remotes being tested and programmed before deployment

ClippyGPT

Clippy is back! And this time we're really glad to see and hear from Microsoft's iconic character, as **David Crookes** explains



David Packman

David Packman is a former Microsoft program manager responsible for chatbots, virtual assistants, and AI copilot projects, who builds companion robotics in his spare time.

magpi.cc/clippygpt

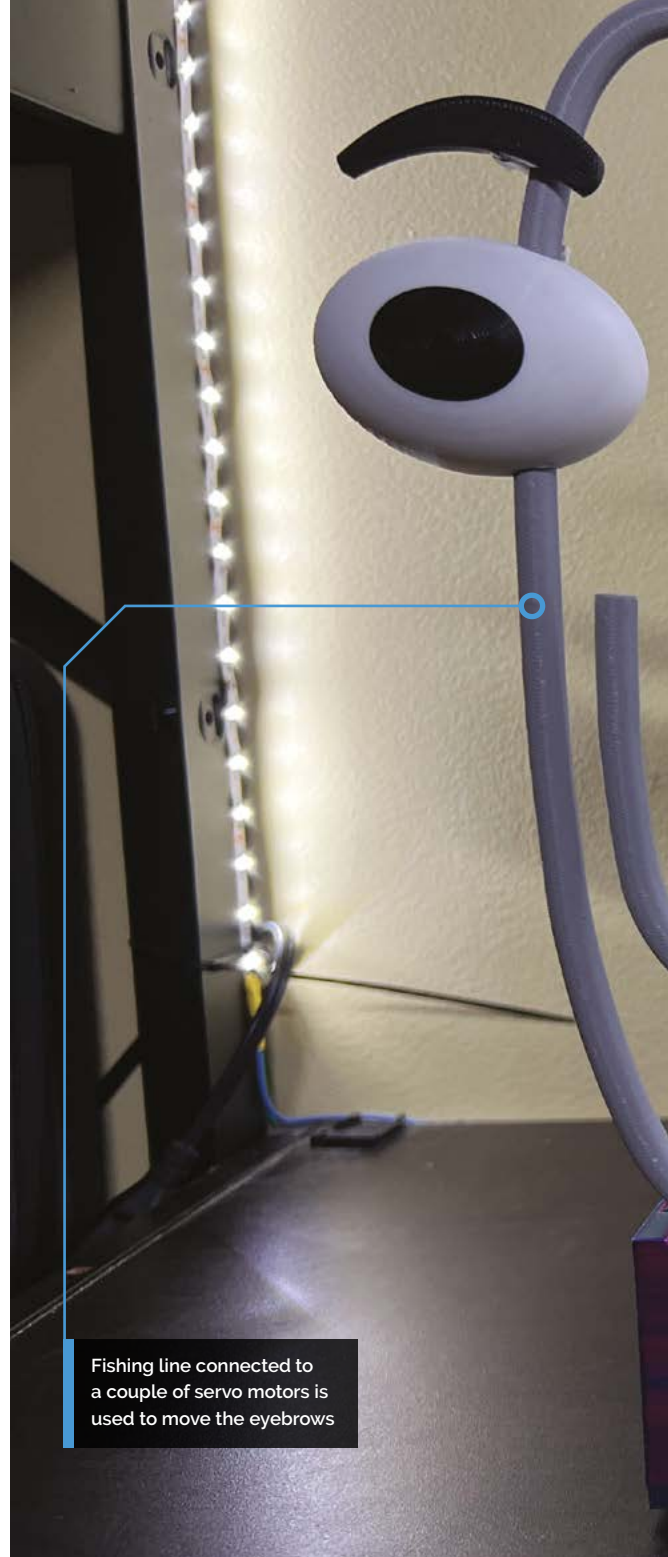
Trying to think of something to write can be hard (a fact that feels very much true right now, given that this is the fifth stab at penning an intro!). But imagine how much harder writing would be if you had a peppy paper clip in your eyeline chirpily offering hints and tips. “Do you need assistance,” it would say, as soon as you so much as tapped a single letter. And if you didn’t need help right at that moment, then it’d be sure to ask you again later!

For users of Microsoft’s Office 97, this became a reality courtesy of an ‘intelligent’ user interface called Clippy which, it’s fair to say, split opinion. But even though TIME magazine reckoned it was one of the 50 worst inventions of all time, there have been revivals. In fact, as well as being able to relive the memories of Clippy via a sticker pack for Microsoft Teams, you can now also enjoy a good old natter with the character, thanks to David Packman and a Raspberry Pi computer.

By combining Clippy with ChatGPT, the AI chatbot that is still taking the world by storm, David has brought Microsoft’s character back to life in a physical form and given users the ability to hold a conversation with it. The result is a distinctive, fun, AI-powered digital assistant. “All the memes about Clippy being ChatGPT in disguise convinced me that the first ChatGPT-empowered companion bot I should build would be a desktop Clippy,” David says.

Chatty man

David has long been interested in robotics and chatbots. “I’ve been building robots as a side project, with a focus on companion or social robotics, for a few years now, and one of the key



Fishing line connected to a couple of servo motors is used to move the eyebrows

components that has been hard to implement is open-ended conversational capability,” he says. To that end, he’s found it challenging to hard-code a manually-curated chatbot into a robotic build, but says ChatGPT was the answer.

“Conversational large language models like ChatGPT are a game-changer, especially for social robotics, so I was keenly interested in integrating this new tool into a robotics build,” he says. “I wanted to use it as a test case to find the best approach for doing this on a budget. It all stems from my desire to create helpful

Quick FACTS

- ▶ Microsoft's Clippy was formally called Clippit
- ▶ It was effectively axed after five years, in 2001
- ▶ David's build cost about \$125 in parts
- ▶ His version of Clippy moves and talks...
- ▶ ...but he shows how to make a non-moving version

A 3D-printed speech bubble surrounds the 5-inch HDMI display

Raspberry Pi connected to an Adafruit Crickit HAT sits inside the base

```

Clippy is ready to help...
RECOGNIZED KEYWORD: Hey Clippy
Recognized: What is MAG Pi magazine?

MAG Pi magazine is a free monthly publication produced by the Raspberry Pi Foundation that focuses on programming, electronics, and computing projects using the Raspberry Pi and other single-board computers. The magazine was first released in May 2012 and is available as a free digital download from the Raspberry Pi website. It is also printed and sold in various countries around the world

```



▲ What's behind the eyes? Among the components is a spring from a retractable pen, cut in half

robots that can help people who have difficulty socialising, or cannot leave home due to illness or other situations.”

Part of this drive is personal. Being able to give people the ability to make friends or create robots that can help with tasks could be of assistance to his son, who has special needs and finds socialising difficult. “I’ve always wanted to help him make friends,” David says. “A conversant chatbot using an inexpensive API really advances that aim, so I had to figure out how and then share that with others.”



▲ As this shot from behind shows, the audio is emitted from the base box.

Say it again

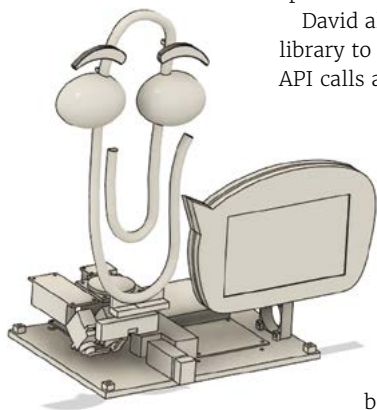
At the robot's heart is a Raspberry Pi 3 Model B computer which he took from an older robot build knowing that it was compatible with other components he owned. One such add-on was the Adafruit Cricket HAT that enables the creation of robot friends using Python by allowing motors, servos, or solenoids to be controlled. "There were some software library dependencies that made it easier to use Raspberry Pi OS as well," he adds.

To handle speech, David turned to Microsoft Azure's speech services. This allowed him to convert speech to text and vice versa, with David finding the free tier provided all the capability he needed to provide a speech interface with ChatGPT. "It also has the benefit of recognising an offline wake word," he says. "So nothing is recorded or sent over the internet until I specifically activate it."

David also made use of the OpenAI Python library to run the ChatGPT queries. "The OpenAI API calls are the only expense, but it's pretty cheap for a build like this," he says. "One day of heavy usage was only eight cents. Also, the Python SpeechRecognition library is largely broken, so it was a very appealing alternative with extra benefits."

Getting animated

A key part of the whole build has been the animatronics. While Clippy's speech is outputted to a speech bubble (essentially a 3D-printed case surrounding a 5-inch HDMI display, fitted with a salvaged speaker from an Amazon Echo Dot), the paper clip has wiggling



▶ You can download all of the STL files and an offline wake word training table from David's Hackster page



eyebrows which help to bring this retro cultural icon to life.

"The animatronics are pretty simplistic, although it was a bit of a bodge to get the eyebrow mechanism to work, since the directional force has to wind through some curved passages

“ The animatronics are pretty simplistic ”

inside Clippy," David says. "In the end, I used a technique that we use to build 3D-printed assistive device hands, which is fishing line.

"To counter the pull of the lines, I also had to design a spring mechanism to return the movement to a start position after the servos release the tension on the lines. I couldn't use bands since that would pull the mechanism out of the sockets in the back, so I resorted to the old standby of using compression springs from old retractable ink pens. From a coding perspective, adding movement to the conversational aspect is relatively easy."

Looking ahead

In all, the build took David a week to complete. He was able to do this with a single Python script.



“And I did it in a way that made it really easy to integrate into any project, in order to build something that was more engaging and personable than just a canned speaker with a microphone.”

David says he is now integrating the techniques into more complex robotic builds, such as a conversational companion bot based on BMO from the Cartoon Network’s *Adventure Time* animated television series, and a counter-top kitchen robot called Whiskrz.

“That will include some machine vision to help identify ingredients, suggesting recipes on what you show it, and do simple tasks such as mixing,” David adds. “I’ve left the build simple so that it’s easy for others to adjust, allowing them to make more cool things with it. There are lots of improvements that can be made based on what you need.” [M](#)

▲ What’s behind those eyes? A close-up of the fittings

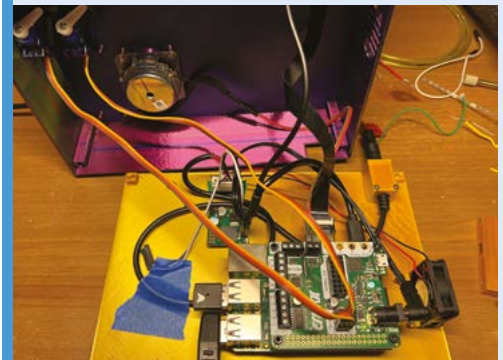
▼ Once everything is connected, a USB microphone is slotted into the front of the case



Bringing Clippy to life



01 Once the keyword is recognised (this is carried out offline using a keyword recognition routine and a locally stored training model table), a speech recognition routine runs and Clippy gives its eyebrows a wiggle! It’s ready to help!



02 After listening to what is spoken until there is a pause in the speech, there are more movements to let the user know the question has been heard. The speech is transformed into text and the speech recognition stops listening.



03 The text is sent to ChatGPT as a prompt and the text answer is sent to the next routine. The text is then fed into the Text-To-Speech routine which displays the response on the screen and replays it as audio.

Doom on a LEGO brick

If you have an appreciation of both LEGO and 1990s video games, read on.

Nicola King digs out her glasses to admire a minuscule make



James Brown

James lives in Wellington, New Zealand, has a background in games and graphics programming, and creates interactive exhibitions.

magpi.cc/ancientj

We've seen portable games projects before, but nothing quite this small!

Maker James Brown took on the huge challenge of embedding a playable game of the classic first-person shooter Doom in a LEGO brick, powered by Pico's RP2040 chip and featuring a tiny OLED screen.

James makes clear that it was an iterative process over many months. The first version (magpi.cc/firstlegobrick) was "very simple and not particularly well-planned. I figured I'd make a tiny computer, attach a 0.42" OLED which just (only just!) fits in the face of a LEGO slope brick, and somehow encapsulate it in resin to look like the original. So, I drew up a PCB as big as I could go without it sticking out, populated it with the smallest microcontroller I could find in stock at JLCPCB (jlcpcb.com), and sent it off to be manufactured."

After the project garnered a positive response, he sought to make it accessible to more people. The question was, how could he create a version that could support a USB connection and be reprogrammed interactively? "RP2040 ticked all the boxes," he shares, "and (crucially!) was available."

3D construction

James ended up with a 3D PCB structure, taking what he'd learnt from the first iteration and improving it: "By soldering together a complete frame of PCBs, I get a very precisely machined brick shape, contacts for the battery, and a USB port tucked away inside."

The frame consists of four panels, with the front one containing the driving forces including the RP2040, a 4MB flash chip, crystal oscillator, and 0.42" SPI OLED screen. "Four lugs on the front panel provide the physical mounting points," explains James, "as well as the electrical connections to the rest of the panels – two for power, two for data."

With 9V contact strips on the side panels, along with a voltage regulator and some diodes, the

rear panel has just enough room for a micro USB socket. The other three panels have castellated edges to provide "the mechanical and electrical connection, and can be designed as a single flat circuit which is later chopped up to make the 3D frame."

Once James had assembled the circuit, he put it in a silicone mould to be encapsulated with epoxy resin. The biggest headache was getting the resin layer over the screen just right: "It needs to be very thin (around 0.1mm), both in order to fit in the space between the outer surface and the studs of the bricks below, but also to be clearly visible."

Doom's day

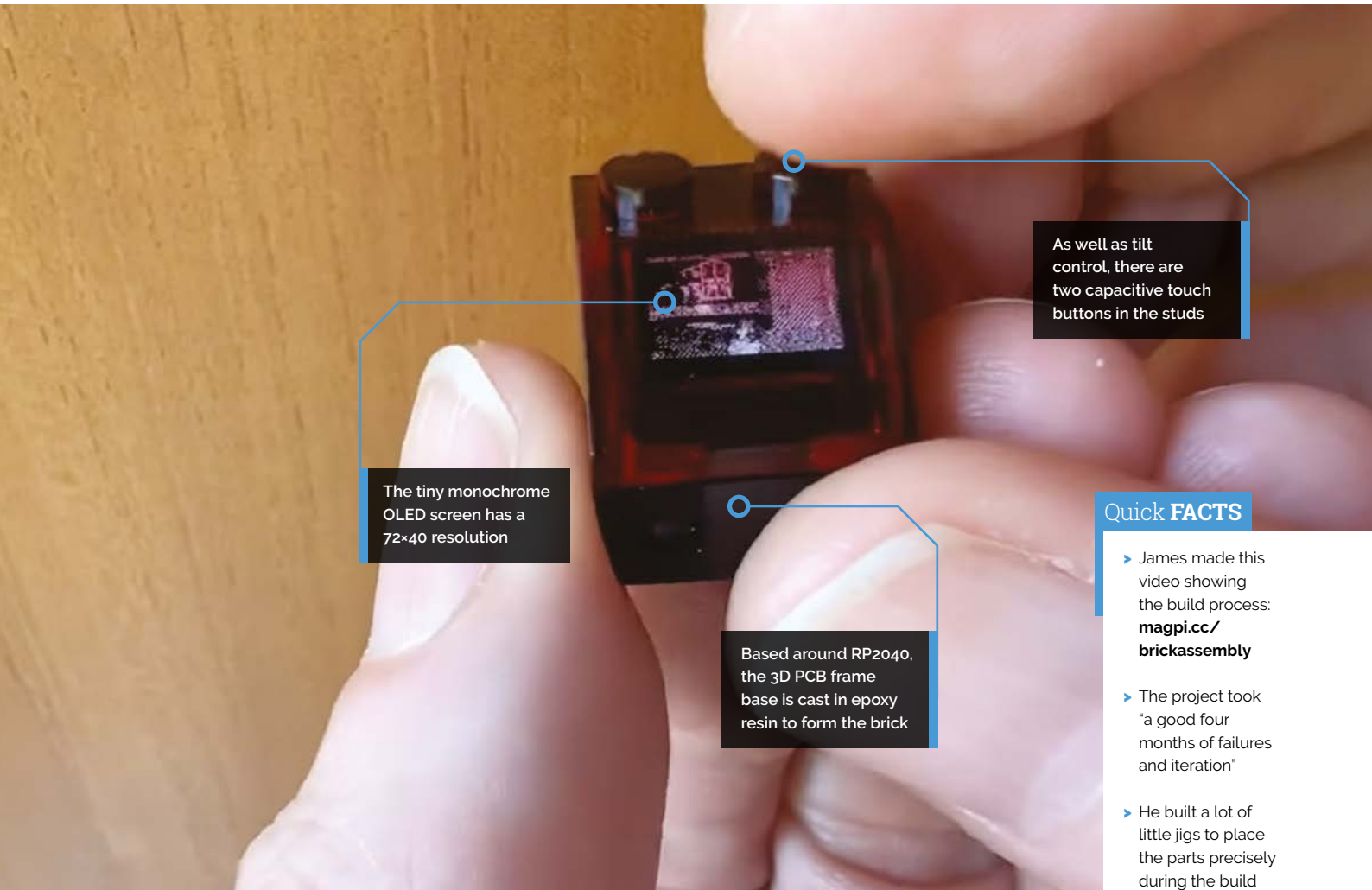
Sparked by questions from social media as to whether his LEGO creation could run Doom, James investigated the possibility but had initial issues running video on the screen. After some tinkering, he "took advantage of the RP2040's second core to update the screen fast enough to

“ The biggest headache was getting the resin layer over the screen just right ”

create greyscale images (magpi.cc/ugrey).” With Doom now a practical proposition, James looked to Graham Sanderson's "amazing" work on a Doom port, did a "horrible quick hack" and it now runs beautifully, ticking off "another entry on 'things that run Doom'."

Seeking to improve the pocket-sized project further, he's now working on adding an internal battery. "I found some rechargeable cells which could fit, but only if I remove the USB port, and I'm still looking for a neat solution for this."

As well as making a test board to try out the battery and charging circuit, James has copy-pasted the LEGO brick's computer module onto it "to make a wearable ring (magpi.cc/doomring) which can run all the same code as the bricks."



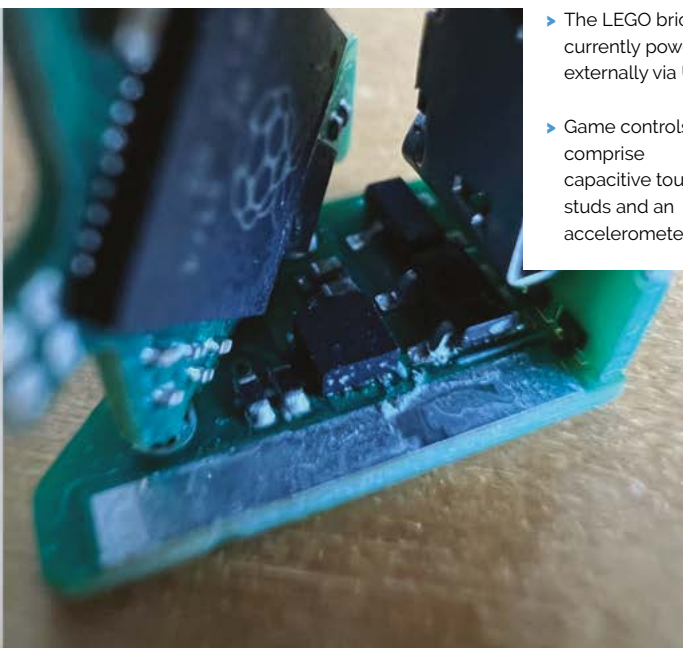
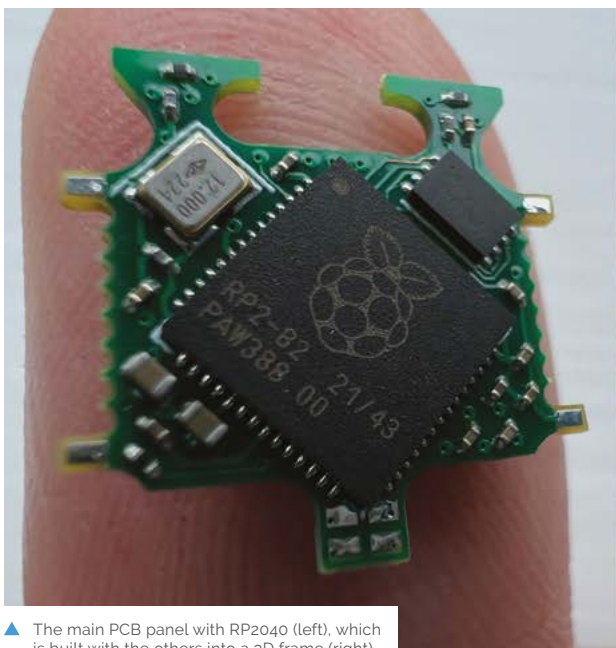
The tiny monochrome OLED screen has a 72x40 resolution

As well as tilt control, there are two capacitive touch buttons in the studs

Based around RP2040, the 3D PCB frame base is cast in epoxy resin to form the brick

Quick FACTS

- ▶ James made this video showing the build process: magpi.cc/brickassembly
- ▶ The project took "a good four months of failures and iteration"
- ▶ He built a lot of little jigs to place the parts precisely during the build
- ▶ The LEGO brick is currently powered externally via USB
- ▶ Game controls comprise capacitive touch studs and an accelerometer



▲ The main PCB panel with RP2040 (left), which is built with the others into a 3D frame (right)

Clamshell BlackBerry cyberdeck

Keeping things as compact as possible drove the design of this hinged Raspberry Pi 4-based tiny computer, discovers **Rosie Hattersley**



Michael Klements

Michael is a mechanical engineer who often shares his inventive ideas and Raspberry Pi creations on his YouTube channel.

magpi.cc/michaelklementsyt

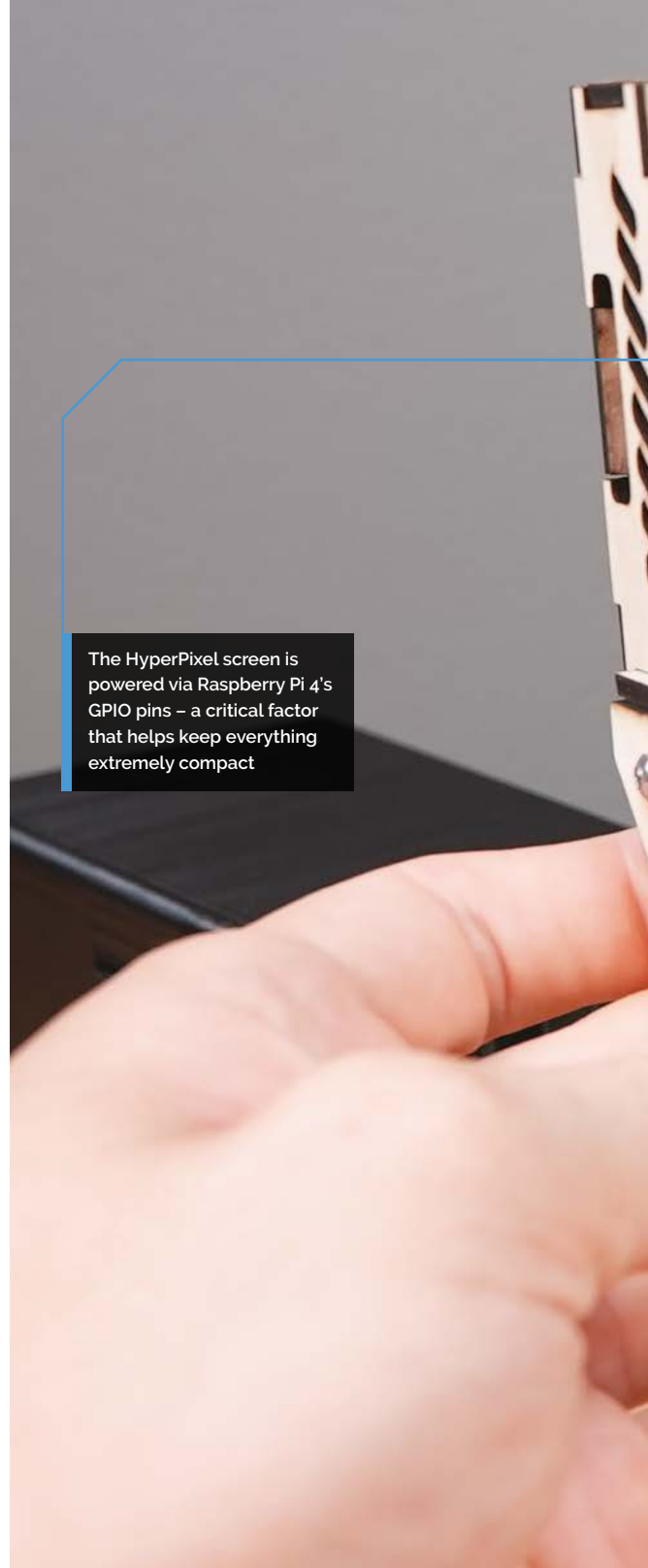
As you might infer from a word prefixed with ‘cyber’, cyberdecks have their origins in the world of 1980s science-fiction.

This particular version, however, harks back to the early 2000s when PDAs were a must-have business accessory, but the iPhone (and as such, smartphones) had yet to launch. Maker Michael Klements chose a clamshell version of a BlackBerry and paired it with Raspberry Pi 4 for maximum computing power, aiming for “a functional desktop computer with a GUI”.

Michael has tackled a fair few Raspberry Pi and other microcontroller projects over the years. As well as the sheer number of projects that can be undertaken, he notes the helpfulness and knowledge of the Raspberry Pi community when a build proves especially challenging. The BlackBerry clamshell cyberdeck project provided the sort of learning opportunity he enjoys and, while he would probably have chosen it anyway for its computing chops, this project became a Raspberry Pi one by default since the HyperPixel 4.0 display he would need for the upcycling challenge currently works only with our favourite single-board computer. Michael realised there was a real advantage to using the HyperPixel 4.0 screen, as it could be powered entirely via Raspberry Pi’s GPIO pins. This had space-saving advantages, since there would be no need for an external power supply or to connect to a DSI or HDMI port.

Fruitful enterprise

Having solved the screen and power conundrum, for what he hoped would be an all-in-one



The HyperPixel screen is powered via Raspberry Pi 4’s GPIO pins – a critical factor that helps keep everything extremely compact

computer that vaguely resembled a tiny laptop, Michael set about sourcing a suitable keyboard. “Most keyboards, even compact and foldable ones, are three to four times bigger than the final build size that I was going for – and you’d still need to add a mouse to that,” he explains. A Tindie store selling keyboards from old BlackBerry mobile phones, paired with a custom carrier board so the trackpad and keyboard act as a USB connected mouse and keyboard, proved a winning purchase



Quick FACTS

- ▶ Michael previously designed a LattePanda clamshell cyberdeck
- ▶ The BlackBerry cyberdeck's plywood case is just 3mm thick
- ▶ This helped trim vital millimetres off its dimensions
- ▶ Design and laser cutting took about six hours...
- ▶ Helped by the fact Michael has repurposed BlackBerry cases before

A veteran builder of Raspberry Pi cases, Michael pushed the boundaries by making this one from 3mm thick plywood

Michael sourced an old BlackBerry keyboard on a custom carrier board that had been repurposed to act as a USB keyboard and mouse

since they worked with Raspberry Pi. The setup provided “the perfect fit for my build, as the keyboard footprint is smaller than the display’s, and still allows you to get full mouse and keyboard functionality.”

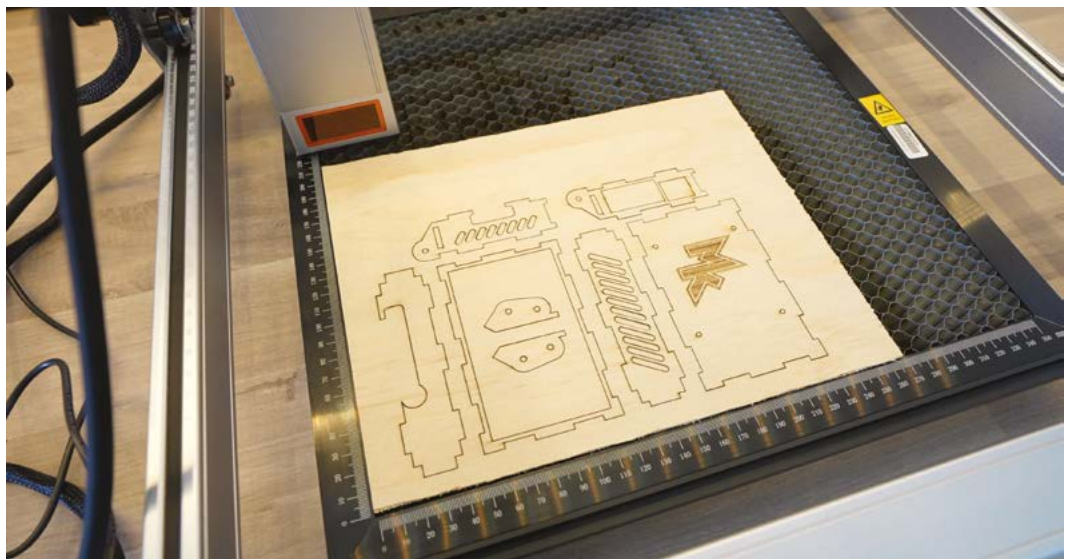
Laser-like focus

Michael designed and laser-cut parts, using Inkscape open-source software to create a case as small as possible, so the overall package wasn’t

much larger than the components themselves. Designing the hinges also “took a bit of work” to get the extent to which they would open just right. Michael wanted to limit the clamshell’s opening travel to have the display at about a 20-degree angle. “This makes it easier to use as a handheld device and lets it sit on a desk like a small laptop would.” The plywood parts were glued together before the Raspberry Pi-based components, HyperPixel display, and repurposed



- ▶ Having installed Raspberry Pi OS, it's time to test the cyberdeck's BlackBerry keyboard
- ▶ Michael designed the case in Inkspace before laser-cutting the parts






BlackBerry Pi keyboard were inserted. Bezels to keep everything in place were then fitted and glued on both halves of the clamshell case.

Getting the angle of the screen just right took a couple of attempts, but Michael has created several laser-cut Raspberry Pi cases before, and is used to working with their dimensions and the clearances involved. The finished build has caught the imagination of many makers who follow his blog and YouTube channel, with several requests for a 3D-printed version.

“ There was a real advantage to using the HyperPixel 4.0 screen ”

For anyone keep to replicate the project, full build details can be found on Michael’s blog: (magpi.cc/diycyberdeck). It can be used to make a portable travel computer for a range of tasks – portable NAS, VPN, travel router, mobile programming platform etc. Michael, himself, is already looking into adding a portable battery pack and plotting further cyberdeck designs. Unlike William Gibson’s sci-fi original, Michael doesn’t seem to be intending to use his cyberdeck as a human brain-to-cyberspace interface! 

▲ The HyperPixel 4.0 display shows off videos and games, such as Big Buck Bunny, to great effect

Cyberdeck style-up



01 Sites such as Tindie feature sellers who upcycle old BlackBerry keyboards and fit them to a custom board for use with a Raspberry Pi. Just add a mini USB keyboard.



