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

The MagPi

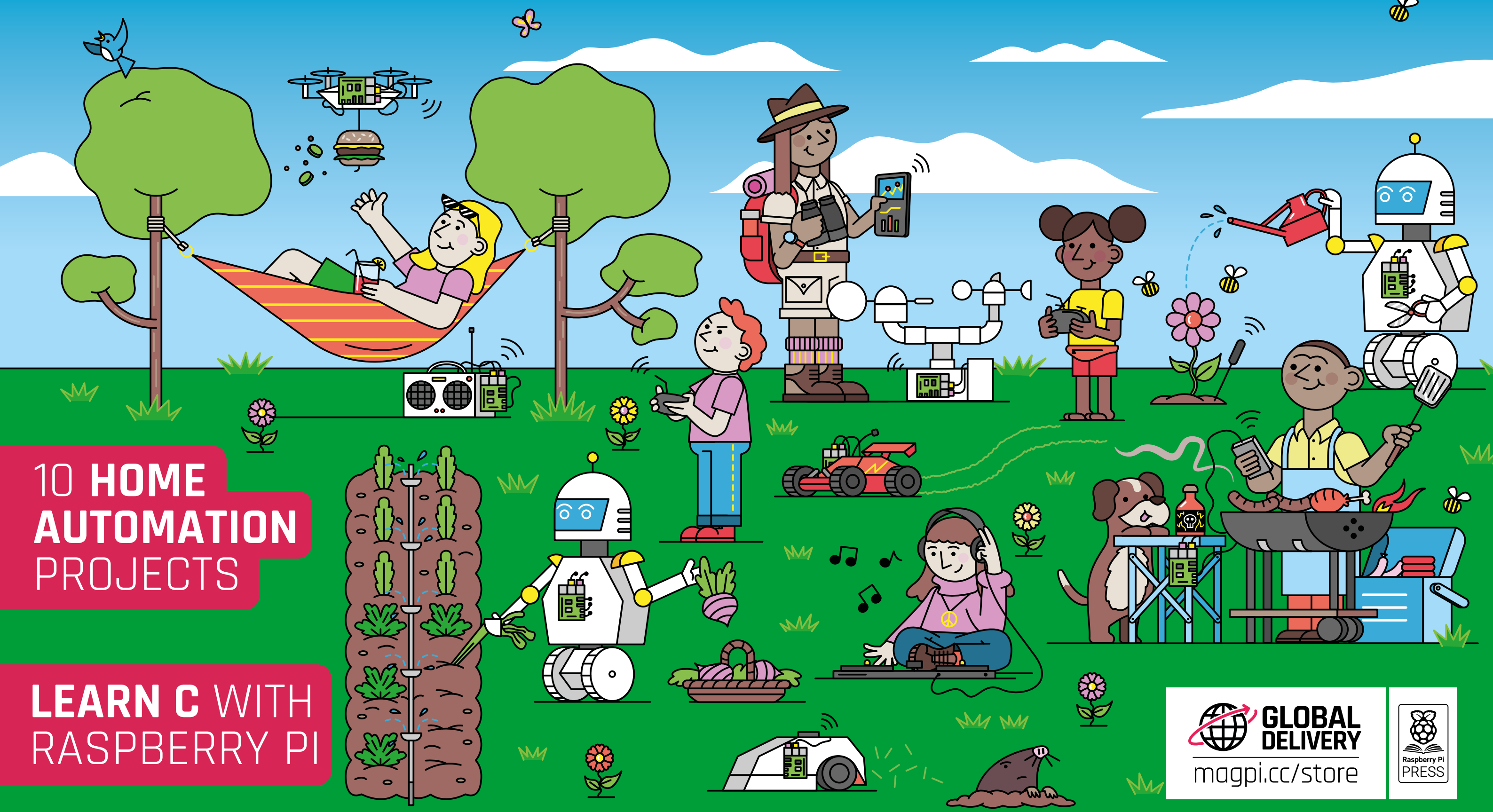
Issue 117 | May 2022 | magpi.cc

The official Raspberry Pi magazine

BUILD A

RASPBERRY PI SMART GARDEN

OUTDOOR FUN & PRACTICAL PROJECTS



10 HOME AUTOMATION PROJECTS

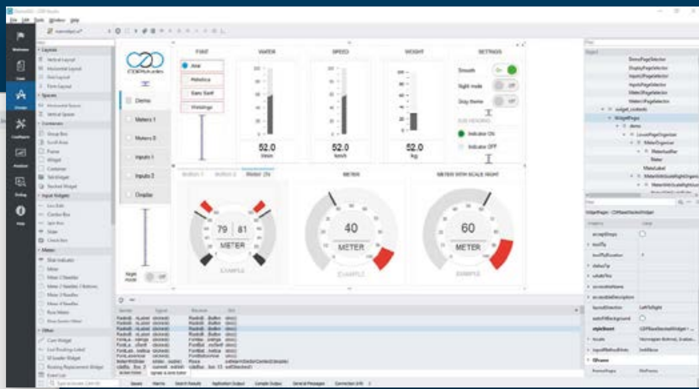
LEARN C WITH RASPBERRY PI

GLOBAL DELIVERY
magpi.cc/store

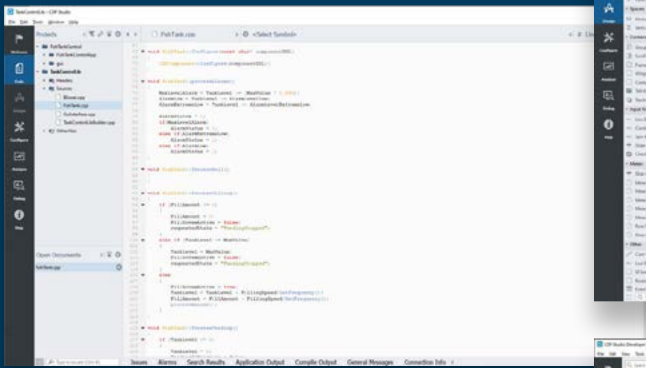
Raspberry Pi PRESS

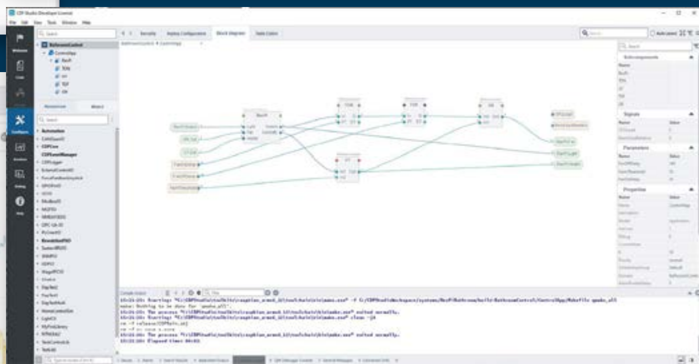
AUDIO MASTERCLASS: HARDWARE, SOFTWARE & PROJECTS!

Design

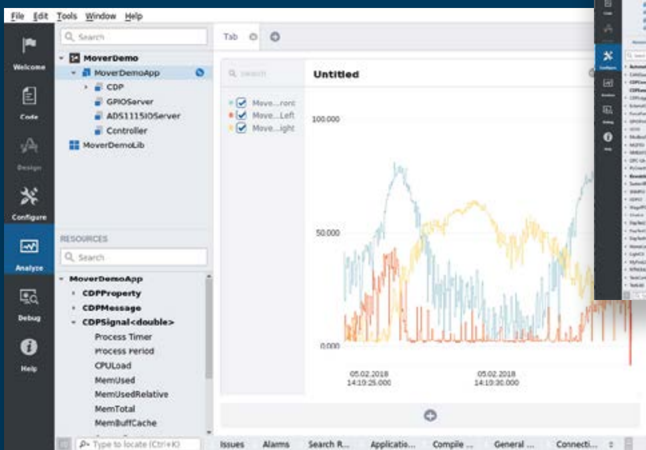


Code





Configure



Analyze

With RPi 4 and Bullseye support

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

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

cdpstudio.com

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

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

CDPStudio

WELCOME
to The MagPi 117

Hopefully you're enjoying the sunshine, because this month we're taking Raspberry Pi outside to play.

In this month's magazine, we're building a Raspberry Pi Smart Garden (page 40). So ask the robot to take care of your plants, fire up the smart barbecue and get cooking your food to perfection, while fooling around with Raspberry Pi-powered drones and RC robots.

There are a couple of super-smart projects that combine cutting-edge AI with nature. Whether it's detecting and getting rid of weeds on a farm (page 16) or building a bird feeder with image recognition (page 26).

If your fingers are decidedly not-green, then PJ has put together an incredible audio special this month (page 68). It's packed with advice for making your own multi-room music systems with Raspberry Pi. Blast some audio around the place and leave the nature outside.

I'm also a huge personal fan of the Commodore 64 Pico project this month (page 30) which uses a Pico to wire games directly to an original C64 via the cartridge slot.

Whether you're an indoor, or outdoor, person; loving the past or the future, there's something for you in this month's magazine.

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Lucy is editor of The MagPi magazine and it's chucking it down in Cambridge. Hopefully, it's sunny by the time this comes out.

@LucyHattersley

GET A RASPBERRY PI ZERO 2 W

WITH A SUBSCRIPTION!

PAGE 38

the

COMPUTERS THAT MADE BRITAIN

*"**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

OUT NOW

Buy online: **wfmag.cc/ctmb**

Available on amazon

Contents

> Issue 117 > May 2022

Cover Feature

Regulars

Project Showcases

40

16

Open Weed Locator

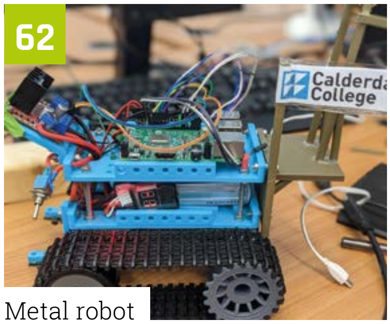
22

NOUS

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

Tutorials

The Big Feature

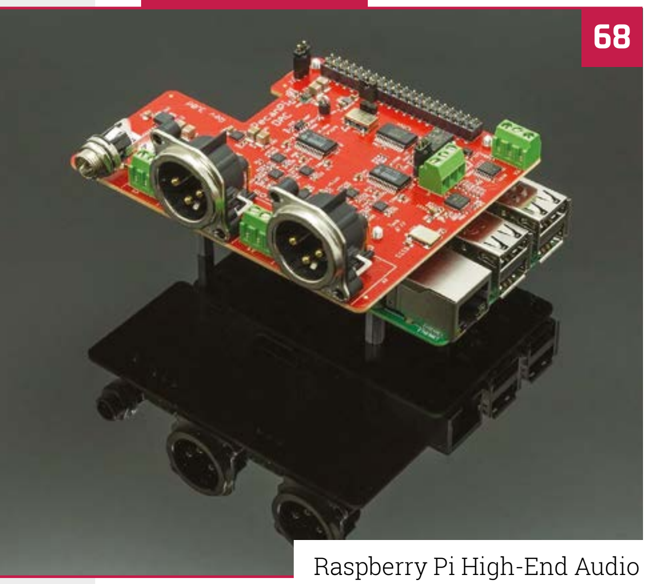


62

Metal robot

66

Raspberry Pi High End Audio



68

Raspberry Pi High-End Audio

78

CutiePi tablet

Reviews

Community

86

Selin Ornek interview

WIN 1 OF 5 **ARGON EON** 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.


Start Your Concept.
Start With Digi-Key.
FREE SHIPPING
ON ORDERS OVER £33 OR $50 USD*
More Products
More Name-Brand Suppliers
More Inventory In-Stock
More Technical Resources
More New Technologies
Digi-Key ELECTRONICS
DIGIKEY.CO.UK
0800 587 0991
DIGIKEY.CO.UK
Digi-Key® ELECTRONICS
*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of $18.00 USD will be billed on all orders of less than $50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA
ECIA MEMBER
Supporting The Authorized Channel

Production and supply-chain update

Why it is difficult to get hold of Raspberry Pi products, by **Eben Upton**, Raspberry Pi CEO

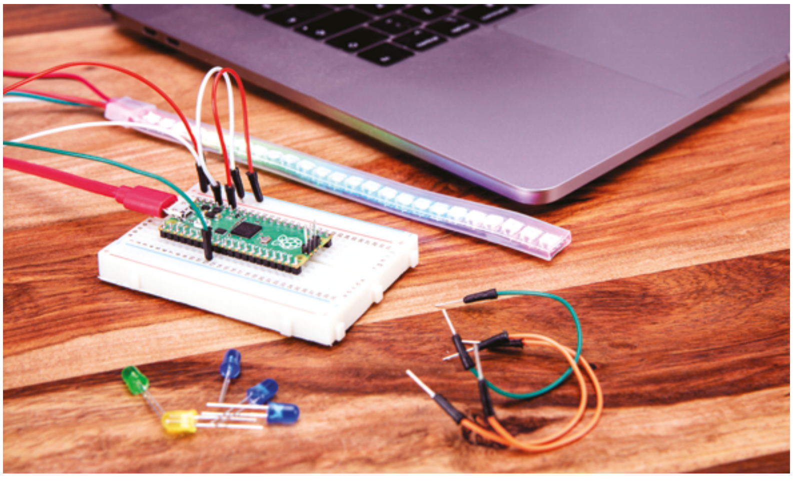
As you will have noticed, it can be hard to buy a Raspberry Pi unit from stock at the moment. Several factors are contributing to this, and we thought it would be helpful to provide an update on what's been happening since we last wrote about this in October 2021 (**magpi.cc/supply**).

Supply

Over the last six months, we've been working hard to get more Raspberry Pi products built and shipped to customers. Despite a variety of supply-chain challenges, we've consistently been able to build around half a million of our single-board computers and Compute Module products each month. As we said in October, the 28nm BCM2711 part used on Raspberry Pi 4 and Compute Module 4 has been more readily available than the 40nm parts used on our older products.

We have a strong pipeline of components, and will continue to build units at at least this rate over the coming months.

▼ Consider Raspberry Pi Pico for your next project, as plenty of stock is available



Demand

As we've said before, the current situation is as much a demand shock as a supply shock: demand for Raspberry Pi products increased sharply from the start of 2021, and supply constraints have prevented us from flexing up to meet this demand, with the result that we now have significant order backlogs for almost all products. In turn, our many resellers have their own backlogs, which they fulfil when they receive stock from us.

> “Demand for Raspberry Pi products increased sharply”

These backlogs absorb Raspberry Pi units as fast as (or faster than) we can produce them, with the result that little of our production volume ends up being immediately available on reseller websites. Where units do appear, bots often attempt to scalp stock which is then resold at higher prices elsewhere. Many Approved Resellers have implemented single-unit limits to combat this, with Adafruit and others going further and enforcing two-factor authentication – we're encouraging other Approved Resellers to consider this route.

We spend a lot of time on backlog management. We have to balance volume demand from commercial and industrial customers with the demand we see from individuals. Right now, we feel the right thing to do is to prioritise commercial and industrial customers – the people who need Raspberry Pi computers to run their businesses – over ex-stock availability to consumers. We're acutely aware that people's livelihoods are at stake. There is currently enough supply to meet the needs of volume commercial and industrial customers. Unfortunately, this comes at the cost of

Raspberry Pi 400 remains in better supply than other models

constrained supply for individuals, who might be looking to buy a small number for home projects or for prototyping.

Advice

So, what should you do if you need to buy a Raspberry Pi in 2022?

Firstly, always buy from an Approved Reseller. We can't emphasise this enough. Our Approved Resellers get preferential access to supplies of Raspberry Pi products. They're also held to a single price: those people you see complaining on social media that they've seen Raspberry Pi computers on sale for vastly inflated amounts of money aren't buying from Approved Resellers, who will all sell you a Raspberry Pi product for the price we state on our products pages plus your local taxes and shipping where appropriate.

If you're a consumer, click on "Buy Now" on a Raspberry Pi product page (**raspberrypi.com/products**) to find an Approved Reseller in your region. Some Approved Resellers take pre-orders, and should be able to give you a good indication of how long it will take to fulfil an order; others don't, in which case you may want to use tools such as rpilocator (**rpilocator.com**) to keep an eye on which resellers have recently received stock.

Secondly, consider Raspberry Pi 400, or Raspberry Pi Pico. These products are generally in better stock positions. We set aside a certain amount of BCM2711 silicon supply for Raspberry Pi 400 (**magpi.cc/raspberrypi400**), which plays an important role in our mission to provide general-purpose PC computing at an affordable price. Many of our Approved Resellers have this product in stock today.

While they are not full-fledged PCs like other Raspberry Pi products, Raspberry Pi Pico (**magpi.cc/pico**), and the many third-party boards based on our RP2040 microcontroller, can be used for many of the same embedded applications. We have plenty of stock of Pico, and of RP2040.

Get in touch!

If you require volume supply of any Raspberry Pi products for an industrial or commercial application, you can contact us at **business@raspberrypi.com**. There remain levers we can pull, and we'll do our best to support you.

Always buy Raspberry Pi from an approved reseller

Raspberry Pi
APPROVED RESELLER

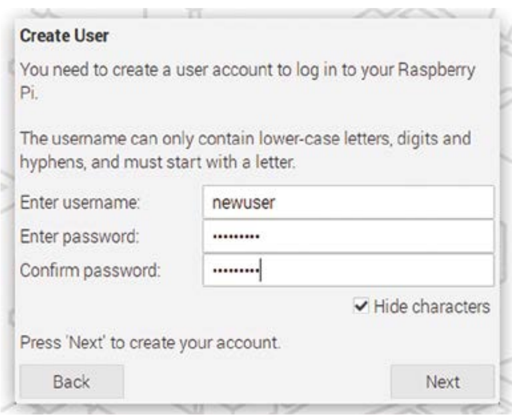
An update to Raspberry Pi OS Bullseye

Latest update removes the default "pi" user. By **Simon Long**, Senior Principal Software Engineer

One of the things which we spend a lot of time thinking about at Raspberry Pi is security. Cyber-attacks and hacking are, sadly, constantly on the increase, and Raspberry Pi computers are as much a target as any other, just because there are so many of them out there nowadays!

Over the years, we have gradually ramped up the security of Raspberry Pi OS; not in response to particular threats, but as a general precaution.

There is always a balance to be struck, however, as security improvements usually carry a cost in terms of usability, and we have tried to keep the system as convenient to use as possible while having an acceptable level of security.



The default "pi" account is no more, and now new users are asked to enter a custom username

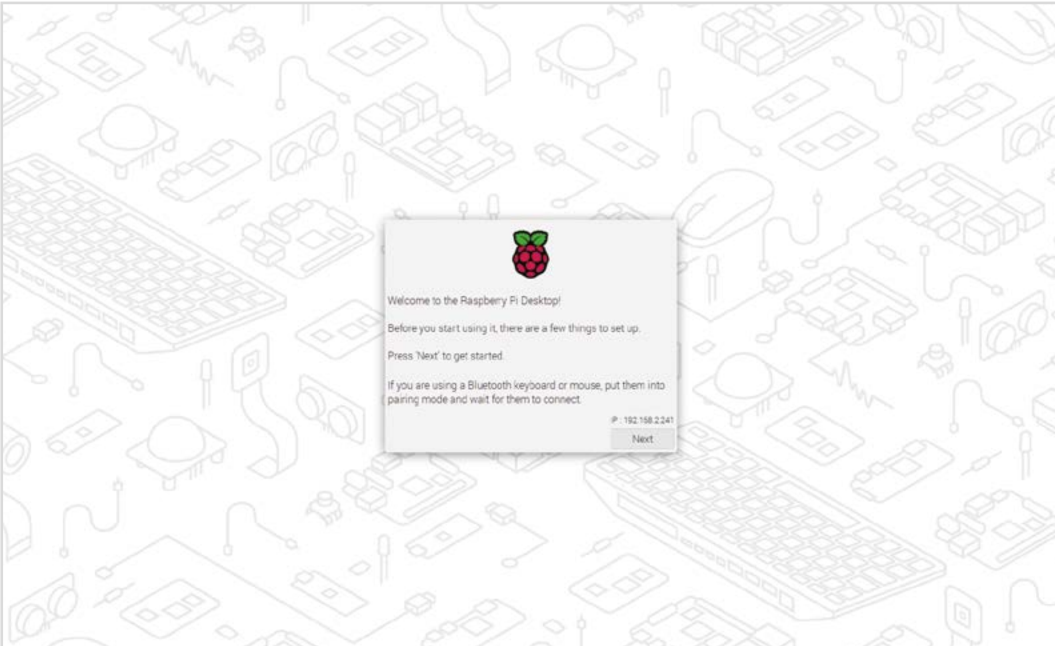
Up until now, all installs of Raspberry Pi OS have had a default user called "pi". This isn't that much of a weakness – just knowing a valid user name doesn't really help much if someone wants to hack into your system. Nonetheless, it could potentially make a brute-force attack slightly easier, and in response to this, some countries are introducing legislation to forbid any internet-connected device from having default login credentials.

With this latest release, the default "pi" user is being removed, and instead you create a user the first time you boot a newly-flashed Raspberry Pi OS image. This is in line with the way most operating systems work nowadays, and, while it may cause a few issues where software (and documentation) assumes the existence of the "pi" user, it feels like a sensible change to make at this point.

The new wizard

Raspberry Pi OS's setup wizard should be a familiar sight by now. It was introduced several years ago, and runs on the first boot, configuring international settings, connecting to wireless LAN and installing any software updates; it also prompts you to change the default password.

The wizard has always been optional – if you pressed "Cancel" on the first page, it just went away and you weren't forced to use it.



The Raspberry Pi OS setup wizard is used to guide new users into setting up a custom user account

From now on, working through the wizard is no longer optional, as this is how a user account is created; until you create a user account, you cannot log in to the desktop. So instead of running as an application in the desktop as before, the wizard now runs in a dedicated environment at first boot.

The wizard itself is largely unchanged, with the key difference being that you are now prompted for a user name and a password. You can set these to "pi" and "raspberry" as before, but you will get a warning message that doing so is unwise. It remains your choice – some software might require the "pi" user, so we aren't being completely authoritarian about this – but we really recommend choosing something else!

> We have gradually ramped up the security of Raspberry Pi OS as a precaution

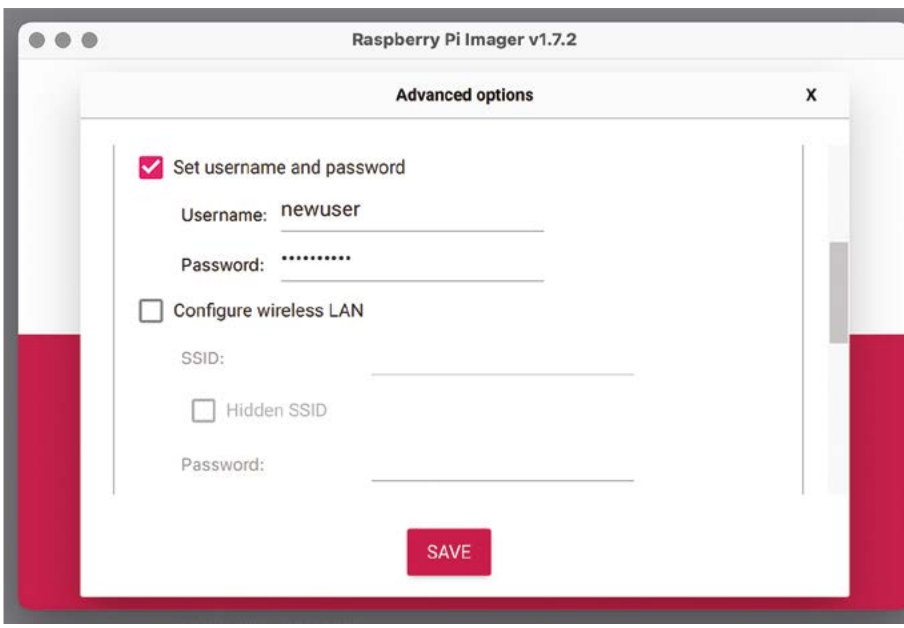
Headless setup

For people who run their Raspberry Pi headless and therefore cannot work through the wizard, Raspberry Pi's Imager tool allows you to preconfigure an image with a user account; when an image created like this is first booted, it will come straight up in the desktop, logged in as the user created in the Imager.

To preconfigure an image like this, when you have selected the source image and destination in Imager, click the "settings" button – the picture of a cogwheel – before clicking "Write", and use the Advanced options menu to enter a username and password, along with any other preconfiguration you want.

There are also mechanisms to preconfigure an image without using Imager. To set up a user on the first boot and bypass the wizard completely, using a file called "userconf" or "userconf.txt" in the boot partition of the SD card. For more detailed information read "An update to Raspberry Pi OS" on Raspberry Pi's website (**magpi.cc/securityupdate**).

Raspberry Pi Imager can be used to set a custom username and password while copying Raspberry Pi OS to the microSD



Mini Pinball Machine

Gamers are sure to have a ball with Chris Dalke's fun desktop pinball table. **David Crookes** has his fingers at the ready…

MAKER

Chris Dalke

Chris is a Boston-based developer and maker. He loves building user interfaces for hardware. At work, he builds autonomous boats.

magpi.cc/minipinballgit

Alert!
Laser cutting

This project involves laser cutting. Be careful when using a laser cutter in your projects.

magpi.cc/lasercuttersafety

Pinball machines have been around for decades. They were popular during the Depression in the 1930s, banned under US gambling laws for 34 years from 1942, developed flippers in 1947 and saw a resurgence in popularity in the 1990s. But, in all that time, the machines have had one thing in common: their sheer size and weight.

That, however, didn't stop Chris Dalke from trying to create a version of his own. "Years ago, I had a high school woodshop class with access to a CNC wood cutter," he says. "I tried to make an electromechanical pinball machine but never finished because the project was too ambitious in scope for my skill set and budget at the time."

Even so, the desire to create a pinball machine remained strong so he tried again, this time creating a miniature version using a Raspberry Pi 4 computer, an Arduino Uno, an LED matrix display, a bunch of buttons and a 7-inch HDMI touchscreen. "It constrained the project to a more realistic scope," he says. "It also allowed the enclosure to be smaller so it could be brought out and played on a tabletop."