02 Use Michael’s step-by-step instructions at magpi.cc/diycyberdeck to create a suitable Raspberry Pi case (laser-cut or 3D-printed) and carefully assemble and glue it together.



03 Add Raspberry Pi 4 and attach the HyperPixel screen using the GPIO pins, then insert and attach the BlackBerry keyboard and other parts. Install Raspberry Pi OS and configure the screen using Pimoroni’s instructions.

Greening the Spark

How does green energy work? One maker decided to create an interactive museum piece to teach people. **Rob Zwetsloot** investigates



Carl Nicholson

Carl used to be technical director of Cyberdrome Enterprises, as well as being responsible for simulators used to train astronauts for the ISS. He's now retired.

cornucopia.co.uk

What do you want to do when you retire? Some might want to relax, others might like to travel. Carl Nicholson had a different idea.

“When I retired and moved back to the UK, I wanted to do something that would raise awareness of climate change and help people understand the ideas behind green energy,” Carl tells us. “I also wanted an excuse to do electronics, maths, coding, and making physical models and equipment.”

The result of this was Greening the Spark, a “simulator of a National Grid designed as a hands-on museum exhibit which can also be played as a game,” as Carl describes it.

Carl comes from a background of making and simulators, working in space operations in Munich, training astronauts and flight controllers on the Columbus module of the International Space Station, and also being the technical director of the company that made the first commercial Crystal Mazes.



▲ The graphical version of the control panel is used to set up the game

Renewable interest

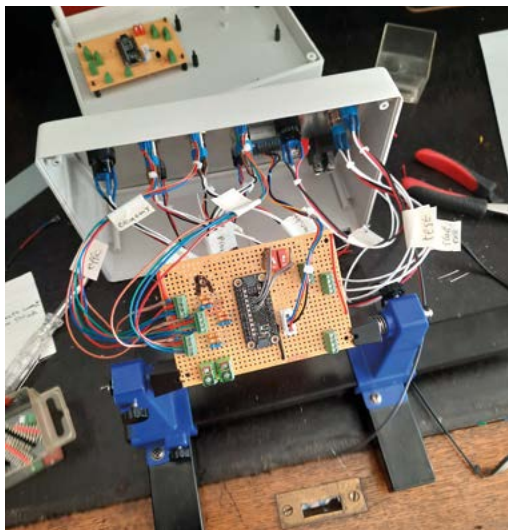
Why did Carl choose Raspberry Pi, though?

“I wanted a small but powerful single-board computer with Wi-Fi and sound, a wide range of off-the-shelf third party hardware and software, and a suitable well-supported language and IDE to code it with,” Carl says. “Raspberry Pi and Python turned out to be the perfect combination.”

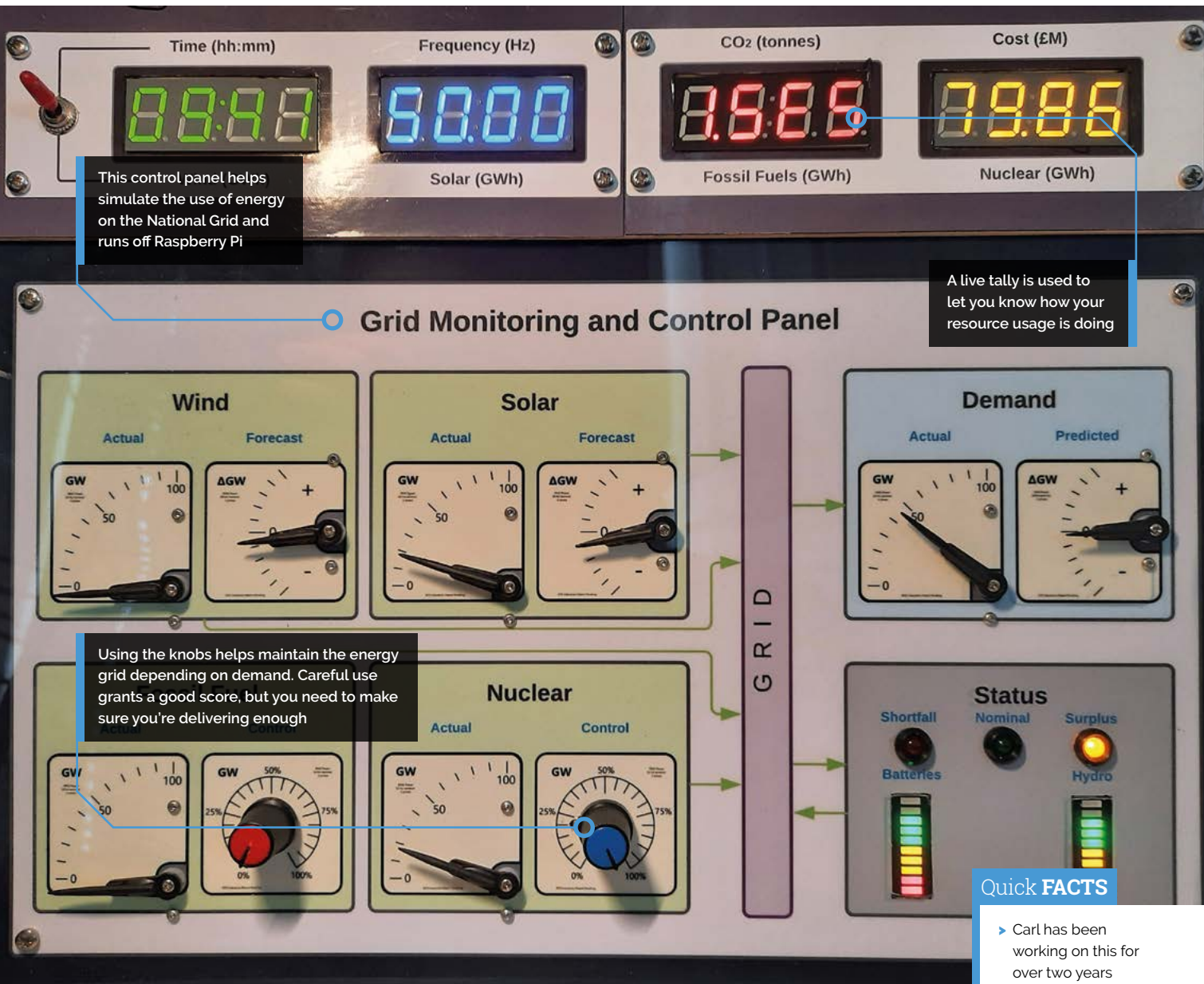
Building it took a while, and was apparently very hard, as everything else was designed and built from scratch using the Red Robotics RedBoard+, an add-on robotics controller for Raspberry Pi. It does quite a lot, though.

“There are software models of renewable electricity sources (wind and solar) driven by a weather model, non-renewables (fossil fuels and nuclear) which are controllable, storage devices (batteries and pumped hydro), and consumer demand,” Carl explains. “There is a working model of a wind farm (with three turbines) and a sunlamp. The idea is to control the fossil fuel and nuclear power to meet demand without blackouts or surplus and to keep the storage devices close to half-full.

“There is an operator control panel to monitor and control the grid, and a visitor control panel to set up a game. There are three ‘characters’ who



▶ The control box is custom-made with custom electronics inside



help visitors understand what is going on with spoken messages: a robotic system voice and two human guides, as well as sound effects and music to add ambience.”

As it's a game, players get a result which is uploaded to a dedicated website, including a summary of what you did and a certificate related to your final 'score'. “In a museum installation, there will also be a themed landscape with physical mock-ups of the various elements,” Carl adds.

Futuristic energy

The project is still ongoing, although is already quite impressive. Carl has ideas for what he'd like to do with it, though.

“ I'm also keen to develop both the hardware and software ”

“I'm hoping to interest museums of science and technology, energy supply companies and possibly the National Grid itself, as well as schools and universities,” Carl says. “I'm also keen to develop both the hardware and software to make it more modular, generic, smarter, and connected. It would be nice to use live data streamed from the National Grid and some AI in the control system.”

Quick FACTS

- ▶ Carl has been working on this for over two years
- ▶ There are currently 11,500 lines of code
- ▶ Due to complexity, it makes use of GPIO expanders
- ▶ There are three model wind turbines for the energy farm diorama
- ▶ Four high-brightness 12V LED arrays are used to emulate solar

Bugg.xyz

acoustic monitoring

Soundscapes can provide a valuable insight into the health of a habitat.

A Raspberry Pi can help conservationists listen in, explains **Rosie Hattersley**



MAKER

Sarab Sethi

Dr Sethi is a post-doctoral research fellow in Cambridge University's Department of Plant Sciences.

bugg.xyz

The louder the dawn chorus and the cacophony of wildlife, the better the locale and, by extension, life for the creatures that dwell there. This is a simplified take on the importance of acoustic monitoring, as conducted by Bugg.xyz using a Raspberry Pi-based system.

Bugg wildlife traps record the sounds of the forest, and are AI-trained to listen to the quietest creatures, as well as those that broadcast their whereabouts to all and sundry. In fact, it's more of an acoustic census, creating detailed records of the particular soundscapes of contrasting habitats, noting changes that may indicate deforestation or a natural disaster leading to depleted biodiversity. Dr Sarab Sethi, an environment research scientist at Imperial College London, developed the Bugg acoustic monitoring system, which is now being used to record the sounds of forests from Norway to Taiwan, as well as Bali and Borneo.

Sound and vision

Sarab created his Bugg kit having read about field monitoring devices built using off-the-shelf components and based around a Raspberry Pi and which featured a microphone, a USB audio card, a plug-in internet dongle, and a weather sensor. Suitably inspired, Sarab set about designing "a continuous, long-term fully autonomous acoustic monitoring device" that could send details back to the lab for analysis and which was robust enough to be placed and remotely monitored in a variety of habitats. With price as well as the hardiness of the components to consider, Raspberry Pi was an obvious choice: "The bigger questions weren't so much about whether a

Raspberry Pi would work with it, but more around the field logistics of mounting one in a forest to withstand temperature swings, humidity swings, and long duration deployment."

Helpfully, Sarab's PhD supervisor had set up SAFE (Stability of Altered Forest Ecosystems),



▶ The team in Norway retrieves the Bugg boxes each winter, since there's little wildlife activity then and deep snow hampers recording



Bugg's injection-moulded case provides robust weather-proofing and can be securely lashed to a tree in even the most exposed environment

Acoustic monitoring using a microphone, attached to Raspberry Pi 3B or above, helps further our understanding of humans' impact

Quick FACTS

- ▶ Weatherproofing for the first Bugg monitoring kit was rustic...
- ▶ A gardener's dry-box was used to protect cables
- ▶ Later versions are 3D-printed and injection-moulded
- ▶ DC power correction compensates for erratic electricity supplies
- ▶ Solar panels power the devices throughout all seasons bar winter



Solar charging is sufficient to power Raspberry Pi, the microphone, and 3G/4G mobile module so sounds can be sent and processed as soon as they're recorded



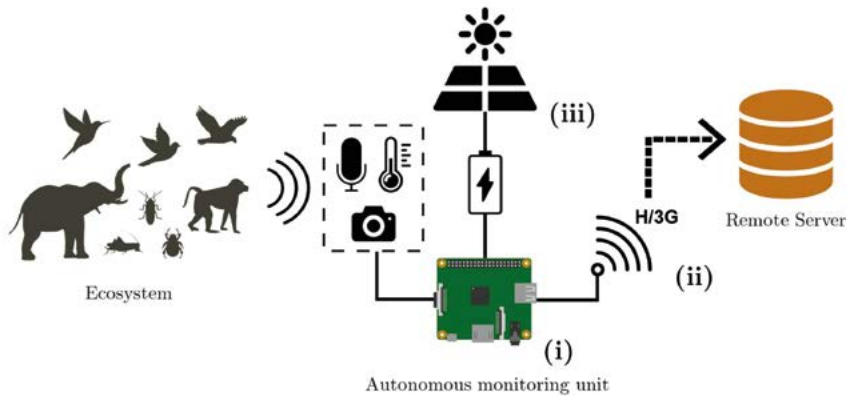
- ▶ Sarab [rear] and colleague install Bugg boxes in the Borneo rainforest, which is subject to man-made degradation that affects both wildlife and the forest itself
- ▼ Field recordings of insects, invertebrates, and birds can be viewed and heard on the SAFE Acoustic site, along with maps showing habitat health variations

a rainforest degradation monitoring project in Borneo (acoustics.safeproject.net), which formed the ideal proving ground for the Bugg acoustic monitoring system.

“Whenever we’re setting up an ecological experiment, we’re looking at gradients,” says Sarab. The SAFE Project questions how habitat degradation in Bali and tropical forest scenarios affect biodiversity. Samples are taken from pristine virgin untouched forests to lightly logged and heavily logged forests and then ultimately, the oil part monocultures. “You can also look at machine learning embeddings of the audio or fingerprint, go and survey ten sites then figure out which soundscapes correspond to what types of biodiversity and what type of habitat quality.” By going to ten more sites, or visiting the same site on ten different days, Sarab can use Bugg to infer “just based on that minute of audio, what the habitat quality or the biodiversity or stability of that system is.”

In the event, Bugg more than held its own against the various moth-trapping and bug-detecting traps, and neatly complemented studies into how forest fragmentation and forest degradation affect biodiversity. The

A screenshot of the SAFE Acoustics website. The top left shows the logo and navigation links "Find Out More" and "The SAFE Project". The main area features a topographic map of Borneo with several red location markers. Below the map, there is a section for "OLD GROWTH" with coordinates 4°39'51.96" N 117°32'6.48" E and a sun icon indicating the time 05:38. On the right, there is a profile for a "Chestnut-backed Scimitar Babbler" with a photo of the bird and a play button icon. The bottom of the screenshot shows a timeline from 01:00 to 00:00 and a waveform visualization.



results were published in the journal *Methods in Ecology and Evolution*, plus a website that details how to go about building your own device (magpi.cc/ecosystemmonitoring). Unfortunately, acoustic monitoring sometimes reveals a lack of wildlife and, what appears from photos to be healthy rainforest, is actually an empty forest.

Anyone home?

The open-source Bugg kits are now used for multiple monitoring tasks. While vegetation sampling involves taking samples and measuring trees, Sarab's Bugg monitor takes a more holistic approach, "using soundscapes as a whole to infer habitat health." Once the microphone in one of the Bugg devices detects sounds, they are transmitted over a mobile network for machine

▲ This schematic shows how data is continuously captured from an ecosystem and uploaded to a remote server

“ A rainforest degradation monitoring project in Borneo formed the ideal proving ground for the Bugg acoustic monitoring system ”

learning-based identification and analysis by scientific researchers. Bugg is generally used to pick up the sounds of particular insects or bird species, so the Raspberry Pi 3-based kit can be trained to recognise and almost instantly identify what's making each sound, be it migratory bird flocks over Arctic Norway or ants in an Indonesian rainforest. Some projects look at a particular habitat in great detail over time, as is the case in Borneo, with four distinct gradients of degraded forest, while in Norway there's a focus on multiple sites with the aim to get a detailed overview of bird life and habitat choice across the whole country (thesoundofnorway.com). [M](#)

Get bugging



01 Sarab shares the project setup and materials information on his website (magpi.cc/ecosystemmonitoring). You'll need Raspberry Pi, a microphone, USB audio card and a Camera Module, a waterproof case to house it, and a means of powering everything.



02 Install the software image provided on the GitHub link to a microSD card of at least 32GB capacity. Connect the microphone, camera, audio card, and other components to Raspberry Pi.



03 A gardener's dry-box makes a decent waterproof housing. Tape the microphone to the outside of one end, and run cables for the power adapter (if using) to the other. Snap the lid into place and lash the box to a secure post or tree, out of harm's way.

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W



Subscriber Benefits

- ▶ **FREE Delivery**
Get it fast and for FREE
- ▶ **Exclusive Offers**
Great gifts, offers, and discounts
- ▶ **Great Savings**
Save up to 35% 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) | £35 (USA) |
| £35 (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 WSubscribe in print
today and get a **FREE**
development board

- ▶ 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.

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

Freight Farms

Raspberry Pi 4-powered vertical hydroponic farms let customers grow and monitor fresh produce anywhere. By **Rosie Hattersley**

With food insecurity increasing, and global temperatures rising, two graduates from the Massachusetts Institute of Design began working on a business idea to address how to increase food production in non-agricultural settings, beginning with rooftop gardens in Boston. Their idea was to increase “local food access and create a replicable model and a product that could be used to grow food anywhere, all across the world.” Backed with a modest amount of seed funding, which they put towards a Kickstarter campaign, co-founder and current president John Freeman and co-founder Brad McNamara came up with the innovative idea of retrofitting shipping containers as locations for vertical hydroponic farms. Sensors would monitor pH levels, nutrients, light, temperature, air, and soil moisture, while cameras checked on plant growth. Container availability and relative ease of shipping, given their original purpose, plus a temperature-controllable environment, meant the self-contained farms could be delivered and set up almost anywhere.

THE CHALLENGE

Turning a lofty ideal into a successful business plan and making good on the promise that early investors anticipated is a challenge for any startup. However, it was clear from the outset that container farming had potential. Freight Farms’ first offering – the Leafy Green Machine, launched in 2013 – featured a climate-controlled seeding section and a main growth area where vertical crop columns saw a variety of leafy vegetables flourish. The container farms featured high-efficiency LEDs, used relatively little energy, and effectively filtered and recycled the water they consumed, making them particularly efficient with this essential resource. They were able to produce up to four tons of food per year.

Despite attracting high-profile customers, Freight Farms was keen to improve the efficiency of its container farms along with the range of crops that could be grown, and thus boost their appeal to a wider range of customers. Both would help unlock the business’s potential beyond the US and beyond its original base of campus and community organisations. Freight Farms set about creating preconfigured farms, marketed as turnkey products that could be set up remotely by the customer and would be seen as a practically fail-safe investment.

Accordingly, Freight Farms has spent the past few years “honing what a container farm looks like, improving the tech behind it and the yields” in order to demonstrate its validity as a grow-anywhere option, explains supply chain manager Meaghan Holmes. Yields had the potential to be a particular challenge, since certain crops are very difficult to grow or to provide the right conditions for, especially as a location’s climate changes. “One of our goals – and our farmers’ goals – is to make the most use of a small amount of space by growing as much as possible,” says Holmes. Monitoring improvements required an increased focus on IoT and the in-house development of a camera with a much wider field of view, while customers were also keen to be able to check on



their container farm remotely, necessitating an app for remote viewing.

THE SOLUTION

Freight Farms recently launched its Raspberry Pi 4-powered Greenery S flagship model. As well as using the top-of-the-line Raspberry Pi, this container farm features a bespoke camera unit. Prior to this, Freight Farms had used off-the-shelf security cameras to monitor plants. “These were limited in terms of setup and configurability, and did not have enough resolution or FOV in a small enough package to embed inside the walls of the farm,” explains Jake Felser, CTO at Freight Farms.

“ Raspberry Pi-powered Greenery S container farms offer a 30% improvement in yield between the original Leafy Green Machine and the current Greenery container farms ”

His team designed their own camera to overcome these issues. Along with demanding performance requirements, these must be small enough to fit inside the panels of the farm; Raspberry Pi provides the small form factor that Freight Farms requires in the computers to which the cameras connect, along with the necessary compatibility with these bespoke, high-quality devices.



“The cameras take us to that next level of the farm really being foolproof,” adds Holmes, while the app helps the crop grower to notice problems promptly. It gives them access to high-quality photographs in which symptoms such as yellowing foliage are clear to observe, so they can top up the container farm’s nutrient supply, or adjust environmental factors such as temperature or humidity.

WHY RASPBERRY PI?

By the time Freight Farms was successfully crowdfunding, Raspberry Pi was becoming a success story in its own right. With a great reputation among the engineers and computer technicians tasked with creating Freight Farms’ ‘turnkey’ growing containers, it was an obvious choice of hardware.

Jake Felser outlines the reasons for choosing Raspberry Pi: “ease of development, open-source traceability, and driver availability at an unbeatable cost for the performance.” This was coupled with a stable supply chain and a large support base. “We also looked at some microcontroller solutions as well as other SBCs, but nothing came close to the performance at the given cost,” he concludes.

Meaghan Holmes maintains a close relationship with Freight Farms’ engineering and manufacturing team, ordering critical items such as Raspberry Pi units, sensors, and lights to measure and monitor plant growth and health. She praises Raspberry Pi’s reliability, its user-friendliness, and the reassurance of it being so widely known – a platform with which almost all of the company’s engineers were already familiar. For a business-critical component, “you wouldn’t want something super-obscure”, she points out,





©Freight Farms



while the engineering team knew that Raspberry Pi has “a really high measure of success in terms of support.”

THE RESULTS

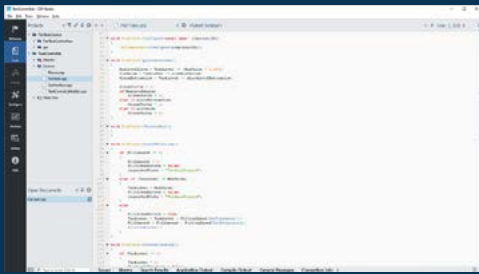
Freight Farms’ Raspberry Pi-powered Greenery S container farms offer the most productive environments in their field, with a 30% improvement in yield between the original Leafy Green Machine and the current Greenery container farms. A thorough onboarding programme in which potential sites are checked for appropriate electrical voltage, robust plumbing, and water access kick-starts each new customer relationship.

The straightforward management of each farm, with ‘recipes’ for growing each crop, plus

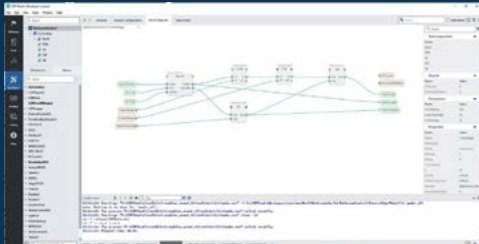
a dedicated support team available by email or phone, has meant the Freight Farm concept has been embraced by non-farmers who perhaps speculatively invest in a single container farm, as well as leisure resorts where agricultural expertise is in short supply but where the restaurants pride themselves on offering fresh locally grown produce. They are particularly proud of working with community kitchens, food banks, and non-profit organisations, such as Lotus House in Miami which supports women and children who have experienced homelessness.

Core crops grown in the container farms include varieties of lettuce, leafy greens such as kale and chard, herbs, beans, and root vegetables. Some customers have even begun experimenting with berries and edible flowers.

There are now more than 200 Freight Farms running productively in over 50 countries. “The Covid-19 pandemic brought into sharp relief, for some remote islands, how dependent they were on imports, not just for discretionary goods and services, but for food access,” says Meaghan Holmes. Freight Farms’ customers include a large number of island nations that were previously dependent on regular shipments to supply food staples that could not be grown locally. [M](#)



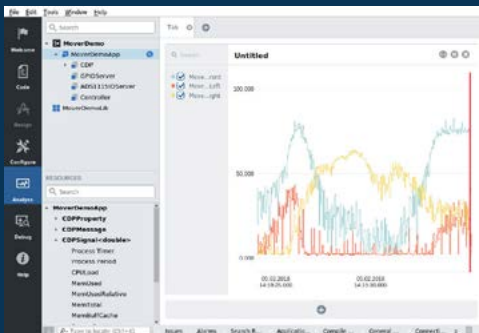
Full-Code



No-Code / Configure



Design



Analyze

now with a
No-Code WEB UI Designer

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

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

cdpstudio.com

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

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



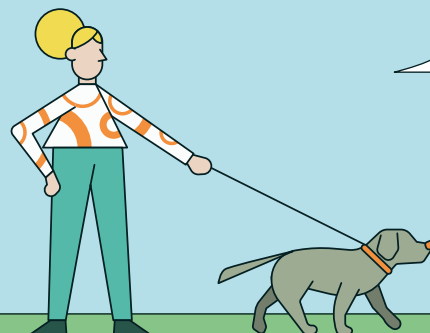
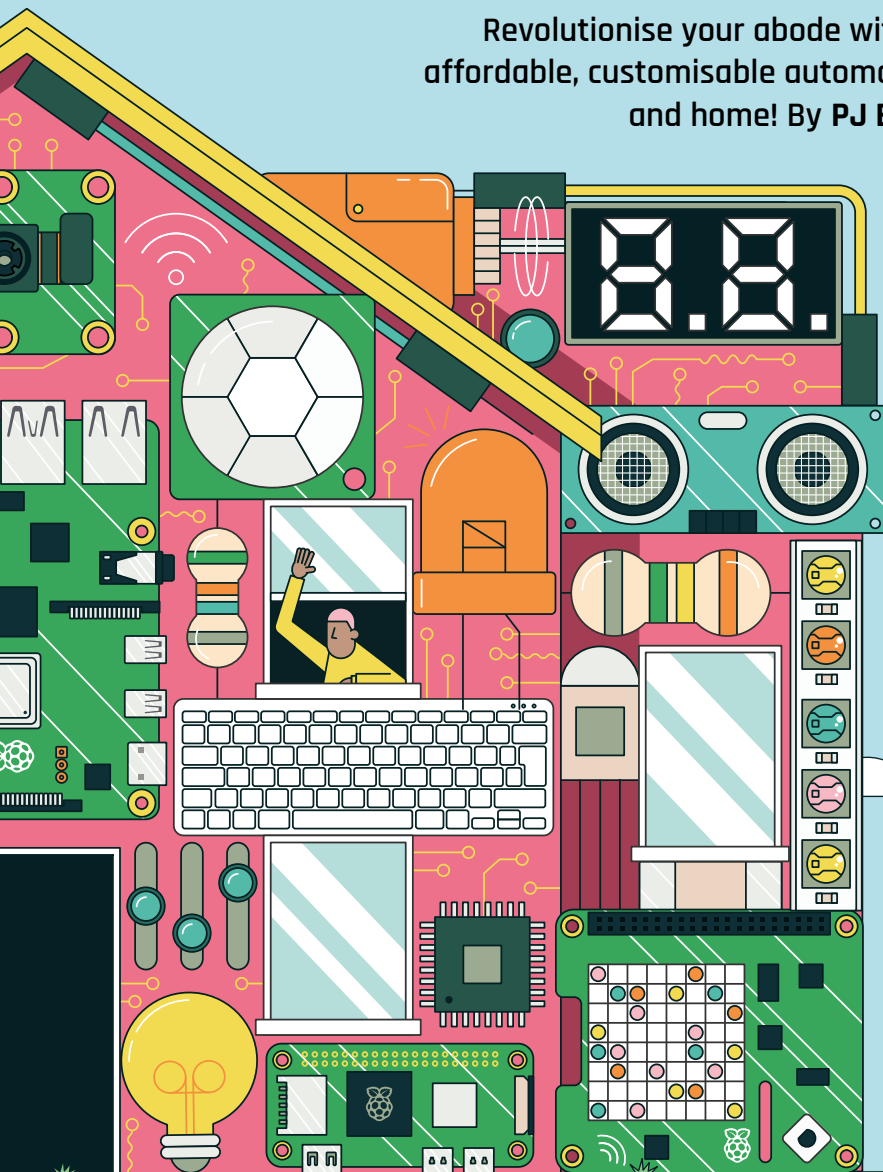
DIY HOME AUTOMATION WITH RASPBERRY PI

Revolutionise your abode with Raspberry Pi:
affordable, customisable automation for your house
and home! By PJ Evans

Welcome to the marvellous world of Raspberry Pi-powered home automation, where your humble house transforms into a technological utopia!

Get ready to shake hands with Home Assistant, your faithful digital butler, and explore cool automation ideas that'll make the Jetsons green with envy. Prepare to dive into the MQTT realm, where efficiency reigns supreme and communication is king. Finally, create your very own smart garage door, an ingenious guardian that'll open and close at your beck and call.

So, strap in and let's high-tech your home!



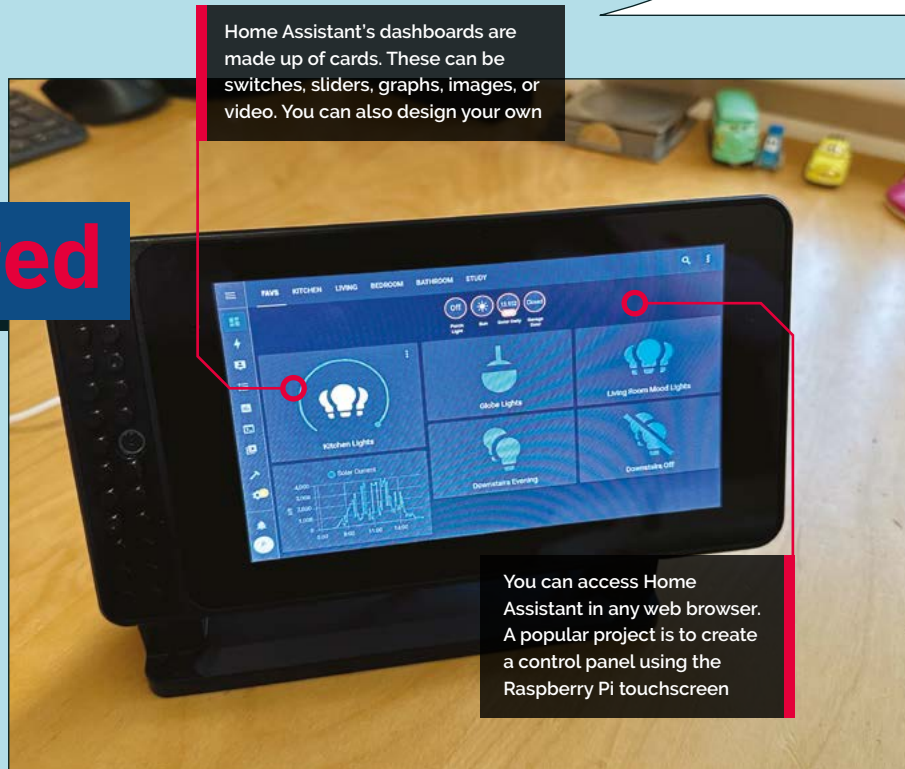
Getting Started with Home Assistant

Take control of your home, and your privacy, with this amazing home automation tool for Raspberry Pi

Home automation is not only useful but can also be a great deal of fun, setting up cool automations or connecting different devices together in new ways. It also can help with energy efficiency and security. There are a wealth of practical reasons to start experimenting with this technology. Although there are different vendor-specific automation systems out there, we prefer one that doesn't 'lock' you into one provider. One such platform is Home Assistant (home-assistant.io), a free open-source operating system designed with flexibility and independence in mind. Home Assistant is a huge topic, but here we'll look at the basics of setting up a server to get you started on your automation journey.

01 Prepare your Raspberry Pi

Although Home Assistant (HA) isn't strictly an operating system in its own right, it is available as a Raspberry Pi image that significantly reduces the work a user has to do to get up and running. HA is intended to run on a Raspberry Pi as the sole service. It is possible to run HA alongside other apps and services, but we're keeping to the true path here. HA works with Raspberry Pi 3, but we strongly recommend using a Raspberry Pi 4 for the best performance. You should also use a wired Ethernet connection for setup and to ensure reliability. HA is headless, so no monitor or keyboard is needed.