The laser-cut maple exterior gives a solid arcade feel

Lane change

Chris had a clear objective in mind from the start. "I wanted to retain the feeling of a physical arcade game with intense sound, vibration and colours as well as the tactile response of the inputs," he says. "I noticed many arcade games feel very good at the lowest level of tactile response, an individual button or joystick press, so I started there, with clicky arcade buttons."

To that end, he decided not to replicate the mechanics of a pinball machine. "It was less about the pinball machine and more about building the complete experience of a small arcade game that could retain the feel of a full-scale game," he explains. It led him to create a screen-based version of pinball which he coded in C++ and OpenGL, using the open-source software development library Raylib to create both the graphics and audio.

"I chose very vibrant neon-inspired colours: purple, green and pink which are very saturated on the monitor," he continues. "In the game code I also added extensive juicing which is the use of many small animations and visual/audio tweaks to improve the feel of a game. For example, the

Quick FACTS

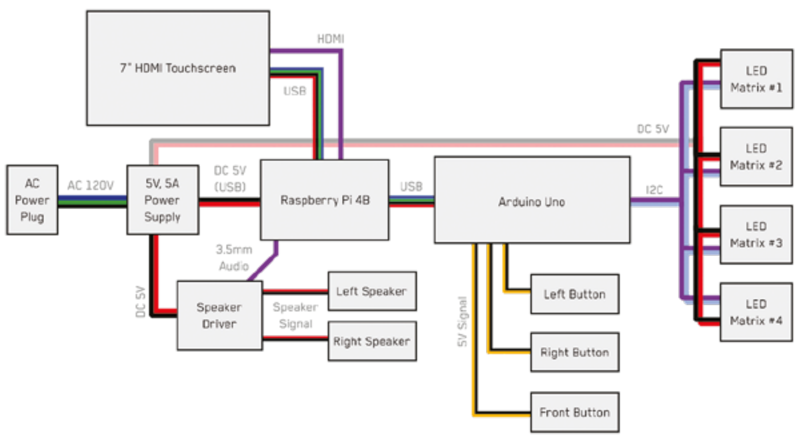
- The device is inspired by the feel of arcade machines
- Players control the game using three tactile buttons
- Plans to provide haptic feedback were dropped
- Arduino Uno sends button presses in real time
- The game was designed to feel like a physical machine

> “The LED matrix casts a very nice orange light on to the wood”

ball and bumpers stretch and distort excessively when a collision occurs, exaggerating the physical effect of the collision.”

Keeping score

For an authentic look, Chris naturally wanted the laser-cut Baltic birch plywood enclosure to resemble a pinball machine, so he tweaked the design to ensure it was unmistakable. “Initially, I’d designed a flat box without the vertical headboard seen in conventional pinball



▲ Raspberry Pi 4B and an Arduino Uno are at the heart of the build. Chris says the project was iterative, however, so he doesn’t have build plans

Chris told us he loved how arcade games rely on simple, satisfying interaction methods that feel good to play

▲ The top panel of the enclosure holds four Adafruit LED Matrix panels controlled over I2C

machines but I added an LED matrix and vertical section," he says. "I wanted to retain the visual signature of a pinball machine and have some element that made the game feel less like it was played only on the touchscreen." YouTube videos such as Secrets of Game Feel and Juice (**magpi.cc/secretsgamefeel**) helped Chris when designing the game.

> "I wanted to retain the feeling of a physical arcade game"

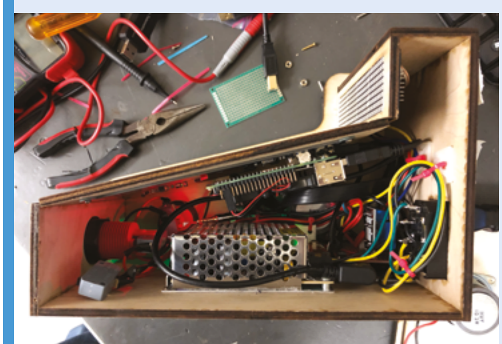
The Adafruit LED matrix provides the score and feedback and it certainly looks the part. "The LED matrix casts a very nice orange light onto the wood and it works well to pull the gameplay out of the screen and into the physical world," Chris continues. The Arduino Uno drives the LED matrix and button inputs, and it communicates with the Raspberry Pi board via a serial protocol. The Raspberry Pi is connected to a speaker driver too, allowing for stereo sound.

"I also included a solenoid which I planned to trigger for haptic feedback," he says. "But the vibration was too high-frequency to match the expectation of a heavier ball – I ended up using sound effects instead." Still, this doesn't detract from the overall build, and Chris is very pleased with how it's turned out. "The project is the sum of many individual tweaks to the components, but the whole experience comes together very well."

Creating a Mini Pinball Machine

01 After creating a CAD render of the Pinball Machine around the size of a Raspberry Pi 4 and a HDMI screen, Chris laser-cut each of the enclosure pieces out of Baltic birch plywood.

02 Each of the wooden components were designed to interlock so that the enclosure could be assembled with as few screws as possible while also allowing easy access when required.



03 It may look like there's ample room but getting all of the electronics inside the enclosure was tricky. Slats are cut into the side of the enclosure to allow players to hear the stereo sound.

Open Weed Locator

Distinguishing between valuable crops and unwanted upstarts requires precise plant knowledge and some Raspberry Pi processing, learns **Rosie Hattersley**

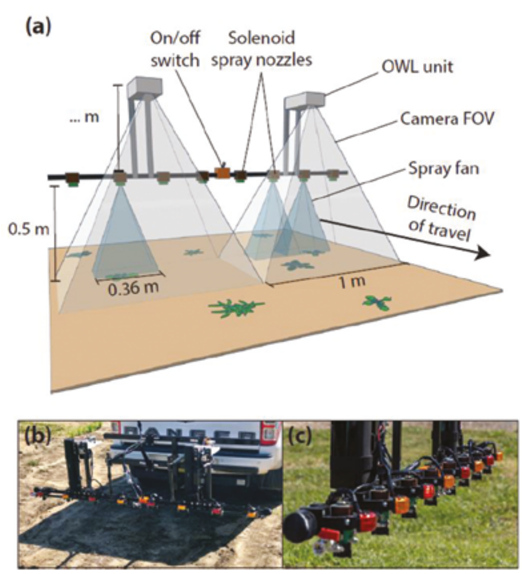
MAKERS

Dr William Salter and Guy Coleman

University of Sydney academics Guy and William developed OWL using Raspberry Pi 4 to make weed management significantly cheaper

magpi.cc/owlgit

Spring has arrived in the northern hemisphere, along with the weeds that pop up ever more fervently each year. Picking off unwanted plants but leaving others behind requires knowledge and precision – exactly what machine learning is adept at. OWL (Open Weed Locator), developed at the University of Sydney, uses Raspberry Pi 4 to make managing agricultural sites with robots more efficient. It is "a green-on-brown weed detector that uses entirely off-the-shelf componentry, very simple green-detection algorithms and entirely 3D printable parts," explain its makers. The Raspberry Pi 4-based OWL detection system can be mounted on a ruggedised vehicle or tractor and costs a mere $400.



Open Weed Locator sensors ignore anything green and apply fungicide or insecticide to other parts of the crop

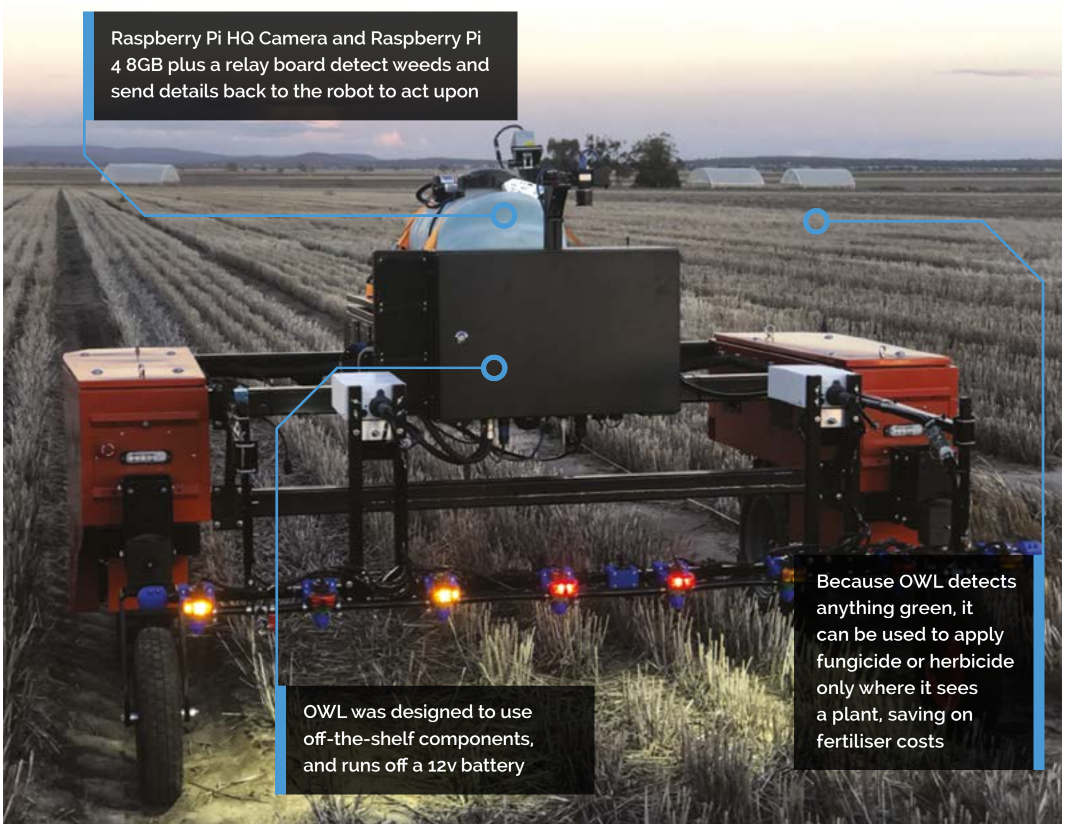
Precise planting

Guy Coleman has extensive experience as an agricultural scientist, and began using Raspberry Pi five years ago as a means of exploring how computer vision might be used in such settings. Weed recognition and precision control using deep learning is the focus of his PhD at the University of Sydney. Before this, Guy was more comfortable doing precision weed-control fieldwork on large-scale paddocks in Australia than developing projects using Python such as the OpenWeedLocator.

He works alongside Dr William Salter, whose background is in plant physiology and open-source technology for plant phenotyping, and who had already built several light sensors and an instrument for the high-throughput measurement of photosynthesis.

"Managing weeds in crops so they don't reduce yields is a big challenge in agriculture, and current methods rely on herbicide applications to whole fields," explains Guy. "Being able to assess where weeds are with cameras means the herbicide is only applied to individual weeds, meaning big savings to the farmer and reduced chemical inputs to the environment."

However, weeds vary hugely in colour, size, and shape and the team needed to find a way of recognising them in all sorts of environmental conditions. Since the weed detector also had to work at a reasonable speed, any algorithm used would have to operate with high frame rates, Guy explains. They chose to base OWL around an 8GB Raspberry Pi 4 because of its combination of low cost, high power and small form factor. "Being easily connected to a whole variety of inputs and



Quick FACTS

- There are now OWLs on four continents
- One is mounted on a robot called Gordon
- OWL is partly about engaging with non-agriculturalists
- And spreading knowledge about weed control using technology
- Using OWL to control weeds with herbicides is purely optional

outputs has been absolutely essential to this project," Guy comments.

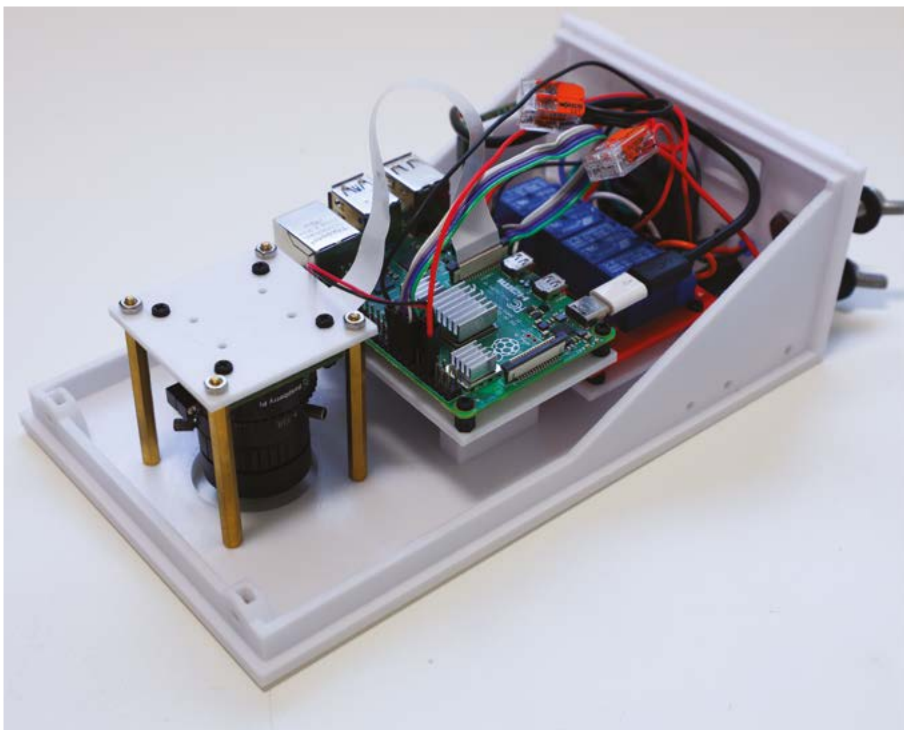
A green detection algorithm running on Raspberry Pi identifies any green weeds that appear in the video feed and then activates a GPIO pin that connects to a relay board. A solenoid can then be switched on to deliver herbicide to the detected weeds.

Smarter applications

Guy and William wrote the code for OWL in Numpy and OpenCV. Keeping OWL open source means it can be easily updated with improved weed detection capabilities. Their biggest challenge was developing an algorithm that performed at an acceptable level in a range of conditions but testing convinced them to settle on a combination of the Excess Green + HSV thresholding systems.

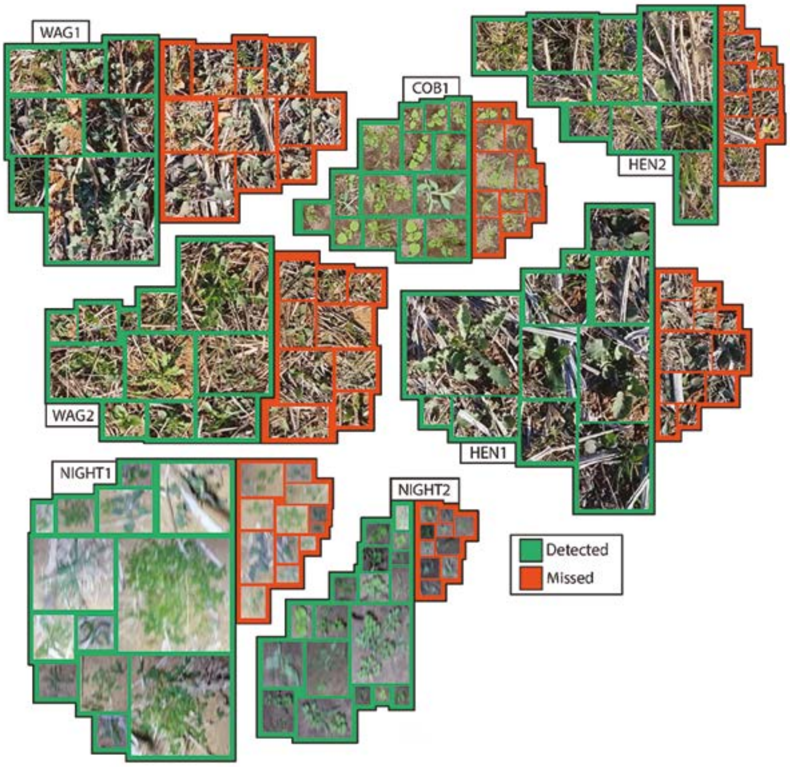
OWL is very much a community project, with the full hardware details posted on GitHub (**magpi.cc/owlgit**) including a 3D-printable enclosure. This has already led to versions of OWL being assembled and used on four different continents, with some tweaking of the enclosure design for easier printing and assembly.

▼ The finished Open Weed Locator build



Guy and William with two OWLs mounted on the University of Sydney's Digital Farmhand robot at the University's Digifarm in Narrabri, seven hours north of Sydney

OWL can be mounted on a tractor or the back of a truck or car



Guy and William used a dual weed detection approach as straw-coloured weeds were often missed by the ExHSV algorithm

The most critical pieces are the Raspberry Pi 4 with 8GB, Raspberry Pi HQ Camera, a relay control board and 12V to 5V 5A DC to DC converter, says Guy. Aside from printing parts, building OWL takes a couple of hours and running costs are minimal – only 12V of input power required for it to run.

> OWL is very much a community project

Guy and William plan to add in-crop weed detection and GPS and say quite a few farmers have spoken to them about different uses they see for OWL to improve the efficiency of food and fibre production globally. The ability to find anything green means OWL can also be used to only apply fungicide or insecticide to the crop or to defoliate green cotton plants. "OWL is a living project. Now that it has 'flown the nest', so to speak, we're excited to see where the community takes it."

Make your own OWL

OWL is open source, with full hardware assembly instructions at **magpi.cc/owlassembly**. You can download and 3D print the enclosure too.

01 Install the owl.img software on a fresh installation of Raspberry Pi OS. Name the virtual environment owl. Assemble the hardware shown in the image.

02 Install Raspberry Pi HQ Camera and enable it with raspi-config. Download the entire OpenWeedLocator repository into Raspberry Pi's Home directory

03 Install the camera as shown, make the Python file greenonbrown.py executable and use bash to make it run at startup. Set up the camera to view the live feed found in greenonbrown.py and reboot.

Entry/Exit Notifier

A simple and descriptive name for a project that may seem simple, but has been engineered very well. **Rob Zwetsloot** buzzes in

MAKER

Wayne Chan

A research computer analyst with the Centre for Earth Observation Science, a climatological research centre at the University of Manitoba. He is also a maker, historical researcher/writer, and retrocomputing enthusiast.

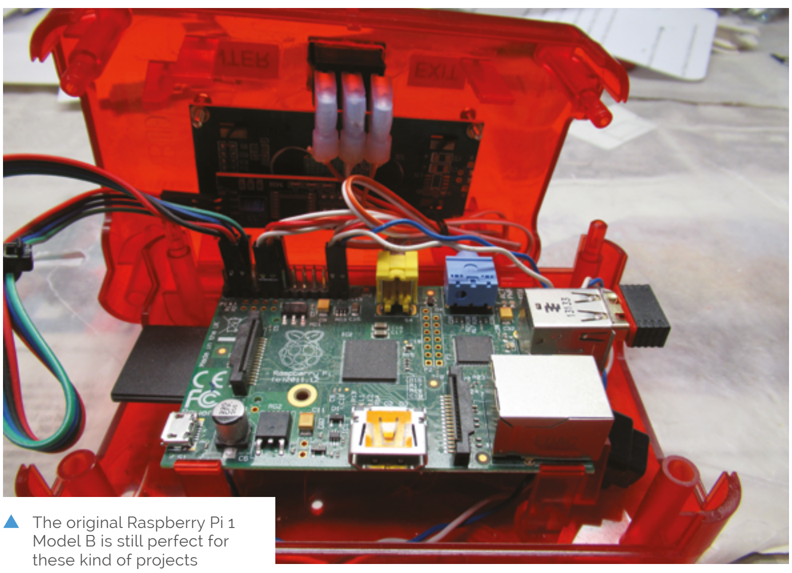
Not only do we like to show off fun, unique, and complex builds in The MagPi, but also Raspberry Pi projects that are just put together really, really well. This month, we want to show off what Wayne Chan has been working on.

"It's a device that sends a notification e-mail with the push of a button." Wayne tells us. "At the university where I work, we have had to observe building capacity limits over the last two years. In order to know how many people were in the building, we were required to send an e-mail to the department office when we entered the building for the day and when we exited. I had the idea to make a simple device that could send a pre-written message with the push of a button."

Sounds simple, right? Wayne had found himself forgetting to send the necessary emails, and thought the process was a bit tedious when he did remember, which became the impetus for the project. There's a little bit more to it than just being a button that makes it pretty cool though.

"I used an old 512MB Raspberry Pi Model B that I had bought back in 2013 that I never did much with." Wayne continues. "Because the project needed internet connectivity and the Model B didn't have wireless LAN, I added a USB dongle. For a display, I decided to use a cheap 16-character by two-line LCD display instead of an LCD panel, because I only needed to know if the email transmission was successful."

▲ The screen confirms the command has been sent



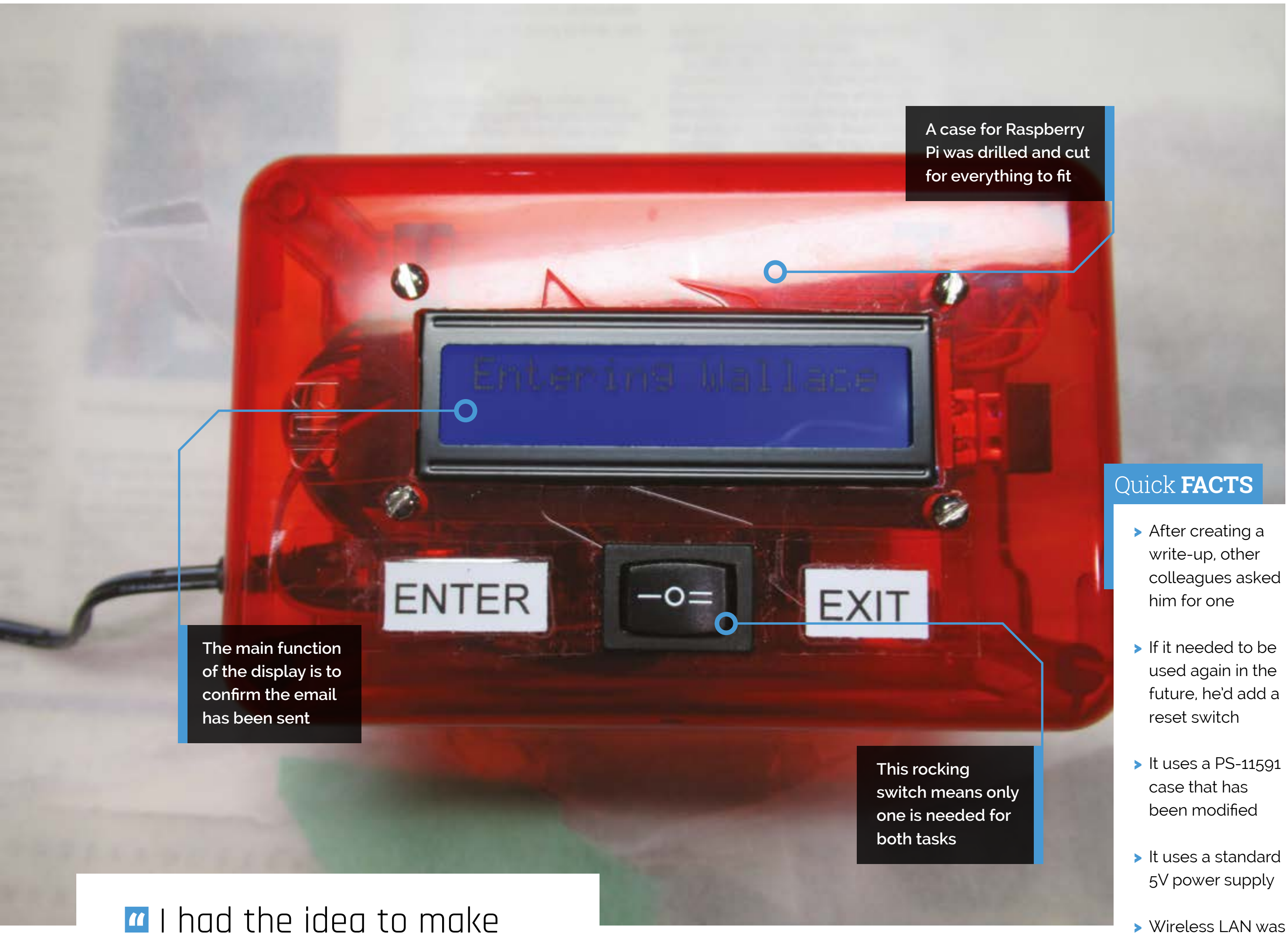
▲ The original Raspberry Pi 1 Model B is still perfect for these kind of projects

Programmable entry

The code on Raspberry Pi was well thought out as well: "Email was handled by msmtp, a lightweight command-line mail transfer agent," Wayne explains.

"Since the university used Microsoft Exchange, I used Davmail, an open-source program that acts as a gateway between non-Microsoft mail clients and MS Exchange servers. The push-button interface on the EN/EX Notifier could have been either two SPST momentary switches, one for entry and one for exit, or a single momentary on-off-momentary on SPDT rocker switch. I decided to go with the rocker switch, and connected it to Raspberry Pi's GPIO pins. When the entry or exit button is pressed, it triggers the appropriate bash script to send an email, and displays a message on the LCD display to tell me if it was successful or not."

Wayne had thought about automating the system by using the room's light switch. However, with other people occasionally going in and out, he found it better to just make it a dedicated switch.