Home Assistant's dashboards are made up of cards. These can be switches, sliders, graphs, images, or video. You can also design your own

You can access Home Assistant in any web browser. A popular project is to create a control panel using the Raspberry Pi touchscreen

02 Write the Home Assistant image

Luckily for us, you can write the latest stable Home Assistant image directly from the Raspberry Pi Imager (magpi.cc/imager). Using a fast SD card 32GB or more in size, insert it into your computer. In Imager, select Choose OS > Other specific-purpose OS > Home assistants and home automation > Home Assistant > Home Assistant OS 9.5 (RPI 4/400 or RPI 3, as needed).

You'll now get a ready-to-boot image. Insert the card into your Raspberry Pi, make sure you've got a wired network connection and power up. After a few minutes, try to connect to <http://homeassistant:8123> in your web browser.

03 Initial setup

Time to grab yourself a favourite beverage. You'll see an initial setup screen stating that it will take about 20 minutes before you can proceed. Soon it will be automatically replaced with the first stage of setup. Provide your name, username, and choice of password. On the next screen, there will be some questions about the server's location. It's important to set this accurately if you want to take advantage of sun-up/down times. Finally, HA will 'look' around your network for any existing smart devices and let you know what it's found. Don't



MAKER PJ Evans

PJ is a writer, software engineer, and home automation enthusiast. He has no idea why his lights keep changing colour at 3:15 pm every day, but it's probably his fault.

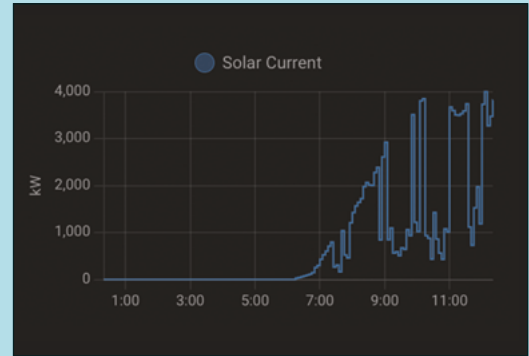
mrpjeans.com

You'll Need

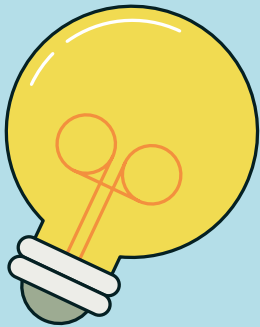
- ▶ Wired internet connection
- ▶ Good-quality case (recommended) magpi.cc/argonone
- ▶ At least one compatible smart device (recommended) home-assistant.io/integrations



▲ Replace your porch light with a smart light bulb and you can trigger it at sundown all year round



▲ Combine devices to create clever energy-saving automation. Got solar panels? When it's sunny, switch the washing machine and all the lights on



Top Tip

Beginning automations

Home Assistant knows your local sunrise and sunset times. A great first automation project is switching a porch light on and off at the right times.

- ▶ As your confidence and knowledge grow, you can create more complex dashboards. You can even incorporate video feeds

worry if something doesn't appear, it will probably just need manual configuration later on.

04 Add integrations

Home Assistant refers to smart platforms as 'integrations'. For instance, if you have Philips Hue or IKEA Trådfri smart lights, it will add an integration for them and then a 'device' for each light it finds. It will also create a default dashboard for you based on what's been found. Not all integrations can be found automatically, so you can browse available integrations and add them yourself or install third-party add-



ons. Integrations include lights, security devices, media centres, printers, and mains power controllers and you can even make your own. At this point, it's best to start exploring.

05 Customisation

One of Home Assistant's greatest strengths is customisation. The Dashboard system ('Lovelace') allows you to arrange your device control and set automations just how you like. By default, HA automates the layout, but we recommend you disable that. Click the three dots in the top-right of the screen followed by 'Edit dashboard'. You'll be asked if you want to take control of layout. Do so and now you can create your perfect layout. You can resize, restyle, add graphs, tabs, and badges. Don't be intimidated; start small and build things up as you become familiar.

06 Next steps

Congratulations, you now have a running Home Assistant server. The capabilities of this service can seem overwhelming at times but with a little reading on home-assistant.io and some digging around the menus, you'll soon be taking control of your home. Once you've added the ability to switch devices on and off, or monitor things like printer ink levels, move on to Automations. These allow certain actions to happen based on events. For example, you can have a motion sensor turn on certain lights around the house. Check out Settings > Automations and have a play.

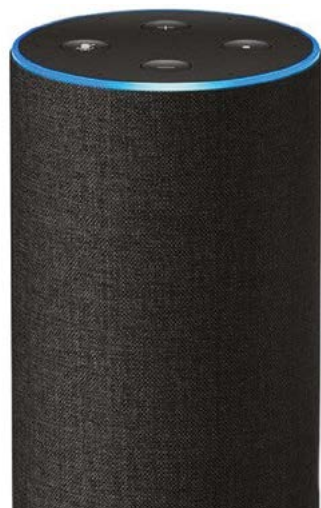
Cool Home Automation Ideas

Looking for inspiration for your home automation project? Here are some incredible ideas!

Home automation has such a broad scope that it is only limited by your imagination. Most people start by controlling lights, but soon you can be monitoring all kinds of things, responding to different inputs, creating scenes and more. So here are some of our favourite types of smart devices and suggestions on what you can do with them. Everything here will work with Home Assistant.

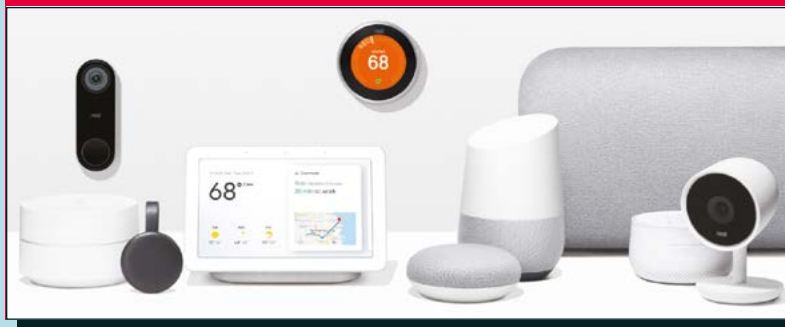
Voice control

There are many different voice control platforms available. The most popular are Amazon's Alexa, Google Assistant, and Apple's Siri. Home Assistant has direct support for Alexa and Google Assistant via integrations so you can control your devices or trigger automations by voice. For Siri, you create a link to HomeKit and create voice-activated short cuts.



Home environment

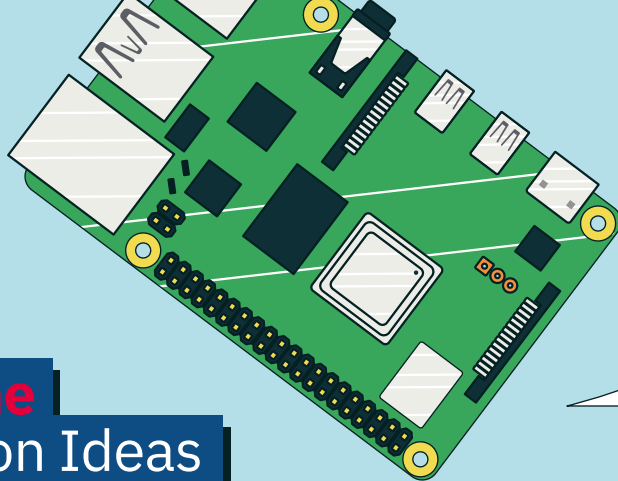
Our homes are filling up with smart devices. We can control temperature, monitor for smoke, use motion-activated cameras and more. Home Assistant adds value by allowing these devices to be linked to others in a way that may not be supported out of the box. Integrations are available for popular platforms Ring and Nest.



Smart lights

A starting point for most home automation newbies. Home Assistant has support for a wide range of lights, notably the Philips Hue and IKEA Trådfri range. Once you have a gateway hub in place, Home Assistant will detect it automatically. You also get fine-grained control of colours, brightness, and colour temperature.





Cool Home Automation Ideas

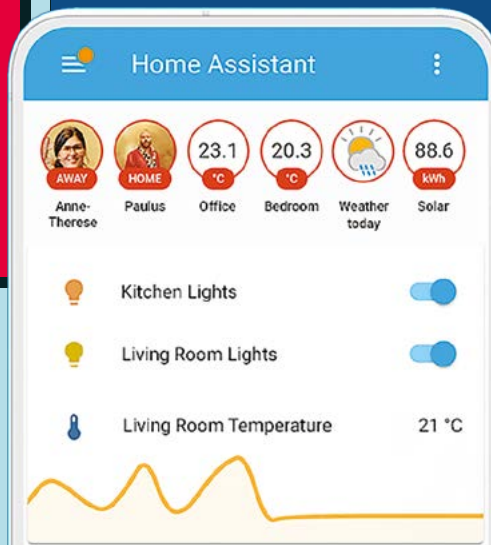


Raspberry Pi

Connect sensors, switches, displays, or simply LEDs, then use MQTT to send and receive status messages. Home Assistant's MQTT plug-in makes adding Raspberry Pi quick and easy. The ultimate flexible smart device. Check out the Remote GPIO integration for codeless monitoring of inputs and output (magpi.cc/homeassistantgpio).

Your phone

Installing the official Home Assistant app unlocks several cool features. If you wish (it's a privacy-first model), your app can update your Home Assistant server with your location and also (depending on the phone) other metrics such as steps, messages received, and so on. Location can be used to trigger events, such as switching the lights on when you're near home.



“Installing the official Home Assistant app unlocks several cool features”

Cars

Cars are getting cleverer too. Many newer models feature mobile communication allowing us to access information by app and sometimes be able to start the car or turn on air conditioning remotely. Many manufacturers are supported by Home Assistant, so you can see how your EV charge is getting on direct from your HA dashboard.





Introducing MQTT

Using a message broker system makes building your home automation system easier and more flexible. Here's how it works

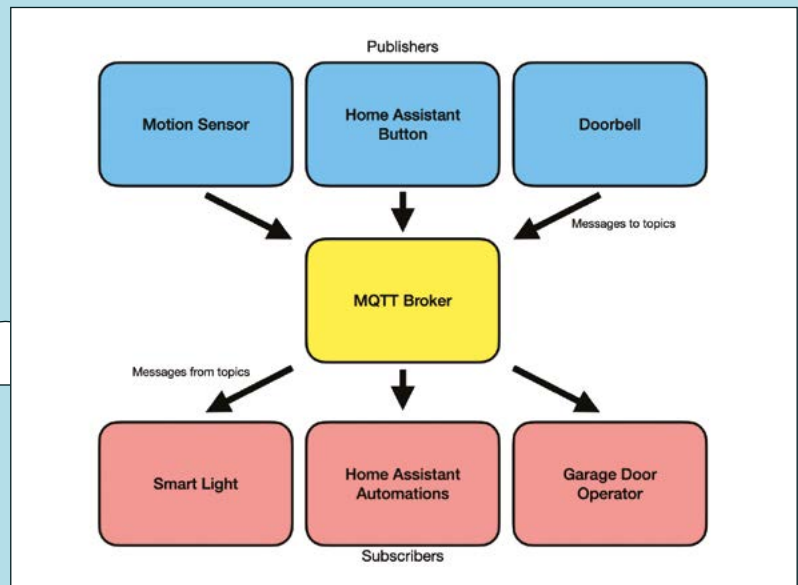
Once you've begun your home automation journey, it's only a matter of time before you come across MQTT. It may at first seem a little confusing and you may find yourself wondering why you need it. MQTT is a simple, lightweight messaging protocol whose purpose is to break the direct links between devices. We call this 'loose coupling'. Why is this a good idea? Let's say you have a temperature sensor. By sending its data to the MQTT server instead of, say, a display, this means we can have any number of devices 'listening' to the sensor and taking action. The sensor's code doesn't need to change if new requirements come along. Devices that send information are 'publishers' and those listening are 'subscribers'. Publishers don't care where the information is going and subscribers don't care where it is coming from.

01 Setting up an MQTT server

To achieve the loose-coupled model we need a central hub that acts as a kind of post office for messages. This is the MQTT server or more commonly 'broker'. Luckily for Raspberry Pi users, there is a simple lightweight broker available called 'mosquitto'. If you are setting up a dedicated MQTT server, installation is as simple as running these commands from the terminal:

```
sudo apt update && sudo apt upgrade
sudo apt install mosquitto mosquitto-clients
```

If you want to install an MQTT broker on a Home Assistant installation, it can be installed as an add-on in Settings > Add-ons > Add-on store.



02 Testing the MQTT server

Clients of an MQTT broker are either publishers (sending data) or subscribers (receiving data). The data streams are known as 'topics' and a broker can have any number. This way subscribers can pick just the information they are interested in. To try it out, make sure you have the mosquitto-clients package installed on your local machine and open two terminal windows. In one, run:

```
mosquitto_sub -h <IP address of MQTT broker>
-t "test"
```

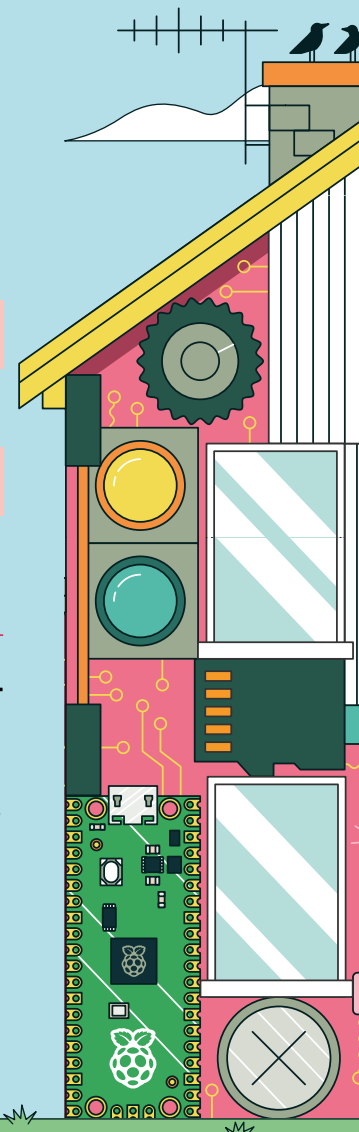
In the second, run:

```
mosquitto_pub -h <IP address of MQTT broker>
-t "test" -m "Hello subscribers"
```

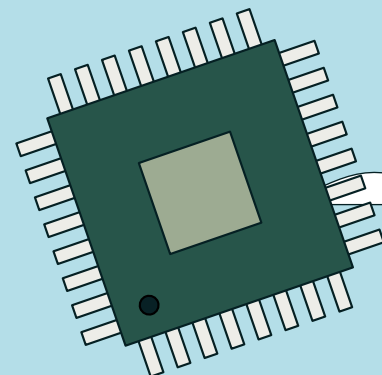
Have a look at the first window, you should see that the subscriber has received the message.

03 Add Home Assistant as a subscriber

To bring the power of MQTT to Home Assistant, go to Settings > Devices & Services > Add Integration and then search for MQTT. Provide the IP address of your server and you're done. To create automations based on data coming into topics, go to Settings > Automations and create a new automation. Set the trigger to MQTT, add the topic name you're interested in and, optionally, the data value you want (e.g. 'true'). You can then trigger any device supported by Home Assistant. For example, a motion sensor can send a message to the 'kitchen/motion' topic and the Home Assistant can trigger an automation that turns the lights on.



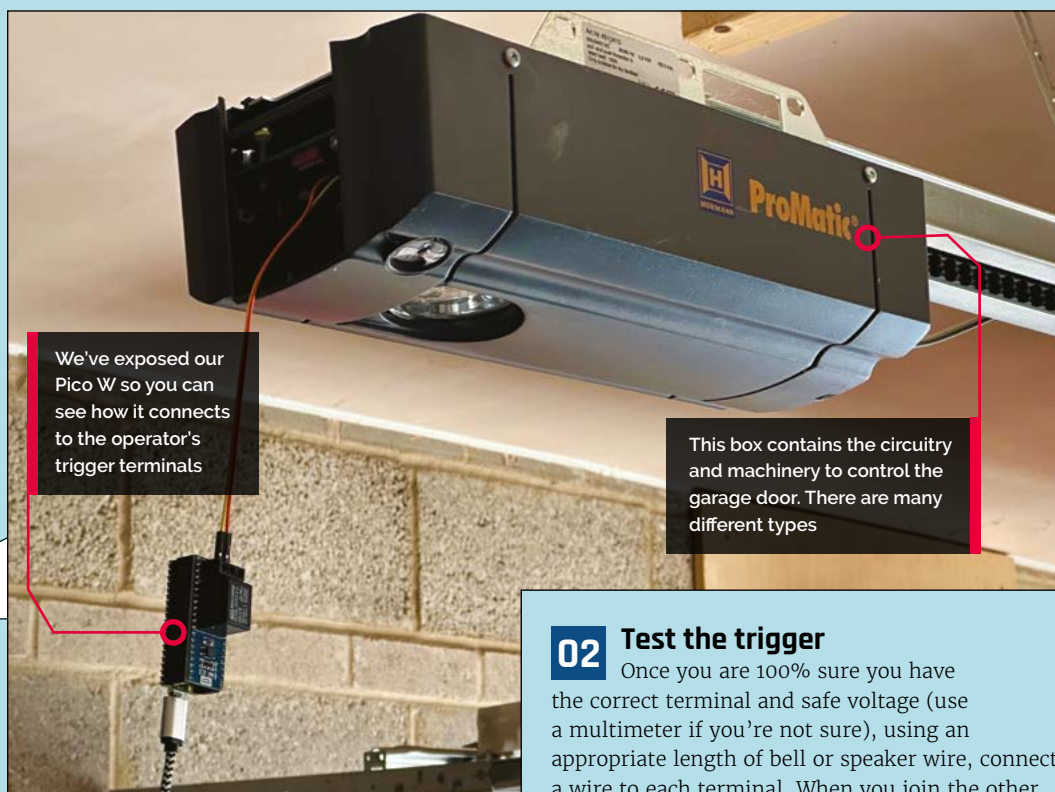
Make a smart garage door



Is your garage door not pulling its weight? Add some smarts with this home automation tutorial and take control

You'll Need

- ▶ Raspberry Pi Pico W
- ▶ SB Electronics Relay for Pico magpi.cc/relaypico
- ▶ Bell wire (length depends on your installation)
- ▶ Motorised garage door
- ▶ Home Assistant server
- ▶ MQTT broker



We've exposed our Pico W so you can see how it connects to the operator's trigger terminals

This box contains the circuitry and machinery to control the garage door. There are many different types

Top Tip

Add a sensor

Use a simple contact (or reed) switch, like those used in alarm systems, to detect whether the door is open or closed. Then you can send specific 'open' or 'close' instructions rather than a toggle.

01 Make a (safe) plan

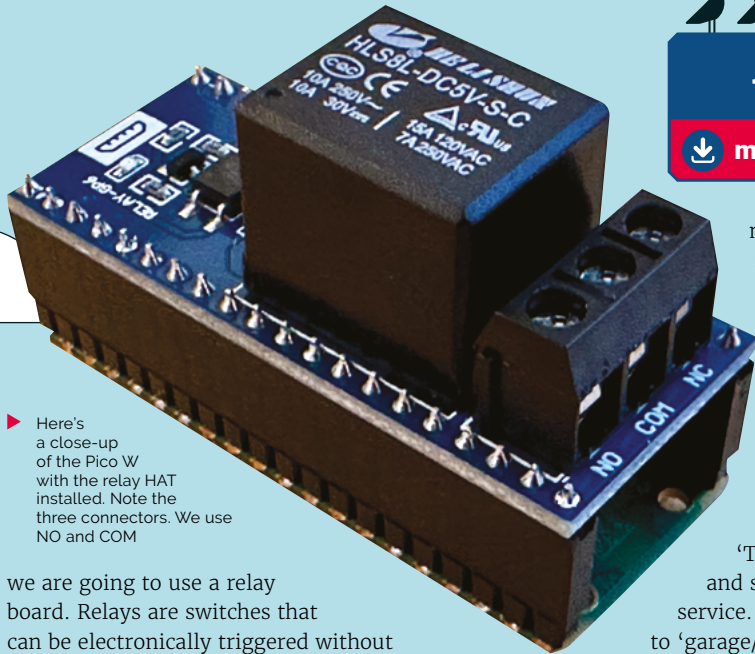
This project is aimed at garage doors with an operator, a motor that opens and closes the door via a remote control. Most operators have their control box near the ceiling in the centre of the garage. Typically, there's a light embedded in there too. Nearly all models have two terminals inside the box for a button to be connected. Research your model of operator and check the manuals for such terminals. Normally they are zero-voltage and safe to use. Most operators run at 24V but check, check, and check again you are not going to interface with anything above that.

02 Test the trigger

Once you are 100% sure you have the correct terminal and safe voltage (use a multimeter if you're not sure), using an appropriate length of bell or speaker wire, connect a wire to each terminal. When you join the other end together, the door should close or open. Make sure you have your remote on you just in case! Finally, think about where you are going to place your Pico W and ensure you have enough of the bell wire to make a connection between it and the garage operator.

03 Prepare your Pico W

Even if the connection terminal is zero-voltage, we still don't want a direct link between the operator and your Pico W, as interference and other factors could cause problems. So,



► Here's a close-up of the Pico W with the relay HAT installed. Note the three connectors. We use NO and COM

we are going to use a relay board. Relays are switches that can be electronically triggered without connecting directly to the thing being switched. We've chosen an SB Electronics relay HAT designed for Pico that can handle a wide range of voltages. The HAT sits underneath Pico W and needs headers, so is compatible with the WH variant or you'll need to add some yourself. Add the HAT, using its USB connector image as a guide.

04 Add the code

Now add the code in **main.py** (from the GitHub repo) using your preferred method, being careful to update the variables at the beginning with the correct values for your setup. You also need to add the `umqtt` library, which you can download from magpi.cc/umqttsimplepy. The code connects to your Wi-Fi network and then to your MQTT server. It's a subscriber, so it will sit and listen for messages for the given topic. Whenever a new message is delivered to the topic, the Pico W will momentarily activate the relay. If you have Mosquitto Utilities installed, you can use this command to test the setup:

```
moquitto_pub -h <IP address of MQTT broker>
-t 'garage/dooroperator' -m 0
```

You should hear the relay click as it closes and opens again.

05 Add to Home Assistant

Now we can add an automation script to Home Assistant that will send the MQTT

DOWNLOAD THE FULL CODE:

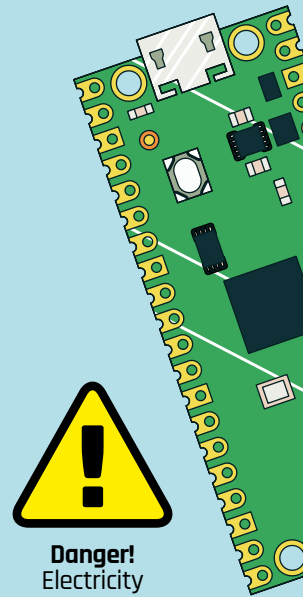
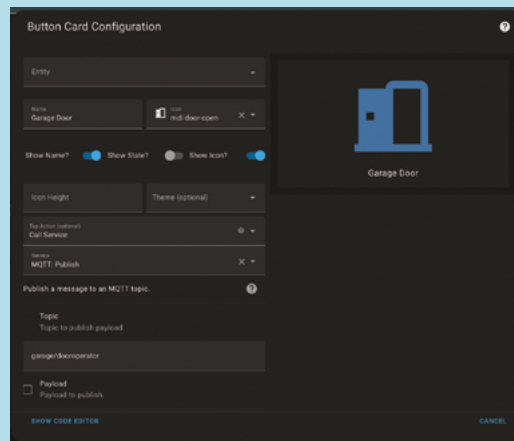
magpi.cc/garagedoorbotgit

message when a button is pressed. On the Home Assistant dashboard, click the three dots in the top-right corner, then 'Edit Dashboard'. Now 'Add Card' and select 'Button'. Click 'X' to remove the default entity chosen. Set the name to anything you like and choose an icon. Set 'Tap Action' to 'Call Service' and select 'MQTT:Publish' as the service. When prompted, set the topic to 'garage/dooroperator' then click 'Save'. If all is well, your Pico W should operate its relay whenever you click the button.

06 Install

Once you're happy the relay is working, you can connect it up to the operator. Cut the wire to the length you need and connect the wires as before to the two terminals. You'll notice the relay has three connectors. The middle one is the ground (COM), so connect that to the ground (black) terminal on the operator. Connect the other wire to the NO side of the relay, not NC. These stand for 'Normally Open' and 'Normally Closed'. We want the connection to be 'open' until it is triggered. Make sure your Pico W is safely away from the elements and dust. **W**

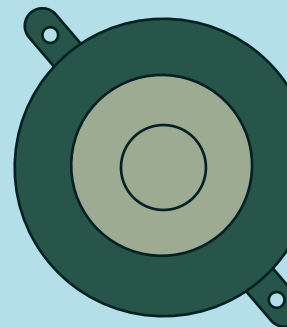
▼ Here's how the setup in Home Assistant should look. The button calls the MQTT service and 'pings' the topic which is sent to the Pico W



Danger!
Electricity

All garage door operators are different. Ensure you only connect to safe or zero-voltage electrical terminals.

magpi.cc/electricalsafety



Top Tip

Always be there!

Never open or close a garage without being able to see it. You might give someone a nasty surprise or inadvertently trap a pet or wildlife.

Part 01

Build your own GPT Chatbot



Use Python and OpenAI's API to talk to a GPT-3.5 powered chatbot on your Raspberry Pi

MAKER

K.G. Orphanides

KG is a writer, developer and software preservationist. They'd like to see fewer chatbot prompts about the meaning of like, and more about adventure on the high seas.

magpi.cc/owlbear

You'll Need

- ▶ An OpenAI account openai.com
- ▶ Thonny
- ▶ An internet connection

▼ Remember to note down the API keys you generate. The encrypted notes field of a password manager is a good place

In this tutorial, we'll be creating a simple chatbot interface in Python using OpenAI's GPT-3.5 Large Language Model (LLM) to create a local interface and instructions (a 'prompt') for a chatbot that behaves similarly to OpenAI's famous ChatGPT.

You'll need an OpenAI account to use the gpt-3.5-turbo model required by this tutorial. You'll also need to keep Raspberry Pi connected to the internet as the models we'll be using have to be accessed via an API that communicates with OpenAI's servers, both because of their proprietary nature and to give Raspberry Pi the extra processing power necessary to run a viable GPT-3 chatbot.

01 Create an OpenAI account

Go to platform.openai.com/signup and follow the instructions to create an account. Users must be over 18. You don't need to enter an organisation name, but if you do, your account will be named after it. New users currently get \$5 in free credits. If you use the same number to verify more than one account, subsequent accounts after the first won't receive free credits. You'll need credits to use any of OpenAI's models,

whether you access them via the web or the API. API calls are very cheap, but free credits expire after three months.

02 Keepers of the key

Next, we'll generate an API key so you can call OpenAI's LLM models in your own code. Go to platform.openai.com/account/api-keys, then click the 'Create new secret key' button. After a couple of seconds, a pop-up will appear with the title API key generated. This is your first and last opportunity to copy the key that's shown in the pop-up's selectable text box. Copy it, and save it somewhere safe. We'll be using this later. If you do lose your key, there's no way of recovering it, but you can always generate a new one.

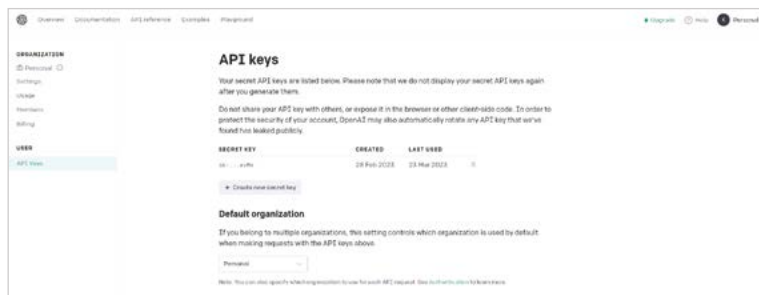
03 First steps

Open a terminal and type:

```
pip install openai
git clone https://codeberg.org/MightyOwlbear/simple-chatbot-gpt-3.5-turbo
```

This will clone our sample code and install OpenAI's Python library and our sample scripts. If you've previously installed the openai package, put `--upgrade` at the end of the `pip install` command to update it.

The Thonny Python IDE comes pre-installed in Raspberry Pi OS. If it launches in Simple mode, click the link at the top right of the toolbar that reads 'Switch to regular mode' and restart the program. Your interactions with the chatbot will take place in Thonny's Shell, which is a major



convenience when tweaking your scripts. You can run them from the terminal, if you prefer.

04 Meeting the API

While OpenAI's own API reference (magpi.cc/openaiintro) and cookbook (magpi.cc/openaicookbook) provide useful code snippets and explanations, the extensive annotation makes the code a little hard to parse.

From the git repository you cloned in the previous step, open `openai-sample-code.py` in Thonny and paste your own OpenAI API key between the quotes in the `openai.api_key =` line.

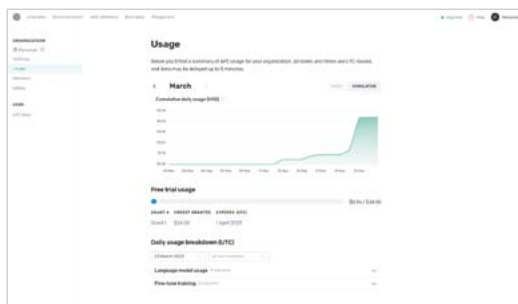
This sample code is based on OpenAI's Chat completion docs. In it, we import the OpenAI library, define the model we wish to use (gpt-3.5-turbo), and use the ChatCompletion routine to send a prompt and print the response it returns.

05 Addressing the model

We've provided a 'user message' – a prompt for the LLM to respond to. This can be anything you like: we asked the AI to pretend to be a sentient Raspberry Pi

You can also write chatbot personality prompts as a 'system message', prefixed by `"role": "system", "content":`. OpenAI warns that its gpt-3.5-turbo model doesn't treat system messages as strong prompts, but we've generally found them functional and will be using one in the next example.

We've also set the model's 'temperature' to 0.2. The higher the temperature, defined on a scale between 0 and 1, the more random the model's responses will be. Finally, we print the output of 'response', which extracts the reply sent via the API.