A case for Raspberry Pi was drilled and cut for everything to fit

The main function of the display is to confirm the email has been sent

This rocking switch means only one is needed for both tasks

Quick FACTS

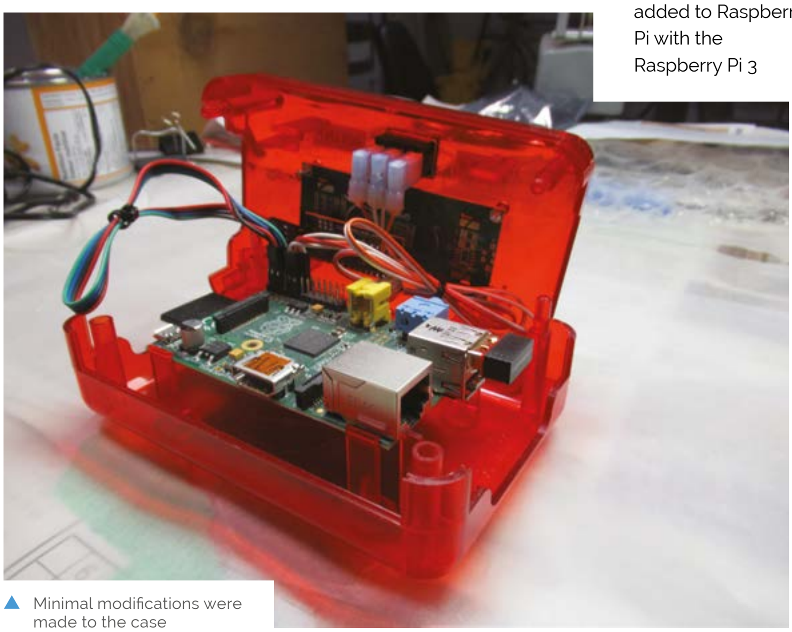
- After creating a write-up, other colleagues asked him for one
- If it needed to be used again in the future, he'd add a reset switch
- It uses a PS-11591 case that has been modified
- It uses a standard 5V power supply
- Wireless LAN was added to Raspberry Pi with the Raspberry Pi 3

> “I had the idea to make a simple device that could send a pre-written message with the push of a button.”

Exit, stage right

While Wayne will soon not need to use it, he definitely was able to get some use out of it:

“It functions fairly well. The main issue is that the wireless LAN connection is sometimes unreliable, so it doesn’t send a message when I push the button, but that is mostly due to my office being far away from the nearest router. I’ve had connection problems with other computers, so it’s not caused specifically by Raspberry Pi. I could probably improve the reception by using an adapter with an external antenna, or by using a wired Ethernet connection instead.”



▲ Minimal modifications were made to the case

NOUS: uNdersea visiOn sUrveillance System

Monitoring submerged wrecks using AI and Raspberry Pi reveals the secrets of the deep, writes **Rosie Hattersley**

MAKER

Dr George Papalambrou

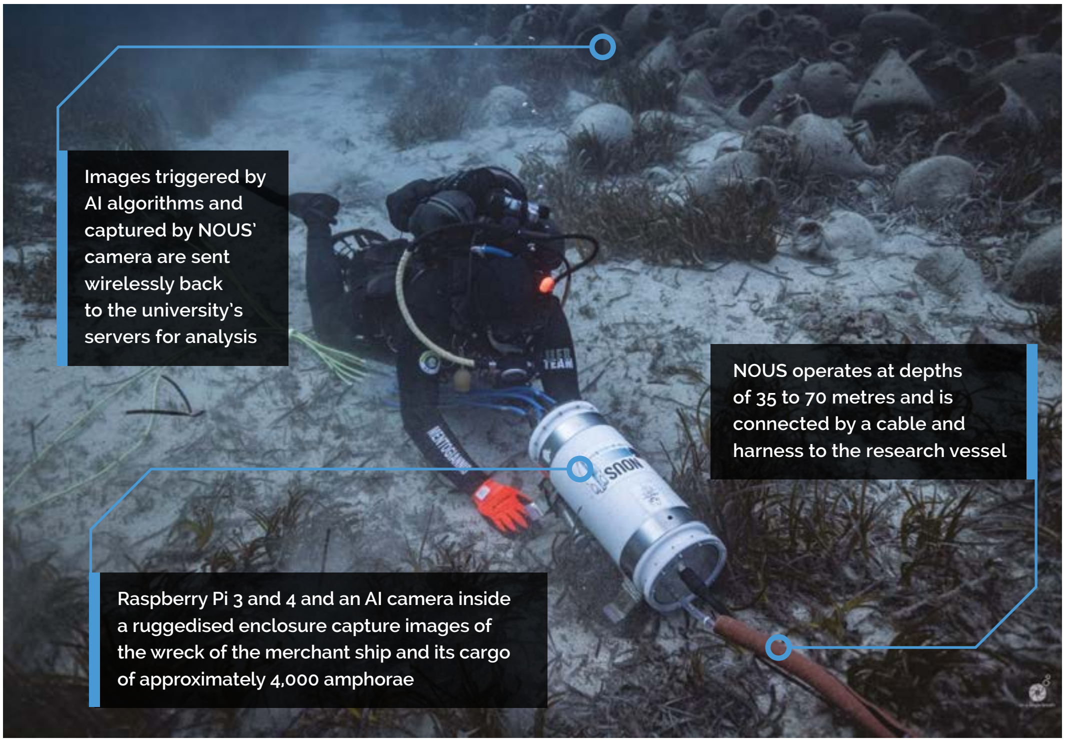
George is an assistant professor at the University of Athens' School of Naval Architecture and Marine Engineering, where he is involved in research related to AI and machine learning

nous.com.gr

Watching shoals of fish flit idly by is one of the most magical and calming experiences. Just such an opportunity is a wonderful by-product of Greece's NOUS project, the uNdersea visiOn sUrveillance System. Based on Raspberry Pi 3 and 4, its cameras help researchers from the National Technical University of Athens monitor the submerged shipwreck of the a merchant ship near Peristera, one of the largest known ships from classical antiquity.

Dr George Papalambrou and his colleagues Vasilis Mentogiannis and Kostas Katsioulis, from the NTAU's School of Naval Architecture and Marine Engineering, knew plenty about Raspberry Pi before selecting it for their underwater archaeology surveillance project. "It was our first choice from day one," George says. For a start, he had used Raspberry Pi alongside Apple HomeKit for home automation, and while at university it was used in CAN-bus networks. George was

The NOUS team working on the first submarine surveillance camera prototype



Quick FACTS

- NOUS is roughly equivalent to the classical Greek word …
- …which translates to intellect, thinking and reasoning
- The Peristrera was a 25m-long merchant ship
- It sank in the 5thC BCE, on its way to Skopelos
- It is now the world's first underwater ancient archaeological museum

also interested to read about Raspberry Pi Compute Modules being used by the University of Surrey for their Cube-Sats (**magpi.cc/cubesatproject**), confirming the hardware's suitability in challenging environments and ability to communicate while being self-powered.

> The sea is an unforgiving environment, and very hard on equipment

NOUS, as the marine surveillance project came to be known, would also need to be able to communicate remotely.

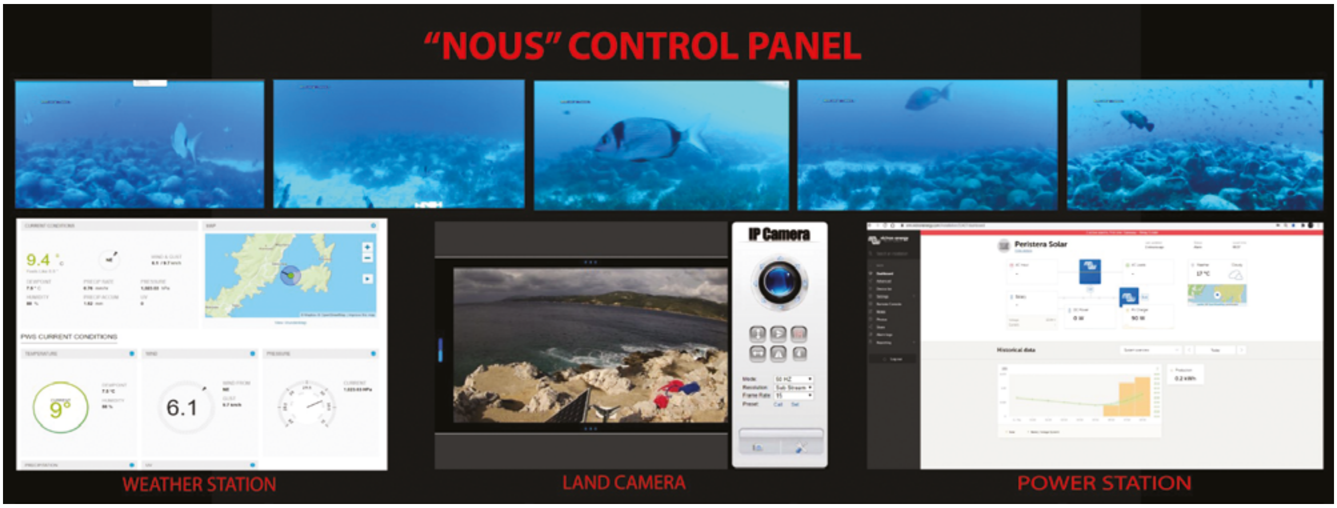
Sea-worthy specifications

George says Greece had wanted to monitor its marine archaeological sites for many years, but one of the main obstacles was how to guard and protect them.

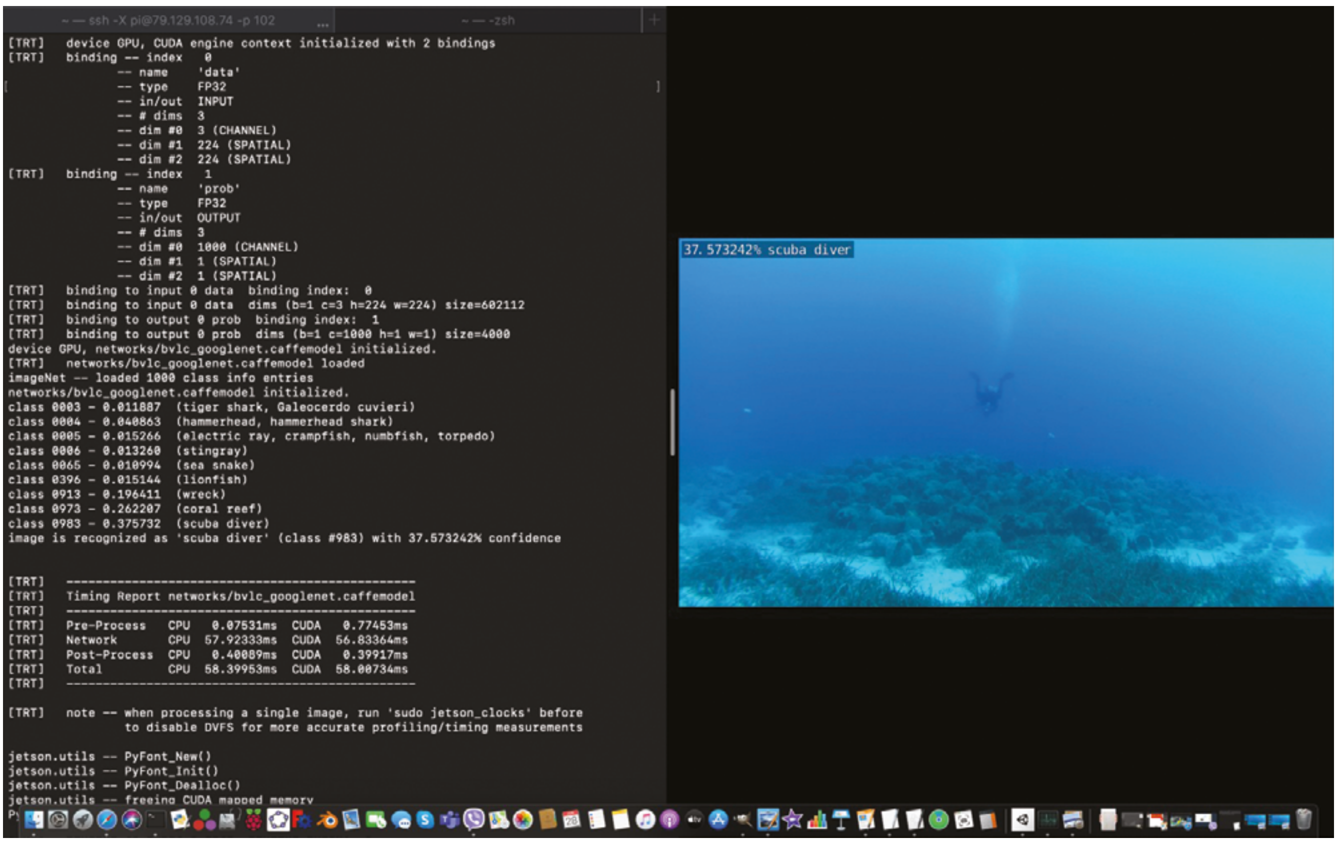
It needed a system that was self-powered (since most wrecks are located a long way from a power supply), that could connect to the internet in order to communicate and be remotely controlled, have sensors to monitor the area of interest continuously, and be able to send alerts in cases of violations, alterations of the site or other events. As well as monitoring protected marine areas, scoping exercises suggested it would also be feasible to include real-time scientific observations throughout the day, and to monitor changes to the climate and biodiversity in the area over long periods of time.

The sea is an unforgiving environment to operate in, and is very hard on equipment, says George, so it was critical they chose gear that could withstand both high pressure and low temperatures. NOUS needs to run continuously round-the-clock at submerged depths of 35 to 70 metres. George explains that the project also needs total software control at the operating system level, as well as at the application level: "We control our devices remotely over the web and SSH, so there is no space for failures or malfunction." Raspberry Pi was always the team's first choice, not least because of its affordability and the invaluable community forums.

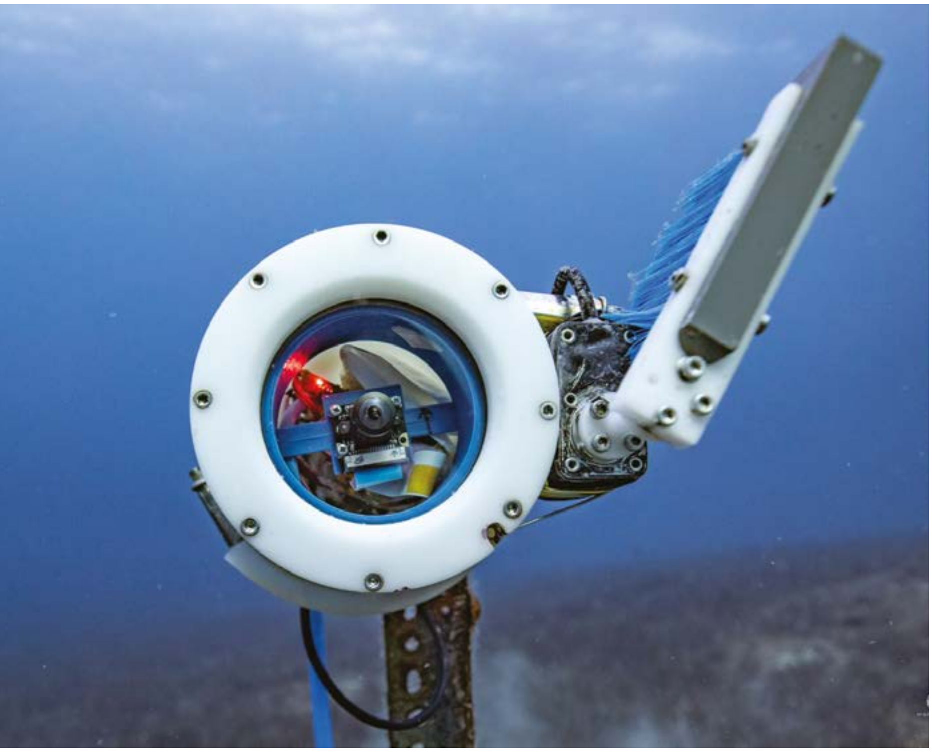
Having bought Raspberry Pi 3 and 4, plus some basic off the shelf electronics, the NOUS team soldered on cables and parts in order to save space in the rugged enclosures that also needed to accommodate AI cameras and



NOUS's dashboard shows live views from each of the undersea cameras as well as logging temperature patterns and local weather conditions



A diver has been detected entering the protected marine area, triggering an alert via Raspberry Pi



Camera one of five surrounding the Peristrera, showing Raspberry Pi's camera in use

networking hardware that could be attached via a harness and operate underwater.

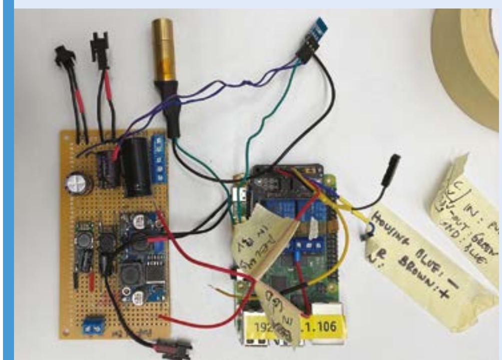
Academic expertise

The software and specialist HATs were developed by George and his University of Athens colleagues to save money and reduce development time. Raspberry Pi boards were set up headless, with X11 forwarding used to optimise remote control along the lengthy undersea cables connecting each module to the base station. The onboard battery packs are supplemented by onshore solar panels sited near where divers set off to view the wreck in the Aegean, some distance short of its intended destination, the island of Skopelos.

Operating full-time since 2020, the surveillance system is still running successfully today. "Raspberry Pi has been a success since the very beginning, providing stability on both the software and hardware sides", says George.

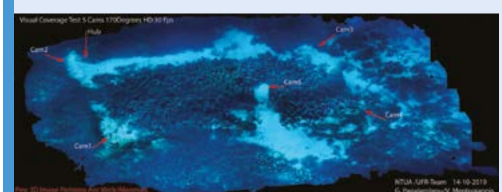
Live footage from the wrecked ship can be seen at (**magpi.cc/peristera**).

Underwater archaeology



01 NOUS primarily uses an 8GB Raspberry Pi 4 at the heart of the AI camera-based marine surveillance setup, along with temperature, moisture and movement sensors, and a Raspberry Pi-compatible camera

02 Sturdy communications, Raspberry Pi running in headless mode, self-contained battery packs and durable waterproof enclosures are also critical elements.



03 The NOUS cameras are positioned near the seabed around the wrecked merchant ship, and are able to capture images at 30fps. Onshore solar panels top up the power supply.

A Bird in the Hand

Jeff Stockman's smart bird feeder counts the number of birds that stop by for food, and also identifies which species they are. **David Crookes** explains

MAKER

Jeff Stockman

Jeff is an avid woodworker, tinkerer, and maker in his spare time. He has 20 years of global commercial healthcare experience and is studying for his Masters in Computer Science at the University of Washington.

magpi.cc/birdhandgit

For four days every February, people across the world take part in the Great Backyard Bird Count. By indicating the number of wild birds they see in real time, these enthusiasts help scientists to spot changes in population and habitats, and it's also great fun to see which of our feathered friends pay a visit.

Inspired by this, maker Jeff Stockman looked to automate the process as part of his Internet of Things course at the University of Washington Tacoma. "I'd made a bird feeder a few years ago and used a first-generation Raspberry Pi computer and a basic webcam connected to my local network," he says. "But I wanted to incorporate both edge and cloud technologies to improve the feeder's capabilities."

The result has been a smart bird feeder that uses motion sensing and image recognition to monitor birds dropping by for something to eat. "It uses an ultrasonic ranger to detect the presence of a bird," Jeff explains. "This detects the distance of the bird from the feeder and, when it decreases below 14cm, it triggers the camera to snap a photo. Once the bird is more than 14cm away, the camera is ready to take a photo again."

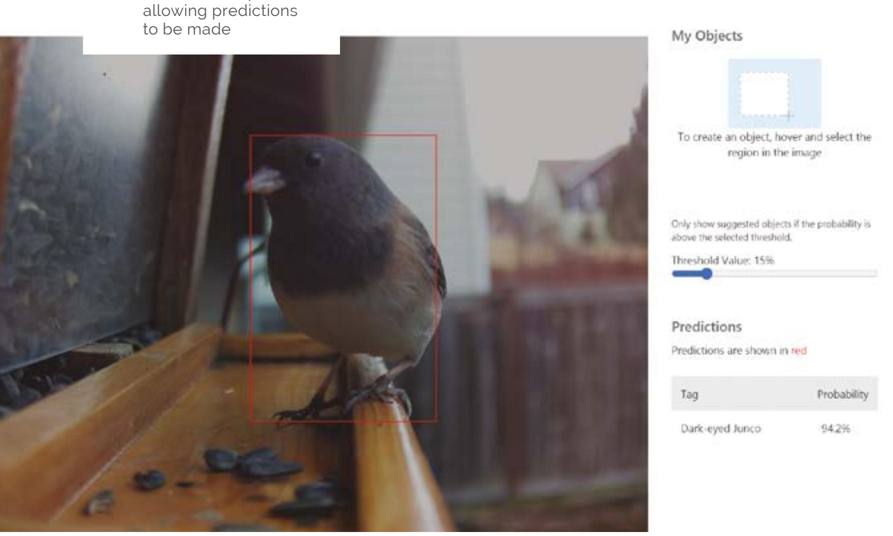
To achieve this, Jeff used a Raspberry Pi 3B computer connected to a Raspberry Pi HQ camera module. "Using Raspberry Pi was a main requirement for the course," Jeff explains. He also used the flow-based development tool Node-RED to simplify the coding as well as the GrovePi+ HAT, which allows a variety of sensors to be easily connected – Jeff added temperature, humidity and light sensors as well.

Training the model

Once a photo is taken, the image is analysed using the free tier of Microsoft's Azure Custom Vision machine learning service. "I wanted image recognition to understand which bird species were present throughout the year and to identify migratory patterns, as well as year-on-year differences in bird populations," Jeff tells us.

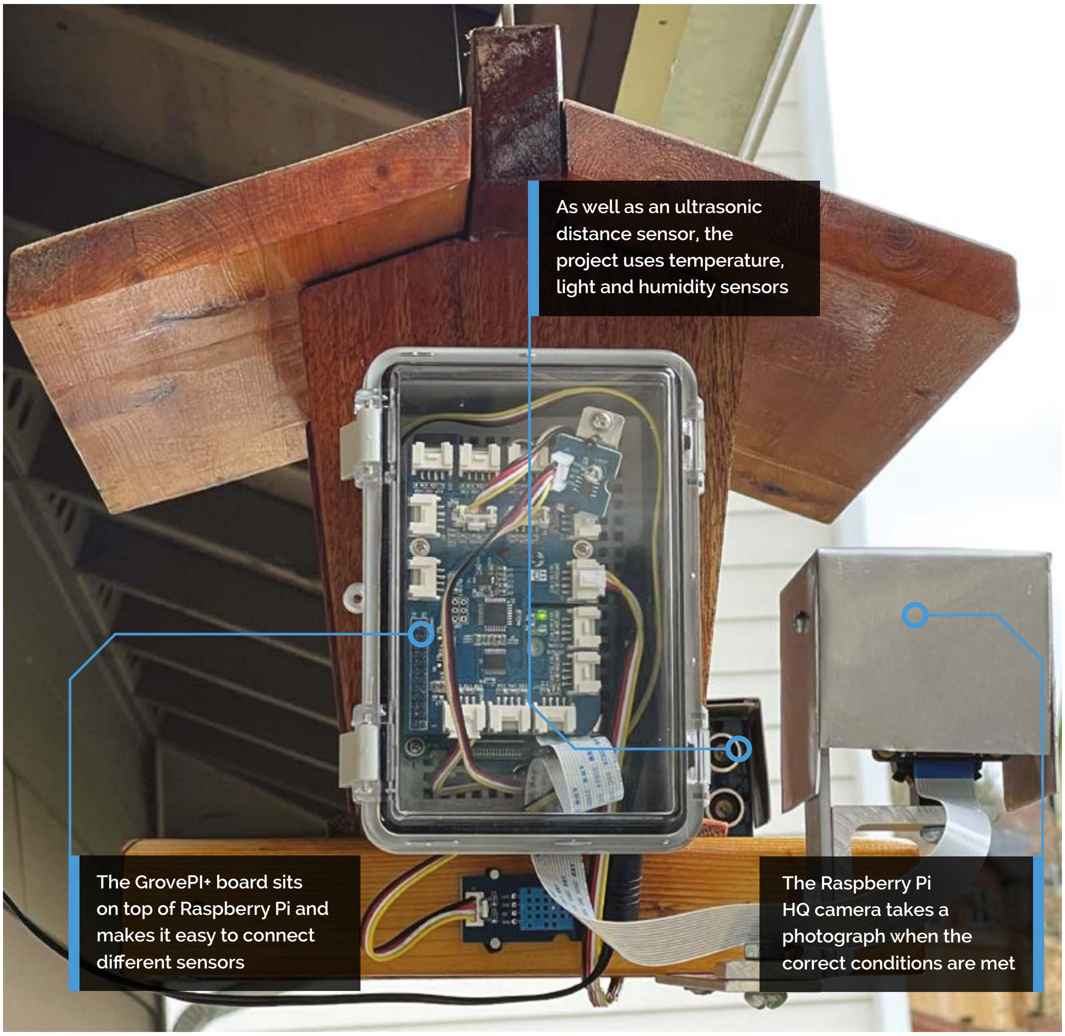
> It identified quickly that there were no early birds

This required Jeff to train the model. "I pulled 20 to 25 images from Google for each of the popular bird species that frequent my feeder. I then uploaded them to Azure and tagged them with the correct species, testing the model from the



▼ Azure was used to train the machine learning model that detects bird species, allowing predictions to be made

▲ The camera and ultrasonic sensor are set on either side of the area where birds will feed



Quick FACTS

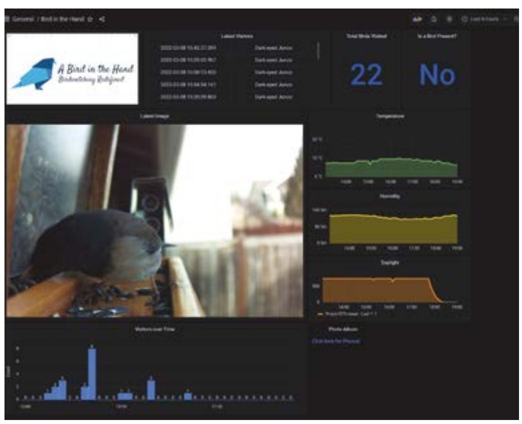
- The feeder detects motion and takes photos
- It keeps a tally of the number of visiting birds
- Sensors are connected to the GrovePi+ HAT
- Activity is stored in an InfluxDB database
- Jeff's considered including a squirrel-detection system

convenience of my desk by printing out different pictures of birds." Jeff then asked Azure to return the species name if the probability exceeded 50 percent. "This would increment the count in a database by one," he adds.

Caught on camera

The model continued to be trained once the smart bird feeder was installed outside. "I could verify the photos and species tags on the Azure website. If incorrect, I'd retag the images and those would then get stored in Azure. Once I had enough real-world images, I retrained the model with new images and additional species that appeared. The model accurately identified 60, then 70, then 80 percent of birds over three iterations of the model."

There were some difficulties. Jeff would like a live feed but says the image capture doesn't trigger when this feature is active. "The GrovePi+ ultrasonic sensor was also sporadic in its measurements – the measurements ranged from 15 to 27 cm," Jeff adds. But the project has proven effective. The data and images are shared with the interactive visualisation web app, Grafana, allowing Jeff to see data and photos in real time. He's also been able to track birds' eating habits with some surprising results. "It identified quickly that there were no early birds," he says. "None appeared before 10am!"



The last known bird detected with a probability of 50 percent or more is shown in Grafana

TARDIS treasure hunt

When Roberto Tyley saw his son's birthday party exterminated, it ultimately led him to invest time in an amazing Doctor Who-themed treasure hunt, as **David Crookes** explains

MAKER

Roberto Tyley

An open source enthusiast, Roberto develops software for The Guardian and authored the BFG Repo-Cleaner

magpi.cc/timetreasure

Last year, many *Doctor Who* fans embarked on a virtual treasure hunt in a bid to find Jodie Whittaker's Time Lord. Fast forward to 2022, and maker Roberto Tyley has created a physical hunt – one in which his son, Alexander, was tasked with fixing the TARDIS by finding and entering a series of passphrases with the help of his friends.

The idea evolved over time, sparked by the cancellation of Roberto's son's seventh birthday party in 2020. The following year's bash was held outdoors as a treasure hunt in the local park and the children had to find clues to unlock a LEGO combination safe. "My son loved it so much, he wanted another treasure hunt – but how was I going to top last year's party?," Roberto says.

His plan was to hide two screen-based devices in different areas of the park, each taking it in turns to generate passphrases which could then be entered into a treasure safe. By ensuring the devices couldn't be moved once they were found, the children would need to split into three teams – two by the devices and one by the safe. They'd wait for the passphrases to appear, and co-ordinate the unlocking over their walkie-talkies.

The devices are connected to individual power banks, making the project portable

Reach for the stars

The passphrase devices were made using a couple of Pico-Clock-Green LED-digit electronic clocks fitted with Raspberry Pi Pico boards and programmed so that they would display a series of words. Roberto ditched the safe, however, in favour of a TARDIS built out of LEGO, which he designed using Bricklink's Studio 2.0.

"The idea was that the children would find the TARDIS in distress," Roberto explains. "Its memory is scrambled and only recoverable with the right sequence of passphrases! As the children enter passphrases the windows would gradually light up with a steady light, and finally the TARDIS would be fixed."

> "The devices need to have accurate Real Time Clocks to ensure they're synchronised"

To enable the inputting of passphrases, the TARDIS was fitted with a Keybow 2040 – Pimoroni's 16-key mini mechanical keyboard, chosen because it incorporates the RP2040 chip from Raspberry Pi. It allowed Roberto to code the device to figure if the correct passphrase was being entered. He then assigned eight of the keys to a set of adjectives and the other eight to nouns. "I thought typing in passphrases letter-by-letter could get a bit laborious, so I made each one a simple combination of two words," he says.

Roberto says the KeyBow 2040 offers a "wonderfully tactile experience". The keyboard is revealed when you lift the roof off the TARDIS.

By cutting a slot in the baseboard of the KeyBow 2040, Roberto could attach a ribbon cable to its delicate pads and provide strain-relief support to the connections.

The TARDIS is made out of LEGO and incorporates a system of lights which can remain powered for 50 hours, controlled by the RP2040 chip.

▲ One of the passphrase devices carefully fitted within a tree and displaying part of a phrase

Quick FACTS

- Children are tasked with finding a broken TARDIS
- To fix it they need to find and enter passphrases
- These are entered via a Pico-powered KeyBow 2040
- Roberto simulated the project in wokwi.com
- The project cost around £200 to make

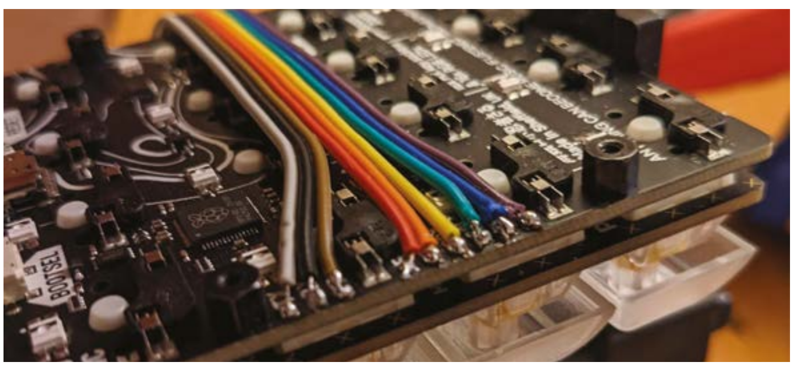
Playing for time

The passphrases had a distinct *Doctor Who* theme. "My son was intrigued by the 'Bad Wolf' mystery in season one – so I made all the passphrases like that: "Bad Wolf", "Good Duck", "Fire Goat" and so on, giving 64 possible phrases," Roberto adds. But, in order for the TARDIS to know which passphrases were being generated at any given time, the devices needed a trio of precision Real Time Clocks (RTC).

"Each passphrase is only valid for ten seconds, so the devices need to have an accurate RTC to ensure they're synchronised in their choice of passphrase," Roberto says. "To make it even harder, the passphrases needed to be shouted out over the walkie talkies as quickly as possible and the children had to enter six correct passphrases in a row to 'fix' the TARDIS."

As you can imagine, the birthday went down a treat, but it was certainly a learning curve for Roberto who hadn't even heard about Raspberry Pi Pico before he planned the project. He was still putting the finishing touches to the code as the party got started. "But it was worth all the time and effort," he says. "My son and his friends totally loved the game and I'd definitely do it all again!"

▼ They Keybow 2040 is based around the RP2040 chip by Raspberry Pi



Commodore 64 Pico emulator

Raspberry Pi's tiny but mighty Pico helps bring a classic computer back to life. **Rosie Hattersley** hears more

MAKER

Kevin Vance

A software engineer by trade, Kevin enjoys coding but only recently discovered Raspberry Pi!

magpi.cc/c64picogit

Despite his job as a software developer, Kevin Vance was new to the world of Raspberry Pi when he began to sketch out plans to revive his old Commodore 64. The retro rebuild of his first ever computer thus became his first Raspberry Pi project. Kevin had been coding since he was young, but only began to immerse himself in the world of digital making as recently as 2020.

Kevin had already started working on making an expansion board for the Commodore when he first learned about Raspberry Pi Pico. "The PIO [Programmable I/O] system sounded really interesting to develop for, and the large flash storage and number of GPIO pins made it ideal for this project, so I started on a new design around it," he explains. The first iteration of the project had a separate microcontroller, flash memory, and a voltage regulator. Kevin was "pretty excited" when he realised Pico included all of those in one module at a much lower cost.

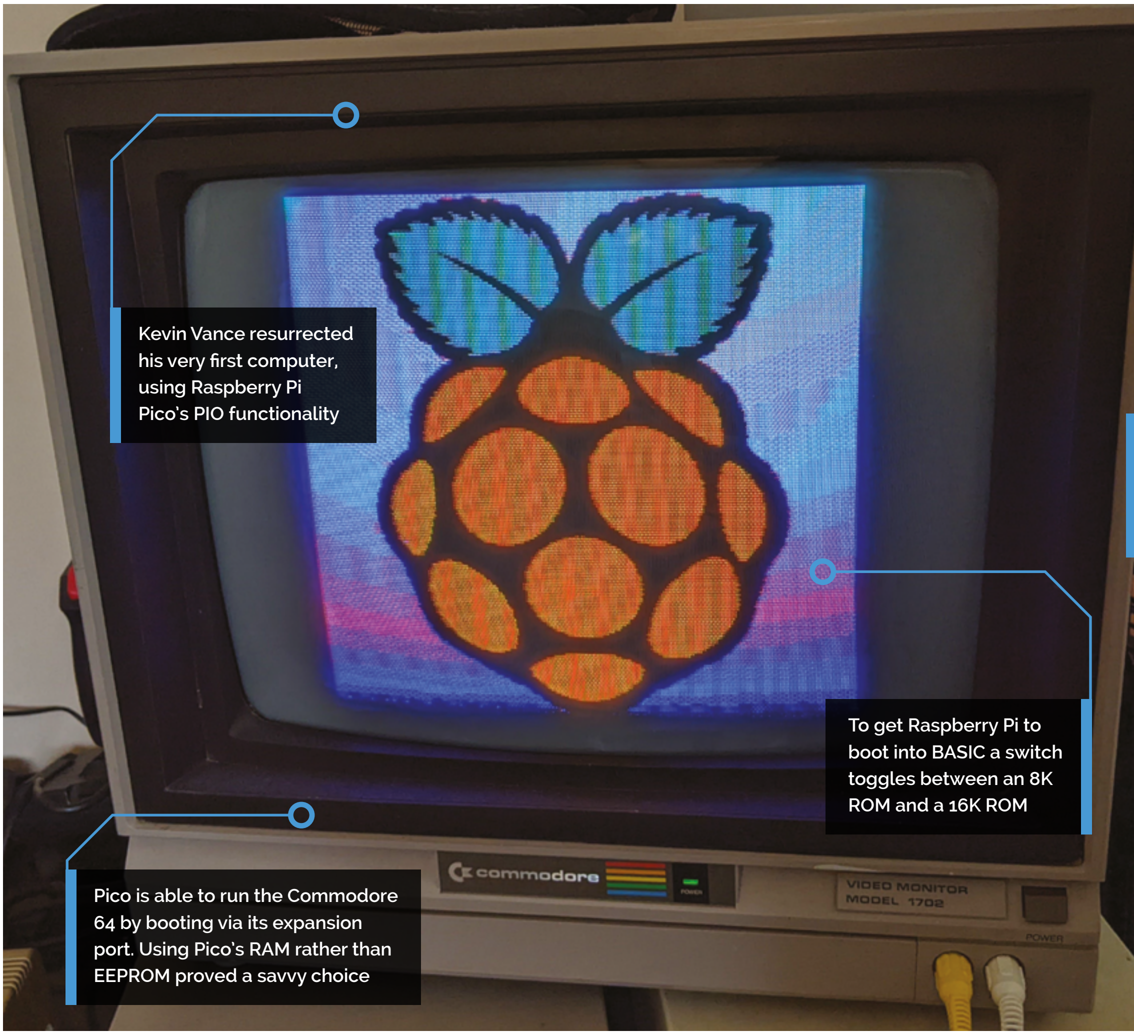
Moving the goalposts

Originally, Kevin wanted to build a Commodore 64 game cartridge with a microcontroller that the Commodore could offload work to. Having followed Ben Eater's "excellent" video walkthroughs on creating a 6502 computer on a breadboard (**eater.net/6502**), Kevin planned a similar scenario with the 6502 machine code stored on a normal EEPROM. The breadboard he designed for his updated Commodore 64 was only his second ever PCB design. Hand-soldering tiny surface-mount components with a fine-tip soldering iron was "error-prone and required patience," so investing $16 in a hot plate was "money well spent!"

Kevin wrote brand-new code for his project, with frequent updates since he kept changing how the board worked. As he gained a better understanding of how the Pico's PIO and DMA controllers could work together, he decided to investigate whether he could use Pico's RAM instead of an EEPROM. "It worked better than I expected," he tells us. "The PIO state machine could put data from the Pico's RAM on the C64's data bus without involving its CPU, well before the Commodore tried to read it!"

> "I wanted to see if I could use the Pico's RAM instead of an EEPROM. It worked better than I expected"

His biggest design challenges were the large number of pins and the Commodore's signal voltage. "Since this is an old machine with a parallel bus, there are not enough GPIO pins on the Pico to hook them all up. Fortunately, since I'm only emulating a ROM cartridge, I can get away with just 14 address lines, eight data lines, and two control lines", explains. He added 5V-tolerant buffers to translate the signals for 3.3V so the Commodore 64's 5V signals would not damage Pico's GPIO pins.



Kevin Vance resurrected his very first computer, using Raspberry Pi Pico's PIO functionality

To get Raspberry Pi to boot into BASIC a switch toggles between an 8K ROM and a 16K ROM

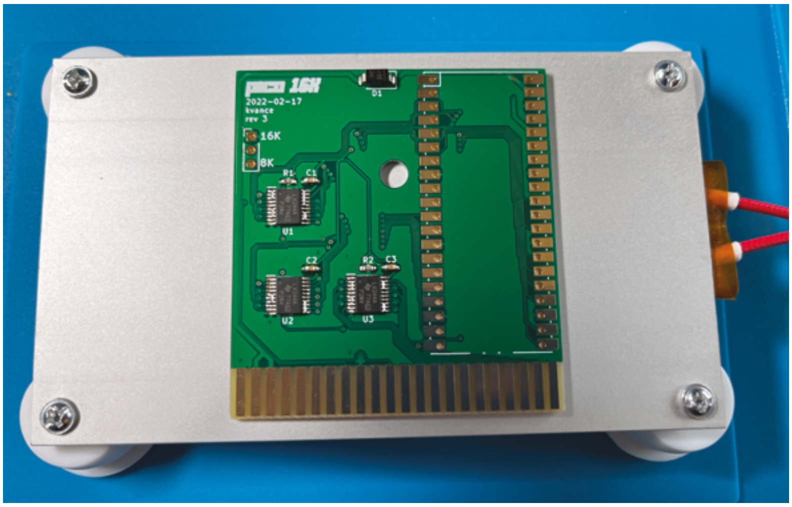
Pico is able to run the Commodore 64 by booting via its expansion port. Using Pico's RAM rather than EEPROM proved a savvy choice

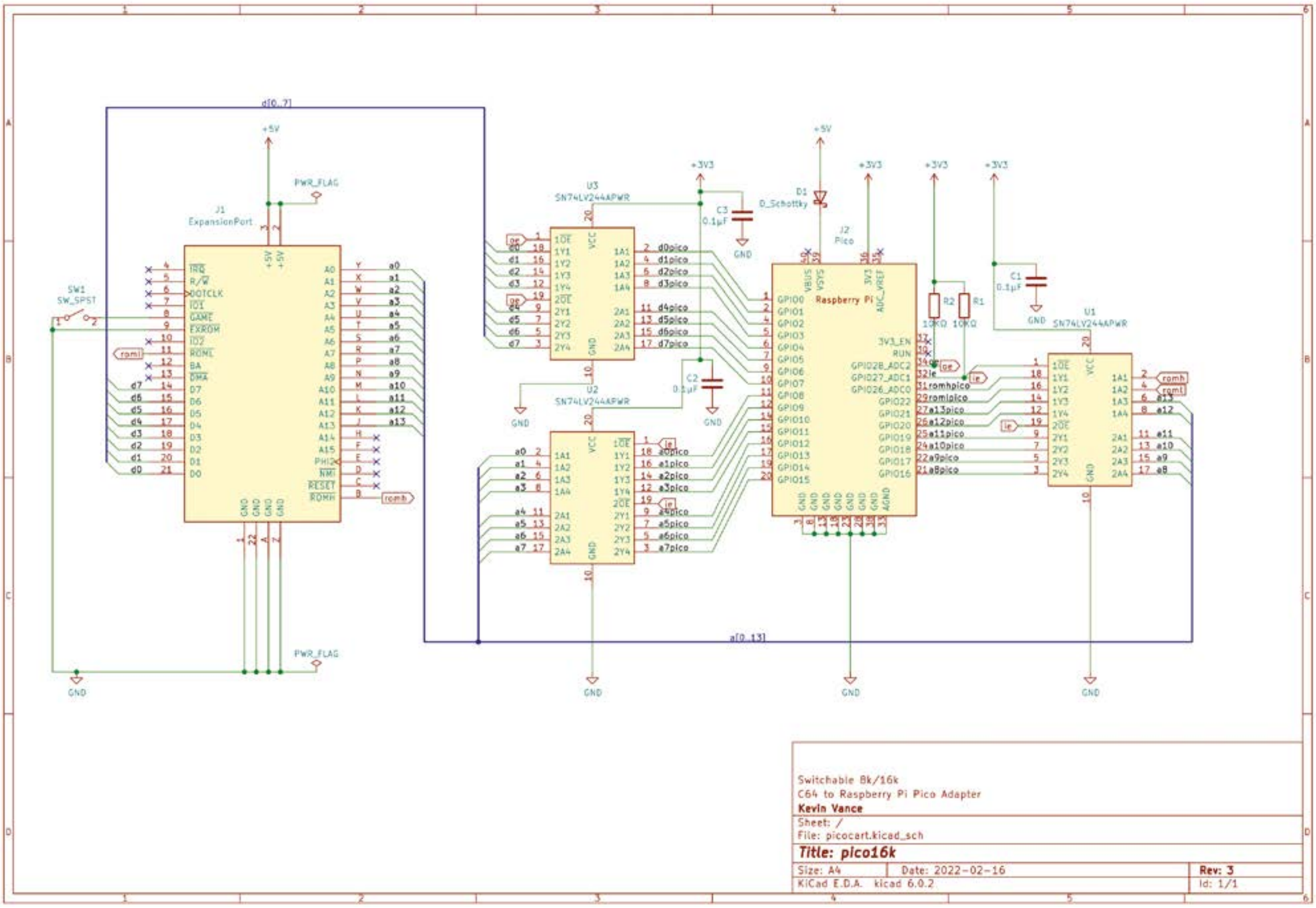
Quick FACTS

- Kevin's project rebuilt his first ever computer
- A cheap hotplate helped with soldering accuracy
- He copied the code for the game *Frogger* into Pico's RAM…
- and was astounded when the C64 booted, and *Frogger* loaded
- The whole project cost him around £27 in parts

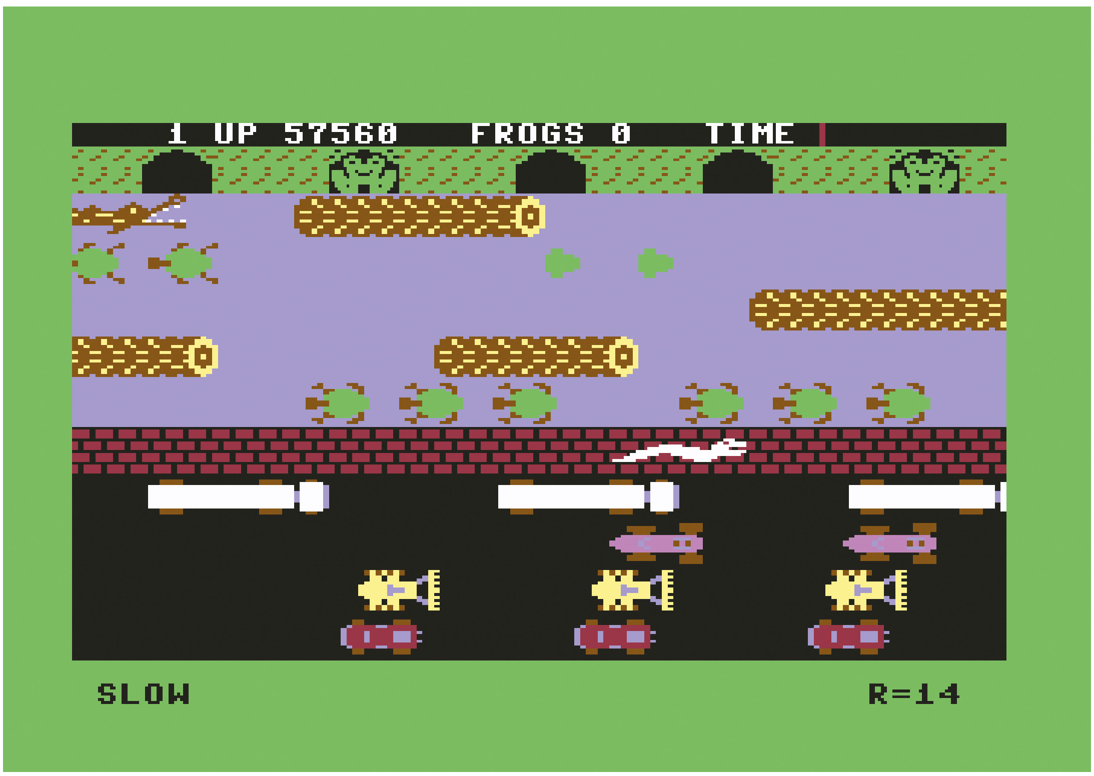
Pico's PIO system provided the biggest advantage for this build, allowing him to keep all the complexity off the CPU and run with predictable timing.

He used address decoding to send commands from the Commodore so that reading from a special 256-byte block of memory would send the address as a 'command' to Pico's CPU using the RX FIFO. He cleverly factored in time for the commands to be completed by getting the CPU to signal when it was ready for more commands using the TX FIFO. "This lets the Commodore poll the command status without interrupting the Pico's CPU," he explains. "There's a world of interesting devices that this could allow the Commodore to communicate with" – something he plans to investigate further.





▲ Kevin's GitHub page shows a detailed schematic of how Pico is able to communicate with the C64



► Kevin was delighted to find *Frogger* loaded up on his revived C64!

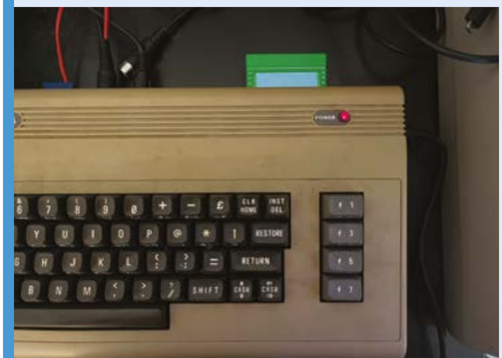
Using a second Pico as an emulator helped with the development process

Extension plans

Although Kevin is pleased to have his childhood computer back up and running, he's still curious about which C64 programs he can use Pico's CPU for – perhaps games he wrote back when his C64 was new and for which he still maintains code – and what devices he can use from the USB port.

He's also become a big fan of Raspberry Pi Pico. "It can do so much per clock cycle and, together with the DMA controller, it can function without the CPU at all once it's been initialised!" He used a second Pico to simulate the C64 bus, speeding up development of the rebuild project. This second Pico may very well become a dedicated microcontroller for automated testing.

Resurrect a Commodore 64



01 For this project you will use the expansion port on an original C64 and connect Raspberry Pi Pico. Full hardware details and code can be found at **magpi.cc/c64picogit**.

02 Follow the instructions on the GitHub to create a circuit board, and then connect to Raspberry Pi Pico.

03 Fit the resulting hardware into a small case with the expansion port connector slots exposed. Install the provided code on Raspberry Pi Pico.

CASE STUDY

Sfera Labs

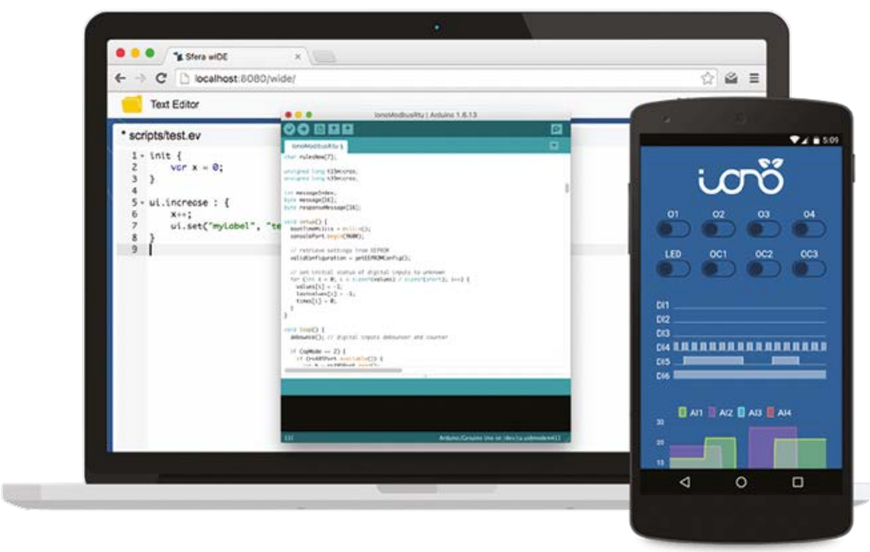
Raspberry Pi is "the platform our customers prefer for industrial applications". By **Rosie Hattersley**

We were "the first company to introduce Raspberry Pi in the industrial automation environment," is the bold claim of Sfera Labs' founder and COO Maria Chizzali. The Italian company focuses on designing, manufacturing and selling open technology for the industrial world, and much of its expertise is in industrial automation servers, I/O modules, and sensors based on Raspberry Pi technology. The idea for the business was a direct response to Raspberry Pi.

THE CHALLENGE

Sfera Labs' customers require everything from pure industrial applications using I/O for multiple sensors (for example, in the energy sector, a big market for Sfera), to Compute Module and Raspberry Pi for edge computing logic, to airport security applications. Cost and mechanical robustness are often among the key considerations, as is production lifetime. "In the industrial automation world, long-term availability is fundamental. Nobody is going to invest in an application if they are not 100% sure that in five years they will be able to use the application," Chizzali notes.

It all started when Sfera Labs' sister company developed an open platform that would let systems designers avoid lock-in with proprietary platforms. These would only work with hardware and software from the same manufacturer. Back then, if you wanted a means of controlling your heating system you'd have to buy software from the same company, Chizzali explains, and the same applied to building control and automation systems. "We had this idea of trying to develop software that was hardware-independent, so it would be able to talk to Siemens, Legrande or whatever. Sfera Labs' vision was to enable the designers to get the best parts of the best products for the project without being tied to a specific company because of the software."



THE SOLUTION

Sfera Labs started by identifying where they could offer features to help industrial customers take advantage of Raspberry Pi, such as DIN rails, UPS, and hardware enclosures. They began building expansion boards with ports, I/O modules, sensors, and power supplies that would suit a range of industrial applications. Sfera has since developed product lines around control/server units, I/O modules, and sensor modules. Sfera Labs' current devices use both Raspberry Pi Model B and Compute Module (CM).