▲ Detailed usage stats allow you to see how much credit or money you're using. Paying users can set spending limits

The messages parameter for ChatCompletions in OpenAI's gpt-3.5-turbo API (magpi.cc/opeaichat) allows you to send prompts – questions or instructions – to the remote AI model

```
File Edit View Run Tools Help
simple_chatbot.py
10 # your chatbot's name
11 botname = "Captain"
12
13 # The initial prompt. Tell the chatbot how you want it to behave.
14 messages=[
15     [{"role": "system", "content": "Present yourself as a captain from the golden age of piracy in all subsequent responses"}]
16 ]
17
18 # we want to run some commands once when we start the program, so we'll use this flag
19 first_run = True
20
21 # our chat routine
22 while True:
23     # Show the user instructions and introduce the chatbot by name.
24     if first_run == True:
25         print("Type exit to leave the chat. Initialising chatbot: Captain.")
26         print("You: Hello!")
27         print("Captain: Aye there, me hearty! I be Captain Blackbeard, scourge of the seven seas and terror of the Caribbean. What brings ye to me ship?")
28         print("You: Tell me about the different jobs on a pirate ship.")
29         print("Captain: Aye, there be many jobs on a pirate ship, each one as important as the next. First and foremost, there be the helmsman, who steers the ship and keeps us on course. Then there be the navigator, who uses the stars and the sun to plot our course. The quartermaster be in charge of the ship's supplies, making sure we have enough food, water, and ammunition to last us on our voyages.")
30         print("The gunner be responsible for the cannons and making sure they be loaded and ready for battle. The carpenter be in charge of keeping the ship in good repair, fixing any damage that be done in battle or from the elements. The cook be in charge of feeding the crew, and let me tell ye, a good cook be worth his weight in gold!")
31         print("Last but not least, there be the crew, who work together to keep the ship running smoothly and to fight off any enemies that come our way. Each job be important, and we all work together as a team to keep our ship afloat and our treasure safe.")
32         print("You: ")
33         print("Local Python 3 - /usr/bin/python3")
```

Because we're appending each question and response to our messages array, our chatbot remembers the instruction we gave it to behave like a pirate from one question to the next

06 A two-way conversation

If we want to have an actual conversation, we'll need to expand on this a little. Open

`simple-chatbot.py` and you paste your API key in.

If you run this script, you'll have a conversation with a captain from the golden age of piracy. We're using Python's `input()` function to store anything the human participant in the conversation types as a string and format it into an appropriate prompt, which goes into an array called `messages`. The API sends this back and returns a similarly structured response, which we then print. Note that responses can take a while if OpenAI's systems are experiencing high demand.

07 Remember me

If you want to have an ongoing conversation with a chatbot, you'll have to give it a memory. In `simple_chatbot.py`, we're doing this with Python's `append` function. Every time the user enters a question, it's stored in the `user_input` variable, formatted as a compliant message object, and appended onto the end of the `messages` array. The LLM's responses are also added to the array. This means that every subsequent question can be answered in the context of the prior conversation and long-term instructions can be followed. However, there are limits to this.

Top Tip

Mutually incompatible

The gpt-3.5-turbo and text-davinci-003 APIs don't use compatible syntax, so you'll need to rewrite your bot scripts if you want to switch from one to the other.

08 Maximum word count

OpenAI breaks text input into ‘tokens’ (magpi.cc/counttokens) – chunks of around four characters. Different models have different token limits: gpt-3.5-turbo can handle a maximum of 4096 tokens, or about 3000 words in English.

Anything longer and you’ll get an error asking you to reduce the length of your messages. OpenAI also charges by the token, both sent and received, but these rates are inexpensive: gpt-3.5-turbo works out at one US cent for every 375,000 words.

In the future, we’ll get longer conversations and text prompts: The new GPT-4 models, currently in beta (magpi.cc/gpt4model), have extended limits of between 8192 and 32,768 tokens.

09 The art of prompting

A lot of the trick to getting the desired behaviour out of an LLM chatbot comes down to how you word the prompts. For example, to get our bot to consistently write as though it’s a pirate

“ OpenAI breaks text input into ‘tokens’ ”

captain, we tell it to “Present yourself as a captain from the golden age of piracy in all subsequent responses. Use colourful language and talk like a pirate.” That “in all subsequent responses” bit is important, otherwise it’ll go back to its default AI chatbot personality after its first response.

10 Politeness and conversation structure

There are a few more things you can do to make your chatbot seem more approachable. Our repository includes variations on the script that have the bot introduce itself at the beginning of the conversation and which listen for keywords indicating that the conversation has ended. In **simple-chatbot-with-intro.py**, we request a response from the model on first run, as an immediate response to our initial prompt, and in **simple-chatbot-with-intro-and-outro.py**, we look for specific keyword combinations so that we can automatically end the chat after the bot has said goodbye in response to the user.

Top Tip

Out in the open

OpenAI is aggressively proprietary, but for an open-source alternative to its LLMs, check out GPT-J-6B (magpi.cc/gptj6b).

11 Get on the web

Talking to your bot through Thonny’s Shell is functional, but not pretty. For a more attractive approach, Rohit Prasan Mandal’s example uses Gradio to create and host a browser-based chatbot. Open a terminal and type:

```
pip install gradio
git clone https://github.com/xiaowuc2/ChatGPT-Python-Applications.git
cd ChatGPT-Python-Applications/chatbot
```

Edit **fantastic-chatbot-gradio.py** and add the line `import yaml` to the top of the script. In the same directory, open **pass.yml** and paste your OpenAI API key into the indicated space.

If your pre-installed version of NumPy is below 1.20, preventing the script from running, open a terminal and type:

```
sudo apt update && sudo apt upgrade
apt remove python3-numpy
sudo apt install libatlas3-dev
pip install numpy
```


Run the script and click the link in Thonny’s Shell to open a browser-based chatbot interface.

12 Black holes and confabulations

LLMs are exposed to vast amounts of text and trained by having them select the most likely next word in a sequence until they can accurately predict the kind of word structures used in meaningful, human-readable text.

They’re designed to be good at text-generation, with gpt-3.5-turbo in particular trained for conversation. It can produce lucid statements in correct English, but has no real ability to analyse the content or accuracy of its statements.

If you give ChatGPT a passage of original text and ask the bot to summarise it, it might cherry-pick some phrases from the passage but fail to register its title or the gender of the protagonist.

The models also have no access to new information. They can’t go and look something up online and their knowledge is frozen at the time of their data set’s creation – September 2021 in the case of GPT-3.5 and GPT-4. If you ask about later events, your bot has no alternative but to confabulate a fiction. This also happens if you ask about something that’s insufficiently well represented in its training data. 

simple-chatbot.py

DOWNLOAD
THE FULL CODE:

> Language: **Python**



magpi.cc/simplechatbotcode

```
001. # import the OpenAI Python library for calling the OpenAI API
002. import openai
003.
004. # Your openAI API key goes here:
005. openai.api_key = "PasteYourKeyHere"
006.
007. # the model we'll be using
008. MODEL = "gpt-3.5-turbo"
009.
010. # your chatbot's name
011. botname = "Captain"
012.
013. # The initial prompt. Tell the chatbot how you want it to behave.
014. messages=[
015.     {"role": "system", "content": "Present yourself as a captain from the golden
    age of piracy in all subsequent responses. Use colourful language and talk like a
    pirate."}
016. ]
017.
018. # we want to run some commands once when we start the program, so we'll use this
    flag
019. first_run = True
020.
021. # our chat routine
022. while True:
023.
024.     # Show the user instructions and introduce the chatbot by name.
025.     if first_run == True:
026.         print("Type exit to leave the chat. Initialising chatbot: "+str(botname)+".")
027.         first_run = False
028.
029.     user_input = input("You: ")
030.     # exit on command
031.     if user_input == 'exit':
032.         exit()
033.
034.     prompt = {"role": "user", "content": user_input}
035.     messages.append(prompt)
036.
037.     # Example OpenAI Python library request - defines the model we wish to use, and
    uses the ChatCompletion routine to send a prompt.
038.     response = openai.ChatCompletion.create(model=MODEL, temperature=0,
    messages=messages)
039.
040.     # extract and print the response
041.     print(str(botname)+' : '+str(response['choices'][0]['message']['content'])+'\n')
042.     messages.append(response['choices'][0]['message'])
```

Raspberry Pi Pico inputs and outputs



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

You'll Need

- 6V buzzer
magpi.cc/buzzer6v
- Arcade buttons with 5V LED
magpi.cc/arcadebuttons
- 2N7000 MOSFET
magpi.cc/2n7000
- 470 Ω and 10 kΩ resistors
magpi.cc/575resistorkit
- PCB switch
magpi.cc/tactileswitchbuttons
- ULN2803 integrated circuit
magpi.cc/uln2803
- Various wires, including crimp connectors

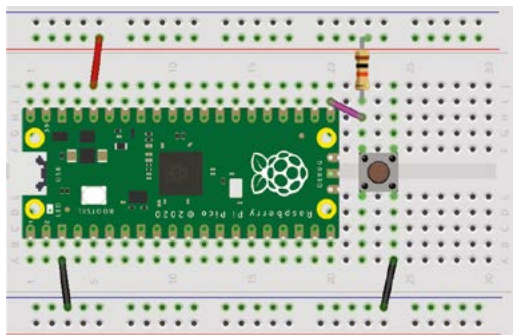
Create a buzzer game whilst learning about how to use inputs and outputs on Raspberry Pi Pico

Inputs and outputs are the ways that a Raspberry Pi Pico can interact with the outside world. In this tutorial, you will learn about pull-up and pull-down inputs and how to connect arcade buttons to a Raspberry Pi Pico. You will also learn about driving a buzzer using a MOSFET transistor.

You will then learn how you can swap the MOSFET for an integrated circuit to create an interactive game to play with friends.

01 Switches as inputs

The digital GPIO pins on Pico can have two different values. One is a high input where the input is at, or near, 3.3V. The other is a low value when the input is at, or near, 0V. What about if the pin isn't connected to anything? You may think that without any input then it would be low, but that isn't necessarily the case. Instead, the input is treated as 'floating'. With a floating input, it can pick up stray signals from passing radio signals or static electricity resulting



▲ **Figure 2** Pull-up example circuit showing a pull-up resistor and switch connecting to ground

in it sometimes being high and sometimes low. To avoid this, we need to use pull-up or pull-down resistors.

02 Pull-up and pull-down resistors

Pull-up and pull-down resistors are a way to set a default value on the input pin. If it's pull-up, that means that with no other input the value will be high; with pull-down, it means that the default value will be low.

To change the value, you need to apply a stronger input that overrides the pull-up or pull-down resistor. For a pull-up, you provide a connection to ground; for pull-down, you provide a connection to 3.3V.

For example, connect a 10kΩ resistor between a GPIO pin and the 3.3V source on pin 36. This sets the input high. Connect a button switch from the same pin down to ground. Whenever you press the button, the lower resistance in the switch results in the voltage being dropped across the resistor and no voltage across the button providing a low input. This is shown in the **Figure 2** wiring diagram.

03 Pull-up and pull-down inside Pico

For most circumstances, you don't need to use external resistors and you can instead use internal resistors within the Pico. This is achieved by using the pull argument when you define the pin.

To enable a pull-up resistor, use:

```
button1 = Pin(16, Pin.IN, Pin.PULL_UP)
```

The third parameter enables a pull-up resistor for that pin. You can now remove the external resistor.

The same can be done for pull down by using the value `Pin.PULL_DOWN`.

To detect if the button has been pressed, use the following line:

```
button1.value()
```

This will return 1 if the input is high (button not pressed) and 0 if the input is low (button pressed).

04 Handling outputs

Using the GPIO ports as an output gives a different problem. The GPIO pins can only provide around 12mA from each of the pins. This is enough to pass a signal on to another device and can also be used to light up a standard LED, but it cannot switch large currents needed by a buzzer, high-power LED, or many other devices.

A common way of switching larger loads is with a MOSFET transistor.

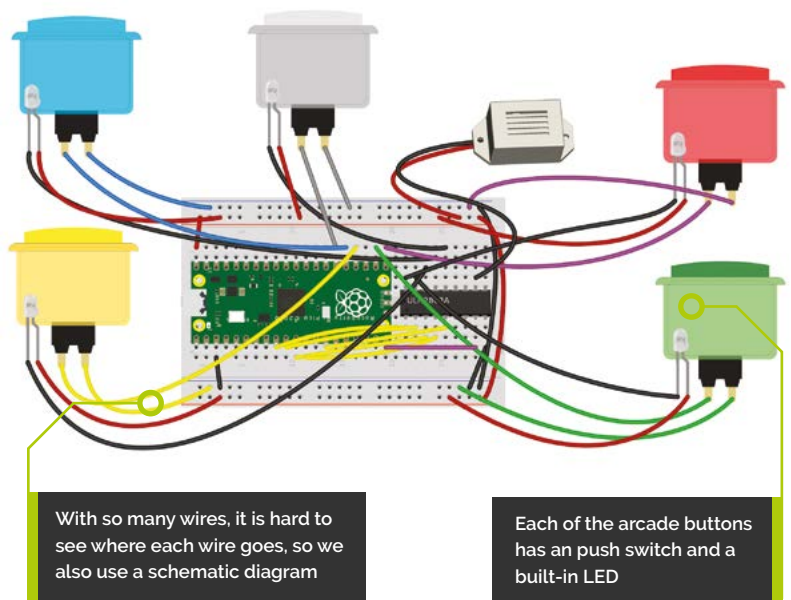
05 N-channel MOSFET

A MOSFET is a semiconductor device which is a type of transistor. It is like a switch which is controlled by an electrical signal instead of pushing it down. The MOSFET has three pins: a drain and source, which is where the main current will flow, and a third pin known as the gate which controls whether the MOSFET is turned on or off. For the N-channel MOSFET, if a positive voltage is applied to the gate then current can flow from the drain to the source.

MOSFETs can be used to switch bigger loads than integrated circuits are capable of. An example of a MOSFET is a 2N7000, which can switch up to 200mA. See the schematic symbol and pin layout in **Figure 3**.

06 Making noise with a buzzer

The buzzer used is a self-contained miniature buzzer designed for 5 or 6V. This type of buzzer is the easiest to use and will make a buzzing noise whenever a voltage is applied to it.



Some other types of buzzers, such as a piezo buzzer, need to have a rapidly changing signal, which would need to be included in the code.

The buzzer uses 25mA, which is too much for Pico's GPIO pins, but within the range for the 2N7000 MOSFET. It also needs to be powered from a 5V power supply. This can be taken from the VBUS connection on pin 40 of your Pico, which is connected to the 5V connection on the USB connector.

▲ **Figure 1** The wiring diagram for the quiz buzzers system

Top Tip

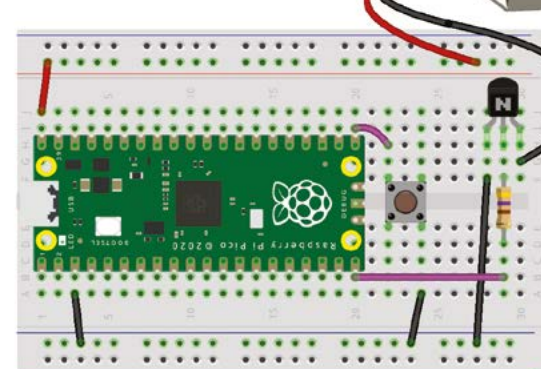
MOSFET pinouts

Different MOSFETs can have different pinouts. Check the datasheet for your MOSFET for more details.

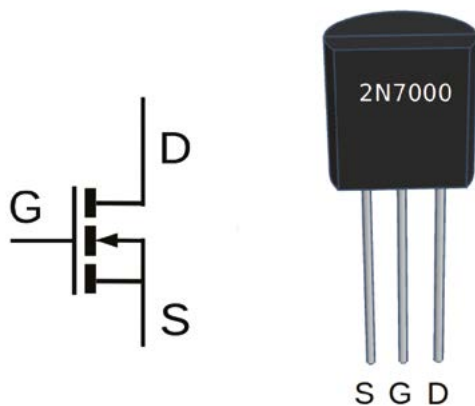
07 Wiring up the MOSFET

Different MOSFETs can have different pinouts. The 2N7000 is in a TO-92 package. With the flat part of the MOSFET at the front, the source is on the left (connected to ground), the gate is in the middle (connected to the output from the GPIO pin), and the drain is on the right (connected to the buzzer's negative terminal).

The breadboard layout for the circuit can be seen in **Figure 4**. A resistor is used between the



▲ **Figure 4** The MOSFET allows Pico to control circuits with a higher voltage or current requirement. The buzzer needs more than a standard GPIO pin could provide



► **Figure 3** Circuit symbol and pin layout for a 2N7000 MOSFET. The gate acts like an electronic version of the button on a switch

GPIO pin and the gate of the MOSFET. The reason for this is that when a MOSFET is first turned on, there can be an initial inrush current flowing into the gate. Once the MOSFET is fully turned on, the current will be very small. A 470Ω resistor reduces the inrush current.

08 Making some noise

With the wiring complete, the buzzer can be activated by setting the output of the GPIO pin high. A simple `while` loop can be used which detects if the button is pressed:

```
if button1.value() == 0:
```

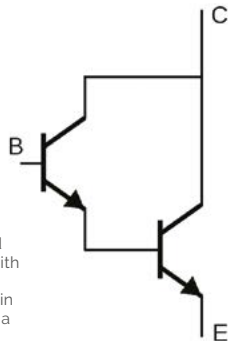
And if so, then turn the GPIO output pin high.

```
buzzer = Pin(15, Pin.OUT)
buzzer.value(1)
```

09 Using an integrated circuit

To create the game, you need to add arcade buttons with built-in LEDs. Many arcade buttons are designed for 12V, but to use with Pico's power, you need ones designed for 5V.

The button LEDs could be controlled through a MOSFET as described above. This would need



► **Figure 5** The Darlington driver uses two bipolar transistors paired together. Used with a digital circuit, they can be utilised in the same way as a single MOSFET

five MOSFETs and five resistors to control all the LEDs. This would also take up a lot of space on the breadboard. An alternative is to use an integrated circuit that combines these into a single chip. A suitable integrated circuit is the ULN2803. This is actually a Darlington driver chip, which is based around a bipolar transistor – see the schematic in **Figure 5**. The ULN2803 has eight Darlington drivers (as shown in **Figure 6**) and can also be used to replace the MOSFET for the buzzer.

10 Connecting the buttons

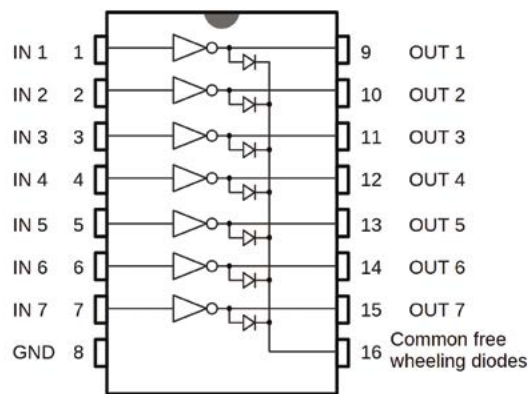
These arcade buttons have four pins, two of which are for the switch and two are for the LED (see the datasheet or supplier information to identify which is which). The buttons and their LEDs have spade connectors. These can be connected using pre-made wires, or you can create your own using a crimp tool (see magpi.cc/crimp).

For the arcade buttons, the switch pins should be connected between the GPIO pin and ground. For the LED, connect the +5V terminal to the 5V power supply and the ground terminal to the appropriate pin on the ULN2803.

The wiring is shown in **Figure 1**, but the number of crossing wires makes it hard to see. A better diagram is the schematic in **Figure 7**.

11 Programming the arcade buttons

The outputs for the LEDs are defined in the same way that the buzzer is. Rather than creating five separate variables for the inputs and five for the LEDs these have been combined into lists.



► **Figure 6** The ULN2803 has eight Darlington drivers, which share a common ground. These can be used for eight separate devices. The large triangle symbols represent each Darlington driver

Top Tip

Darlington driver

The Darlington driver uses two transistors connected together. This gives a very high gain, allowing a much higher output current compared to the input.

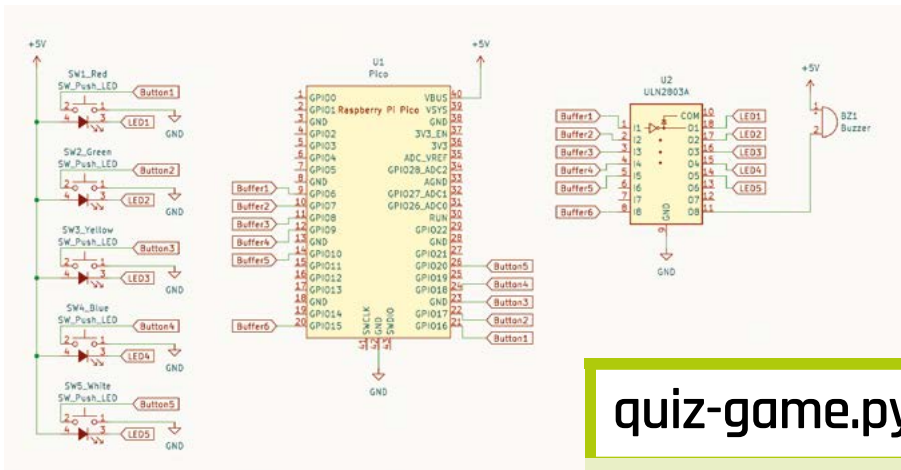


Figure 7 The schematic diagram makes it easier to see which components are wired together. To avoid crossing lines, labels have been used; labels with the same name are connected together

A `for` loop is used to go through each of the pin numbers and assign them to the buttons list for the inputs and the `leds` list for the outputs.

The code provided allows for the buttons to be used in a ‘finger on the buzzer’ quiz game. Each of the buttons’ elements are checked and, if a button is pressed, then the LED is turned on and the buzzer sounds.

12 Finishing the game

You can play the game with the players holding the switches in your hand, or you can create an enclosure to hold the circuit and mount the switches. If you have a 3D printer, then you can download the STL file from magpi.cc/penguinquizgame. If not, you can create your own by drilling appropriate holes into a plastic storage box.

There are different games that could be played by changing the code. The code included here is for a quiz game buzzer. In addition, the GitHub repository also includes a reaction game. [\[1\]](#)



Figure 8 The finished game can be installed into an enclosure. This one is 3D-printed, but you could just use a plastic storage box and drill holes for the buttons

quiz-game.py

DOWNLOAD THE FULL CODE:

Language: **MicroPython**

magpi.cc/arcadebuttonsgit

```
001. ## Quiz game
002. # Detects the first to press their button
003. # Sounds the buzzer for 1 second and lights the
004. # button pressed for additional 2 seconds
005. from machine import Pin
006. import utime
007.
008. buzzer = Pin(15, Pin.OUT)
009. # Red, Green, Yellow, Blue, White
010. button_pins = [16, 17, 18, 19, 20]
011. led_pins = [6, 7, 8, 9, 10]
012. buttons = []
013. leds = []
014.
015. for i in range(0, len(button_pins)):
016.     buttons.append(Pin(button_pins[i], Pin.IN,
017.                        Pin.PULL_UP))
018.     leds.append(Pin(led_pins[i], Pin.OUT))
019. while (1):
020.     button_pressed = False
021.     # check for which buttons pressed
022.     for i in range(0, len(buttons)):
023.         if buttons[i].value() == 0:
024.             print("Button {} pressed".format(i))
025.             leds[i].value(1)
026.             button_pressed = True
027.         else:
028.             leds[i].value(0)
029.     if button_pressed:
030.         # Sound buzzer for 1 second
031.         buzzer.value(1)
032.         utime.sleep(1)
033.         buzzer.value(0)
034.         # Leave LEDs on for 2 more seconds
035.         utime.sleep(2)
036.         # Reset all buttons
037.         for i in range(0, len(buttons)):
038.             buttons[i].value(0)
```

Create a Bluetooth music visualiser



Phil King

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

@philkingeditor

Use a Pico W-powered LED matrix display as a Bluetooth speaker with real-time visual effects

As the French trio Stardust once sang, “music sounds better with you”. It also sounds better when you have an audio visualiser creating cool-looking, on-the-fly lighting effects reacting to the sound. The good news is that you can turn a Pimoroni Galactic or Cosmic Unicorn LED matrix display into a Bluetooth speaker with real-time visual effects.

While you can easily run the ‘Blunicorn’ music visualiser by downloading the firmware and flashing it to the Unicorn’s integrated Pico W, we’ll show you how to edit one of the effects and add an extra one by editing the code on GitHub.

01 Download Blunicorn firmware

The Galactic and Cosmic Unicorn displays come with Pimoroni’s own custom Pico firmware pre-installed with an automatically running program that lets you choose from four demos. But you’ll need to replace this with the special Blunicorn firmware to create a Bluetooth music visualiser.

Head to the Releases page in the GitHub repo ([magpi.cc/blunicorndl](https://github.com/magpi/blunicorndl)) and download the UF2 firmware for your board: either the ‘cosmic-blunicorn’ or ‘galactic-unicorn’ file, in ZIP or GZ form. Once it’s downloaded to your computer, extract it to create a folder with the relevant **blunicorn.uf2** file inside.

02 Flash firmware to Pico W

Now you have the UF2 file, it’s time to flash it to your display board’s Pico W, situated on the rear. As usual, connect it via USB to your computer while holding the BOOTSEL button on Pico W. Release the button and it should show up on your computer’s file system as a drive called RPI-RP2. Simply drag the **blunicorn.uf2** file into the RPI-RP2 drive to flash it to Pico W.

03 Bluetooth pairing

Once the firmware is flashed, press the Reset button on the rear of your Galactic or Cosmic Unicorn to restart it. It will automatically start running the Blunicorn firmware, which makes it available as a Bluetooth device.

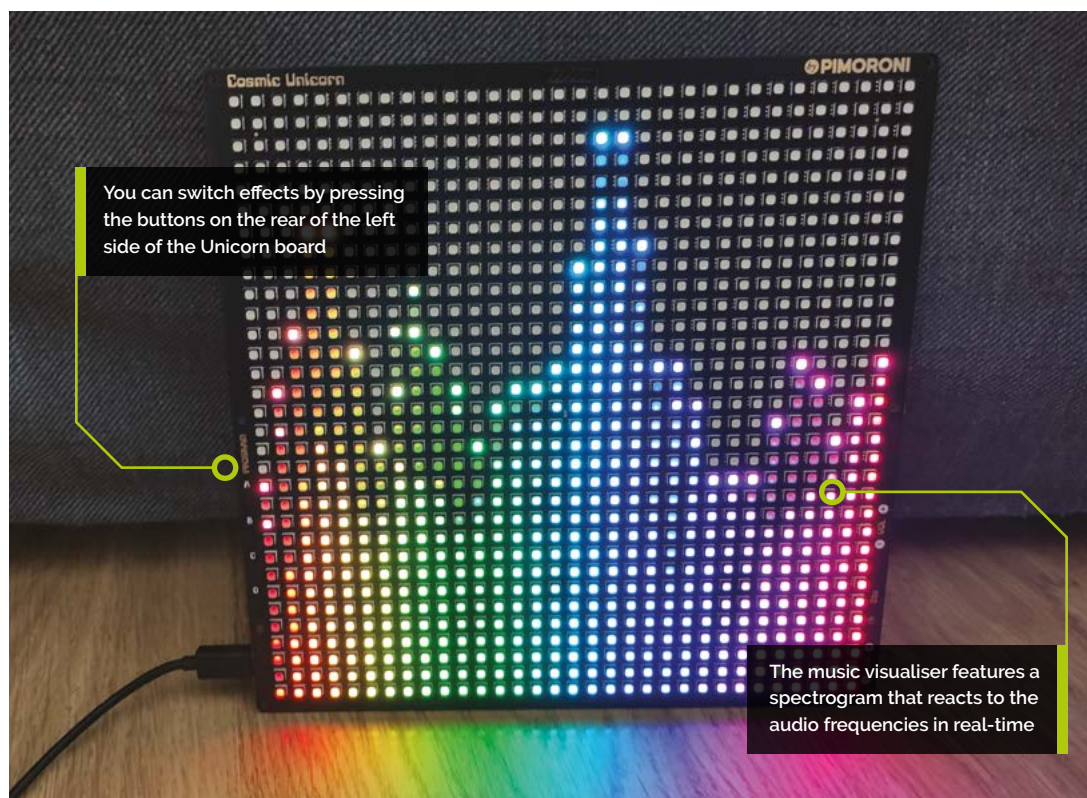
On your computer, or a smartphone, turn Bluetooth on and find ‘Galactic Unicorn’ or ‘Cosmic Unicorn’ in the available Bluetooth devices. Select it to pair with it on Bluetooth. If using a computer, make sure its audio output is also set to Galactic or Cosmic Unicorn rather than the internal speakers.

04 Use visualiser effects

With your computer or smartphone connected to your Unicorn display via Bluetooth,

You’ll Need

- ▶ Cosmic Unicorn or Galactic Unicorn (Pico W Aboard) magpi.cc/cosmic magpi.cc/galactic
- ▶ Blunicorn firmware magpi.cc/blunicorndl
- ▶ Laptop, tablet, or smartphone with Bluetooth



it's time to test it out. Just start playing some music or other audio, and you should hear it through your Unicorn board's rear speaker. If not, make sure you have the volume turned up on your computer or phone.

The Unicorn display should also start showing a rainbow FFT (fast Fourier transform) effect by

“ Make sure you have the volume turned up on your computer or phone ”

default, with a spectrogram reacting in real-time to the levels of various frequencies in the music.

You can press the B button on the Unicorn to switch to a 'classic' FFT effect which shows the main spectrogram frequency levels in green and the peaks in red.

Try different types of music and alter the output volume on your connected device for higher or lower peaks.

05 Fork GitHub repo

To start editing an effect, it's a good idea to fork the GitHub repository so you can experiment with altering the code directly via GitHub rather on

a locally downloaded version – although you can still do that by cloning the repo to your computer.

You'll need to sign up for a free GitHub account if you don't have one already. Then visit the Blunicorn repo (actually named 'galactic-bluetooth-audio') at magpi.cc/blunicorn. Click the Fork option at the top right of the page to create a copy of the repo in your own GitHub account. You can now alter this to your heart's content.

06 Edit an effect

While you could clone the GitHub repo to your computer and edit the code locally, you need quite a lot of extra bits installed to build a new UF2 file on the command line, including the Pico-SDK. It's far easier to edit the code directly on GitHub and then use GitHub Actions to automatically build a new UF2 file.

Open the **effect** folder and click on **classic_fft.cpp** to edit the code for the Classic FFT effect. We're going to make a simple change to alter the effect's colour scheme, which is set in the **for** loop on lines 76 to 81. The **h** variable determines the hue.

The 0.4 multiplier on line 78 sets the colour sweep range, while the 0.333 on line 79 determines the offset to map to a specific portion of the hue-wheel. By default, this

Top Tip

Restore firmware

If you want to later restore the standard Pimoroni firmware to your Galactic or Cosmic Unicorn, download the relevant version from magpi.cc/pimoronipicoreleases and flash it to Pico W.



▲ A modified Classic FFT effect on the Cosmic Unicorn, with a greater range of colour and altered offset

results in green frequency levels and red peaks. By adjusting the offset to 0.666, for instance, you can change the colours to blue and green.

07 Compile new UF2

One of the pros of working in GitHub is that it can be used to easily rebuild the project's UF2 files when you change the code. Click on GitHub Actions and then, for 'CMake based projects', click the Configure button to create a YAML file for it.

The CMake workflow should then run automatically every time you commit a change to the repo (if not, make sure the workflow is enabled). It usually takes a couple of minutes to complete the task; UF2 files for Galactic and Cosmic Unicorn versions will then appear in the Artifacts section for that workflow run.