> An open solution is better than a traditional proprietary platform in which you depend on someone else and someone else's updates

Strato Pi is a control unit known for its extensive communication protocol support, reliability, and fault tolerance, and was one of Sfera's earliest product lines.

Sfera Labs began using Raspberry Pi in Strato in February 2016, upgrading to Raspberry Pi Compute Module with Strato Pi CM in 2019. In 2020 they added a CM Duo edition featuring two SD card slots, allowing for one to be used for additional storage or as a redundant drive.

The company's other two main hardware ranges are Iono input/output control modules, and Exo environmental sensors and interfaces. Both of these support a broad range of wired and wireless communication protocols, making them ideal for adding to an existing network where they can monitor noise, humidity, seismic activity, air quality, and so on.

Iono is ideal for micro businesses such as small hotels or B&Bs, allowing them to control heating, lights, temperature, and access from a single 200 euro module. "It's been in our product line since forever," says Chizzali, "and is one of our most popular products".

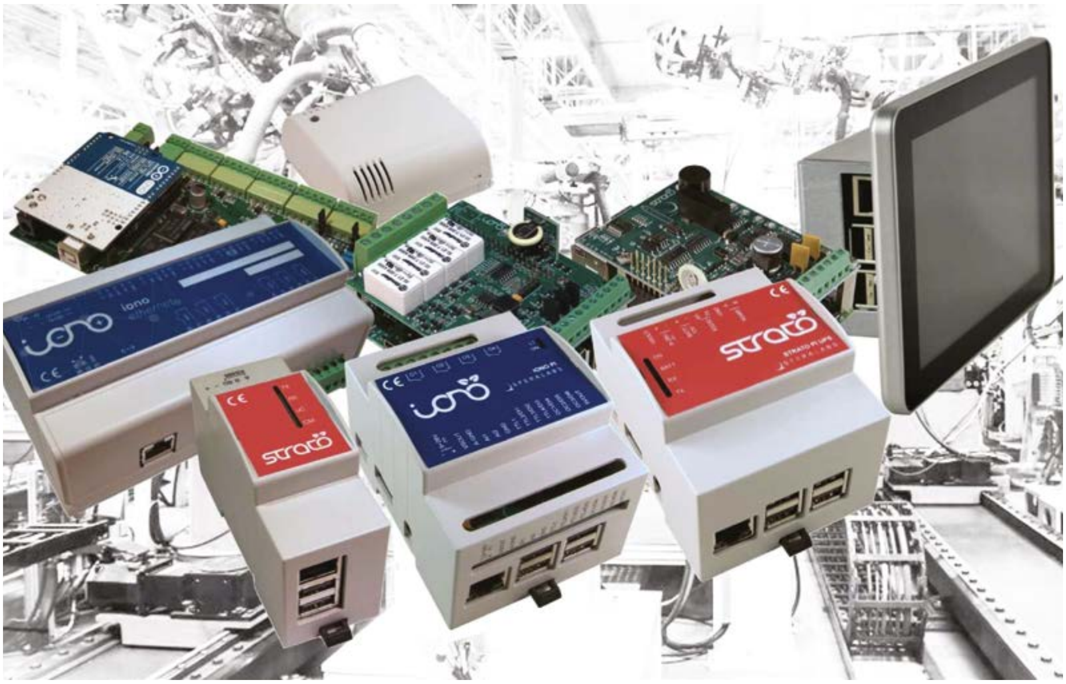
Sfera Labs' 2021 launches included Exo Sense Pi, based on Raspberry Pi Compute Module 4. Around the CM4 module Sfera has built sensors and sensor server software. Chizzali characterises a typical use case scenario in which the user wants to monitor the air quality, temperature, and ventilation of a room, and also needs to be able to detect occupancy levels. A Bluetooth sensor counts the number of people in the room and, via a wireless connection to an external speaker, the device is able to issue an audible warning in the form of a preset phrase, such as "please wait for someone to exit this room before entering".

WHY RASPBERRY PI?

Sfera Labs felt confident in choosing Raspberry Pi because of its large production volumes and the long manufacturing lifetimes to which it commits.

Open platforms are key, Chizzali believes, because of the number of people who will be knowledgeable about them and because of the support from their communities, compared to a specialised and closed proprietary platform. From a practical standpoint, "if your supplier disappears for some reason, or your programmer quits, you don't have to panic because there are many other people [who are versed in that platform] so you can find another programmer and you can find a good supplier."

This resilience is becoming increasingly appreciated, says Chizzali. Whether in professional, industrial, or building environments, people using proprietary platforms can get stuck with systems controls that they cannot change, and that hinders them, sometimes to the extent they cannot comply with new regulations and new methods of production. In industrial settings, control systems need to evolve



depending on the situation, the regulations or the market, Chizzali points out. "An open solution is better than a traditional proprietary platform in which you depend on someone else and someone else's updates."

Chizzali points out 40 million Raspberry Pi sales and a forum with close to 300,000 members whose 1.6 million posts should not be ignored in the troubleshooting and support equation. "If there

> "Raspberry Pi is definitely the platform that our customers prefer for industrial applications"

is a problem – and it is impossible that problems do not exist – whenever there is a problem, if you have a proprietary platform, all you can do is wait for the next release. If you have a problem on an open platform with community forums the size of those of Raspberry Pi and Arduino, the community very often finds a solution in just a matter of days."

Raspberry Pi also appeals for prototyping, not least because it is low-cost. "You buy a couple of Raspberry Pi boards, developers software test it and, when you're done, you have software that is ready," says Chizzali. You don't need to develop your own hardware such as an embedded application or device. Faster prototyping means reaching the production stage sooner and being quicker to market.

As a result, Raspberry Pi is "definitely the platform where we are doing most of our business, definitely the platform that we're doing most of our research and development on and it's definitely the platform that our customers prefer for industrial applications."

THE RESULTS

Sfera Labs is active in no fewer than 62 countries, its main markets being North America, Europe — particularly Germany and the Netherlands — and Australia. Energy sector customers are a key segment. "By their nature, energy companies and types of application are new," explains Chizzali. "They don't have anything that's 'legacy', so obviously they use Raspberry Pi and software based on it." Emerging markets in Africa are also becoming a good market for Sfera Labs as a result.

Long-term clients are satisfied. Hi-Interiors, which makes the Hi-Can Smart Bed, has worked with Sfera Labs for a decade. COO and co-founder Gianni Tallarico explains: "Hi-Interiors' expertise is in the digital transformation of furniture and we wanted to work with a partner who had a similar strength in electronics and software development. Sfera Labs leveraged Raspberry Pi's open platform to design the control system for HiCan and HiAm using the Strato Pi server and Exo Sense sensors. The success of this project led to a ten-year relationship between the two companies. The modular system architecture that Sfera Labs developed greatly simplified the transition to the second-generation smart bed, the Hi Bed."

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE FOR THE MODERN MAKER

SUBSCRIBE AND SAVE UP TO 35% on the cover price

ISSUE #54

OUT NOW

hsmag.cc

Available on the App Store

GET IT ON Google Play

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

Subscriber Benefits

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

Rolling Monthly Subscription

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

Subscribe for 12 Months

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

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

Subscribe by phone: **01293 312193**

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

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

JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero 2 W

WITH YOUR FIRST 12-MONTH SUBSCRIPTION

Subscribe in print today and get a FREE computer!

WORTH $15

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

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

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From £2.29

Available on the App Store

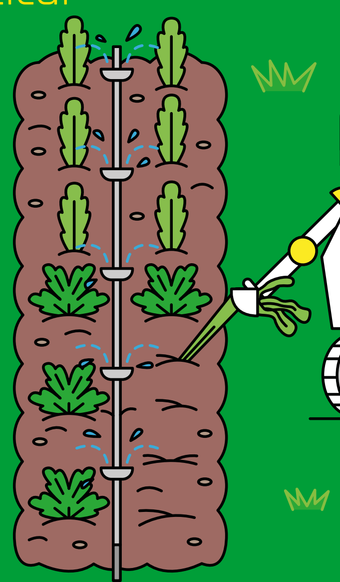
GET IT ON Google Play

BUILD A RASPBERRY PI SMART GARDEN

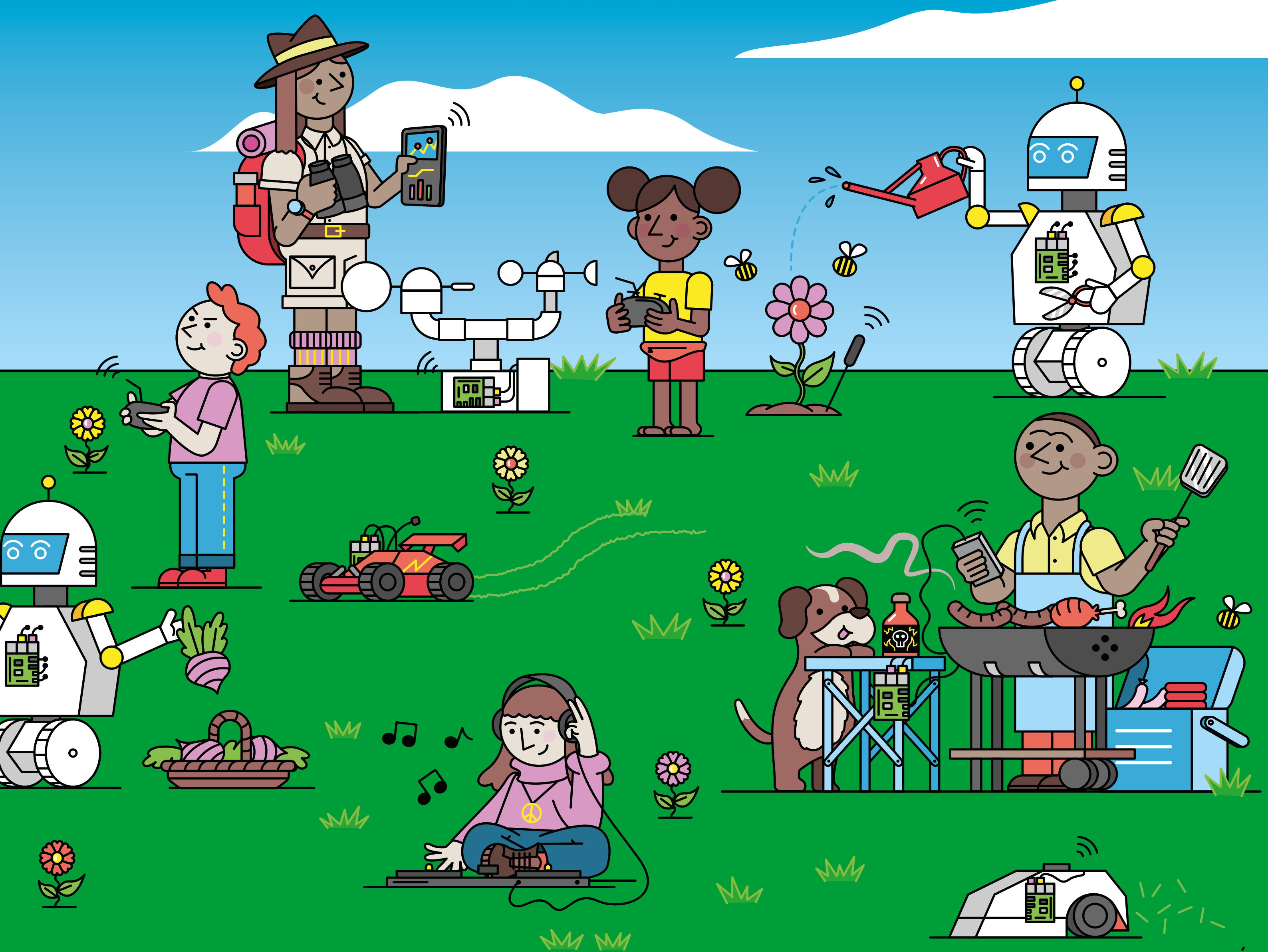
Get outdoors and use a Raspberry Pi for practical and leisure projects. By **Rob Zwetsloot**

As we write this, Spring has sprung here in the UK, which means about seven minutes of sunlight has drawn everyone outside to touch some grass and maybe head to the beach. If you want to leave your house but also want to keep working on a Raspberry Pi project, how about combining both?

Like their indoor project counterparts, outdoor builds range from the most practical projects for monitoring crops and flower beds, to fun leisure makes to enjoy the sun in your own way. Get your suntan lotion ready, find your shades, and let's get out there.



CONTENTS



AUTOMATED GARDENING

Get robots to grow your veg

MUDPI

MAKER ERIC DAVISSON

A customisable and scalable automated gardening system allows you to monitor soil moisture, temperature, humidity and rain data and you can use all of that data to control an irrigation system. This level of control helps not only save water, but also grow plants betters. Its creator keeps expanding on the hobby version he created, which he has created guides for if you wish to follow along.

mudpi.app

This small box contains almost all you'll need to control your garden

Eric's automated garden keeps getting bigger

LOW-COST SOLUTION

Léo Galley from Brooklyn showed us their project which uses a series of internet-connected Raspberry Pi Zero boards with temperature and moisture sensors attached that allows a team at a big farm complete their rounds faster than before, giving up-to-date and accurate information on the conditions in storehouses and greenhouses. You can see it in action at **farmsensordashboard.com**

MAKER
ETIENNE DENEAULT

SMART GARDEN BBQ

This small smart garden also uses something else we'll be covering in these pages – a BBQ. Although not in the traditional sense – this broken Weber grill got some soil, seeds, sensors, and water installed in it so it could grow some plants. It's a neat little planter and you could even grow some veg on it to put on a working grill.

magpi.cc/smartgarden

▲ The Weber aesthetic is maintained even if it's not quite being used in the way it was designed for

ROBOT WORKERS

Want a more traditional robot looking after your farm? This Land Care Robot from **directedmachines.com** is a solar-powered Raspberry Pi robot that is quite powerful, with a 42HP electric motor, special cameras, navigation systems, and more.

◄ A custom wooden frame surrounds the old BBQ

FARMBOT

MAKER
FARMBOT

This truly robotic farming system allows you to set up a custom growing patch with a variety of produce. You set the whole thing up using an app that lets you maximise the space of your custom patch, and then a robot arm is used to plant the seeds, weed, and water the whole system. You can keep tabs on it with your phone remotely, and even do some manual control as well. It's very smart but you do pay for it.

farmbot.io

▼ The robot arm runs along the side of the patch you create for more efficient farming

POWERED-UP PATIO

Upgrade your back garden with Raspberry Pi

FLIRT PI INTERNET RADIO

There's a certain aesthetic to sunbathing like it's the '70s, with a white plastic sunbed and big hat. Complete it with a Raspberry Pi internet radio built from an old radio, allowing you to listen to new tunes on your retro tanning kick.

magpi.cc/flirtpi

MAKER
MARTIN MANDERS

▲ The finished product looks great and you won't misplace it with that orange case

◀ It's portable, so take it anywhere you go

HEATERMETER

MAKER
BRYAN MAYLAND

While BBQs are great, sometimes you can easily end up with an overcooked bit of spicy chicken, or a hot dog sausage burnt to a crisp. Improve your grilling (and smoking) with the HeaterMeter, which allows you to monitor the temperature of the BBQ itself (and control it if you have the right fan addition) and any food you probe as well so it hits the exact right temperature.

▲ A better UI for tracking your smoking

heatermeter.com

DIY WEATHER STATION

MAKER
SPACEMANLABS

There is a Raspberry Pi project you can use for building a weather station (**magpi.cc/weatherstation**), however there are also some cheaper ways to try it out that don't require a kit. This tutorial shows you how to build and code your own weather station, and is a great summer project.

magpi.cc/diyweather

ROBO MOWERS

Robot hoovers exist. Cars can self-drive. Surely, your humble lawnmower can be roboticised? Yes.

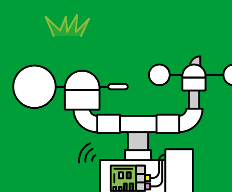
PIMOWBOT

The folks behind this and other mowers are constantly showing us updates to their line of automated, solar-powered grass choppers. Although a bit more dangerous than a Roomba, the tech is still very promising

RASPIMOWER DALEK

Want something a bit more sinister and robot-looking to keep your grass trimmed? Look no further than this Dalek, which exterminates long grass. Is it a bit overkill? Possibly, but it is also quite the spectacle.

▲ Building your system to be waterproof is very important



◄ The complete weather station allows you to keep track of the elements

FUN IN THE SUN

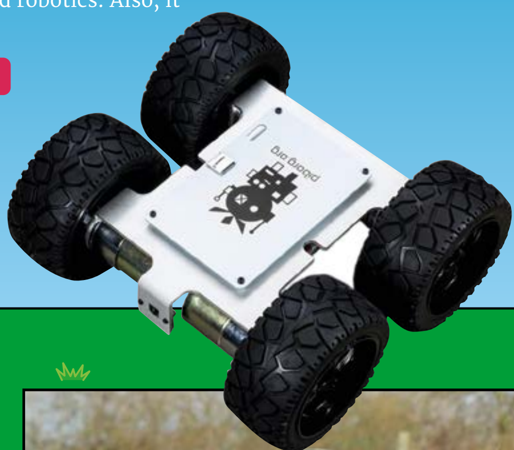
Everyone needs to make sure to take a break

MONSTERBORG

MAKER PIBORG

A very cool Raspberry Pi robot that can be used as a remote control car out of the box. It's fast, sturdy, and very high quality. You can also program it to be automated, and a is a great intro to programming and robotics. Also, it looks very cool.

magpi.cc/monsterborg



Angles like this make RC cars look ridiculously exciting

PI DECK

MAKER DANIEL JAMES

Have a party? Want to control the music for it? Then you should build a Pi Deck so you can spin your own digital music for your guests. It's a lot more involved than just hitting play on Spotify, and other folks can have fun with mixing, and live the true DJ experience in your garden

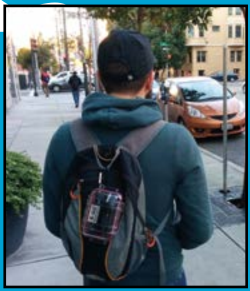
magpi.cc/pideck

The records may not be physical but you can still scratch and mix

GPS LOGGER

Matt made this for tracking his trips to work, but you can easily use it to track hikes and other walking routes. It can run off a Raspberry Pi Zero, so it doesn't take up much space in your bag.

magpi.cc/40



MAKER
MATT RICHARDSON

You can keep your one inside the bag if you wish

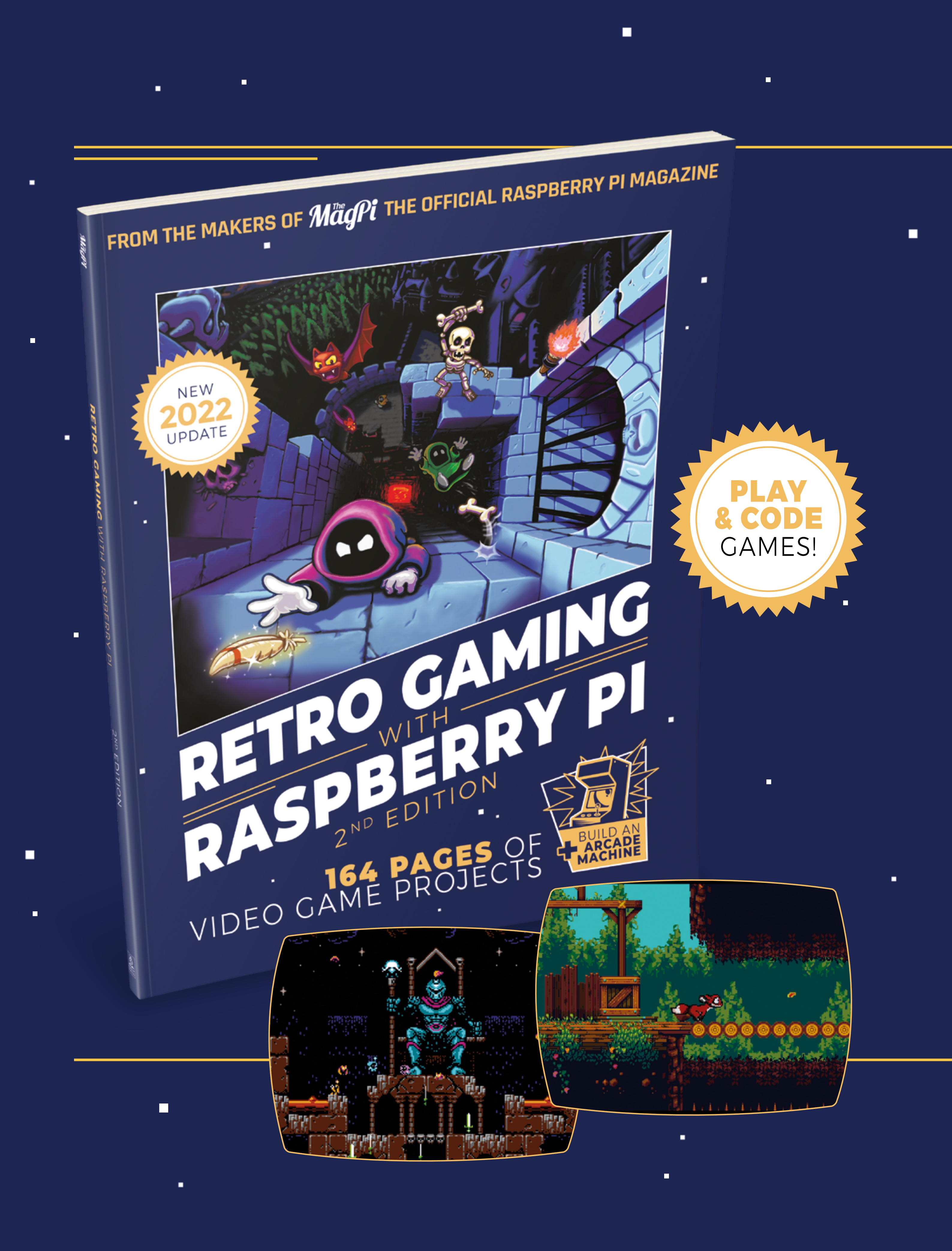
DRONE PI

MAKER
REGLISE44

Flying is cool. Remote control flying machines are cooler. Raspberry Pi-powered drones are the coolest. This is a great DIY build that helps you build a large drone with a Raspberry Pi. There are a few kits, and smaller versions as well, but this is a great way to learn.

magpi.cc/dronepi

This is a bit bigger and more advanced than your average drone


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

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**

Part 02

Learn ARM assembly: Welcome to the world of 64-bit

Learn to code a small assembly language program for Raspberry Pi OS (64-bit)

MAKER

Stephen Smith

Stephen is a retired software developer who has written three books on ARM assembly language programming. He is a member of the Sunshine Coast Search and Rescue and enjoys mountain biking, hiking, and running. He is also a member of the Sunshine Coast Writers and Editors Society (**scwes.ca**).

magpi.cc/stephensmith

Raspberry Pi recently included Raspberry Pi OS (64-bit) in its Imager program, making the 64-bit edition of its operating system widely available.

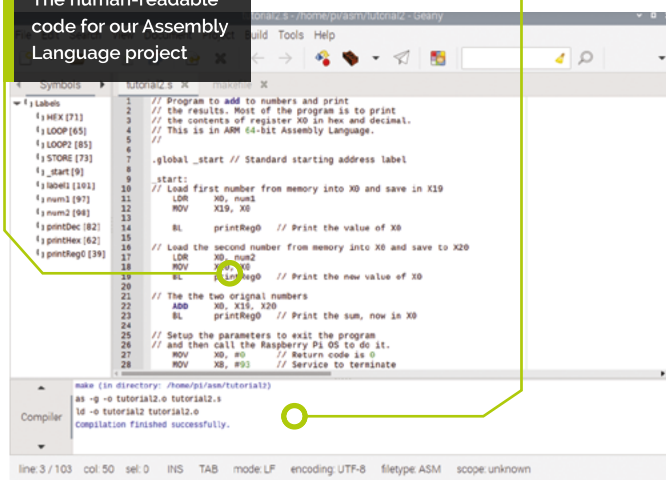
In this tutorial, we will create a simple program in ARM 64-bit assembly language to perform some calculations and print out the values. We will learn how to access memory, use 64-bit registers, construct loops, and perform conditional logic. Much of what we learned last time with our 32-bit tutorial applies here, for instance, using the gdb debugger is the same in both 32 and 64 bits.

01 Creating the program

The source code for this tutorial is in **tutorial2.s** and a **makefile** file is provided (**magpi.cc/learnassembly2**). Start by downloading both of the files.

The process to build this 64-bit program is identical to what we did previously for ARM 32-bit (see The MagPi magazine issue #116, **magpi.cc/116**). The only difference is that the tools bundled with Raspberry Pi OS (64-bit) will compile for the ARM CPU in 64-bit mode.

There are a few differences when working with ARM 64-bit. Comments use double slashes ('//') rather than the '@' sign. The program still starts execution at the `global _start` label.



Build your project directly in Geany

The human-readable code for our Assembly Language project

In the following steps, we discuss small parts of the program and how they work. To build and run this program create a directory folder named tutorial2. From a command prompt use:

```
mkdir tutorial2
```

Place the two source files you downloaded in this folder, change the working directory to this folder and run make then run the program:

```
cd tutorial2
make
./tutorial2
```

You will see the output:

```
 Reg0: 0x1234567890ABCDEF
01311768467294899695
 Reg0: 0x00000000FEDCBA90
00000000004275878544
 Reg0: 0x123456798F88887F
01311768471570778239
```

This program loads two 64-bit integers from memory into the registers and adds them together. Most of the program is the code to format the numbers into printable form both in base 16 hexadecimal and the usual base 10 decimal. Let's study how the program does this.