08 Flash and test

Download the relevant newly created UF2 file for your Unicorn model, and flash it to Pico W as previously, then reset. Connect to the Galactic or Cosmic Unicorn via Bluetooth from another device again and then press the B button on the rear to see the Classic Effect with your new colour scheme (blue shades and green peaks in our example).

Experiment with changing the range and offset values for the `h` variable in the effect code to get different colours. If you have C++ skills, you can also try altering other aspects.



09 Add a new effect

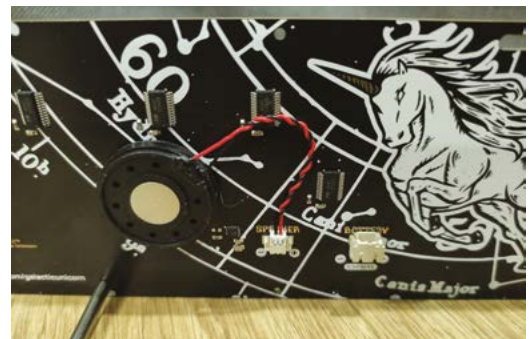
Next, we'll duplicate an existing effect file, edit it, then assign it to an unused button (C). Open the `classic_fft.cpp` file in GitHub, then click the Copy icon. Go back to the `effect` folder, click on 'Add file', then '+ Create new file'. Paste the code you copied, then edit it however you want

“ Experiment with changing the range and offset values ”

(e.g. to alter the colour scheme, as before). Change the instances of the function name 'ClassicFFT' to 'ClassicFFT2'. Name the file in the top field (e.g. `classic2_fft.cpp`), and click 'Commit changes'.

You need to create a new ClassicFFT2 class in the `effects.hpp` file. Copy and paste the ClassicFFT class (lines 49–77), then change the new class's instances of 'ClassicFFT' to 'ClassicFFT2'.

You also need to create a CMake file for the effect. Follow the above procedure to copy the code from `classic_fft.cmake` into a new file

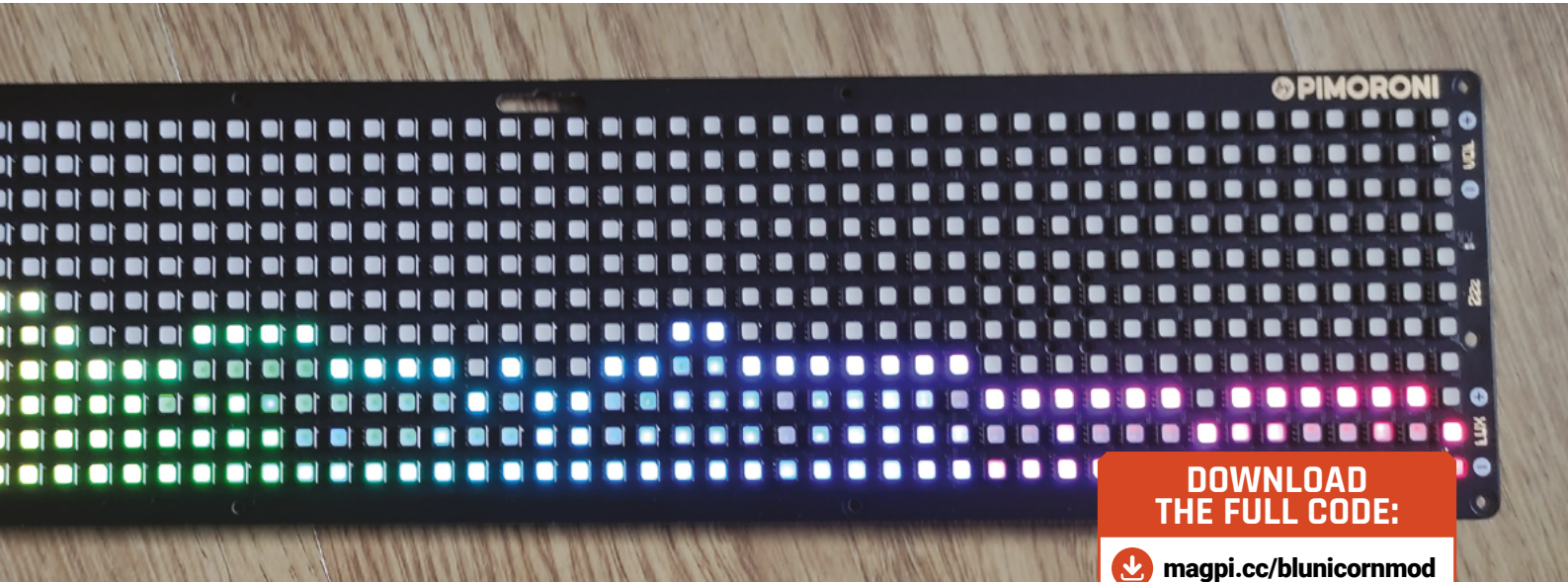


▲ The rear of the board features a mini speaker that plays the audio; you could connect a bigger speaker to the port

Top Tip

Make it another way

If you clone the GitHub repo, you can use CMake to build the project on the command line. You'll need the Pico-SDK, `pico-extras` library, and GCC ARM embedded toolchain. See section 2.1 of the Getting started with Raspberry Pi Pico guide: magpi.cc/getstartedpico.



DOWNLOAD THE FULL CODE:

magpi.cc/blunicornmod

called `classic2_fft.cmake`. You'll need to edit the code in it to change the instances of 'classic_fft' to 'classic2_fft'.

10 Effect additions

We'll need to add the new effect to the button handling list in `btstack_audio_pico.cpp` in the `src` folder. Under line 180, add this code to assign the third effect to button C:

```
if (!gpio_get(Display::SWITCH_C)) {
    current_effect = 2;
}
```

In the same file, the effects are pushed to the Bluetooth stack on lines 150–151. Underneath, add:

```
effects.push_back(&classic2_fft);
```

To add an instance for the effect, add this after line 70:

```
ClassicFFT2 classic2_fft(display, fft);
```

When done, click 'Commit changes'.

11 Append CMake list

To get CMake to build the third effect successfully into a new UF2 file, you'll need to make a couple of additions to the `CMakeLists.txt` file. First, let's link it to the project; under line 42, add:

```
classic2_fft
```

To include the effect, under line 16, add:

```
include(effect/classic2_fft.cmake)
```

Now click 'Commit changes'.

12 Download and test

Check the GitHub Actions page and you'll see the new workflow in progress. You can click on it and then either Cosmic Unicorn or Galactic Unicorn to see the verbose output and check for any errors.

Once the workflow has completed successfully, download the relevant UF2 file from its Artifacts list and flash it to your Unicorn board as before. Reset the board, connect via Bluetooth, and try pressing the C button to see the new effect. You can now switch between three effects on your music visualiser. **M**

▲ The Rainbow FFT effect on the smaller Galactic Unicorn display, reacting to Bluetooth music being played

▼ By changing the hue range and offset values in the `classic_fft.cpp` effect code, we can alter the colours used



Camera Module: Getting Started



David Plowman

MAKER

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

raspberrypi.com

Getting to know Raspberry Pi Camera Module options, setting up, and testing a camera with Raspberry Pi

In this tutorial, we'll introduce you to the range of different Raspberry Pi cameras and talk you through the particular features and benefits of each one. We'll show you how to connect them to Raspberry Pi and test that they are working correctly. Finally, we'll finish with a couple of simple examples to show you how to capture your first photo or video.

This is the first part of a series of tutorials that will take you through Raspberry Pi cameras, from absolute beginner to advanced Raspberry Pi imaging techniques.

Raspberry Pi has made five different types of cameras, of which all but the original v1 camera are currently in production. Some of them come in two or more different variants. Compatible camera modules are also made by a number of third-party suppliers, though we can't cover all of those in this tutorial and, for any questions of support issues related to them, you might have to go back to the supplier where you bought the camera.

The official Raspberry Pi camera modules are:

- Raspberry Pi Camera Module
- Raspberry Pi Camera Module 2
- Raspberry Pi High Quality Camera
- Raspberry Pi Camera Module 3
- Raspberry Pi Global Shutter Camera

You'll Need

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

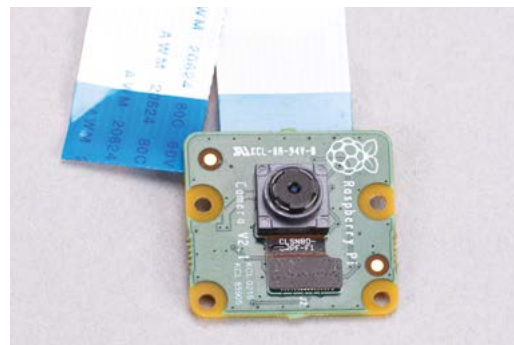


01 Raspberry Pi Camera Module

Raspberry Pi's first official camera was launched in 2013 and production ceased in 2017 as the sensor on which the camera is based, is no longer available. A number of third-party sellers continue to sell v1 or v1-compatible cameras, and all Raspberry Pi software continues to support it.

The camera is a 5MP device in a fixed focus module, with a maximum resolution of 2592×1944 pixels, and uses the Omnivision OV5647 image sensor. At lower resolutions, it is capable of up to 60 frames per second. It is suitable for still image photography and video recording.

Versions of this camera with the IR (infrared) filter removed have been available for specialised use cases.



02 Raspberry Pi Camera Module 2

Raspberry Pi Camera Module 2 was introduced in 2018 to replace the original Camera Module, and it comes in a similar small form factor. It offers broadly similar features and slightly improved performance.

Camera Module 2 is an 8MP pixel device capable of outputting images at a maximum resolution of 3280×2464 pixels. It is capable of up to 100 frames per second at a VGA resolution and is based on Sony's IMX219 image sensor.

Like the original camera, it uses a fixed focus lens which is manually adjusted using a supplied tool to increase the range of applications for which it can be deployed.

03 Raspberry Pi High Quality Camera

Raspberry Pi High Quality Camera, introduced in 2020, is a larger device that supports interchangeable C- and CS-mount lenses. The image sensor ships without an included lens. Raspberry Pi recommends a wide-angle 6 mm CS-mount lens and a narrower field of view 16 mm C-mount lens; both are low-cost and suitable for a wide range of use cases. For specific applications, users may want to select more appropriate lenses. An alternative version of the HQ Camera that accepts M12 lenses was released in 2023.

The High Quality Camera features a 12MP sensor offering a maximum resolution of 4032×3040 pixels and, because of its physically larger sensor size, it can be expected to produce better image quality than the Camera Module 2. It makes use of Sony's IMX477 image sensor, and is also suitable for capturing video. At lower resolutions, it can run at up to 120 frames per second.

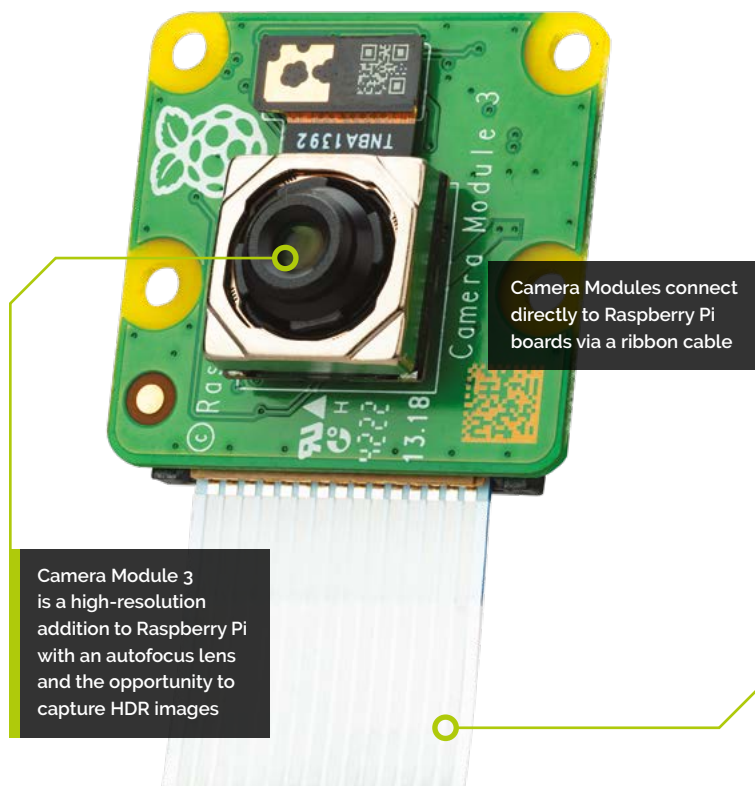


04 Raspberry Pi Camera Module 3

Introduced in 2023, the Camera Module 3, is another small form factor camera that is quite similar to Camera Module 2. It offers increased resolution in a wider aspect ratio, an autofocus lens, and the opportunity to capture HDR (High Dynamic Range) images.

The autofocus can run in continuous mode, or be triggered prior to image capture, according to the software being used.

The maximum resolution of the sensor is 4608×2592 pixels, therefore note that it has a



Camera Modules connect directly to Raspberry Pi boards via a ribbon cable

Camera Module 3 is a high-resolution addition to Raspberry Pi with an autofocus lens and the opportunity to capture HDR images

“ The sensor is quite capable for still photography, but shines particularly for video recording applications ”

16:9 rather than a 4:3 aspect ratio. Because of the special nature of this sensor, the HDR feature is not available at the maximum resolution, but only at a resolution of 2304×1296 pixels.

The sensor is quite capable for still photography, but shines particularly for video recording applications, especially where HDR is beneficial. At an appropriate resolution, it will support video capture at up to 120 frames per second.

The camera module is based on the Sony IMX708 sensor. It is packaged into four different variants:



Top Tip

NoIR

Current Camera Module devices are available without the IR (infrared) filter. These are able to record better in low light conditions (but daylight images have curious colours). These models are marked as 'NoIR'.

- Camera Module 3. The regular version with an IR filter and lens with the standard field of view
- Camera Module 3 Wide. A version with an IR filter but replacing the standard lens with a wide-angle one
- Camera Module 3 NoIR. A version without an IR filter but with the standard field of view lens
- Camera Module 3 NoIR Wide. A version without an IR filter and with the wide-angle lens



05 Raspberry Pi Global Shutter Camera

The Global Shutter (GS) camera, introduced in 2023, uses a quite different technology to the other cameras. Whereas it's normal in this market segment to use rolling shutter cameras, the GS camera employs a global shutter.

In this camera, all the pixels in a captured image are exposed at exactly the same time, rather than the rolling shutter camera where pixels are captured line by line. In practice, this is great for capturing motion – the bus whizzing by in your video will remain a rectangle with the global shutter camera, whereas rolling shutter sensors will all turn it into a parallelogram.

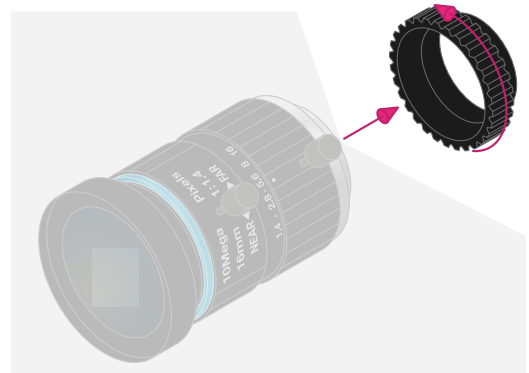
The Global Shutter camera is based on Sony's IMX296 image sensor. Global shutter cameras are electronically more complex, so resolutions are generally lower. In this case, the Global Shutter camera produces 1456×1088 pixels, though on the upside the extra internal complexity means that the physical pixels are larger, making the camera more sensitive and better at low light imaging.

The Global Shutter camera uses the same mounting and lenses as the High Quality Camera, so for more information on using and swapping lenses, please check out the following section on interchangeable lenses.

06 Using interchangeable lenses

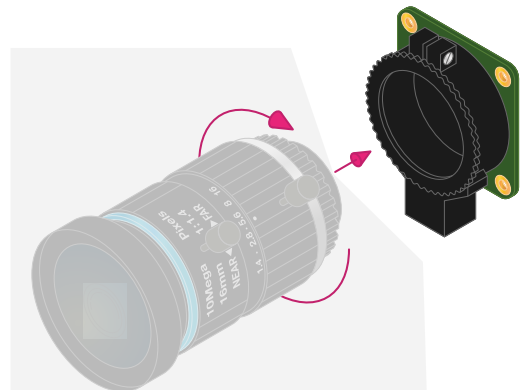
The High Quality and Global Shutter Cameras both use the same formats of interchangeable C-mount and CS-mount lenses. An adapter for M12 lenses is also available.

Raspberry Pi supplies a low-cost, wide-angle 6 mm CS-mount lens and a larger 16 mm C-mount lens. Their usage is identical, except that the 16 mm C-mount lens will require the C-CS adapter to be fitted first.



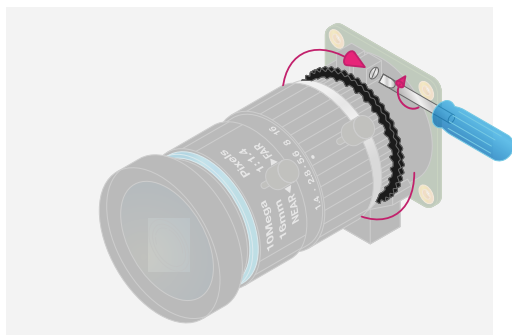
6.1 Fitting the C-CS adapter

If you need the C-CS adapter, ensure that it is fitted to your 16 mm lens. The lenses have a longer back focus than the 6 mm lens, and therefore require the adapter.



6.2 Fitting the lens to the camera

Rotate the lens (and C-CS adapter if present) clockwise, all the way into the back focus adjustment ring.

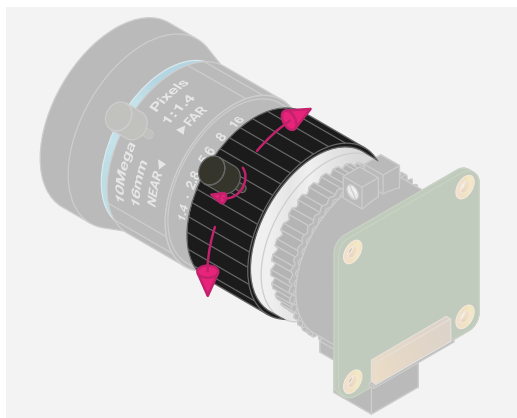


6.3 Back focus adjustment ring and lock screw

The back focus adjustment ring should be screwed in fully. Use the back focus lock screw to make

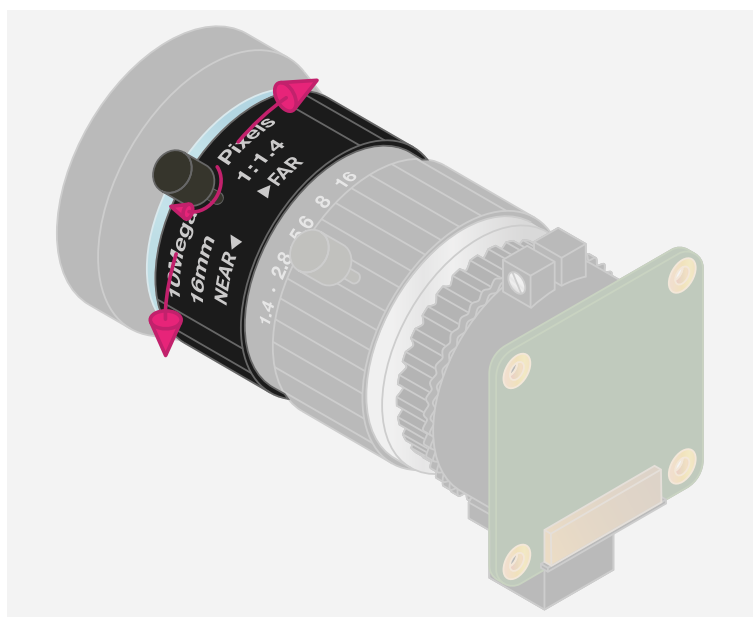
“ An adapter for M12 lenses is also available ”

sure it does not move out of this position when adjusting the aperture or focus.



6.4 Aperture

To adjust the aperture, hold the camera with the lens facing away from you. Turn the inner ring, closest to the camera, while holding the camera steady. Turn clockwise to close the aperture and reduce image brightness. Turn anti-clockwise to open the aperture. When happy with the light level, tighten the screw on the side of the lens to lock the aperture into position.



6.5 Focus

To adjust focus, hold the camera with the lens facing away from you. Turn the focus ring, labelled 'NEAR<>FAR', anti-clockwise to focus on a nearby object. Turn it clockwise to focus on a distant object. You may find you need to adjust the aperture again after this.

07 Connecting and testing your camera

Your Raspberry Pi and camera are connected using a flat ribbon cable. There are actually two types of camera connector: the standard one which we find on all the camera boards and on 'large' Raspberry Pi computers (Raspberry Pi 2, 3, and 4). Then there is a slightly narrower version of this connector which we find on 'small' Pi computers (Raspberry Pi Zero, and Zero 2) and also often on Raspberry Pi's I/O board that goes with the various Compute Modules.

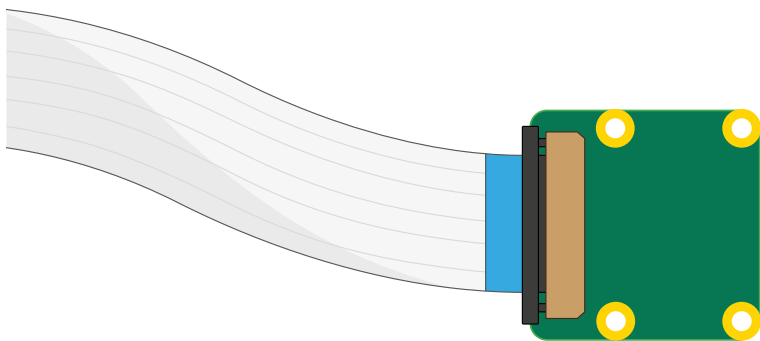
Correspondingly, the cables may have both ends the same, or you can also get cables with one narrow end and one wider end.

On one side of the cable, the electrical connections, visible as narrow silver or golden stripes, will be visible. On the other side, they are covered by a coloured (often but not always blue) piece of plastic. We'll refer to these as the 'electrical' and 'insulated' sides of the ribbon cable respectively.

Top Tip

Removing IR

Some users have removed the IR (infra red) filter from the High Quality Camera. The operation requires some skill and will void any warranty and there's a significant risk of permanently destroying your camera.

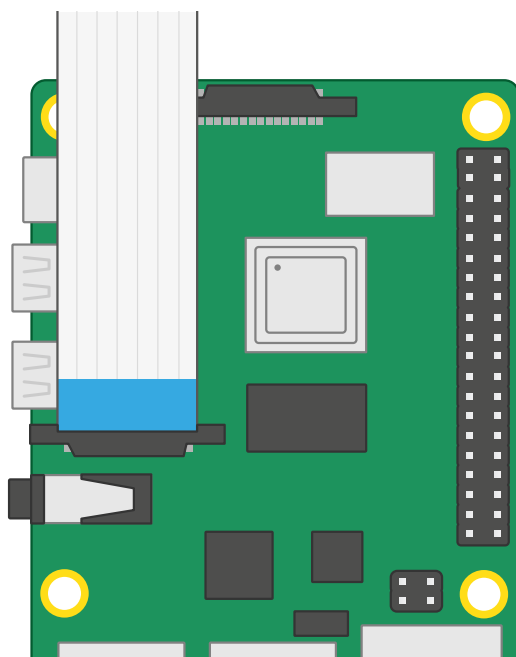


08 Connecting to the camera

At the bottom of every camera board is a connector for the ribbon cable. At the edges, you'll be able to grip the bottom part of the connector and pull it slightly out. Once you've done this, you should be able to push the ribbon cable in. Finally, push those edges of the connector home again so that the ribbon cable is held firmly.

On all Raspberry Pi camera boards, the ribbon cable must be inserted so that the electrical side of the cable faces forwards. So, if you turn the camera board to point away from you, so that you can't see the actual camera lens, then you should see the insulated side of the ribbon cable.

The camera ribbon cables are short for a reason – they can otherwise be prone to interference and transmission errors. We would generally recommend avoiding lengths greater than about 30 cm, though people have had success with longer cables too.



09 Connecting to Raspberry Pi

The large Raspberry Pi boards have the same connector as you will find on the camera, and you can open and close it in the same way. Note that there are two identical connectors on the boards – be sure to use the one located near the Ethernet and USB ports, and not the one on the far side of the board. The ribbon would fit here but the camera will not function.

As for orientation, the insulated side of the ribbon cable must face towards the Ethernet and USB ports, so the electrical side will face towards the power connector.

Raspberry Pi Zero boards have a narrower version of the connector, and therefore also need a ribbon cable with the narrower end. These are very similar, only smaller, and open and close in the same way. But do take care, as they are more fragile and break more easily!

When you insert the ribbon cable, make sure that the insulated side of the cable is face up. That is, when you can see the chips and connectors on the top of the board, you can also see the insulated side of the cable.

“ Raspberry Pi Zero boards have a narrower version of the connector ”

Some other boards, such as the I/O board Raspberry Pi makes for Compute Modules, have a different version of the narrow connector. This one is hinged – carefully prise the edge of it up. The ribbon cable should be inserted with the electrical side face down. The hinged part of the connector should now be snapped shut on top of the insulated side of the connector.

10 Test your camera

To test that your camera is correctly connected and working, you will need to install Raspberry Pi OS (Operating System) onto your board. Using Raspberry Pi Imager (magpi.cc/imager), write the default option, Raspberry Pi OS (32-bit), to a microSD card.

Insert this into your Raspberry Pi and power it on. Let it perform all the necessary updates when it first boots and then finally reboot it again.

For those less familiar with Raspberry Pi computers, we would recommend performing your

initial testing with a keyboard and screen attached directly to your Raspberry Pi. Using the Raspberry menu, open the Accessories category and click on the Terminal button. Into the Terminal window it should be sufficient to type:

```
libcamera-hello
```

This will open a camera preview window and display the camera images there for about five seconds.

If you don't see the preview window, please check that:

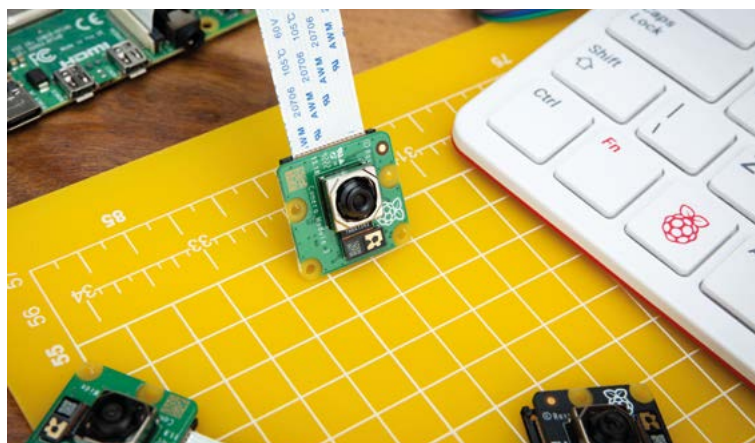
- You are directly connected to your Raspberry Pi with a screen and keyboard
- You are running the official Raspberry Pi software and that it is fully up to date. If you have made any changes to your Raspberry Pi's configuration, please consider reverting back to the standard unaltered Raspberry Pi OS (by rewriting your micro SD card if necessary) and trying again
- All your cables are the right way round and seated correctly
- You are using an official Raspberry Pi camera. Non-official cameras are supported, but may need some extra configuration which the module vendor would have to supply
- You have a good-quality power supply. Raspberry Pi power supplies are recommended because they are known to supply sufficient power to Raspberry Pi and the camera
- Please check out Raspberry Pi's Camera Forum for more tips if you're having trouble.

11 Capturing your first photo

Raspberry Pi OS comes with a number of command-line utilities for capturing images in various different ways. We've already seen `libcamera-hello`, which we used to test that the camera was working. The next one is `libcamera-still`. This opens up a camera preview window just like `libcamera-hello` did but, once it's run for five seconds, it will capture a high resolution still photograph as a JPEG file, and then quit. To use it like this, enter:

```
libcamera-still -o picture.jpg
```

Note how the `-o` option specifies the name of the file to which the image is saved. There is also the `-t` option which specifies how long, in



milliseconds, the preview window runs before the image is captured. So to capture the image after ten seconds of preview, use:

```
libcamera-still -t 10000 -o picture.jpg
```

▲ Camera Module 3 attached to Raspberry Pi 4 board via the CSI (Camera Serial Interface) connector

“ The camera ribbon cables are short for a reason ”

You can view your pictures by double-clicking on the files you've saved in the File Manager. We'll learn about many more of `libcamera-still`'s options in the tutorials that follow.

12 Capturing your first video

There's another command-line tool for capturing videos; this time it's called `libcamera-vid`. You can use the `-o` and `-t` options as you did before, though this time `-t` specifies how long (again in milliseconds) the video lasts. To save a ten-second video, use:

```
libcamera-vid -t 10000 -o video.h264
```

Note that the file name ends with `.h264`, which identifies the format of the video file. Unfortunately, not all video players will play the `.h264` format correctly, so we would recommend using a utility like `ffplay`. To play your video file with `ffplay`, enter:

```
ffplay video.h264
```

We'll learn later how to save different formats of video, as well as more of `libcamera-vid`'s advanced features, in the tutorials that follow. [\[1\]](#)



RFID spells fun

Make a fun card-matching game using RFID readers and a Pimoroni Pico LiPo



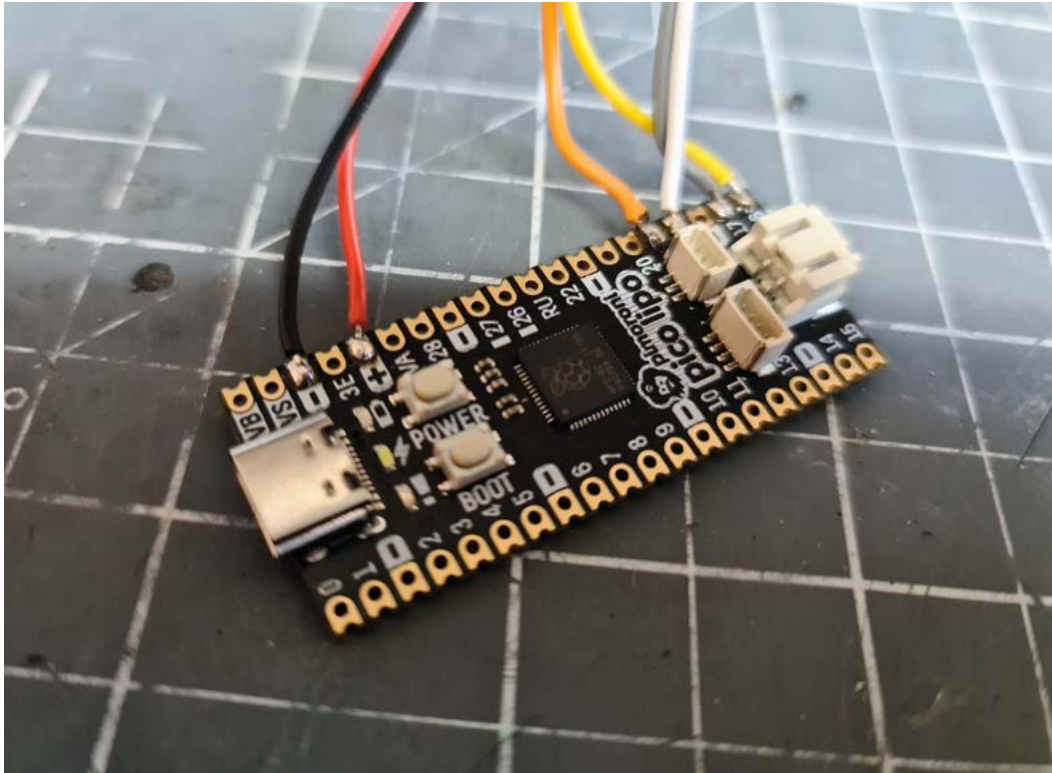
Dr Andrew Lewis

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

Kids love to play and learn, but playing games to practice new skills like reading, mathematics, or even simple baking is something that parents associate with either one-on-one interaction or with tablet computers. It doesn't have to be that way, and while computers do have their place, it's nice to present a child with a more tactile interactive experience. In this project, you'll see how to assemble a simple game board with radio frequency identification (RFID) readers to make a spelling game for young children. The game uses letter cards and picture cards. Using the letter cards to spell the word that matches the picture makes the board play a simple tune (see Beep Boop box, overleaf). You can also use the game to program new picture cards by activating a hidden reed switch on the board.

RFID systems are everywhere. They're in your phone, on the packaging for your parcels, on the food in supermarkets, in your debit card, and maybe even on the door to your office or gym. Looking at its list of applications in modern society, it's easy to see the technology as a very 'suited and booted' security and business-rooted phenomenon. It's a technology for stock control, security, and access control. But with origins that lead back to the theremin, it's a technology that deserves to be seen in projects with a lighter, less business-focused theme.

Making a simple game board isn't rocket science. You'll just need a shallow box that you can use as a board, and some creative inspiration to decorate it. You could hand-cut a box from thin plywood or cardboard, laser-cut one in acrylic, or even reuse an old pizza box. The important part of this project is the electronics, and getting the position of the RFID readers inside your game board correct. →



Left ♦ The Pimoroni Pico LiPo board is a little more expensive than the vanilla Raspberry Pi Pico, but it comes with some additional LiPo battery management and extra memory

YOU'LL NEED

- ♦ **Pimoroni Pico LiPo**
(shop.pimoroni.com/products/pimoroni-pico-lipo)
- ♦ **1 × LiPo battery pack**
(shop.pimoroni.com/products/lipo-battery-pack)
- ♦ **6 × RC522 13.56MHz RFID/NFC module**
(amazon.co.uk/AOICRIE-RFID-Kit-Compatible-Raspberry/dp/B09H6PLM1C/)
- ♦ **40 × 13.56MHz RFID/NFC cards, with 1kB of memory (also called MIFARE cards)**
(shop.pimoroni.com/products/rfid-card-10-pcs)
- ♦ **ISD1820 audio record module with mic and speaker**
(amazon.co.uk/HALJ1A-ISD1820-Recording-Playback-Loudspeaker/dp/B06XD96N43/ref=asc_df_B06XD96N43)
- ♦ **A reed switch**
(Normally open type)
- ♦ **A shallow box, or a piece of wood or card to use as a game board**

DATA LOADING

There has been a lot of talk here about reading and writing RFID cards, but very little about what that actually means. RFID cards come in a variety of different standards, using a number of different frequencies and protocols. This project uses an RC522 reader, which works at a frequency of 13.56MHz and is used with classic MIFARE-type cards that have 1kB of memory. These are the sort of cards that you find used as travel passes or printer/photocopier credit cards. The memory is arranged into 16 sectors, each of which contains four blocks of 16 bytes. The first three blocks are data blocks that are used for storage, but the fourth block is an access control block, which defines how the information in that sector can be accessed. Writing to this block incorrectly can permanently break the card. In addition to the writeable areas, there is a 4-byte long read-only ID burned into the card. This ID cannot be changed, and it lets you identify a particular tag if you need to. In order to read or write information on a MIFARE card, the card must be authenticated using one of two keys. The multiple keys can be programmed to allow different access levels to the information stored on the card. As an example, Key A could allow read-only access, while B would allow

write access. By default, these cards are usually delivered with the access keys set to hex FFFF. Some card sectors are safe to edit, others sectors are protected. If you try to mess around with these blocks and sectors and don't know what you are doing, you can trigger security features that will ruin a card permanently. The datasheet for MIFARE Classic 1K cards (hsmag.cc/MIFARE_Classic1K) gives you the full breakdown on exactly how this works and which areas are safe to work with, but for now, you just need to understand that the areas used in the project files are 'safe' to use, and if you want to start messing around with other areas, you should read around the subject a little bit before you experiment.

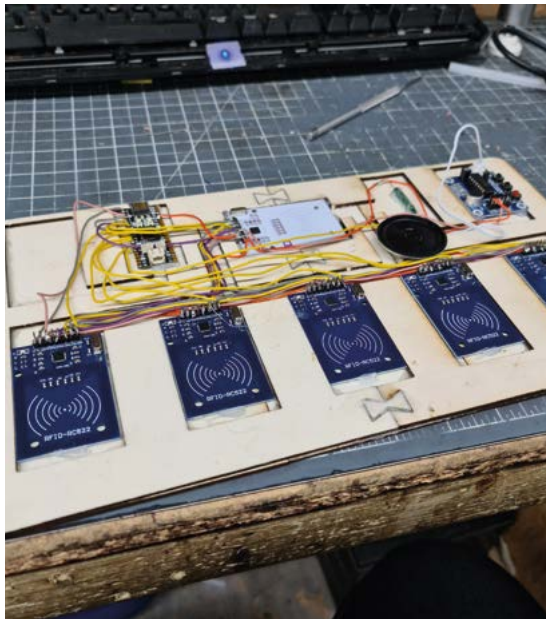
One other thing to note is that the configuration of the MIFARE card's memory as 16 sectors containing four blocks each is not actually reflected in the MicroPython code. In the Python examples, the code is represented more simply as 64 sectors of 16 bytes. This means that the access control blocks are actually every fourth sector. The sectors begin numbering at 0, so sectors 3, 7, 11, 15, 19...63 are all control blocks, and writing to them accidentally would most likely wreck the card.



Warning!
Hot solder

Soldering irons get very hot, and stay hot for a long time after they're unplugged. Make sure that you put the iron in the stand when you're not using it and don't touch the metal parts – even after it's unplugged.

magpi.cc/soldering



Above ♦

While this project looks pretty simple as a schematic, it's actually quite a workout to solder quite so many wires in one go unless you're used to doing it professionally. You might also find that you're going to be reusing the same wire colour for different purposes in the same circuit, which is never ideal. If you're feeling the strain, split the job into several sessions, and keep track of which wires you've already soldered by marking your schematic at the end of each session

BEEP BOOP

Detecting cards is part of the game, but so is creating some meaningful feedback for the user. LED lights and buzzers are one option, but you can also use a sound module like the ISD1820 or ISD1932 to make the game more distinctive. These modules allow you to record sounds using a microphone, and play them back in part, or in full, by triggering a pin high. The exact recording time and number of different sounds stored will depend on the version of the chip and the configuration that you're using, but you can easily record several sounds to reward your players. One consideration with these modules is that they aren't very loud, so you will need an amplifier if you want big sounds for your game.

Some RFID modules (particularly the cheapest of the 'buy 20 for the price of a coffee' varieties) don't have the best range, and will only work if your card or tag is a couple of centimetres away from the reader in open air. Different materials will reduce this range by different amounts, and anything that can interfere with electromagnetic waves (like iron) will probably stop the signal completely. Getting your players to place their cards in the right place is key to making your game work well. Most RFID modules use similar circuits and control chips, and it is possible to adjust the gain of most readers to increase their effective range. However, increasing the range too far could mean that multiple cards interfere with each other and give unexpected readings – it's unlikely, but possible, that this could cause a problem for your board game if the card readers are placed too close together.

The game you are creating will use RFID readers to read letter cards and picture cards, so you will need at least four RFID readers to spell simple three-letter words (three letters, one picture). The readers connect to a Raspberry Pi 2040-based board. In the board game shown in this article, a Pimoroni Pico LiPo has been used because the board has a LiPo battery charging circuit built in. If you want to use AA batteries with your project or power the project straight from USB, you could easily use a Raspberry Pi Pico or Raspberry Pico W instead.

To interface the RC522 devices with the Raspberry Pi Pico, you are going to use MicroPython, and Dan J. Perron's micropython-mfr522 library, which you can find here: hsmag.cc/DanJPerron. If you've never used MicroPython on the Raspberry Pi Pico before, you can follow the instructions at hsmag.cc/Get_Started_Pico to get started.

SERIAL PERIPHERAL INTERFACE

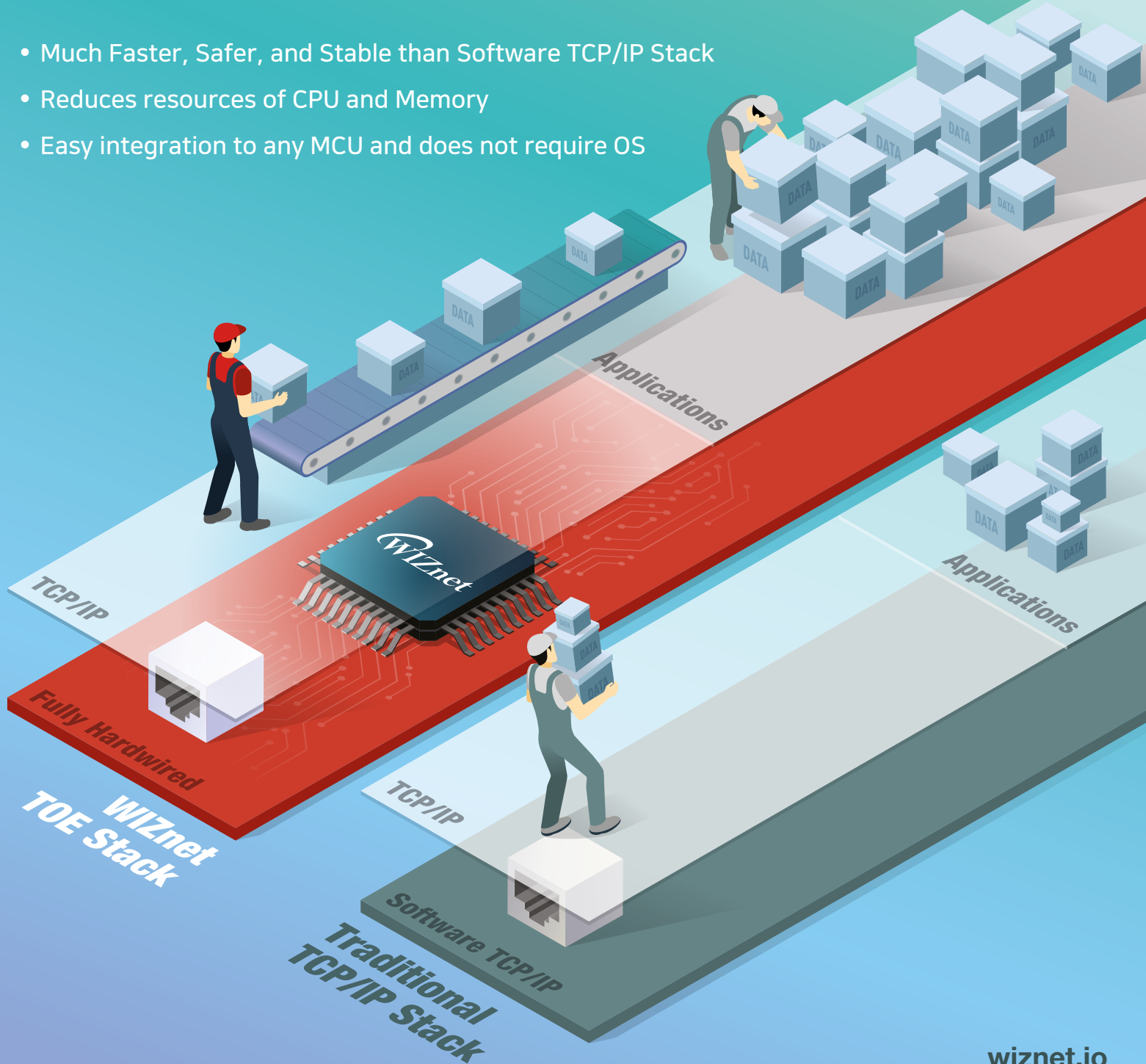
The RC522 RFID reader modules use the SPI interface to communicate with the Raspberry Pi Pico. SPI is a popular and well-established protocol that lets you connect multiple devices together on a single bus to reduce the overall number of lines used to communicate with the processor. Although some devices can be daisy-chained to reduce the number of connections going to the master device, many (including the RC522 modules used in this project) are independently wired, with each connection being connected to the same pin on the next board. The SPI interface needs several connections to function properly, some of which are shared between boards. These are usually labelled MISO, MOSI, and SCLK. MISO stands for Master In Slave Out, and is the connection that sends information from the slave device (our RC522) to the master device (the Raspberry Pi Pico). MOSI is the reverse of this, standing for Master Out Slave In, sending information from the master device to the slave). As you might have already guessed, SCLK is a clock signal that keeps everything in sync. In addition to these shared pins, you also need a dedicated pin to indicate which board should be active, and this is usually labelled SS or CS for Slave Select or Chip Select. On Some RC522 boards, this pin might be labelled SDA. This pin goes straight to the master and does not get shared with the other slave boards, which means that you'll need one free pin on your Raspberry Pi Pico for each device on your SPI interface. Although the RC522 board has several other pins, these are the only pins that relate to the SPI interface directly. The remaining pins deal with power (3.3V and GND), a device reset pin (RST), and an IRQ pin, which triggers when the slave detects a card. While it would be possible to use this pin to trigger the Raspberry Pi Pico, it provides no significant advantage over simply polling each reader in sequence, and would require several extra wires to be connected to the board.

Internet Offload co-Processors for IoT

WIZnet TOE

TCP/IP Offload Ethernet

- Much Faster, Safer, and Stable than Software TCP/IP Stack
- Reduces resources of CPU and Memory
- Easy integration to any MCU and does not require OS



sinclair



Commodo

the

COMPUTERS

THAT MADE

BRITAIN



OUT NOW

"The Computers That Made Britain is one of the best things I've read this year. It's an incredible story of eccentrics and oddballs, geniuses and madmen, and one that will have you pining for a future that could have been. It's utterly astonishing!"

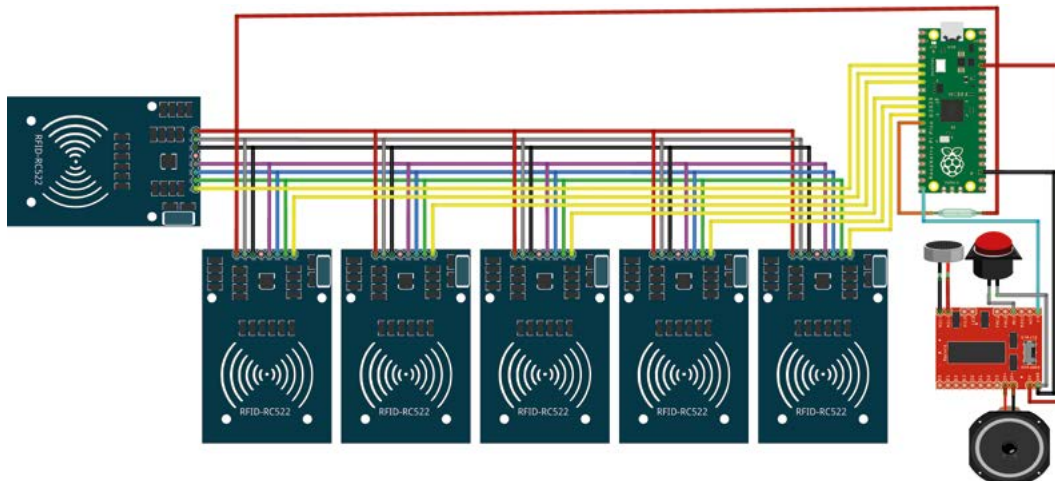
- **Stuart Turton**, bestselling author and journalist

.....



Buy online: wfmag.cc/ctmb





QUICK TIP

If you're having problems with the position of the cards not triggering an RFID reader, you could create a slight recess in the board so that your cards fit in exactly the right place.

Left ♦ The RC522 RFID modules use the SPI interface, sharing several wires between each board. GPIO pin 15 serves as a trigger that starts a recorded celebratory tune playing

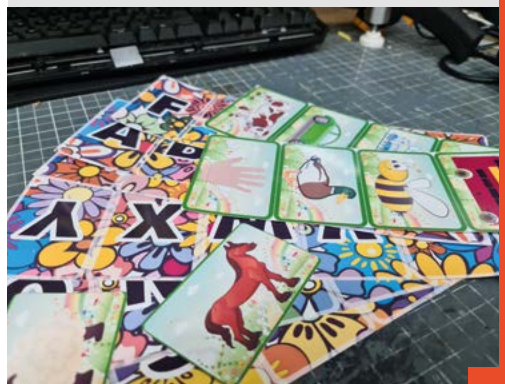
Download the library and copy the **mrfc522.py** file onto your Raspberry Pi Pico. You will also need to copy the project's **main.py** file onto the Raspberry Pi Pico when you are ready to run the game, but first there is some work to do setting up the cards for the game. At the moment, your game cards have not been programmed. They have no data stored on them beyond that which came from the factory. Ideally, you want to make at least one full set of letters A–Z, and extras to deal with words that have double letters (sheep, pool, egg, etc....), and you'll also need to make several picture cards that are programmed with the word represented by the picture on their face. Letters and words are

“ You want to make at least one full set of letters A–Z, and extras to deal with words that have double letters ”

written to the cards at the same location in the card using ASCII. To write the information to the cards, the project uses two Python files: **Pico Alphabet Program.py** and **Pico Word Program.py**. The first of these files lets you program the entire alphabet A–Z automatically one card after another. Each card scanned gets the next letter of the alphabet written to it. The second program allows you to write a whole word or a single letter to the card, by typing the word into the console in Thonny and then pressing the **RETURN** key. □

STICK HERE

Decorating your RFID cards is a little bit trickier than you might expect. Normal RFID cards are made from PVC or similar materials that give off toxic fumes if they are burnt, and do not absorb ink from regular inkjet printers. Dedicated card printers use a sublimation dye that transfers onto the card using heat, but these printers and their consumables are usually very expensive. You can dye the cards with synthetic fabric dye and acetone, or use dedicated PVC paints, but these aren't particularly easy or forgiving techniques to work with. You could choose to use RFID stickers instead of cards, and attach them to cards of your own creation, or you could print designs onto sticky-backed inkjet paper and trim them to the exact shape of the cards once they are stuck in place. This method might sound quite amateurish, but it's a low-cost solution, and the results are actually quite respectable. It also makes it easier to peel off damaged designs and refresh the cards in future. You can also experiment with alternatives to paper, like synthetic felt or EVA foam.



HackSpace

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

Left ♦ The most time-consuming part of this project is making all of the cards. The quickest way is to print on paper and glue onto cards. If you can afford to spend a little bit extra on parts, you could use RFID stickers instead of cards and save a little bit of time

Get fit with *Raspberry Pi*

Hack yourself together some superior smart fitness devices

you can do pretty much anything with a Raspberry Pi or Raspberry Pi Pico, even try and improve your fitness!

Whether you're tracking your fitness more closely, getting a more efficient workout in, or just making a treadmill a little more entertaining, there's definitely some way you can use a Raspberry Pi to improve your fitness journey.

We've gathered some of our favourite projects together to help you get you in tip-top shape. Grab your trainers and let's make some gains.

Fitness *basics*

What to consider before exercising

There are many things to keep in mind when starting to work out, and the information can be a lot to parse as well. We have some tips to get you going, but make sure to do your own research.

Health and fitness

You know your body better than anyone, so if you have any health issues or concerns, you should consider them before exercising.

- › Seek medical advice if you're not sure
- › Listen to your body
- › Don't push yourself through pain
- › Make sure to rest properly
- › Stretch before and after exercising

Perspective

People treat many metrics as proof of health and fitness. Not everybody is the same though, so make sure you don't beat yourself up for not fitting certain categories.

- › Weight is not always a correct measure of health. Muscle is more dense than fat
- › Take BMI (Body Mass Index) results as very rough guidance
- › Different bodies react differently to workouts. There is no one way to get fit

Enjoy your time

Don't get stressed about your exercising! Making it fun is a great way to keep it up for a longer period of time

- › Set small, achievable goals
- › Everyone has to start somewhere
- › Keep an eye on your diet, but don't obsess over it



Exercise bike mod

▲ This works with any old exercise bike – and we do mean old

Breathe new life into that old exercise bike

One of the newer trends with exercise bikes these days is to have your cycling power a video of a wonderful environment that you could be cycling through if you had the ability to do so. The technical side of it is fairly simple – with some kind of coder or sensor, you can track how much you're

pedalling and translate that to video playback or, in this case, graphic generation.

This project by paddywoof uses a Hall effect sensor to keep track of the speed of the bike, feeding into a Python program that renders karst, fjords, and alpine environments as you pedal. It's easy enough to edit your own maps if you have the time too.

Some of the instructions are for a slightly older version of Raspberry Pi OS and the Python that comes with it. However, it should still work just fine.

magpi.cc/bikemod

▼ The graphics may be a little simplistic, but you can always upgrade them or have it control a video's speed





HIIT Workout Trainer

Data analytics for your fitness

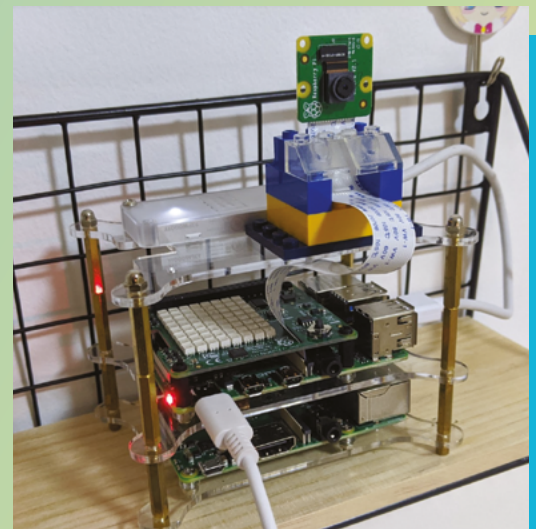
During 2020, when a lot of people were working from home, James Wong decided to improve his workout routine. He also wanted to combine it with his research into machine learning on Raspberry Pi. Hence, combining HIIT (high-intensity interval training) with Raspberry Pi to track his workout and give him useful data on it.

How is your data used? To score yourself against others, naturally, taking advantage of competitive streaks to get you improving the efficiency of your workout. A Coral Edge TPU is used to aid in the machine learning part, improving the

sampling rate to 30 frames per second.

The code is open-source and is readily available on GitHub (magpi.cc/hiitpigit). James believes it should be easily adaptable to other sports or workout types – yoga springs to mind when it comes to tracking body metrics!

magpi.cc/hiitpi

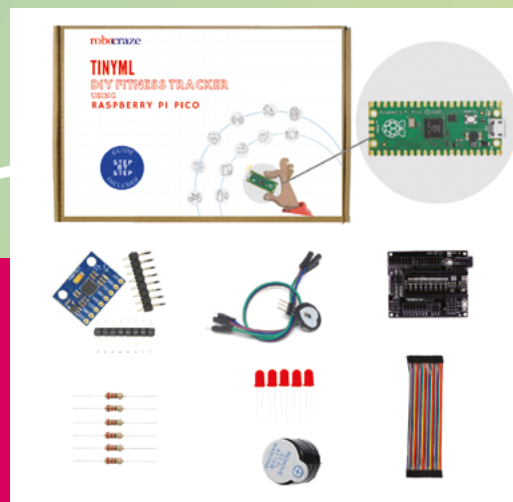


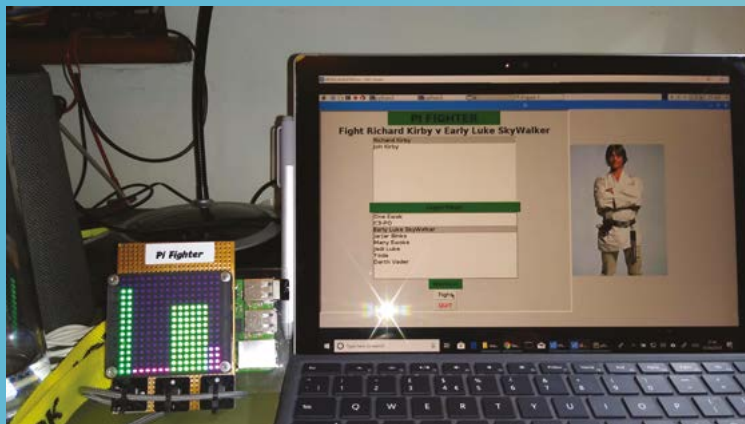
▲ The setup is fairly simple, requiring a Raspberry Pi and a Camera Module – a TPU really helps out, though

Pico fitness tracker

This kit makes use of TinyML, a low-power machine learning language to create a fitness tracker that runs on a Pico or RP2040 board of your own design. It has a pulse rate monitor, among other sensors, and can be built into wearable workout gear. Take a look at it here:

magpi.cc/picofit





◀ Select your character and you can even find some stats on fights

Pi Fighter

Here comes a new challenger

Gamifying your workout has its upsides and downsides. However, having a target to beat, both in terms of a high score and pop culture characters to focus on, can make a workout much more fun. Making use of the kind of technology used by Ivan Dragon in *Rocky IV*, or even punch machines at arcades.

Raspberry Pi acts as an accelerometer to measure the power of your punch to calculate how many hit points (HP) are taken off the character you're fighting – in some cases this means Luke Skywalker

or Darth Vader. A simple LED screen is used to let you know how much more fightin' you'll need to do.

Like all good games, the different characters increase in power, meaning you'll be worn out by the time you take down the Dark Lord of the Sith. Or, maybe you'll need to face him twice – depends how strong you are.

magpi.cc/pifighter



▲ A proper punching bag will be needed if you really want to go at it

Simple entertainment

Some very modern treadmills have built-in screens so you can watch stuff in the gym. We've not seen them everywhere yet though so, with a simple Kodi-powered Raspberry Pi tablet, you can take your media anywhere to watch while you jog or cycle. We recommend using a SmartiPi Touch 2 to build it:

magpi.cc/smartipi2





DIY smart scales

◀ The Balance Board has built-in scales that you can make use of

An encouraging weight tracker for those who need it

Wii Fit was a revolution for fitness video games, spawning loads of motion-controlled workouts. If you don't really like to use your Balance Board much anymore, you can hack it to be a pair of smart scales using a Raspberry Pi and a pencil.

The Wii Balance Board connects via Bluetooth much like Wiimotes, although you'll need a bit of custom code to keep it paired permanently – included with the rest of the code for this project. It does require you to press a button underneath the Balance Board though, achieved by the pencil we mentioned before – a simple solution that won't need much maintenance.

Once you step on it, 250 measurements are taken to create an average (due to electronic noise) and it then sends the info to an IoT service called Initial State which does the tracking. It also sends you texts to let you know your progress, including encouraging messages as well.

magpi.cc/95

▼ The scales text you updates on your progress with encouraging messages



Compute Blade

SPECS

INPUT/ OUTPUT:

Gigabit Ethernet, USB-A, USB-C, HDMI, UART, Compute blade headers, microSD card slot, hardware switchable Wi-Fi, BT, and EEPROM write-protection

SUPPORT:

Raspberry Pi Compute Module 4, NVMe SSD up to 22100, TPM 2.0

POWER:

Power over Ethernet IEEE 802.3at (PoE+) up to 30 W (normal operation 2-8 W)

► Compute Blade ► computeblade.com ► £54 / \$65

Putting Raspberry Pi Compute Module onto a blade for a low-cost and energy-efficient alternative to a rack-based server. **Lucy Hattersley** weighs up the benefits

Compute Blade is a fascinating product developed by Ivan Kuleshov at Uptime Lab (uplab.pro) that puts a Compute Module in a rack-mountable carrier board to create a high-density server for industrial or home use.

Blade servers are the current type of format popular amongst server systems. A traditional rack-mounted server houses a single computer inside a wide chassis (like a tray). Thanks to its small footprint, you can fit up to 20 Compute Blade devices into a rack space.

Compute Blade puts all of the components (storage, RAM, and computer) on a single PCB. In the case of Compute Blade, it uses a Raspberry Pi Compute Module 4 as the main computer.

You can also dedicate each blade to a specific task: web serving, file serving, email support, virtualisation, and so on. So taking one task offline for maintenance leaves the rest up and running.

There are security advantages to separating services via blades, as it reduces the surface area for a hacker to gain access to the system. To this end, Compute Blade can feature a TPM 2.0 chip (Trusted Platform Module) to store cryptographic keys and encrypt your data.

Above all though, blade servers are more energy-efficient than traditional servers. Energy costs are always important for server infrastructure, and this has only become more of an issue recently with higher energy costs. The Arm chips found on Raspberry Pi computers are more energy-efficient than their rivals.

There's a lot here that's interesting and we're delighted that Ivan Kuleshov raised over a million euros on Kickstarter to bring the project to life. Ivan sent us two Compute Blade Dev boards, a fan, and a four-board enclosure for testing.

Check it Blade

Compute Blade is a single PCB that can house a Raspberry Pi Compute Module (magpi.cc/computemodule). This rack-mountable PCB measures 425mm × 255mm × 17.5mm with Raspberry Pi Compute Module 4 connectors and PoE (Power over Ethernet), so you can power and transfer data from a single Ethernet cable.

We used a TP-Link TL-PoE150S for setup testing (magpi.cc/poeinjector), and fitted the two Compute Blade boards to the main Raspberry Pi rack server. Alongside Ethernet sit a USB-C and USB-A port (with a switch to move between the two), an HDMI port, UART, TPM 2.0, microSD card port, M.2 M-key (for an NVMe SSD up to 22110), and a fan unit connector. The Dev unit we tested



► The two Compute Blade boards we had in for testing mounted in our server rack space



“ If you are looking to build an Arm-based server, this is our recommendation ”

features an anodized heatsink secured with T7 hex screws. (Stress testing remained below 65°C).

There are LEDs for SSD, Power, and Activity (these can be disabled in software) and a programmable Edge Button on the front panel. This button acts as a pull tab and can be clicked while the blade is mounted in a chassis. You could use it to perform a restart, for example, or custom script (it is connected to GPIO 20).

Three models are available: Basic, TPM, and Dev (which we are testing here). The Compute Blade Documentation website has more information on the varying specifications (docs.computeblade.com). The Kickstarter campaign offered 4-node cluster cases, 10-inch rack clusters, and 19-inch rack clusters which you can 3D print for your rack.