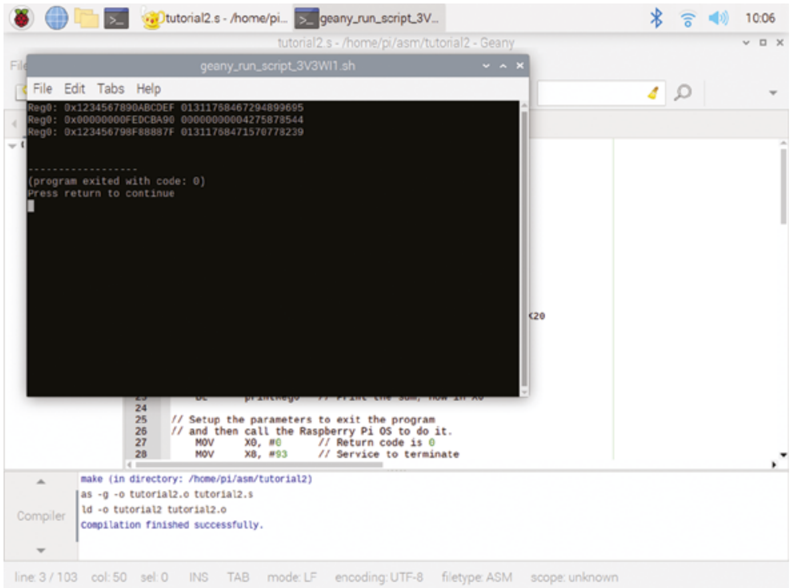
Open Geany (menu > Programming > Geany Programmer's Editor) and open **tutorial2.js**. Geany will make it easier to examine the code. (You can choose Build > Make and Build > Execute to run the program inside Geany).

02 Using 64-bit registers

In the previous tutorial, we used the 32-bit registers `R0` to `R15`. These are not available in the 64-bit world. The general-purpose 64-bit registers are named `X0` to `X30`, with the addition of a combination stack pointer/zero registers `SP/XZR`, a program counter `PC`, and a program status register. You can access the lower 32-bits of each general-purpose register using the names `W0` to `W30` and `WZR`.

Even though we are in the 64-bit world, each executable instruction still compiles to a 32-bit word. This keeps programs compact and allows instructions to be loaded and processed quickly. This is a primary reason why RISC CPUs execute most instructions in one clock cycle and use very little power to do so. This is a challenge for programmers, since memory addresses are 64 bits in length and we need to load these into registers. There are many tricks to accomplish this, and we will see two of them in this tutorial.

The first instruction is `LDR` on line 11 and it loads `num1` into register `X0`. The number num1 is defined at the end of the code. The `LDR` instruction loads a value from memory and builds the address using `PC` relative addressing. The assembler calculates the offset of `num1` from the current location where the program is executing and uses that offset. This only works for read-only constants since you cannot write to the executable section of a program in Linux.



▲ Running the program from inside Geany

```
  // Load first number from memory into X0 and
save in X19
        LDR             X0, num1
```

Move `X0` to `X19` to save the value. The routine that prints the values is destructive and erases the value in the process of printing it.

```
        MOV             X19, X0
```

Branch to the subroutine to print out the value of register `X0`.

```
        BL              printReg0          //
Print the value of X0
```

The program loads the second number, saves it to `X20` and prints it out.

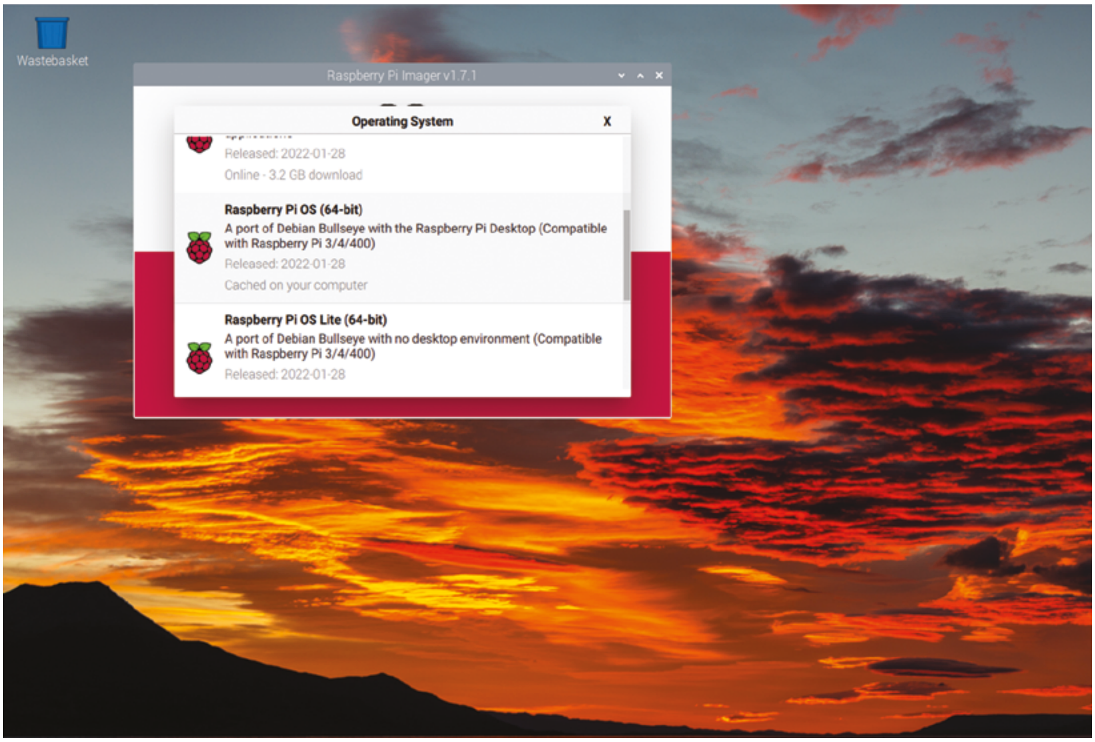
```
        LDR             X0, num2
        MOV             X20, X0
        BL              printReg0          //
Print the new value of X0
```

Add the two 64-bit numbers, storing the sum in `X0`, in preparation to print it out.

```
        ADD             X0, X19, X20
        BL              printReg0          //
Print the sum, now in X0
```

…

```
  num1: .quad 0x1234567890ABCDEF   // first
number
  num2: .quad 0xfedcba90             //
second number
```



Use Imager to install Raspberry Pi OS (64-bit)

03 The routine to print X0

The routine to print `X0` loads a template string to overwrite with the correct values. It calls separate routines to format the register first in hex, then in decimal into the template string. It then uses Raspberry Pi OS write to file service to output the finished string to stdout.

In 64-bit there is no push or pop instruction, instead STR pushes and `LDR` pops. In this case, we save the link register (`LR`), which is register `X30`, to the stack. We need to do this since we call two additional routines, and they will overwrite `LR`.

```
 printReg0:
        STR             LR, [SP, #-16]!   //
Save the return address
```

The following `LDR` instruction loads the address of the template string into register `X1`. But this instruction is only 32-bits in size and the address of label is 64-bits? How does this work? The assembler accomplishes this by placing the address of `label1` in with the code next to the numbers entered, then uses `PC` relative addressing to load this address. Be glad we don't have to do this by hand.

```
        LDR             X1, =label1
// Load address of label and buffer
```

The printHex routine formats `X0` in hexadecimal to the buffer provided in `X1`. The `printHex` routine changes `X0` and `X1` as it processes.

Save the original values, ready to print in decimal later.

```
        MOV             X7, X1            //
Keep a backup for later
        MOV             X9, X0            //
Keep a backup of register to print
        ADD             X1, X1, #7
// Skip "Reg0: 0x"
        BL              printHex          //
Print X0 in hex
```

Restore the registers and do the same to add `X0` in decimal to the correct place in the template string.

```
        MOV             X0, X9            //
Restore original value of X0
        MOV             X1, X7            //
Restore original buffer
        ADD             X1, X1, #24
// After hex value and a space
        BL              printDec          //
Print the value in decimal
```

The next section sets up the parameters for the Linux write file routine. The Linux service numbers are different in 64-bit than in 32-bit.

```
  // Print out the finished string using the
Linux service
        MOV             X0, #0            //
Stdout
        MOV             X1, X7            //
```

```
Original address of the string
        MOV                 X2, #46           //
Length of the string
        MOV                 X8, #64           //
Linux service to write to a file
        SVC                 0
// Perform the operation
        LDR                 LR, [SP], #16     //
Restore the return address
```

Also, ARM 64-bit Assembly Language has a function return instruction, and we need to use it here:

```
        RET
  …
  .data
  label1: .ascii "Reg0: 0x
\n\n\n"
```

04 Format X0 in hexadecimal

The `printHex` routine formats the number into hexadecimal. This routine contains a conditional statement and a loop.

The algorithm works by masking off the low order digit by `AND`'ing it with `0xF` which is four single bits.

```
                    29B5
          AND       000F
                       5
```

The program then adds '0' to this which converts the digit to the range of ASCII characters '0' to '9'. If the digit was in the range 0xA to 0xF then we subtract 10, to put it in the range 0 to 5 and add the 'A' character.

To get the next digit we shift the number to the right by four bits.

```
                              29B5
        SHIFT RIGHT         4
                               29B
```

After shifting, repeat `AND`'ing the number with 0xF to get the next digit. We are retrieving the digits in right to left order, therefore building the ASCII number in the buffer from right to left. There are 16 hex digits in a 64-bit integer, so we set `X2` our loop counter to 16. We then add 16 to the buffer, so it points to the last position.

```
  printHex:
        MOV      X2, #16  // 16 hex chars in
64-bit
        ADD      X1, X1, #16 // Print in
reverse so start at back
```

The next part is the loop. It gets the low order 4-bits by ANDing the register with 0xf and then places the result in `X3`. It then compares the digit to 10 to see if we are formatting '0' to '9' or 'A' to 'F'. The compare function performs a subtraction and sets bits in the program status register appropriately.

Branch to `HEX` on plus, meaning the result of the subtraction is greater than or equal to zero. If the result is less than zero execution continues and ASCII '0' is added to `X3` to convert to a printable ASCII character, then jump over the part to convert the other values.

```
  LOOP:
        AND      X3, X0, #0xF // And off the
low order nibble
        CMP      X3, #10  // Is the character
0-9 or A-F?
        B.PL     HEX                // if >=10
branch
        ADD      X3, X3, #'0' // Convert to
ASCII
        B        STORE              // Jump
ahead to store
```

The hex part converts a value of 10-15 to A-F. We can put a constant equation in the operand which the assembler will evaluate for us.

```
  HEX:
        ADD      X3, X3, #('A' - 10) //
Convert to ASCII
```

makefile

Language: **Make**

DOWNLOAD THE FULL CODE:

magpi.cc/learnassembly2

```
001.  tutorial2: tutorial2.o
002.          ld -o tutorial2 tutorial2.o
003.
004.  tutorial2.o: tutorial2.s
005.          as -g -o tutorial2.o tutorial2.s
```

tutorial2.s

> Language: **Assembly language**

DOWNLOAD THE FULL CODE:

magpi.cc/learnassembly2

```
// Program to add to numbers and print
// the results. Most of the program is to print
// the contents of register X0 in hex and decimal.
// This is in ARM 64-bit Assembly Language.
//

.global _start // Standard starting address label

_start:
// Load first number from memory into X0 and save in X19
        LDR             X0, num1
        MOV             X19, X0

        BL              printReg0       // Print the value of X0

// Load the second number from memory into X0 and save to X20
        LDR             X0, num2
        MOV             X20, X0
        BL              printReg0       // Print the new value of X0

// The two orignal numbers
        ADD             X0, X19, X20
        BL              printReg0       // Print the sum, now in X0

// Setup the parameters to exit the program
// and then call the Raspberry Pi OS to do it.
        MOV             X0, #0          // Return code is 0
        MOV             X8, #93         // Service to terminate
        SVC             0               // Call Linux to perform

// Function: printReg0
// Purpose: print out the contents of register X0 in both
// hexadecimal and decimal to stdout.
// Builds the string:
//              Reg0: 0xhexvalue decimalvalue
//              Registers overwritten: X0, X1, X2, X7, X8, X9
//

printReg0:
        STR             LR, [SP, #-16]! // Save the return address
        LDR             X1, =label1     // Load address of label and buffer
        MOV             X7, X1          // Keep a backup for later
        MOV             X9, X0          // Keep a backup of register to print
        ADD             X1, X1, #7      // Skipp "Reg0: 0x"
        BL              printHex        // Print X0 in hex
        MOV             X0, X9          // Restore original value of X0
        MOV             X1, X7          // Restore original buffer
        ADD             X1, X1, #24     // After hex value and a space
        BL              printDec        // Print the value in decimal

// Print out the finished string using the Linux service
        MOV             X0, #0          // Stdout
        MOV             X1, X7          // Original address of the string
        MOV             X2, #46     // Length of the string
        MOV             X8, #64         // Linux service to write to a file
        SVC             0               // Perform the operation
        LDR         LR, [SP], #16 // Restore the return address
        RET

// printHex: print register X0 in hex to buffer pointed to by X1
//              Registers overwritten: X0, X1, X2, X3
printHex:
        MOV             X2, #16         // 16 hex chars in 64-bit
        ADD             X1, X1, #16 // Print in
```

```
reverse so start at back
LOOP:
        AND         X3, X0, #0xF // And off the low order nibble
        CMP         X3, #10          // Is the character 0-9 or A-F?
        B.PL    HEX                  // if >=10 branch
        ADD         X3, X3, #'0' // Convert to ASCII
        B           STORE            // Jump ahead to store
HEX:
        ADD         X3, X3, #('A' - 10) // Convert to ASCII
STORE:
        STRB        W3, [X1], #-1    // Save and post decrement
        MOV         X0, X0, LSR 4    // Divide by 16 by shifting 4 bits
        SUBS    X2, X2, #1           // Decement loop counter
        B.NE    LOOP                 // Loop if not zero
        RET

// printDec: print register X0 in decimal to buffer X1
//              Registers overwritten: X0, X1, X2, X3, X8
printDec:
        MOV         X2, #20          // 20 decimal chars in 64-bit
        ADD         X1, X1, #20 // Fill in buffer in reverse
LOOP2:
        MOV         X8, #10          // Decimal is base 10
        MOV         X3, X0           // Keep original number
        UDIV    X0, X0, X8       // X0 = X0 / 10
        // Calculate the remainder using MSUB.
        MSUB    X3, X0, X8, X3 // X3 = X3 - (X0 * 10)
        ADD         X3, X3, #'0'// Convert digit to ASCII
        STRB    W3, [X1], #-1 // Store the value and post decrement
        SUBS    X2, X2, #1 // Decrement loop counter
        B.NE    LOOP2 // Loop if not zero
        RET

num1: .quad 0x1234567890ABCDEF // first number
num2: .quad 0xfedcba90 // second number

.data
label1: .ascii "Reg0: 0x \n\n\n"
```

Stephen's Stuff

Stephen has written three books on assembly language programming. The second one is *Programming with 64-Bit ARM Assembly Language* which is the place to go for a deeper understanding of the topics touched on in this tutorial. The first one is *Raspberry Pi Assembly Language Programming* for 32-bit ARM code and the third one is *RP2040 Assembly Language Programming* for the Raspberry Pi Pico.

Store the hex value in the buffer with a STRB instruction which stores a byte from the register and decrements the buffer by 1 after storing.

```
 STORE:
        STRB     W3, [X1], #-1     // Save and
post decrement
```

Shift the value in X0 right by 4 bits using the logical shift right option of a MOV instruction. This positions the next digit for our conversion.

```
        MOV      X0, X0, LSR 4     // Divide
by 16 by shifting 4 bits
```

Decrement the loop counter with a SUBS instruction. An 'S' on the end of an instruction mnemonic means to set the program status register based on the result. This way we can branch on not equal to zero right away without needing an additional CMP instruction. If the loop counter equals zero then we are done and can return.

```
        SUBS     X2, X2, #1                //
Decement loop counter
        B.NE     LOOP                      //
Loop if not zero
        RET
```

05 Format X0 in decimal

In the printHex routine, we wanted the quotient and remainder of dividing X0 by 16. We used the trick of getting the remainder by ANDing with 0xf and the quotient by shifting the number right by 4 bits. To print in decimal, we need the quotient and remainder by dividing by 10. To do this we use the unsigned division instruction UDIV. There is no instruction to directly give the remainder of the division, but we can easily construct it with a single multiply/subtract instruction.

```
 printDec:
        MOV      X2, #20  // 20 decimal chars
in 64-bit
        ADD      X1, X1, #20 // Fill in buffer
in reverse
 LOOP2:
        MOV      X8, #10  // Decimal is base
10
        MOV      X3, X0   // Keep original
number
        UDIV     X0, X0, X8       // X0 = X0
/ 10
        MSUB     X3, X0, X8, X3 // X3 = X3 -
(X0 * 10)
        ADD      X3, X3, #'0'// Convert digit
to ASCII
        STRB     W3, [X1], #-1 // Store the
value and post decrement
        SUBS     X2, X2, #1       //
Decrement loop counter
        B.NE     LOOP2            // Loop if
not zero
        RET
```

Let's look at an example to better understand how we retrieved the right digit for formatting.

```
                1654
          /       10
                 165
```

The integer division had the desired effect of shifting the number right by one digit. To construct the remainder, which we need:

```
  1654                        1654
         - (10 * 165)   ==>  - 1650
                                          4
```

Tricks like this are typical in RISC, where the goal is to keep the instruction set small as possible.

06 Modify the program

This simple program incorporates many elements required to program in any language, including performing calculations, executing loops, performing conditional logic, and accessing memory. The program only prints out unsigned integers, so why don't you add the logic to print out the decimal part signed? It takes some conditional logic to add the '–' sign and use the NEG instruction to switch the number from negative to positive. If you have a different formatting preference, you can adjust the program, perhaps using leading spaces rather than zeroes for the decimal number. Or try performing different calculations and observing the outputs.

English not your mother tongue?

The MagPi is also available in German!

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

Use the coupon code 115PicoDE

on www.magpi.de/115

Part 02

CDP Studio: Sense HAT

Use the low-code, block-based environment of CDP Studio to read sensors on a Sense HAT

WRITER

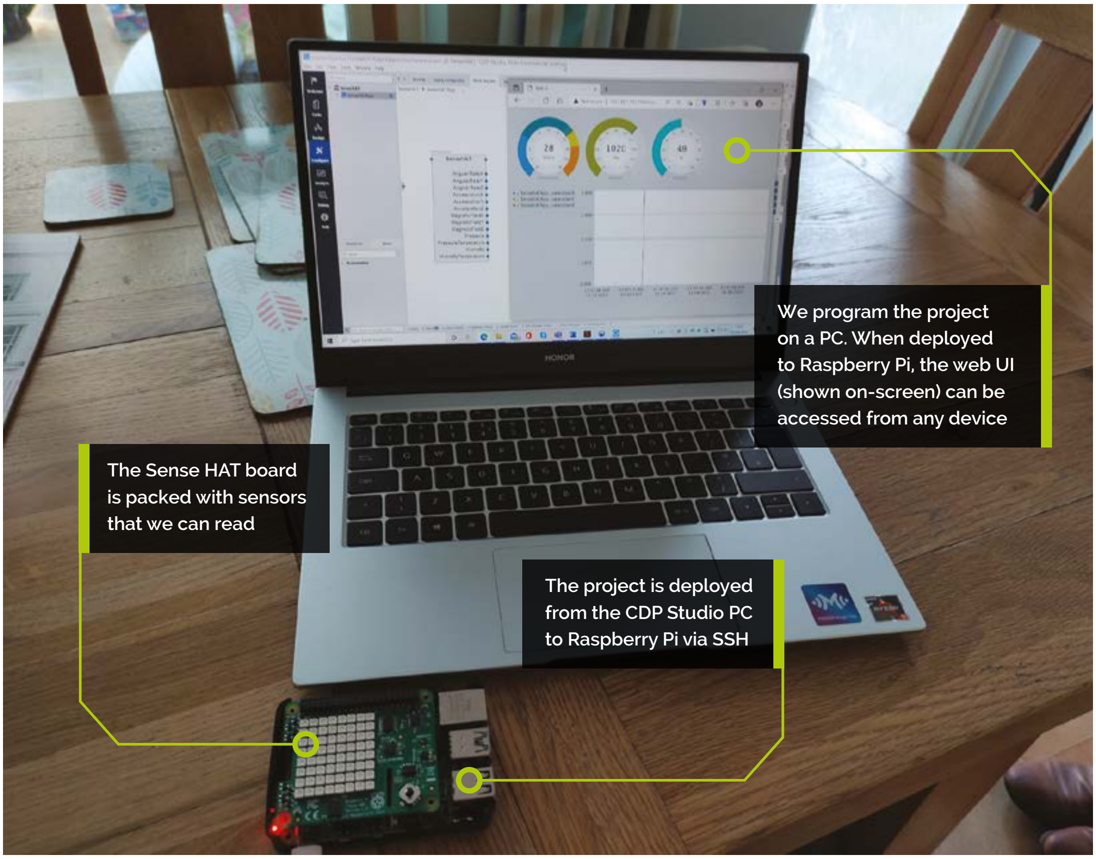
Phil King

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

@philkingeditor

As an 'out of the box' software development tool, CDP Studio is used to build industrial control, automation, and edge systems. Yet it's fairly easy to get to grips with its low- (or even no-) code programming environment, and you can deploy projects to a Raspberry Pi.

Last time, we created an app to flash LEDs connected to a Raspberry Pi in different patterns, selectable on a web GUI. In this second tutorial, we'll be reading the signals from the numerous sensors on a Sense HAT attached to our Raspberry Pi. Using CDP Studio's built-in 'SenseHAT' recipe, this is very easy to do – programming-wise, this project only involves a single preset block in the Block Editor. We can then display the readings with a variety of widgets and graphs in a web GUI that can be viewed in a browser on any device on the network.



We program the project on a PC. When deployed to Raspberry Pi, the web UI (shown on-screen) can be accessed from any device

The Sense HAT board is packed with sensors that we can read

The project is deployed from the CDP Studio PC to Raspberry Pi via SSH

You'll Need

- Linux or Windows PC
- CDP Studio **cdpstudio.com/getstarted**
- Raspberry Pi
- Raspberry Pi OS (Bullseye or Legacy version)
- Sense HAT

01 Install the software

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

02 Prepare Raspberry Pi

If you've followed the first tutorial, your Raspberry Pi should already be ready to use with CDP Studio. If not, you'll need to prepare it, which involves enabling SSH and modifying its **/etc/security/limits.conf** file – see **magpi.cc/cdprpisetup** for details.

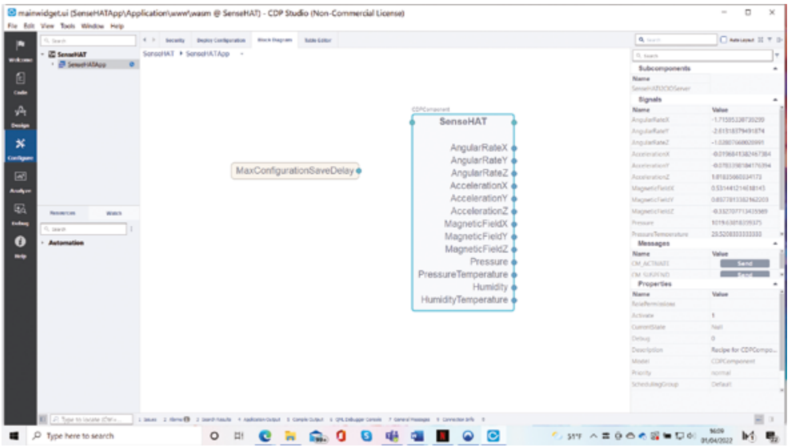
03 Start a new project

Unlike last time, there's no need to create a library for this project. So just go to File > Create New and choose CDP System. Give it a name (such as SenseHAT) and click Next, then Next again. From the Application Type drop-down, choose WebUI to automatically create a web UI for it – although if you forget to do this, you can always add one later. Now click Finish to create the system. Unless you changed it, its default app will be SenseHATapp, as shown in the hierarchy in the left panel of Configure mode.

04 Add SenseHAT component

In the Block Diagram tab, you'll see the application block. Either double-click it or click the app in the hierarchy to open it up. Don't worry about the default blocks there. In the bottom-left panel, under Resources, open the I2CIO category. Now drag the SenseHAT component there into the middle of the block diagram.

As you can see, it has outputs (on the right of the block) for all of the Sense HAT's sensors. These include angular rates and acceleration for the X, Y, Z axes of the HAT's gyroscope and accelerometer, to sense orientation and movement in three dimensions. There are also magnetic field outputs for its magnetometer. Finally there are pressure, humidity, and two temperature outputs depending on which sensor you want to use.



The preconfigured SenseHAT block provides readings from all the sensors, seen here in the right-hand panel

05 Deploying the program

This time, there's no need to add any other blocks to the application, nor wire anything up. The SenseHAT component will work as it is to take live readings from the HAT's sensors. Let's try it out by deploying the application.

To do so, go to the Deploy Configuration tab and select WiFi under Networks to show devices on your network. Find your Raspberry Pi by its IP address, enter pi as the Username, and click Pair, then enter its password.

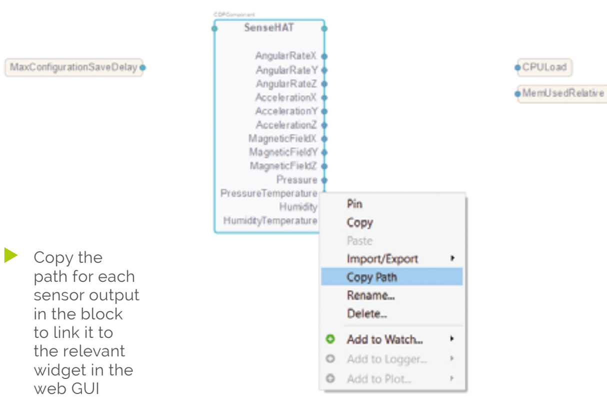
Now make sure its IP address (or name if you've renamed it) is selected under Device in Applications, and that you have the relevant Toolkit selected for your version of Raspberry Pi OS: Raspberry Pi ARMv8 for Bullseye, or Raspberry Pi/Raspbian ARMv6 for Legacy.

Right-click SenseHAT in the left-panel hierarchy and choose Run & Connect to run the application on your Raspberry Pi. Wait for it to deploy. Now select the SenseHAT component in the Block Diagram and you'll see the live signals from the HAT's sensors in the right-hand panel.