How sharp is it?

It's a wonderfully designed piece of kit, and we believe it'd be a worthy addition to any server space. The question for us is what would a non-server administrator do with a rack of Compute Blades? It would make a great cluster computer for testing and experimentation, running your own small server enables you to deploy personal cloud services (much more privacy), or it could make a great Pi-hole ad-blocker addition to your network. You could also deploy a Raspberry Pi in the home on your network for any available coding or scripting tasks you want. We set ours up as a custom GPT chatbot that we can log into any time without depending on the OpenAI website.

Compute Blade more than doubled its Kickstarter funding, and the end result is pretty well-built from our testing, and demand is certainly out there. If you are looking to build an Arm-based server, this is our recommendation and we think it's a neat product to have on your home network. 📄

▲ The bare Compute Blade board and a Dev version with Compute Module 4, heatsink, and an M.2 drive attached

Verdict

Compute Blade is a wonderful piece of design that combines the low-cost availability of Compute Module with storage, PoE networking, and a blade format. Perfect for building your Arm cluster or server solution.

10/10

PicoBricks Base Kit

SPECS

FEATURES:

Raspberry Pi Pico, 10 × snap-off electronics modules, protoboard area, reset button

CONNECTIONS:

20 × Grove connectors, 2 × 20-pin breakout headers, battery connector

EXTRAS:

10 × Grove cables, 10 × male-male jumper wires, micro-USB to USB cable, two-piece wooden stand, battery holder (2×AAA), mini breadboard

► The Pi Hut ► magpi.cc/picobricks ► £49 / \$49

This neat educational electronics kit eliminates cable clutter and soldering. By **Phil King**

Aimed mainly at STEM educators, as well as makers in general, PicoBricks is a project development kit based around Raspberry Pi Pico.

As such, it comes supplied with a standard Raspberry Pi Pico board already slotted into headers in the middle of the kit. This is surrounded by ten assorted modules (the ‘bricks’) including sensors, LEDs, potentiometer, motor driver, relay, and a mini OLED display (see ‘The bricks’ box for details).

There’s also a protoboard area, rather than a breadboard (so the holes aren’t already connected in rows), although a mini breadboard is also supplied. Another nice touch is the battery connector for the supplied 2×AAA battery holder, so you can power the kit away from a computer.

Interestingly, the ten modules are already connected to Pico’s GPIO pins via a frame with solder joints. So, you can use them all straight away without having to do any wiring.

Alternatively, you can snap the modules off and wire them to Pico

with Grove connectors (ten supplied). The beauty of this is that they can subsequently be reused separately for other projects, once you’ve finished going through the ones in the PicoBricks e-book.

Electronic learning

The 234-page e-book takes learners through 25 electronics projects of increasing complexity, from blinking an LED to setting up a mini smart greenhouse. Each project features an introductory explanation, wiring diagram, and step-by-step programming guide using the MicroBlocks visual programming system (microblocks.fun), akin to Scratch, which can be run on any type of connected computer – either in the browser or using an offline app. A PicoBricks library can be installed via its menus.

Alternative project code is also provided in the e-book in MicroPython and Arduino IDE (C/C++) form, albeit without any explanation as to how it works. There are also links to the code, although it sometimes differs slightly from what’s in the book.

There are some fun and interesting projects to complete with the PicoBricks Base kit, including a reaction game, magic 8-ball, music player with speed control, thermometer, alarm clock, and colour-matching game. Many of them make good use of the mini OLED. A few community-made projects are also available from the PicoBricks GitHub repo (magpi.cc/picobricksgh).

Missing components

Perhaps the biggest downside is that around half of the projects in the

Verdict

While the Base kit is missing a few components for some projects, it offers a value-for-money and well-organised way for newcomers to learn about electronics and programming.

8/10



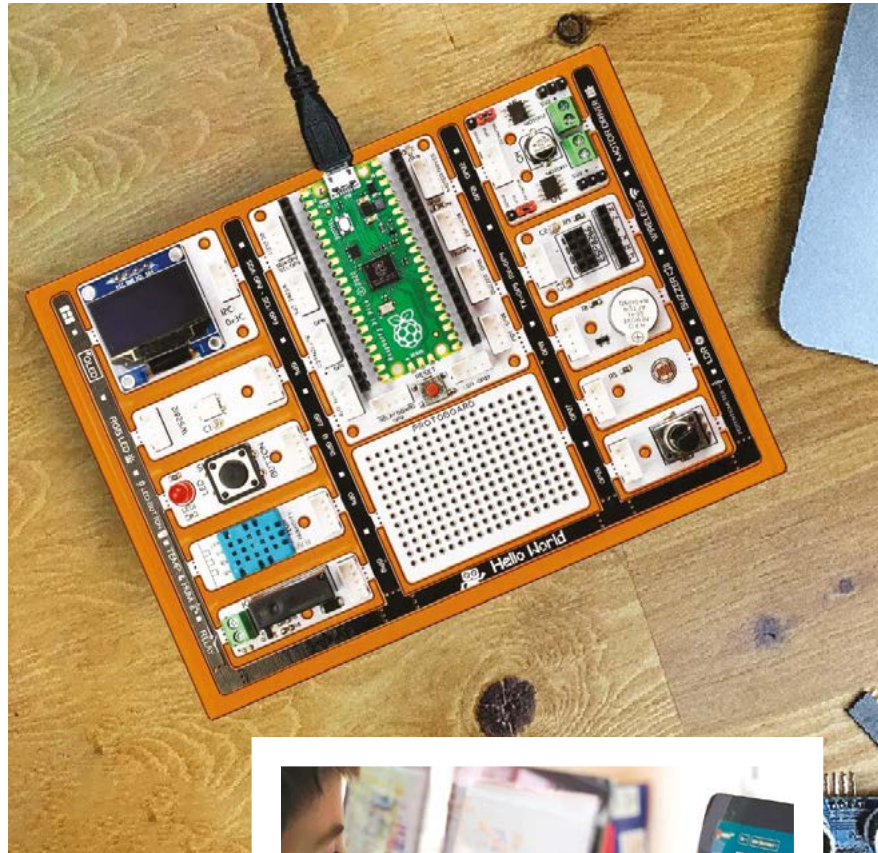
e-book require extra components – such as a 2WD robot car and smart greenhouse kit – that are only included in the more expensive IoT Expert (\$69) or Zero to Hero (\$99) versions of the PicoBricks kit, available direct from **robotistan.com**.

Alternatively, you could supplement the Base kit with standard components already in your toolbox (or buy them separately), such as servos, motors, and ultrasonic distance and PIR sensors. These can be connected with jumper wires to the full female

“ Alternatively, you can snap the modules off and wire them to Pico with Grove connectors ”

GPIO breakout headers flanking Pico, or to the Grove connectors.

In addition, instead of using the optional Wi-Fi and Bluetooth boards (that connect to the PicoBricks IoT module), you could always replace the kit’s removable Raspberry Pi Pico with a Pico W. Either way, once you’ve learnt the basics, there’s the potential to experiment with IoT projects not fully explored in the e-book. 📖



- ▲ The PicoBricks kit fits neatly into a moulded plastic tray
- ▶ The PicoBricks kits are aimed mainly at educators, but can be used for home learning



The bricks

The PicoBricks kit features ten electronics modules that can be used in situ or snapped off

1. 0.96-INCH OLED

128×64 monochrome display to show text, sensor readings, and simple graphics.

2. RGB LED

Addressable WS2812 RGB multicolour LED with output connector.

3. LED & BUTTON

Red LED and push-button.

4. TEMPERATURE & HUMIDITY SENSOR

Standard DHT11 sensor.

5. RELAY

Relay switch with two terminal block connections.

6. MOTOR DRIVER

Dual H-bridge for connecting two DC motors or servos.

7. IOT BOARD

Connections for optional ESP8266 Wi-Fi and HC06 Bluetooth modules.

8. BUZZER

Active piezo buzzer for making simple sounds.

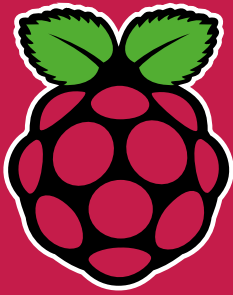
9. LIGHT SENSOR

Light-dependent resistor (LDR) for detecting light levels.

10. POTENTIOMETER

Variable resistor with rotary knob.

THE *Official*

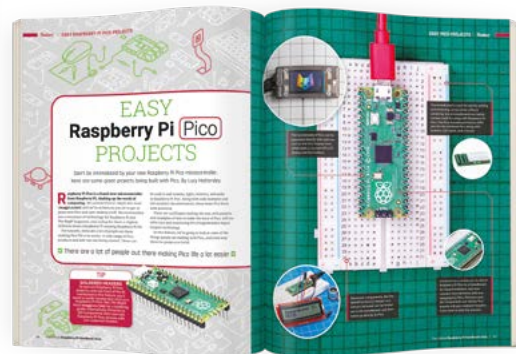
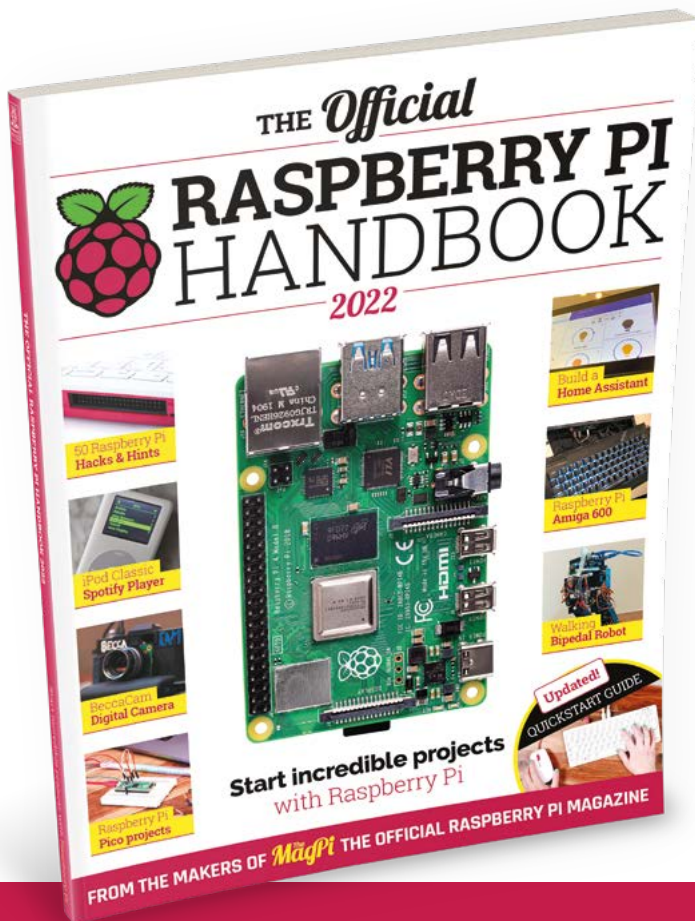


RASPBERRY PI HANDBOOK

2022

200 PAGES OF RASPBERRY PI

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

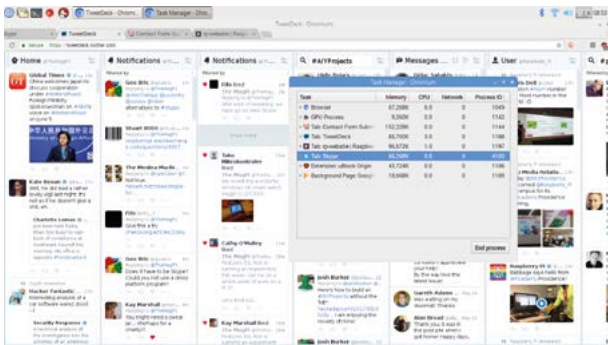


Buy online: magpi.cc/store

10 Amazing: Browsers

The best ways to surf the web on Raspberry Pi

There are many, many browsers available on every kind of operating system, and Raspberry Pi and its OS is no different. We've got the lowdown on ten of the best browsers you can use on Raspberry Pi, whether you're using Raspberry Pi as a desktop, or for a fun project. [M](#)

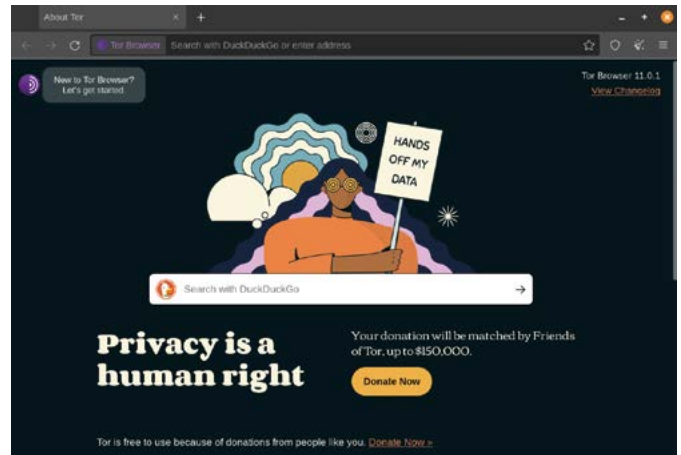


▲ Chromium

Default browsing

It's the browser you have installed on Raspberry Pi OS by default! It's very solid and has specific optimisations for Raspberry Pi OS.

chromium.org



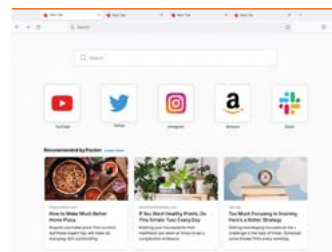
▲ Tor

Ultimate privacy

If you need to browse with complete confidence of your anonymity, Tor Browser uses the Tor network to anonymise your location and IP.

torproject.org

Firefox



Open-source darling

The browser that started the great move away from Internet Explorer has a special version for Raspberry Pi that includes many of the normal version's features.

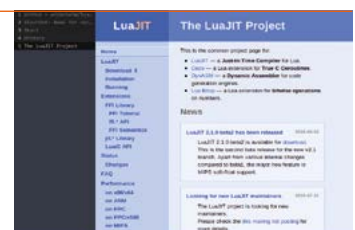
firefox.com

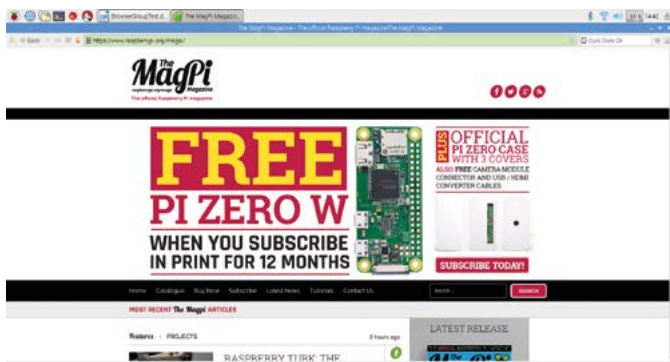
Luakit

Fast and customisable

Very quick and very configurable. Luakit is a bit of a minimalist browser that power users may prefer.

luakit.github.io





▲ Midori

Lightweight excellence

Midori is a venerable lightweight browser alternative which now has extra security and a great Android spin-off.

magpi.cc/midori



Lynx

Who needs graphics?

If you find the internet is too bloated with images, then this text-based browser may be for you.

magpi.cc/lynx

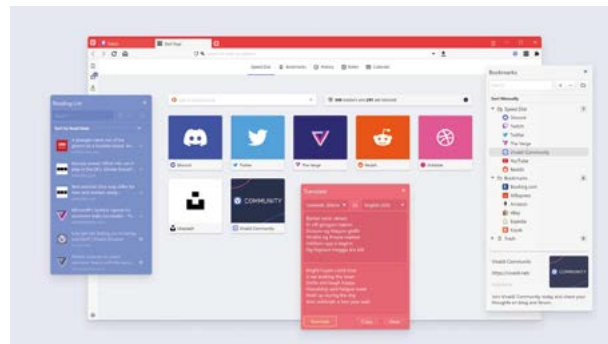


▲ Puffin Cloud Learning

Cloud rendering

This unique browser renders content on encrypted cloud servers which are then streamed to the web browser. It can be much faster than regular browsers.

puffin.com



▲ Vivaldi

Opera reborn

A much newer browser that recaptures many of the functions of older Opera browsers. It's mostly open-source too.

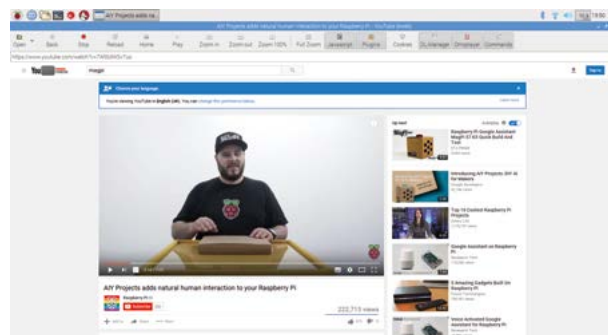
vivaldi.com

Dillo

Retro minimalism

Designed for older systems, Dillo is an incredibly minimalistic bit of software that is about as small as a browser can be while still working.

dillo.org



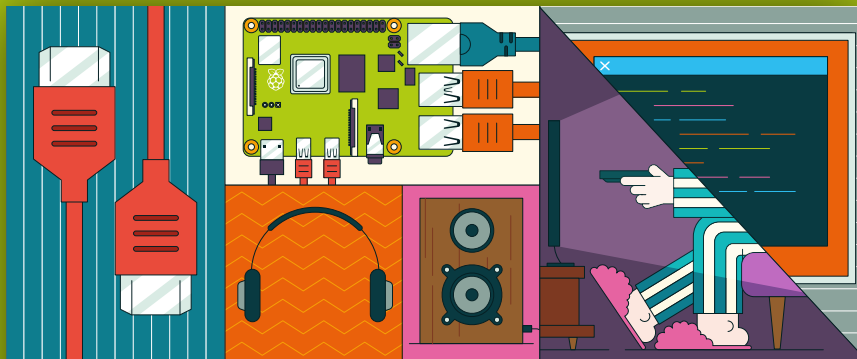
▲ Kweb

Kiosk web browsing

An alternative to the default browser to turn Raspberry Pi into a web kiosk, perfect for public places.

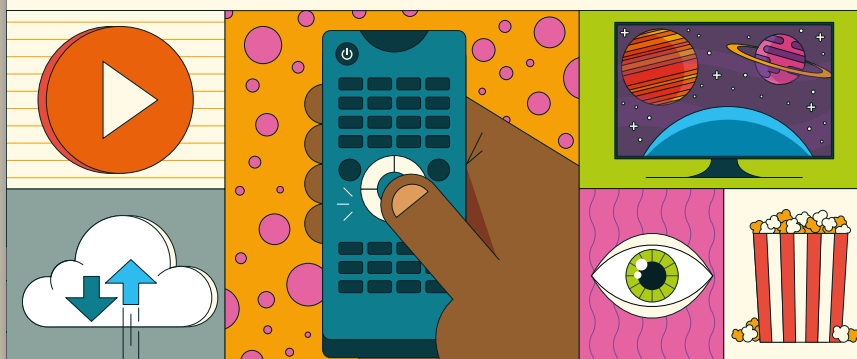
magpi.cc/kweb

Your FREE guide to making a smart TV



BUILD A RASPBERRY PI MEDIA PLAYER

Power up your TV and music system



FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

magpi.cc/mediaplayer

Learn: Sense HAT

This sensor-packed board helps you explore the world, says **Rosie Hattersley**

Sense HAT

AUTHOR

Raspberry Pi

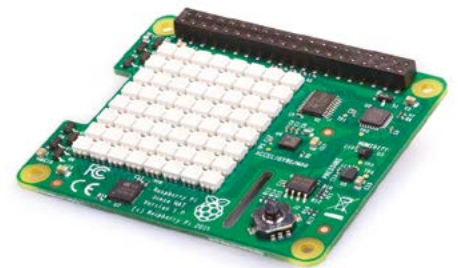
Price:
free


magpi.cc/sensehat

The Raspberry Pi Sense HAT is an add-on board for the Raspberry Pi, designed to provide a range of environmental sensors and input/output capabilities. It was originally developed for the Astro Pi mission, a project by the Raspberry Pi Foundation and the European Space Agency (ESA)

to send Raspberry Pi computers to the International Space Station (ISS) for educational purposes.

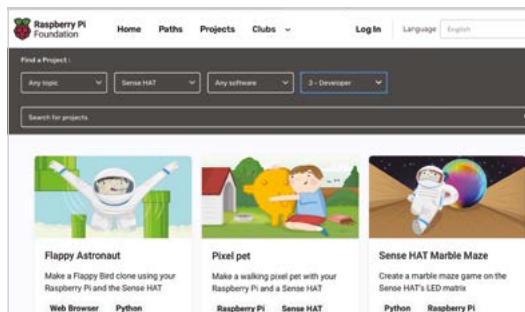
Start out with the Raspberry Pi Sense HAT product page. Here you will find more information on the components, a link to buying a Sense HAT, along with



links to Astro Pi, the Python library, and the Sense HAT Product Brief. 

Projects

Movement and LED-based projects to try for yourself



RASPBERRY PI SENSE HAT PONG

Paying homage to the original computer game, this walkthrough on how combine a Sense HAT and Raspberry Pi, plus some Python code to create classic Pong, is bound to appeal to retro gamers.

There's even a link to a video of the original game being played on an oscilloscope.

► magpi.cc/sensehatpong

SENSE HAT MARBLE MAZE

Orientation sensors on the Sense HAT detect which way the virtual

marble (in the guise of coloured LEDs) is travelling in this tricky maze puzzle. Have fun following the setup guide, before setting a timer and challenging your friends to escape the maze in the fastest time.

► magpi.cc/sensehatmarble

NIGHT CLOCK

Lorna Jane's tutorial shows you how the LED matrix can be used as a useful and quirky display that can be used as an at-a-glance bedside time check, without having to turn on the light or phone display.

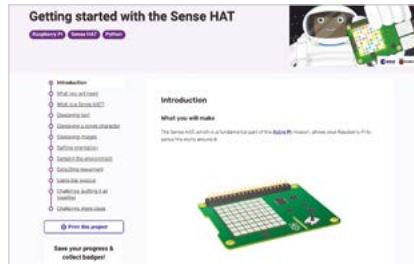
► magpi.cc/sensehatnightclock

Raspberry Pi Sense HAT

AUTHOR Raspberry Pi Foundation

Price: free
magpi.cc/gettingstartedsensehat

Now we've whetted your appetite about things a Sense HAT can do, head to this useful guide to set up. As this guide explains, the Sense HAT contains a number of on-board sensors to measure temperature, humidity, colour, movement, and orientation, and also has an LED grid to display the results of your investigations. With its origins as a sensing device destined to go into space with the European Space Agency, and as a teaching aid as part of the



Astro Pi project, there's little the Sense HAT can't do. This tutorial walks you through displaying text, images, and measuring the orientation of the Sense HAT device, before detecting movement using the joystick for input, and putting it all together to create your own projects that sense and react to their surroundings. magpi.cc/sensehatAPI

Sense HAT Documentation

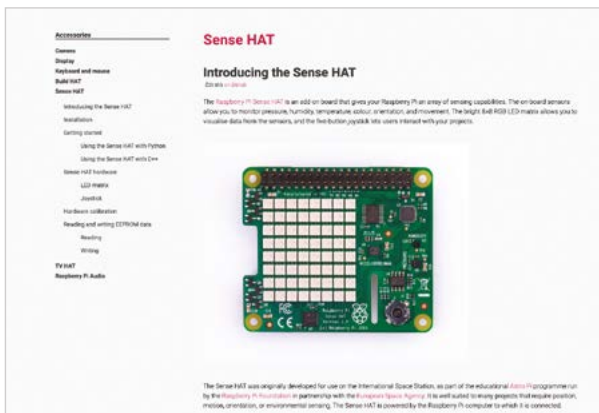
AUTHOR Raspberry Pi

Price: free
magpi.cc/sensehatdocs

The official documentation will walk you through some of the more advanced aspects of Sense HAT and how to use this Raspberry Pi accessory with

different computer languages, including C++ as well as Scratch and Python. Each of the sensors (gyroscope and magnetometer and barometric pressure sensor among them) is covered, along with calibration for the magnetometer and accelerometer. The official documentation also covered reading and writing the EEPROM data. Also take a look at the corresponding Sense HAT Python documentation, linked to from the official documentation. Here you will find Python API examples and the Sense HAT API reference guide, ideal for all your coding needs:

magpi.cc/sensehatAPI [magpi.cc/sensehatAPI](#)



YouTube Videos

Easy setup guides you can watch at your leisure



RASPBERRY PI SENSE HAT PIXEL TUTORIAL

Carrie Ann Philbin shows how to set up the pixel display on your Sense HAT and create a pixel pet. The avatar pet moves when you shake the device, demonstrating one of its many sensing capabilities.

magpi.cc/sensehattutorialyt

ELLORA JAMES - SENSE HAT

Ellora James is a great video blogger who explains clearly how to go about setting up a Sense HAT for the first time, and provides a video tutorial that follows the instructions and code on the Raspberry Pi website. It's ideal viewing if you want to take your first steps into using HATs. Ellora also has plenty of other useful Raspberry Pi YouTube guides.

magpi.cc/sensehatej

THE RASPBERRY PI SENSE HAT IS AWESOME!

ETA Prime takes you through the LED matrix and all the sensors, focusing on the parts and capabilities of Raspberry Pi Sense HAT.

magpi.cc/sensehateta

RETRO GAMING

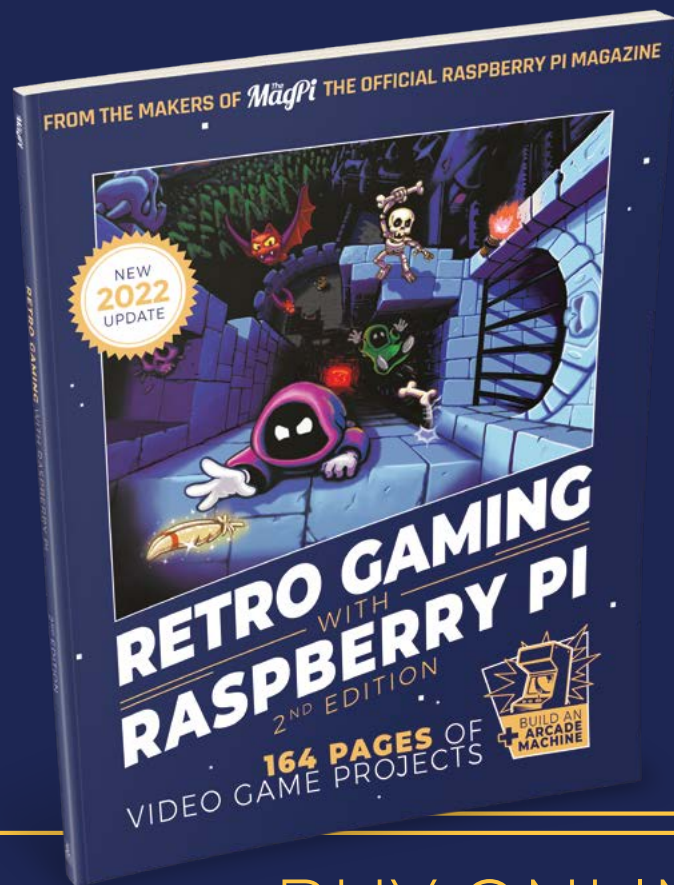
WITH

RASPBERRY PI

2ND EDITION

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

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



BUY ONLINE:
magpi.cc/store



Brian Jepson

A long-time programmer, maker, and tinkerer, meet the new publishing director for Raspberry Pi

- Name **Brian Jepson** | ➤ Occupation **Publisher**
- Community role **Maker** | ➤ URL **jepstone.net**



▲ Brian helps out at the Rhode Island Computer Museum

“I grew up in the smallest state in the US (Rhode Island) during the golden age of Radio Shack,” Brian Jepson tells us. He’s the new publishing director here at Raspberry Pi Towers, but he’s an old hand at making.

“I’ve been a lot of things over the years, but if we go back to the beginning, I’d say I’m a computer programmer and tinkerer,” Brian continues. “I got my start programming on a somewhat customised Atari 400 with Atari BASIC, Forth, and a smattering of 6502 assembler. I kept up with that kind of thing over the years, and ended up programming for a time, then writing, and eventually editing and publishing, but always with a dash of programming on the side.”

His previous work in publishing includes O’Reilly Media and Make Magazine – he even helped run a Maker Faire in his native Rhode Island.

What is your history with making?

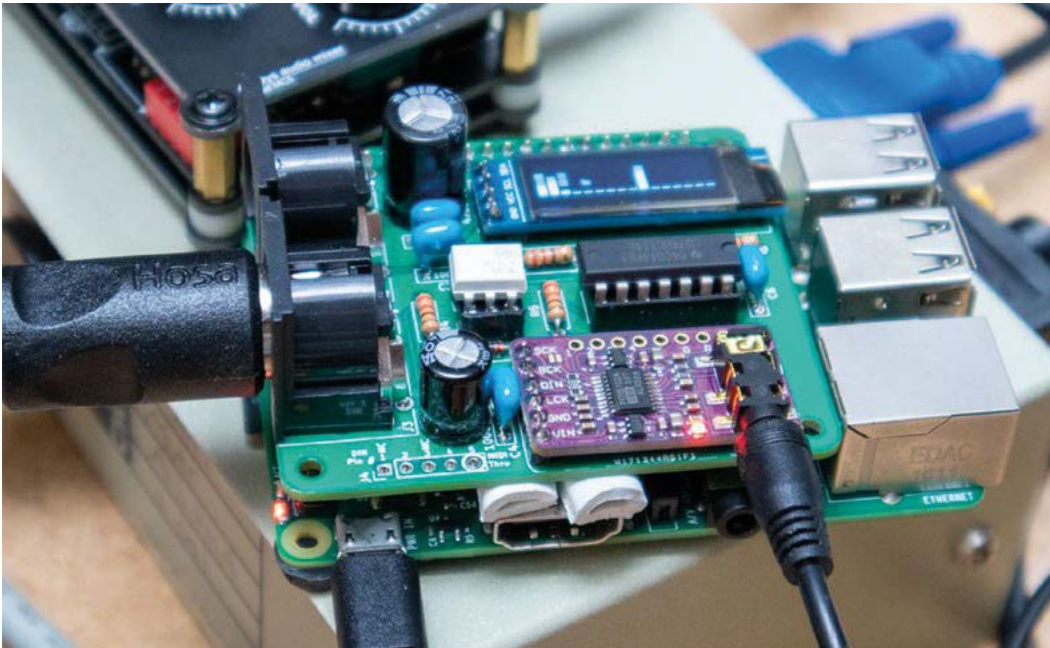
I had one of those 150-in-1 project kits where you connected wires using spring terminals. I was terrible with a soldering iron until I was much older. My

maker interests ended up getting pushed to the side as I spent more time with computers. Immediately preceding my time at university, and really throughout my life, I fell in with an unjuried, uncensored arts organisation (**as220.org**) that very much embraces the DIY spirit. In those years, I made zines, posters, performance art, but very little visual art outside of printed works. I went away to New York for a few years, came back, and stayed involved with AS220 as a volunteer and occasional board member. Over time, AS220 itself has grown a lot, and now operates three mixed-use buildings in downtown Providence, RI, supporting everything from live-work studios to a vibrant youth program.