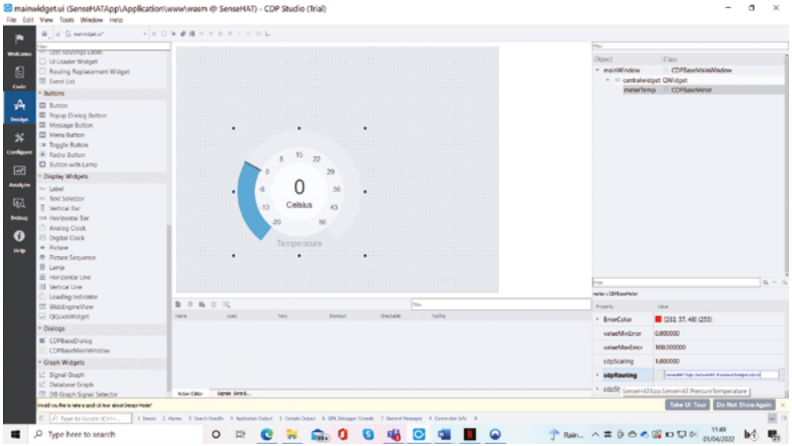
Top Tip

Deploy on PC

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



Copy the path for each sensor output in the block to link it to the relevant widget in the web GUI



Creating the web GUI in Design mode. Paste the routing for the sensor output into the meter widget's cdpRouting field

06 Design a web GUI

That's all very well, but it's not the most user-friendly way to view the sensor readings. Right-click the system or app name in the hierarchy panel and select Stop.

Got to Design mode to start designing a web GUI to show your readings. In the left panel, you'll see a variety of elements that you can add to the GUI canvas in the middle. We'll start by adding some widgets to show the temperature, pressure, and humidity readings from the Sense HAT.

We opted to use a standard Meter widget for ours, but you could use a Vertical Bar (under Display Widgets) for one or more readings if you prefer. Simply drag the widget from the left panel onto the canvas to add it. In the bottom-right properties panel, you can then alter its minimum and maximum values (in minValue and maxValue) and text labels (textPrefix and textSuffix). For our temperature meter, we set min and max values to -20 and 50, and labels to Celsius and Temperature. To send the sensor data to the meter, you will need to add the routing for it. Go back to the Block Diagram and right-click the SenseHAT output you want – we chose Pressure Temperature (the temperature reading from the pressure sensor – and Copy Path. Back in Design mode, paste it into the field for cdpRouting at the bottom of the properties panel.

07 Test your first meter

Let's check our GUI and temperature meter are working correctly. As before, right-click the system name in the hierarchy and choose Run & Connect. Once it has been deployed, the Application Output panel at the bottom will show the URL for the web GUI. It will be your Raspberry Pi's IP address followed by **:7869/index.html** – for example, ours was "**192.168.1.112:7689/index.html**."

Open it in a web browser and you should see the meter in your GUI showing the current temperature reading.

08 Add more meters

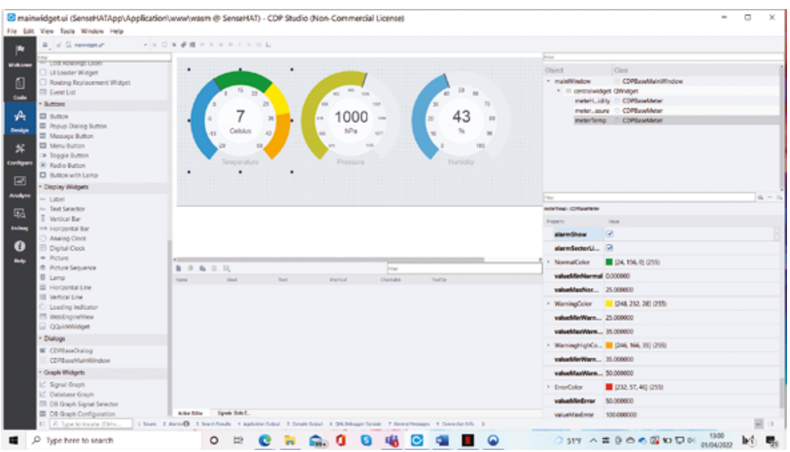
Stop the application from running. We'll now add meters for pressure and humidity to our web GUI. In Design mode, drag a meter over for both (or copy and paste your temperature meter). Then change the min and max values and text labels in the properties panel, as we did before.

For our pressure meter, we opted for min and max values of 870 and 1100, with text labels of hPa and Pressure. For the humidity meter, the min and max values should be 0 and 100 (as it's always a percentage), while the text labels are % and Humidity.

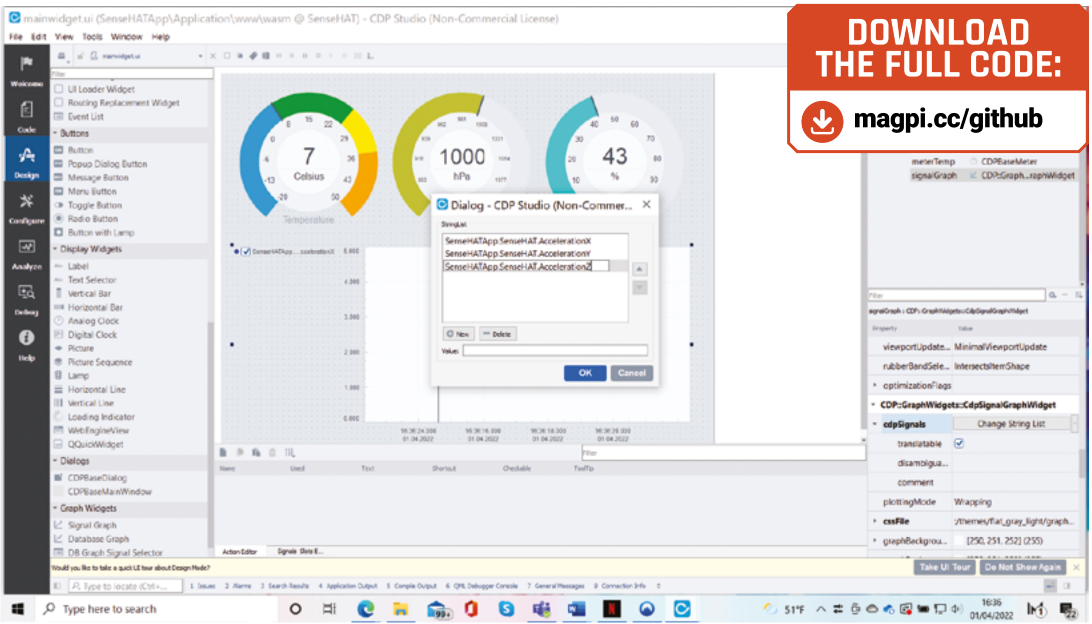
Again, you will need to add the routing for each meter so it displays the correct sensor signal. So right-click the relevant output (Pressure or Humidity) of the SenseHAT block in the Block Diagram, select Copy Path, and – in Design Mode – paste it into the cdpRouting field of the properties panel for the meter.

If you want to get creative, you can alter the fillColor values in the properties to change the colour of each meter's fill colour so they're not all blue. For our temperature meter, we also set min and max values for NormalColor, WarningColor, and WarningHighColor to allocate the colours to a range of temperatures on our meter.

When ready, run the application again and you should see all three meters in the web GUI, showing the relevant sensor readings.



You can alter the fill colours for the meters if you like, and even add coloured sectors for different value ranges



DOWNLOAD THE FULL CODE:

magpi.cc/github

◀ Creating a graph in Design mode. Paste the routing paths for the sensor outputs into the String List

▼ The finished web GUI, with meters for temperature, pressure, and humidity, plus a graph plotting the three axes of the accelerometer

09 Add a live graph

Meters and display widgets aren't the only way to display data in the web GUI. Let's add a Signal Graph to our GUI. You'll find it under Graph Widgets in the left panel of Design mode. Drag it into the GUI canvas (you may need to expand the latter) and alter its size accordingly.

We opted to show the readings from the Sense HAT's accelerometer for our graph. In the graph's properties panel, scroll down to find cdpSignals, then click Change String List next to it. This brings up a dialog where you can paste the routing for each sensor output you want to add. As before, you can obtain the routing by right-clicking the output of the Sense HAT block in the Block Diagram and selecting Copy Path. Our paths were SenseHATApp.SenseHAT.AccelerationX and so on (replacing X with Y and Z).

We also set the yAxisMinValue and yAxisMax Value for the graph to -2 and 2 – since we found them to be the min/max G values read from the Sense HAT's accelerometer. Of course, you could add other sensor readings to the graph if you want, or even add a second graph for something that needs a different vertical scale.

10 Test it out

Our web GUI now features meters for temperature, pressure, and humidity, along with a signal graph to show live readings for the three axes of the accelerometer. Let's try it out.

As before, Run & Connect the application to deploy it to your Raspberry Pi, then visit the web GUI URL in a web browser – don't forget to refresh it so you see the revised GUI. Under your three meters, the graph should show the current accelerometer readings. With Raspberry Pi lying flat, the Z axis reading should be 1G (the others 0). If you tilt Raspberry Pi up to stand on its short side, the X reading will be 1; stand it on its long side and the Y value will be 1. To get higher readings, try shaking Raspberry Pi!

Taking it further

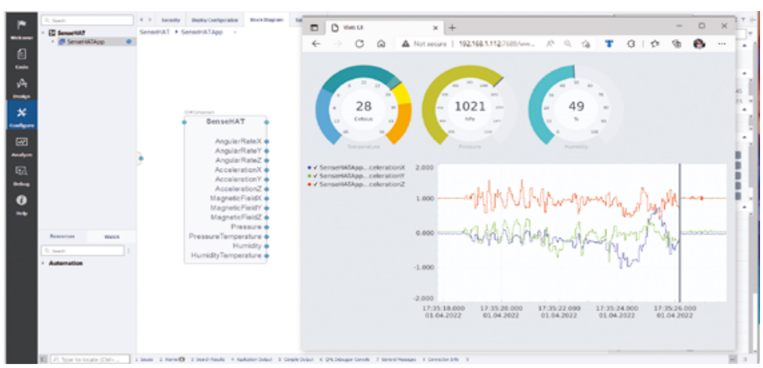
We now have a web GUI showing live readings from the Sense HAT. You could add more sensor readings to it, alter its layout, or jazz it up however you want. You could also log your sensor data using a CDPLogger block – see **magpi.cc/cdplogger** for more details – and plot it on a graph using the Database Graph widget in Design mode.

You're not restricted to the Sense HAT either: you could read values from sensors connected to the GPIO pins on Raspberry Pi using a GPIOServer block (as we used in part one). CDP Studio also features built-in support for I2C devices and the ADS1115 ADC – see **magpi.cc/cdpi2cio**.

Top Tip

Random issue

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



Building a sheet metal rover body shell

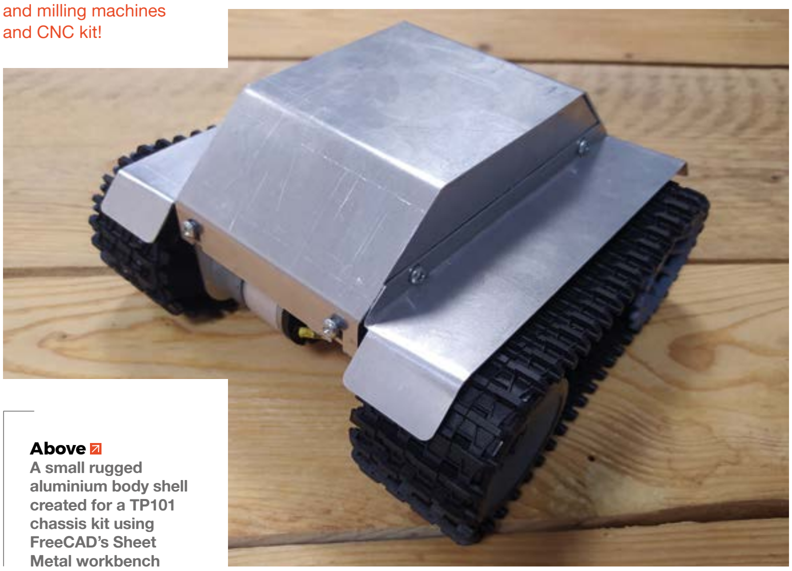
Putting together a range of skills in FreeCAD, CNC routing, and metalwork to make a metal rover body shell

Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

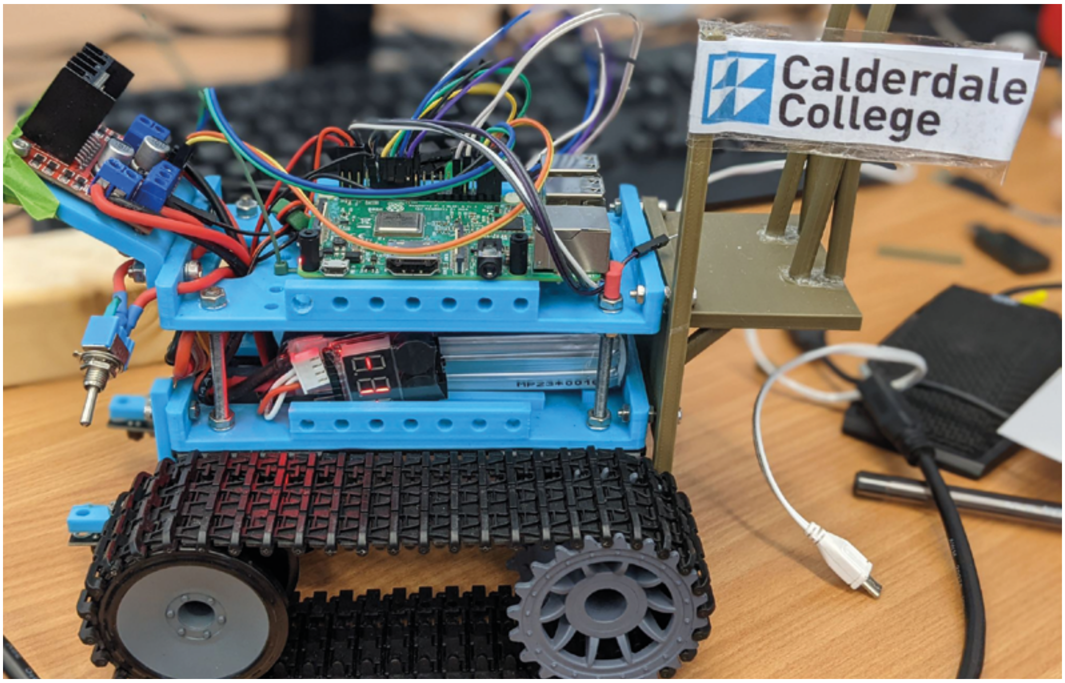
In one section of the FreeCAD tutorial series, we looked at a brilliant FreeCAD add-on: the Sheet Metal workbench. You might recall from issue 50 that it enables the dedicated 'CADista' to create a basic sheet of metal and then build a design by adding attached folded sections. Whilst amazing in itself, the real secret sauce of this workbench is that after you have designed something, you can then flatten the design. From this, you can then create CAM toolpaths, laser cutting files, and more. As I wrote that tutorial, I recalled a project from one of the early dedicated maker-type books from years ago. The first edition of *101 Spy Gadgets for the Evil Genius* had many cool ideas – in one, they took a rugged 4WD RC car, hacked it back to the bare chassis, and then created a sheet metal body shell for it in an attempt to make a rugged robot for covert surveillance missions! Ever since, I've wanted to build a metal-clad rover.



Above A small rugged aluminium body shell created for a TP101 chassis kit using FreeCAD's Sheet Metal workbench

I've previously designed the open-source modular tracked vehicle (MTV) robot which uses a cheap TP101 chassis, track, and motor kit as its base. It then adds a stack of reconfigurable 3D-printed parts to make a really modular platform for experiments. The original story with links to the MTV files can be found in issue 32. I experimented with numerous ways of rover control for the MTV, including direct radio control using hobby RC gear and brushed speed controllers, a hybrid system using an Arduino receiving the RC signals and using an L298N motor driver, as well as using a pair of micro:bits as a transmitter and receiver coupled with a cheap micro:bit motor-driving add-on board. Finally, I even wrote an article (issue 35) about controlling the MTV robot via dual-tone multi-frequency (DTMF) technology, opening up interesting ideas for controlling a robot from anywhere in the world with a telephone! I really like this chassis; it's cheap and rugged, and the tracks, although they take a bit of tinkering with, work very well. As an aside, it's great to see others using the MTV robot design; for example, Calderdale College used the design to compete successfully in the excellent Pi Wars competition, a set of challenges for Raspberry Pi-powered robots.

With all this in mind, I really wanted to make a robust shell for the TP101 chassis. A good starting point was to accurately measure and model the TP101 chassis and, in particular, the position and size of the available chassis mount points. I began by taking some accurate measurements and then drew the outline of the upper surface of the chassis. Making this rectangle the BaseBend object in the Sheet Metal workbench, I then added the folds to emulate the chassis sections where the motors and the guide wheels for the tracks mount, but stopped



Warning!

This project features multiple cutting and drilling techniques. Use power tools safely.

magpi.cc/diysafety

Left Lovely to see other people and organisations using the MTV robot design which also uses the TP01 chassis kit

Below Creating an accurate model of the TP101 chassis and mount points in FreeCAD

short of modelling all the mount points for the motors and wheels.

I then did the longer task of drawing and constraining a sketch on the chassis surface which accurately modelled all the holes and slots that the TP101 chassis comes with. This is a useful job to do for designing all kinds of parts to this chassis, as well as the body shell I was aiming for in this project.

I was quite interested in creating a symmetrical body shell for this rover – this would reduce the number of parts I needed to model as I could use an Assembly workbench to add multiples of the same part from file to lay out the rover body shell. I kept the model of the chassis open in one project tab, then, in a new project, I began to model the sidewall. As background, I had a few thoughts on how I would rig the control system for this rover, and I gave a little thought to what components I might use. I am a fan of the cheap, but rugged, L298N motor driver modules which come with a fairly large heatsink on the board – these would probably be the tallest module I'd choose to go inside. With this in mind, I made a basic sketch of the sidewall and added dimensional constraints. Again, using the Sheet Metal workbench, I added the folded section underneath that would mount to the chassis, and also some small folded tabs that would serve to mount the roof of the body shell. Interestingly, those small tabs are a good example of something that is quite easily added in the Sheet Metal workbench that can be tricky to actually make, but more on that later.

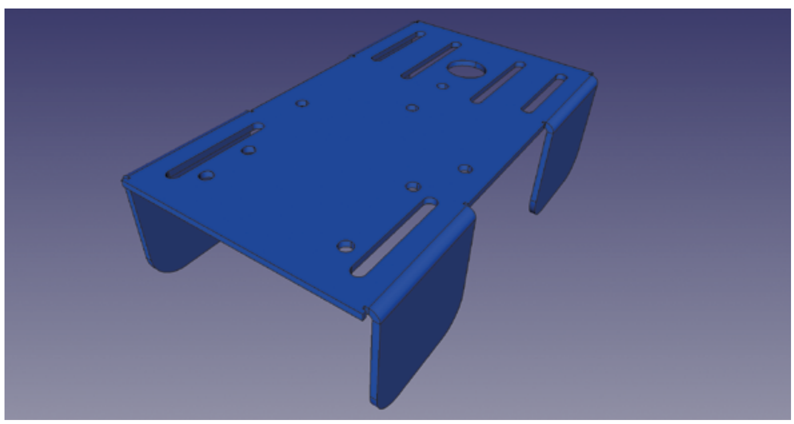
I added some mount holes to the project and saved the model file. I then opened a new FreeCAD project and used the A2plus workbench to create an assembly of the chassis and the sidewall. Of course, I could add two copies of the sidewall and constrain them into position on either side of the chassis – we covered the A2plus workbench in issue 43.

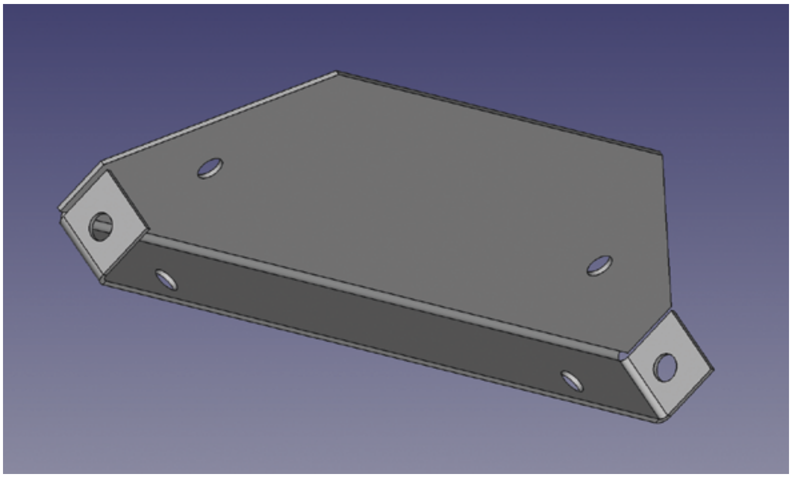
> "I am a fan of the very cheap but rugged **L298N motor driver modules**"

Next, I created some mudguards. I didn't want these to cover the full width of the tracks, but protrude around 30 mm to provide enough cover to stop mud/moisture being thrown up onto the body shell. This means any →

QUICK TIP

All the FreeCAD techniques used in this build are covered in the FreeCAD tutorial series.





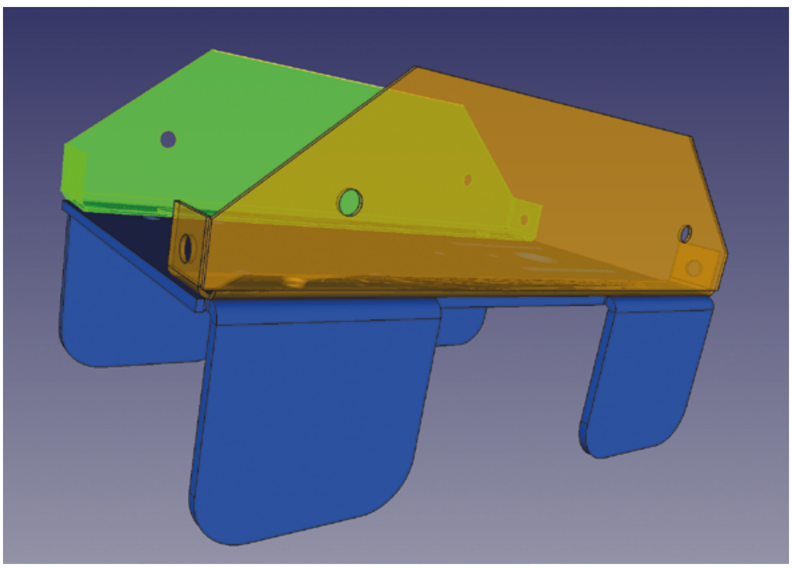
Above The Sheet Metal workbench makes complex parts really easy to achieve. This sidewall only has one basic sketch and then the folded sections are added

Below Using the A2plus workbench means that I could create an assembly of the chassis and panels, checking how things fit and look together

Right Here we can see both the 3D flattened object and the projected wire sketch of the object. These objects can be exported in a variety of file formats

external equipment, cameras, or sensors should be well protected. The other use is that the mudguards can provide a small platform for mounting objects onto directly. I drew the small section that attaches to the sidewall first, and then added the mudguard section as a fold. A good tip here was I drew the basic sketch for the part that attaches in the same file as the sidewall, importing useful edges from the sidewall object to snap the sketch too, then you can simply copy the sketch into a body in a new project to create a new part and delete the sketch in the sidewall project. To finish off the mudguard, I added some end folds at 45 degrees and jumped back to the Part Design workbench to add some fillets to round the edges and pocket the mounting holes.

Again, as running checks and to see the body shell develop, I pulled two copies of the mudguards into the project where I was creating an assembly of the parts. All that remained was to create a layout for what I called the 'canopy' – the roof section that covered the sidewalls. To do this, I returned to the sidewall sketch and made a copy of the base sketch in a new project. I then deleted most of the sidewall sketch constraints and constrained the length of the lines that followed where I wanted the roof to go. I then constrained those remaining lines as horizontal, copied the line, and created a rectangle. This meant that I had the rectangle for my BaseBend object, but also that I had geometry that I could import into further sketches to create the fold lines.



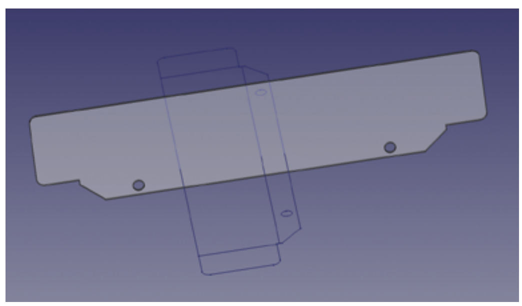
TO HAVE AND TO FOLD

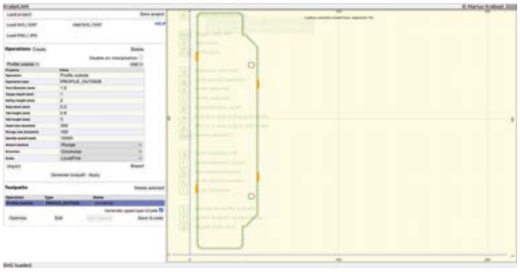
Once more, we covered this in the Sheet Metal workbench tutorial, but I then made a collection of four separate sketches attached to one face of the canopy object which were simple lines at the points I wanted to create the folds. Jumping momentarily back into the original sidewall project, I made some quick angle constraints (which created a duplicated constraint error) and then deleted them, just making a note of the angles I wanted to fold the canopy section too.

Again, a quick bit of folding in our canopy project, and the adding of some mount holes, and we could add the canopy into our assembly project to see the whole body shell created.

With the shell design complete and positioned in the assembly project, I then went back to each individual part project and used the Sheet Metal workbench tools to flatten each part. The flattening process creates both a flattened 3D part and a projected wire sketch of the part. I could have used the 3D model and moved to the Path workbench to create some CNC toolpaths, but I decided to export the wire sketches as flattened SVGs, and then I could make some changes in Inkscape, and then I could do a variety of processes with the files.