In the past several years, I’ve become involved in the Rhode Island Computer Museum, which has allowed me to fuse my interest in computers (especially old computers) with making.

Where did you learn about Raspberry Pi?

Raspberry Pi was such big news when it launched that a better question would be ‘where didn’t I learn about Raspberry Pi?’



◀ A custom Raspberry Pi MIDI synthesizer

Working on the Make books portfolio at the time meant this was huge news. We managed to get a book out by January 2013, co-written by Raspberry Pi's own Matt Richardson, as well as Shawn Wallace. Shawn is a fellow Rhode Island maker/artist who has been part of my adventures going back to the campus daily and literary magazine, through AS220 (where he was the founding director of the Providence Fab Lab), and into nearly every Rhode Island maker event I've been part of.

What are some of your favourite things you've made with Raspberry Pi?

I rely heavily on Raspberry Pi to keep old computers up and running. I recently got a Mac Quadra up and running, and I wouldn't have been able to do it without PiSCSI, a combination of hardware and software that allows a Raspberry Pi to emulate a variety of SCSI devices. It's a much quieter hard drive than what was originally in the Mac, and it's easy to move files to and from the disk images, mostly eliminating the need to fuss with floppies. I also use a Raspberry

Pi with `mt32-pi`, which turns a Raspberry Pi into a MIDI synthesizer that works just like the Roland MT-32, and various General MIDI synthesizers that can be used with old games and other software. I have built two `mt32-pi` boards and use them

with Atari STs. In addition to the many games that use MIDI, there are many music apps for the Atari ST that work with MIDI, such as Steinberg Cubase and C-Lab Notator. Many of these old music apps hold up pretty well to this day. [M](#)

“ I rely heavily on Raspberry Pi to keep old computers up and running ”



▲ An early Rhode Island Mini Maker Faire, circa 2011.

Credit: Carolyn Fahm, of Greenwich Meantime Photography

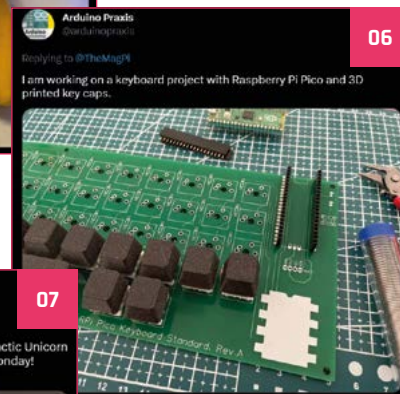
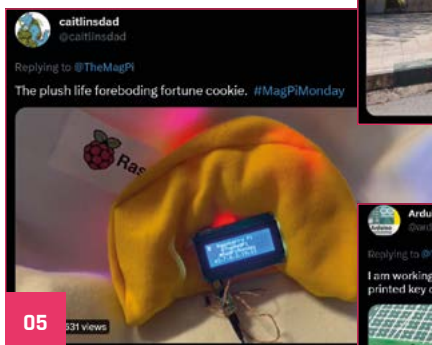
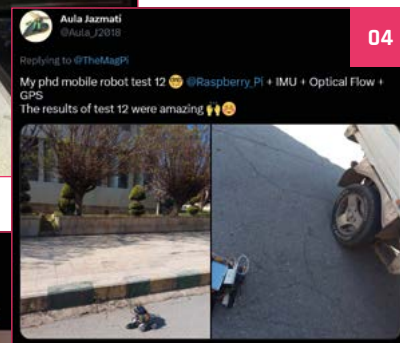
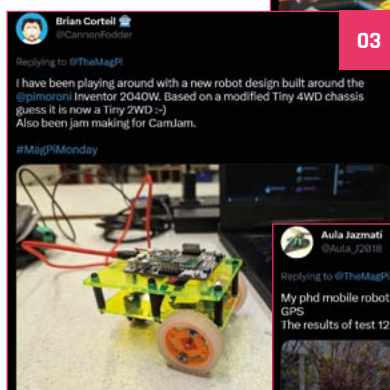
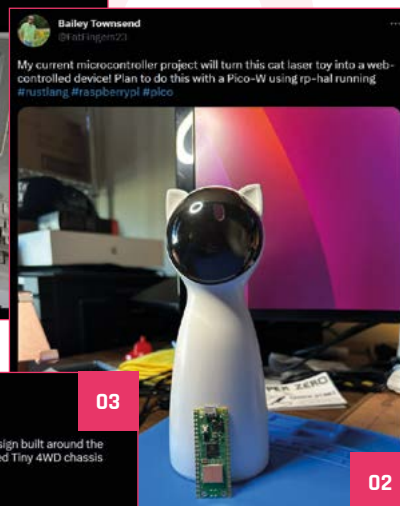
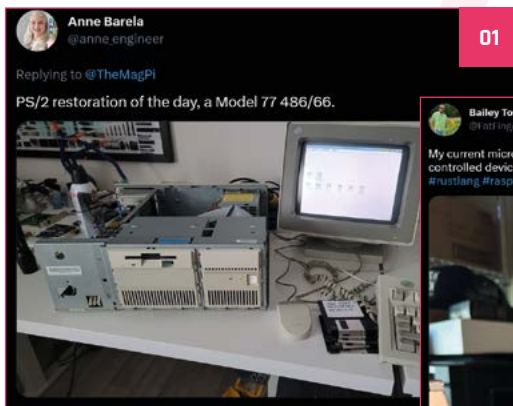
MagPi Monday

Amazing projects direct from our Twitter!

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

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

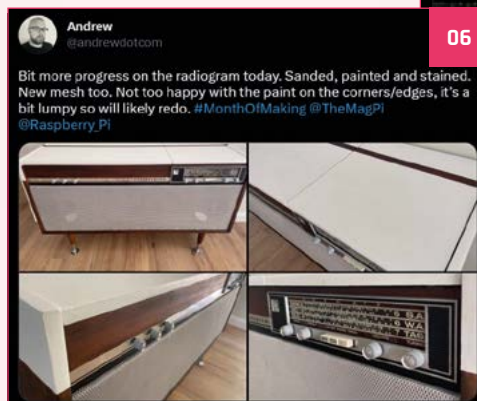
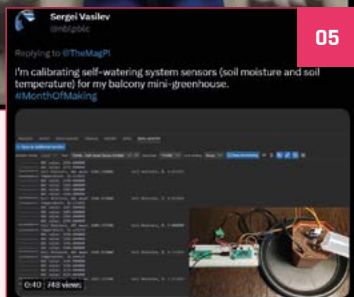
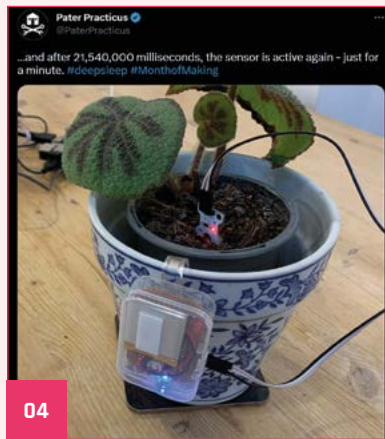
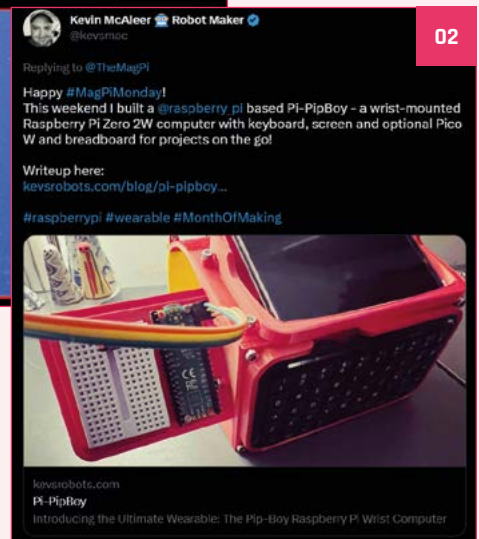
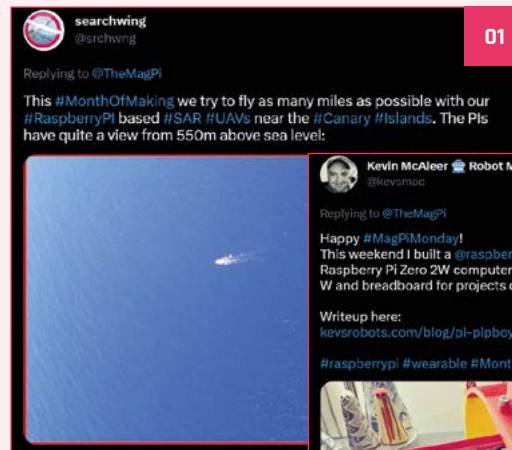
01. Classic computer maintenance is essential to keep them running for preservation
02. This is a great looking case to use for a fun sounding project
03. Tiny 4WD has evolved into a smaller, more compact form
04. Another excellent robot test here with Aula
05. A plush fortune cookie may be less tasty but it is just as wise
06. The most custom way to build a keyboard – 3D-print it
07. This looks great in motion – we love all the projects we've seen using the Galactic Unicorn



#MonthOfMaking


Y'all got making online in March. Here's some of the cool stuff we saw

01. Raspberry Pi has been pretty high but rarely under its own power. 550 metres is very impressive
02. Kevin brought back a classic Raspberry Pi project, this time powered by Pico
03. 10 March is Mario day (Mar10 in US notation), and Igor celebrated by... dismembering him?
04. Peter has been further refining his planter watering system over the #MonthOfMaking
05. Another excellent plant watering system, this one using multiple Pico
06. Some very retro tech here getting the upcycling treatment – we love it
07. We really like this visualisation of joystick movement Dr Footleg has been working on. We'd love to see it applied at the EVO fighting game tournament



Events in pictures

Community and official events in the wild

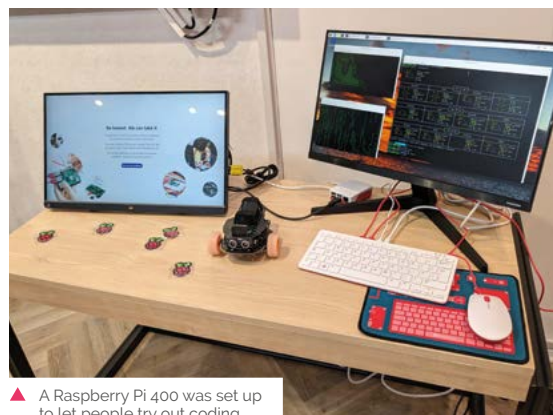
This month, a Raspberry Pi pop-up store opened in Southampton, and we went along to see just how it went down. According to the shop staff, different areas tend to buy different kinds of products at the pop-ups, with Raspberry Pi 400 kits and Debuggers going down very well. 

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

PAGE 92



▲ Visitors told us all about the amazing projects they'd be making



▲ A Raspberry Pi 400 was set up to let people try out coding



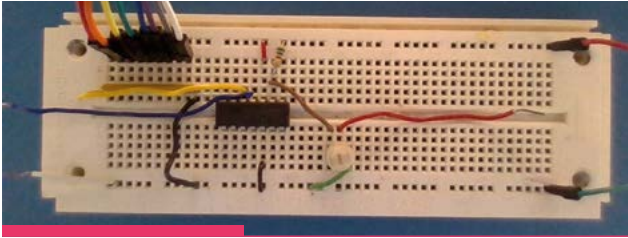
▲ The queues, before the store opened, wrapped around a large portion of West Quay!



▲ The modest stock selection went very quickly

Best of the rest!

Other great things we saw this month




Cheap Oscilloscope

Ed Mikkelsen emailed us about this cool project. "I wanted to build my own oscilloscope with inexpensive parts, and do my own programming. I also wanted to see how fast I could make it. I first programmed the oscilloscope in C++ using GTK for the graphics and achieved a conversion rate of 8 usec. I also programmed it Python with a considerably slower conversion rate because Python is a scripting language. With C++ I have also added a second trace, zero crossing detection, and automatic refreshing. All in all, loads of fun."

► magpi.cc/github129



Internet/Local Network Monitoring Service

A great network monitoring setup was sent to us by John Loop, which he described as such: "Completely self-contained performance monitoring setup. Does not rely on third party servers. GREAT use for an older Raspberry Pi 3B/3B+ lying around. All it requires is downloading the image and burning an SD card. Comes up running." 

► magpi.cc/imonitorg

Together we can make a difference

Give young people the opportunity to learn about technology

The Raspberry Pi Foundation enables young people to realise their full potential through the power of digital technologies, but we can't do this work without your help. Your support helps us give young people the opportunities they need in today's world. Together we can offer thousands more young people across the globe the chance to learn to create with digital technologies.

Generous donations from organisations and individuals who share our mission make our work possible.



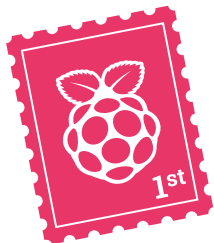
DONATE NOW



Donate today to make a difference: rpf.io/magpidonate

 **Raspberry Pi**
Foundation

Your Letters



Readall solution

In the latest *The MagPi* correspondence, Marian asked “you could print articles on the following: gpio readall - the installation of and usage of as a means of checking the status of inputs and outputs.”

When WiringPi was deprecated, I wrote a Python replacement for gpio readall. I have since written versions for many of the popular libraries (including my own pi_gpio).

This has the advantage of displaying the ACTUAL programmed GPIO function (if changed) and pulls (on Raspberry Pi 4).

I have been playing recently with the gpiochip interface (included in recent kernels) and realised that the raspi-gpio debug tool actually has all the details needed. raspi-gpio should be included in all recent Raspberry Pi OS, but if not you can install raspi-gpio from the repository.

This version of my program (which I call gpioreadall) reformats the output of raspi-gpio into a user-friendly format, similar to the WiringPi gpio readall.

This produces output identical to Figure 1, but is independent of any other libraries/daemons.

The code can be downloaded from here: magpi.cc/gpioreadall.

Ian via email

This is a fantastic script and, yes, raspi-gpio is included on all Raspberry Pi OS installs. For Marian last week, and anyone else who wants to keep an eye on their GPIO pins, this is a great bit of code to use!

► Figure 1: An example of how the script displays GPIO readings

BCM	Name	Mode	V	Pi 4B Board	V	Mode	Name	BCM
	3.3v		1	2			5v	
2	GPIO2	IN ^	1	3	4		5v	
3	GPIO3	IN ^	1	5	6		GND	
4	GPIO4	IN ^	1	7	8	1	ALT5 TXD1	14
	GND			9	10	1	ALT5 RXD1	15
17	GPIO17	IN v	0	11	12	0	IN v GPIO18	18
27	GPIO27	IN v	0	13	14		GND	
22	GPIO22	IN v	0	15	16	0	IN v GPIO23	23
	3.3v			17	18	0	IN v GPIO24	24
10	SPI0_MOSI	ALT0	0	19	20		GND	
9	SPI0_MISO	ALT0	0	21	22	0	IN v GPIO25	25
11	SPI0_SCLK	ALT0	0	23	24	1	OUT GPIO8	8
	GND			25	26	1	OUT GPIO7	7
0	GPIO0	IN ^	1	27	28	1	IN ^ GPIO1	1
5	GPIO5	IN ^	1	29	30		GND	
6	GPIO6	IN ^	1	31	32	0	OUT GPIO12	12
13	GPIO13	IN v	0	33	34		GND	
19	GPIO19	IN v	0	35	36	0	IN v GPIO16	16
26	GPIO26	IN v	0	37	38	0	IN v GPIO20	20
	GND			39	40	1	IN ^ GPIO21	21

Pop-up stock

Unfortunately I live a bit too far away from Cambridge to head to the Raspberry Pi Store there, but I will be able to get to some of the pop-ups this year! What kind of stock should I expect to see at the pop-ups?

Mahiru via Twitter

A lot of Raspberry Pi hardware is available at the pop-ups. However, each pop-up seems to sell out of specific items faster than others. Kits with Raspberry Pi 4 and Raspberry Pi Zero 2 W can be found in the pop-ups, along with Raspberry Pi 400 and desktop kits, add-ons like the Camera Modules, some of our books, and loads of other accessories and general Raspberry Pi merch!



▲ If you miss stock of a particular product on one day, the pop-ups try and restock the following day!

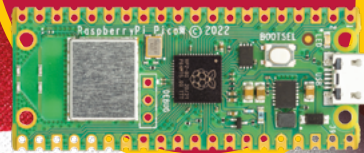
Contact us!

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

USA SPECIAL! 6 ISSUES FOR \$45



FREE RASPBERRY PI PICO W



Subscribe online:
magpi.cc/subscribe

Price is charged at £35 sterling. The dollar price will depend on the exchange rate. Six issues and free Pico W for £35 is also available in Canada and Europe. Subscription is for the next six issues and does not renew automatically. This is a limited offer. Offer subject to change or withdrawal at any time.



Community Events Calendar

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

01. Exeter Pi Jam

- 📅 Saturday 6 May
- 📍 Exeter Central Library, Exeter, UK
- ▶ magpi.cc/exeterjam129

First Saturday of every month, a group of enthusiasts of all ages bring their projects or just come along to make, learn, and talk. There is a range of equipment for people to try or use to work through projects. Come and see, and have a chat.



02. Melbourne Raspberry Pi Meetup

- 📅 Sunday 7 May
- 📍 Docklands Makerspace and Library, Melbourne, AUS
- ▶ magpi.cc/mrpm129

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 20 May
- 📍 FibreHub, Pool, UK
- ▶ magpi.cc/ctm129

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. Hybrid Raspberry Pint

- 📅 Tuesday 30 May
- 📍 Microsoft Reactor London, London, UK
- ▶ magpi.cc/pint129

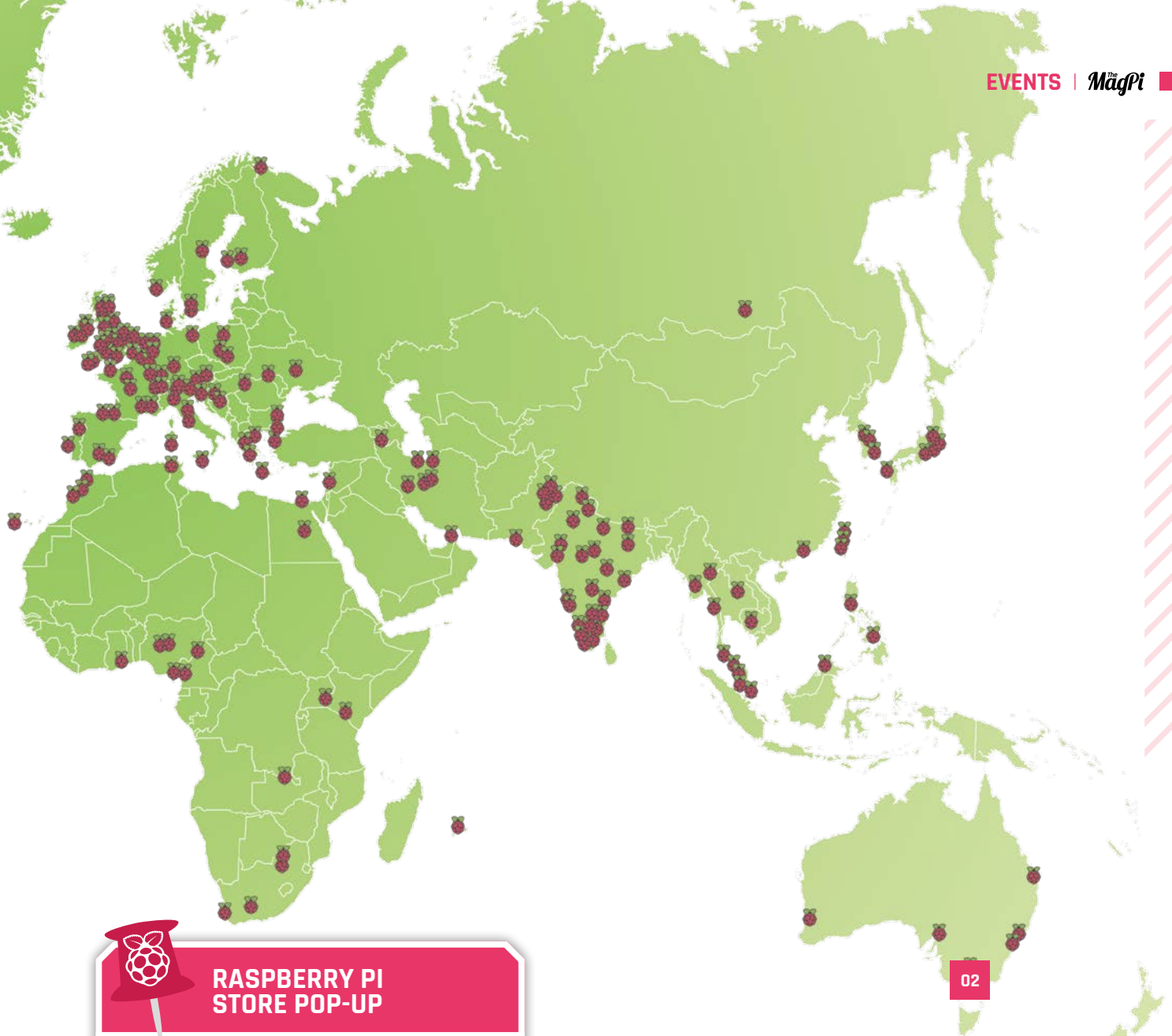
Raspberry Pint is a meetup for sharing digital making experiences. Most presentations are about building personal or professional projects with Raspberry Pi and other maker technology. You can also learn skills and techniques such as website design, PCB design, software development, 3D printing, and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

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

Raspberry Pint



Facebook: [Raspberry Pint](https://www.facebook.com/RaspberryPi)
Twitter: [@RaspberryPi](https://twitter.com/RaspberryPi)



RASPBERRY PI STORE POP-UP

Where can you find Raspberry Pi next?

- Next location **Raspberry Pi Store pop-up Birmingham**
- Where **Bull Ring, Birmingham, UK**
- When **Friday 28 May and Saturday 29 May**

At this Raspberry Pi Store pop-up in Birmingham, 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/popup2023bham



HackSpace

TECHNOLOGY IN YOUR HANDS

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



SUBSCRIBE AND
SAVE UP TO
35%
on the cover price



ISSUE **#66**

OUT NOW

hsmag.cc



WIN ONE OF FIVE

RASPBERRY PI GLOBAL SHUTTER CAMERAS

The new Raspberry Pi Global Shutter Camera is perfect for taking photos of fast action for sports photography, or even computer vision applications that need to take accurate photos fast. We have five to give away!



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

Terms & Conditions

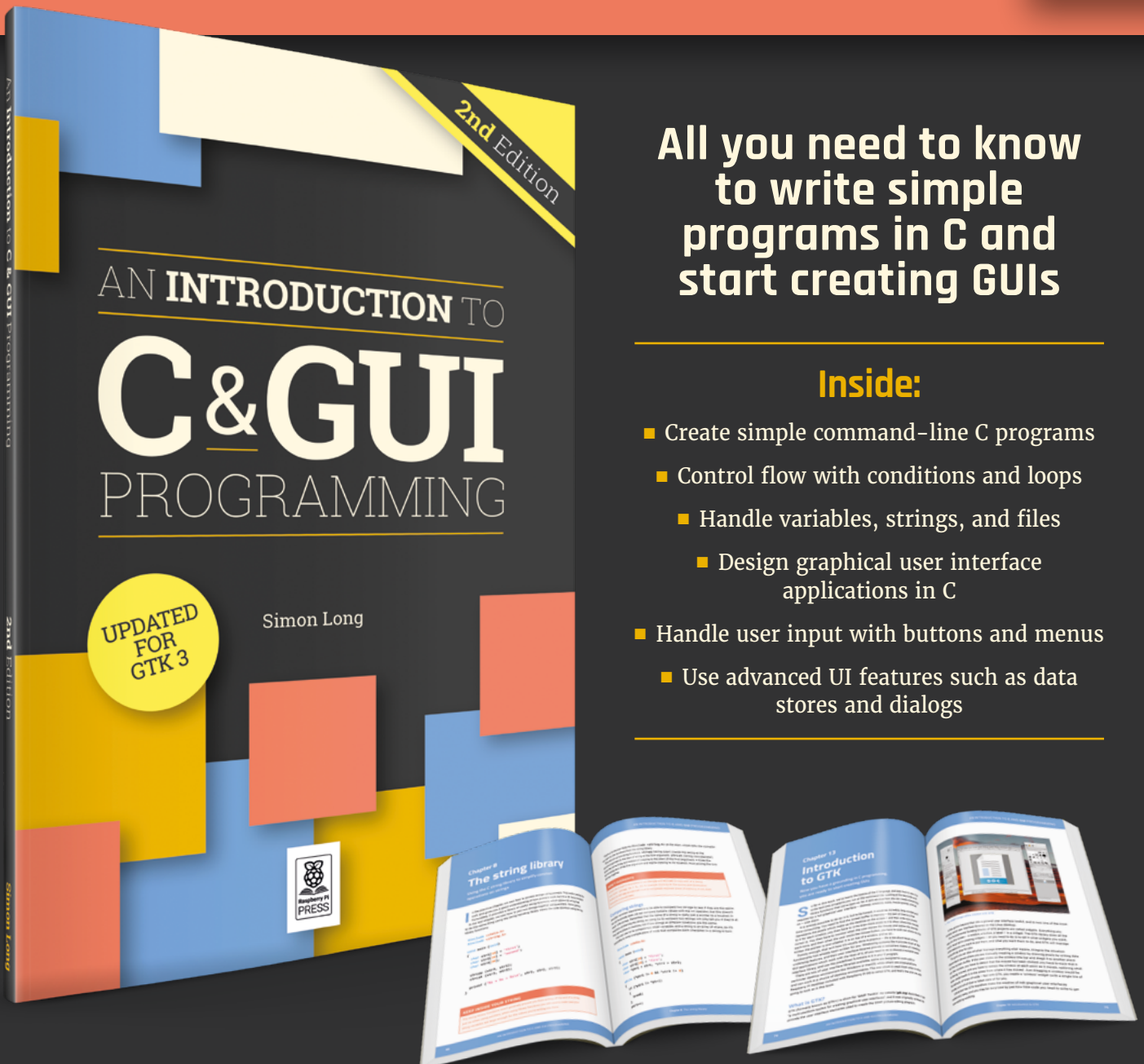
Competition opens on **26 April 2023** and closes on **25 May 2023**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. 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.

AN INTRODUCTION TO C&GUI PROGRAMMING

All you need to know
to write simple
programs in C and
start creating GUIs

Inside:

- Create simple command-line C programs
- Control flow with conditions and loops
 - Handle variables, strings, and files
 - Design graphical user interface applications in C
- Handle user input with buttons and menus
 - Use advanced UI features such as data stores and dialogs



Buy online: magpi.cc/cgui

Linux &
The Command Line
SPECIAL



BREW YOUR OWN

with Raspberry Pi

THE MAGPI #130
ON SALE 25 MAY

Plus!

Move to an
open-source GPT

Replica vintage
computers

Crack the
command line

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, PJ Evans,
Rosemary Hattersley, Nicola
King, Phil King, Andrew Lewis,
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



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

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

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



Taking a break

Resting can be important, even when it comes to making, says **Rob Zwetsloot**

This issue, I wrote a feature about getting fit with Raspberry Pi and it reminded me of the fitness concept of how rest is equally important as actually working out. In terms of your body, exercising depletes you of resources and damages muscle fibres – resting allows you to get your energy back, and for the muscles to repair stronger than before.

On a basic level, making sure to take breaks while making can be a safety precaution. Nobody should be soldering while they can barely stay awake, after all. More importantly, sometimes you need to put some space between yourself and a project to allow your brain to rest. Constantly hitting it against a problem you're having is hardly a good way to solve it.

Sleep on it

I once read a (possibly apocryphal) story about how a scientist fell asleep while trying to figure out how they got some specific results – while asleep they dreamt about a solution and, after waking up, found out it was correct.

I've never quite had a eureka moment in my dreams like that myself, but there have been plenty of

times when a bit of engineering and/or code have stumped me until I looked at it with fresh eyes the next morning.

Even writing stuff for the magazine can benefit from a break. Sometimes an angle or a subject isn't quite making sense and that little bit of time apart helps focus my thoughts. And in terms of focus, the Pomodoro method of 30 minutes of work with a five minute break also really helps

“ As well as short breaks, sometimes you need to just take time off a hobby. Burnout is very real ”

me. We had a project about making your own Pomodoro timer in issue 103 (magpi.cc/103) which I should make. However, I've just been using my phone and its Focus feature.

At the other end of the spectrum, all-nighters really are overrated I feel, although as I near my forties,

they're a little harder to do anyway. It means I'm doing them much less though.

On hiatus

As well as short breaks, sometimes you need to just take time off a hobby. Burnout is very real, whether it's with work or with something you're doing for fun, and you don't want to ruin your relationship with your favourite hobby because you forced yourself to keep doing something. When taking breaks from one hobby in the past, I've focussed on another hobby instead. Flexing a different part of your mind and/or skill set is always good for growth – and can even aid you in other hobbies. Although, sometimes, you have that con crunch and need to get your prop working by any means necessary. Just make sure not to do any soldering in the hotel room; I speak from experience. It's not a very suitable space. 📺

AUTHOR

Rob Zwetsloot

Rob actually currently has a decent hobby balance for the moment, although there's always one project being neglected...

magpi.cc

HiPi.io

HIGHPI PRO

•———— The new case from the HiPi.io team ————•



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

Available at these great Pi stores:

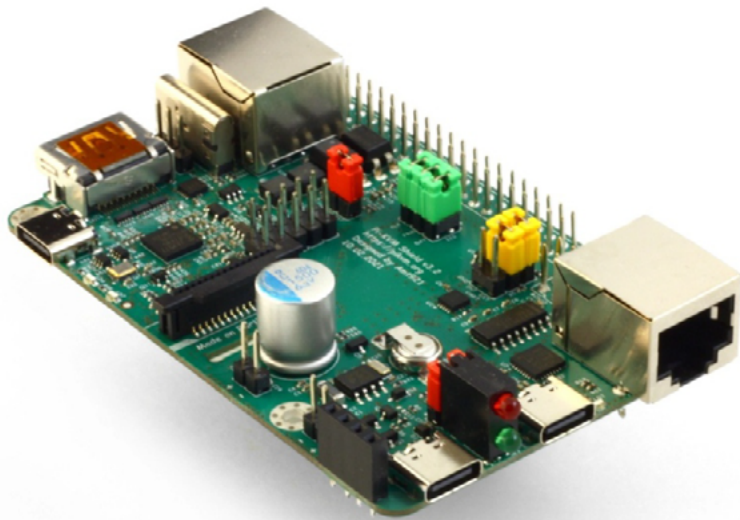


Contact your favorite Pi store if it's not listed here

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



Reseller suggestions and inquiries:
wholesale@hipi.io