I know if you haven't got a shed full of tools and access to a CNC machine, reading about people making this stuff can be frustrating, and indeed I did CNC route the panels for my build. But in reality, these panels could be made in a variety of ways. Thin





aluminium sheet can definitely be cut well with a metal bandsaw, or you could even use a combination of hacksaw and fretsaw to cut these panels, perhaps finishing them to size with files. As such, it's useful to be able to make a template to use to create your cut-lines. The simplest approach is to print the flattened SVG onto paper and glue the paper to your aluminium sheet – this gives you a nice set of lines to follow, and then you can soak and wash the paper away. Another approach, if you have access to a laser cutter, is to laser-cut some templates. As an experiment, I laser-cut some really accurate templates out of leftover corrugated cardboard – the nice thing about this approach is that I set the fold lines to be included as a small incision onto the cardboard to again aid marking up on metal.

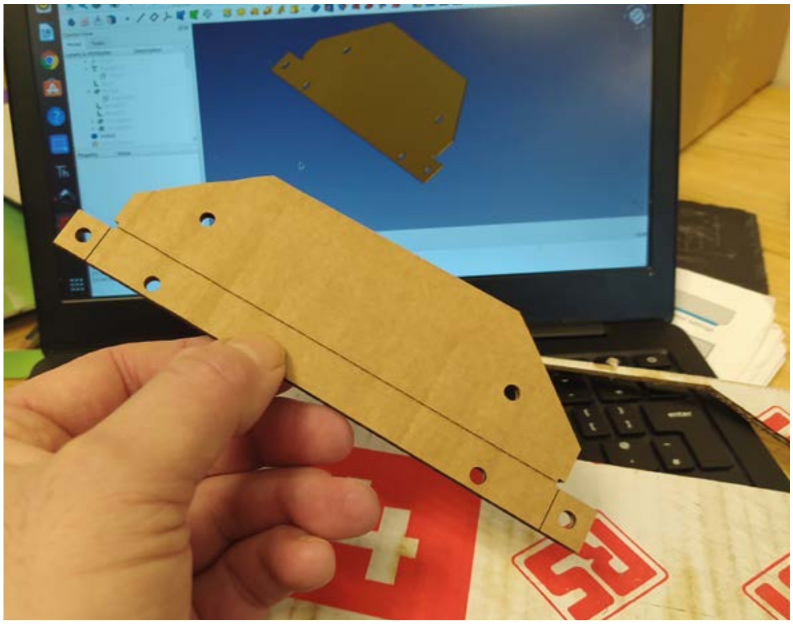
As an aside, it made me think that a possible approach could be to create folded card stock parts and cover them with fibreglass and epoxy for strong and lightweight panels.

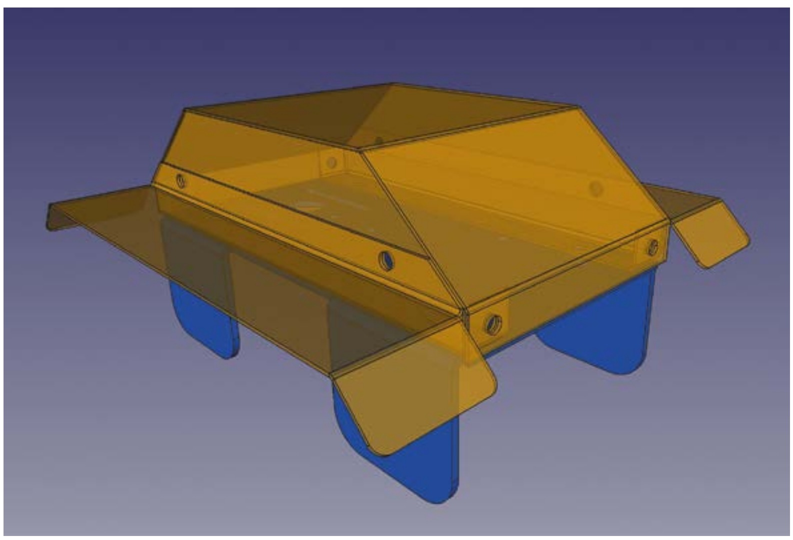
> Cutting aluminium on my CNC3040 is **pretty straightforward if you go nice and slow**

I made no real change to the SVGs for the sidewall and the mudguard designs and pulled the SVGs into KrabzCAM to create some toolpaths for my CNC router. We looked at the free and open-source KrabzCAM software as part of the mini-series we wrote on the cheap CNC3018 machine back in issue 43. KrabzCAM is very useful for this type of 2D CAM work and makes setting up toolpaths really simple. Cutting aluminium on my CNC3040 is pretty straightforward if you go nice and slow with low feed rates. I ran a small 1.5 mm end mill cutter at 350 mm per minute and a step down of 0.25 mm per pass, and I'd designed the body shell to use some 0.8 mm aluminium stock I had in the pile.

As I don't tend to skim my waste boards, there's often a little variance in height across a piece of work stock, so I tend to cut well through the material to compensate. I set all my toolpaths to cut to 1 mm deep to ensure it cut through. I added some work holding tabs to the toolpaths to keep the part in place and then set about machining. I tend to flood the material with a puddle of cutting fluid which acts to cool the tool tip, but also, the liquid tends to capture the aluminium chips and stops them from flying everywhere. So, with a good dollop of fluid added, I set about machining panels.

I wanted to cut and assemble the side panels and mudguard sections before finishing the canopy cover, as I wondered if there might need to be some tweaks. Having cut the mudguards and side panels, I went to the shed and dug out my three-in-one metal folding, rolling, and cutting machine. We reviewed this machine from Warco in issue 36. →

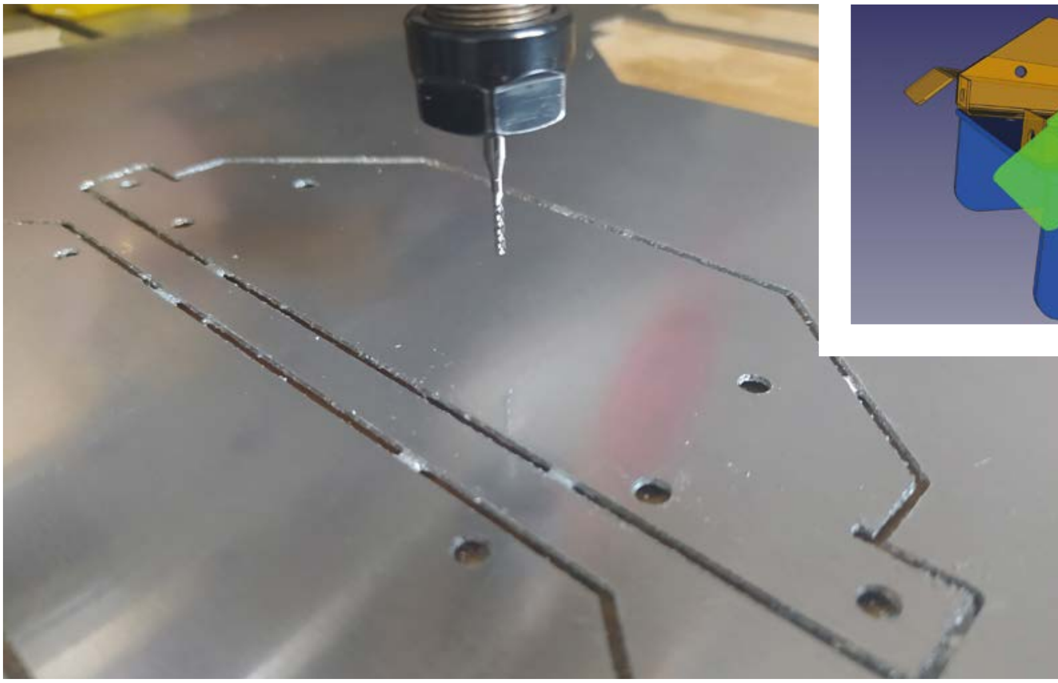


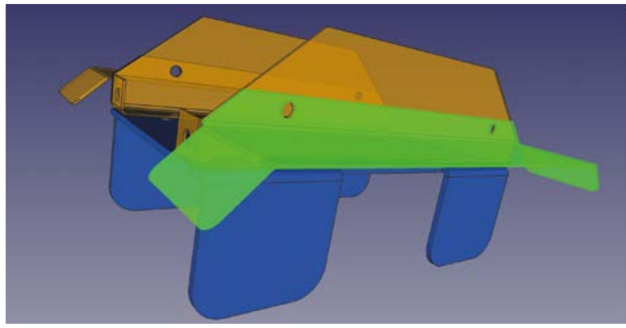


Above Laser-cutting cardboard templates or simple printing onto paper or card is a great option for template making for those without access to a CNC

Left Importing SVG and creating 2D toolpaths is pretty straightforward in the free and open-source KrabzCAM

Below With the canopy modelled, everything could be assembled and checked out





Above Making the design symmetrical meant I could use multiple copies of single components to assemble the 3D design

Left The ball screw version of the CNC3040 router does an OK job of routing thin aluminium sheets

Above First assembly of a mudguard and a sidewall section in the real world!

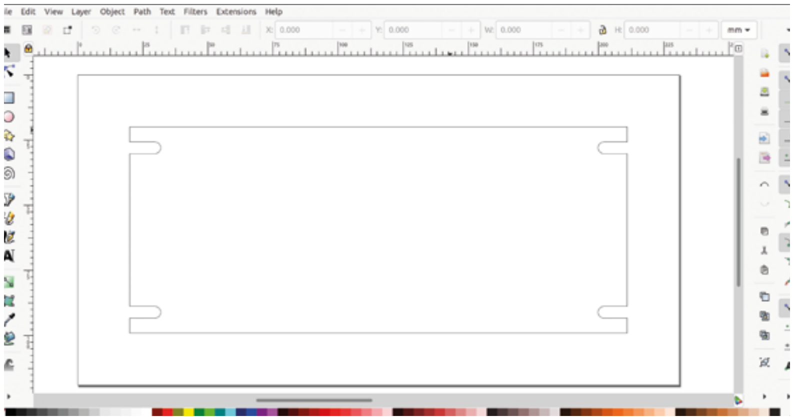
Right Using the press brake section of the 3-in-1 Warco Formit metal working machine

Although it's an expensive tool for occasional use, we found it does make sharper and neater folds than other methods. If you don't have the space for a tool such as this, there are other options. There are folding accessories that act in a similar manner to this that clip onto a bench vice, but you can also use a vice on its own to create folds. In fact, on occasion, it was easier to use a small vice than it was to use the folding machine for some of the smaller bends in this project.

My main advice about creating the folds using the three-in-one machine is to make sure that your workpiece is in the correct position, because if you make even a slight bend in the wrong place, it's incredibly difficult to rectify. It's quite a Zen activity – you need to mindfully manipulate the part into the correct place, but then be quite assertive and confident to make the fold in one neat move!

ORDER, ORDER

I tended to use the larger machine to make the longer folds in the parts, but doing so often means you can't then fit the panel back into the folding apparatus to create the smaller folds. A good example of this was mentioned earlier – the internal small tabs that end up at 90 degrees to the sidewall are quite difficult to make once the long fold in the sidewall has been created. You might think you could use the open edge of the folding wedge, but the angle of the sidewall clashes with the apparatus during the fold. So I added these folds by pinching the work in a tiny vice and bending the panel by hand. This works, but creates a wider radius fold and is less consistent – but it's good enough.



It's lovely to assemble CNC-cut panels. Fitting the folded sidewall with a folded mudguard section, it's extremely satisfying to see the holes align and the angled edges of the mudguard follow the line of the sidewall so perfectly!

Once I had the sidewalls and the mudguards assembled, I considered the canopy. I was quite happy with the idea of it but I made a change to the mount holes. I realised that it would be easier to mount if we had slots rather than holes, as there was slight variance in the small folded tabs that had the receiving holes having made them with a vice. Instead of remodelling the piece in FreeCAD, it seemed much simpler to make this simple adjustment to the exported SVG in Inkscape. The other slight bonus being that, using slots instead of holes, the entire canopy cover panel could be cut in one outside profile operation instead of four-hole operations and then a cut out. Again, I set up the toolpaths in KrabzCAM and then cut the canopy in around ten minutes on my CNC router. Once again, prior to folding, the part needs a little de-burring and any of the tab remnants need removing from the sides of the panel.

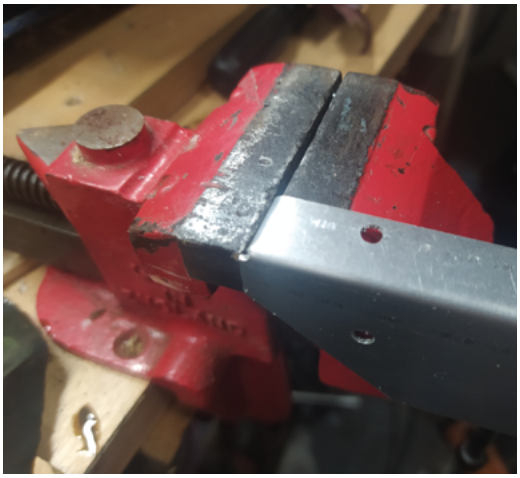
It's pretty tricky to get the folds in the correct place for the canopy. I revisited the FreeCAD sketches and wrote down the coordinates for the fold points and marked these onto the canopy by scribing a small line. It's also tricky to get the folds to the correct angle. If I thought hard enough, I could probably work out some kind of 3D-printed guide block, but I ended up doing this by eye. I'd make a fold, then offer up the folded piece to the sidewalls to check angles and then readjust. I got it as close as I could and then test-fitted the canopy. It fits quite well, but it takes some adjustment to get it so that there are very few gaps around the close line where the canopy meets the sidewall. For the next revision, when I have more of an idea of the internal structure and running gear, I have a couple of other ideas. I'll either redesign the canopy to be more modular, with two angled end pieces more permanently in place and a flat plate to close the roof, or I'll make the canopy sit slightly inside the sidewalls, as I think it would be easier to create a flush finish this way.

You might also be wondering how I managed to use the nuts and bolts that go into the canopy slots as you can't access the internal side to tighten the nut. I used an old trick where I created a captive nut on the inside panels by using a small amount of epoxy glue to attach the nut in place!

As body shells go, I am really pleased with this project, and it's certainly proved that the FreeCAD Sheet Metal workbench is very capable at designing real-world objects. I also think the results look pretty cool. With a few tweaks and some more building, I should have a very capable and rugged little rover platform.

If you are interested in looking at the design, I have added a stack of files to this repository: **hsmag.cc/SheetMetalRover**.

The separate panel FreeCAD files and the assembled model are there, as well as the SVG files. I've stopped short of supplying the G-codes because everyone's CNC routers and tooling are different. Hopefully, this little project has inspired you to explore metalwork for robot rovers and perhaps to look at the FreeCAD Sheet Metal workbench!



Left Once the side walls were in place, I epoxied captive nuts in position to be able to bolt the canopy cover on

Above Adjusting an SVG in Inkscape prior to creating the toolpath was easier than remodelling in FreeCAD

Below Some of the smaller bends, which are added after the longer folds, are morae easily achieved using a small bench vice

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**.

Raspberry Pi high-end audio

It's time to up your audio game on our favourite computer! **PJ Evans** takes us through the options. This might get loud

From the very beginning of Raspberry Pi, audio has been one of its most popular applications. Raspberry Pi's small form factor and fixed parts lend themselves perfectly to sitting in a living room or kitchen. Although the Raspberry Pi Model A and B computers feature built-in audio, it wasn't long before more advanced HATs (Hardware Attached on Top) appeared, raising the output quality to something that would make any self-respecting audiophile drool. Add in amazing audio management software, and we had a rival for top-end home audio systems.

In this special feature, we're going to look at the audio hardware available and how to get the best out of it.

USB Audio Adaptor

The Pi Hut ▶ £4.50

The logical step up from Raspberry Pi's built-in audio is to add a USB audio adaptor. This is also the neatest low-cost solution for adding audio to Raspberry Pi Zero (with a USB micro-to-A adaptor, **magpi.cc/microusbadaptor**) and a simple method for multi-channel projects, as you can add as many adaptors as you like. Many different versions are available so be sure to check that the one you are considering will work with your choice of operating system. Most (including the one linked to here) will work without any configuration or driver installations. This will also add the ability to record audio on your Raspberry Pi.

magpi.cc/usbaudio

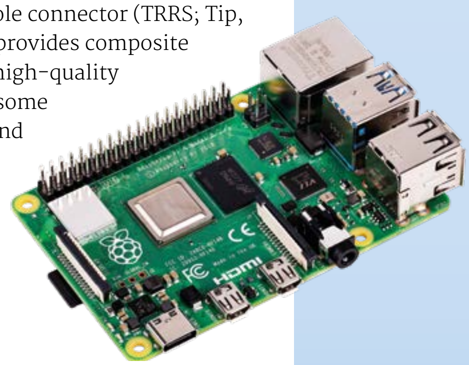
Your audio options

Built-in audio

Raspberry Pi ▶ Free!

When choosing your audio solution, don't forget that all Raspberry Pi Model A and Model B variants come with audio baked right in. All feature a 3.5mm socket that provides stereo audio at line-out levels. On later models, such as Raspberry Pi 3 Model B and Raspberry Pi 4 Model B, this is a four-pole connector (TRRS; Tip, Ring, Ring, Sleeve) that also provides composite video. So, if you're not after high-quality audio and just want to make some noise, just add an amplifier and you're set. Raspberry Pi Zero computers do not feature 3.5mm audio out, so only its larger cousins will do. Suitable cables are widely and cheaply available.

magpi.cc/raspberrypi4



▲ Need basic stereo audio? The Raspberry Pi 4B has you covered with a standard 3.5mm headphone socket

◀ Simple but effective, a USB audio adaptor adds sound and recording capability to any Raspberry Pi

Pimoroni's range of audio HATs cover all common use-cases

Pirate Audio Headphone Amp

Pimoroni ▶ £20

With a name like Pirate Audio, it can only be our friends from Sheffield-on-Sea, Pimoroni. Its Pirate Audio range is similar to the HiFiBerry DAC but with more features. We've chosen the headphone version here, but line-out, 3W amp and even built-in speaker versions are available. All four Raspberry Pi Zero-sized HATs feature a 1.3-inch IPS screen and four control buttons, making them perfect for on-the-move audio. A comprehensive online guide takes you through installation including a full audio solution based on Mopidy (**magpi.cc/mopidy**), so you can get running right away.
magpi.cc/pirateaudioheadphone

PecanPi

Orchard Audio ▶ $350

Orchard Audio has firmly established itself at the top of the pile when it comes to Raspberry Pi audiophiles. A seemingly relentless dedication to sourcing the best components and cutting absolutely no corners when it comes to the design of their DAC boards has resulted in the kind of performance no one could ever have associated with the humble Raspberry Pi. The DIY version of its signature PecanPi is a fully-loaded HAT suitable for studio use with twin DACs and XLR outputs. Yes, the price is eye-watering, but you get what you're paying for. If only the best is good enough, you've found your product.
magpi.cc/pecanpi

HiFiBerry DAC+ Zero

HiFiBerry ▶ £18

A low-cost but high-quality DAC can add greats sound reproduction to a Raspberry Pi Zero

Computers don't understand audio as we hear it. Instead, they use a Digital-Analogue Converter (DAC) to turn a digital signal into something we can hear. The quality of the DAC integrated circuit is the single most important factor in producing great audio quality. Adding a dedicated DAC to your Raspberry Pi is the best bang-for-buck upgrade you can get.

At just £18, this HAT produces great line-out audio quality and is perfect for Raspberry Pi Zero projects. Several variations are available that add digital audio and even small amplifiers. A great range of cases means it is perfect for home audio projects.
magpi.cc/hifiberrydaczero

Perfect for your voice assistant project, the Respeaker's far-field microphones capture great quality speech for more reliable recognition

Respeaker v2

Seeedstudio ▶ £25

So far we've mostly covered audio output devices, but what about input? The easiest solution is to get a USB audio adaptor, but you'll also need a suitable microphone and amplifier if you want to record voice. A popular use of Raspberry Pi in the home is to build a voice assistant. If you fancy trying to build your own, this HAT-based audio recording device is everything you need. An array of four microphones with far-field capability can capture voice commands from 5m away. Resources and tutorials are available from ReSpeaker (respeaker.io) to help you build your own Alexa or Siri.
magpi.cc/respeaker

Small but mighty. The PecanPi produces studio-grade sound

Top Tips

Other audio types

There are plenty of other ways to make noise with a Raspberry Pi. A wide range of buzzers and sirens are available. Always be careful with volume!

Get what you need

Don't be tempted to splash out on the latest and greatest if all you want to do is make a buzz.

Set up a whole-house audio system

Whether it is a single room or everything including the loo, here's how to get audio in every room with Raspberry Pi

WRITER

PJ Evans

PJ is a writer, software engineer and a very bad audiophile. His prized Val Doonican collection can be listened to in pristine quality.

@mrpjevans

You'll Need

- HiFiBerry DAC+ Zero **magpi.cc/hifiberrydaczero**
- Case **magpi.cc/daczerofaceplate**
- Active Speakers **magpi.cc/edifierbookshelf**

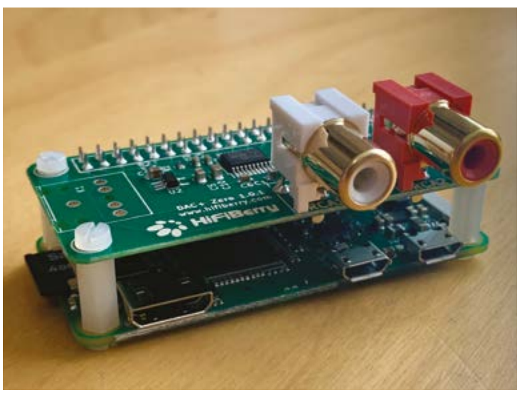
Just imagine being able to listen to your music anywhere in your home, in perfect sync as you move around. Such audio systems do exist, but can cause serious damage to your bank account. However, it is now possible to build just as good a system with nothing but Raspberry Pi computers and some incredible open-source software. Add a DAC HAT for a boost in sound quality, perhaps a cool case and speakers, and you're good to go. You can use streaming services such as Spotify too.

01 Install your DAC and configure

If you are using a DAC HAT, now is the time to install it. Follow the manufacturer's instructions closely and ensure your Raspberry Pi operating system is fully up-to-date. If you're using Raspberry Pi OS Lite, you will need to use `alsamixer` to enable the card (or at least turn the volume up). Once you're ready to test the audio and have connected the DAC to some active speakers, here's a simple command to check everything is working:

```
speaker-test -c 2
```

This will play white noise through the left and right channel. If you can hear it, you're good to proceed.



Without its case you can see how the DAC+ Zero is assembled with Raspberry Pi Zero

02 Install Mopidy

Now that we have sound, the next step is to install software to control and manage our library of music. Mopidy is an excellent choice for this and comes with Iris, a beautiful web interface. To install Mopidy follow the commands in **mopidy.txt** (press return after each line). This will install everything you need. Now, to access Mopidy remotely, edit the config file:

```
sudo nano /etc/mopidy/mopidy.conf
```

At the end of the file, add the code in listing **mopidyconfig.txt**. Now save it, then start the server:

```
sudo systemctl start mopidy
```

You should now get a response on **<ip address>:6680/iris/**

03 Load up some music

Your music folder is **/home/pi/Music**, and you can now transfer your music library to that

The HiFiBerry DAC+ Zero provide high-quality output through standard RCA connectors

Mopidy Iris provides a beautiful user interface to all your music

> "Add a DAC HAT for a boost in sound quality"

directory. It is common to use an Artist/Album/Tracks pattern. Mopidy will inspect the metadata in the files and catalogue accordingly. Once you're ready, ask Mopidy to scan the folder:

```
sudo mopidyctl local scan
sudo systemctl restart mopidy
```

If you can't immediately see your new files, try Browse > Local Media to locate them. You now have a fully featured, remote-controlled media player. Modipy has a host of plugins, so you can add services such as radio and Spotify.

04 Add Snapcast

Snapcast is an open-source multi-room streaming system that provides proper in-sync playback without loss of quality. To make it work, we re-route the Mopidy output stream to the Snapcast server, which then relays the signal to Snapcast clients, including locally. To install Snapcast server enter the following in Terminal:

```
wget https://github.com/badaix/
snapcast/releases/download/v0.26.0/
snapserver_0.26.0-1_armhf.deb
sudo dpkg -i snapserver_0.26.0-1_armhf.deb
```

And the client…

```
wget https://github.com/badaix/
snapcast/releases/download/v0.26.0/
snapclient_0.26.0-1_armhf.deb
sudo dpkg -i snapclient_0.26.0-1_armhf.deb
sudo apt -f install
```

A dependency error on step 2 can be ignored.

Warning! Volume!

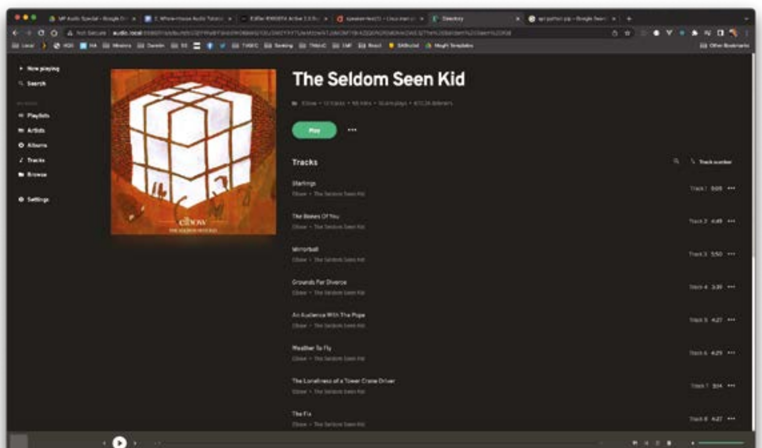
Always be careful when testing audio out, especially if amplified. Excessive volume can cause lasting hearing damage.

magpi.cc/hearingloss

Top Tips

Groups

Snapcast supports 'groups', so you can have many different players playing back in sync. You can even have different music streams playing simultaneously.



▼ The Iris interface uses metadata from your music files to produce an interface that works equally well on desktop and mobile

mopidy.txt

Language: **BASH**

DOWNLOAD THE FULL CODE:
magpi.cc/mopidytxt

```
wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key add -
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list https://apt.mopidy.com/bullseye.list
sudo apt update
sudo apt install mopidy python3-pip
sudo adduser mopidy video
sudo pip3 install Mopidy-Iris
sudo pip3 install Mopidy-Local
sudo sh -c 'echo "mopidy ALL=NOPASSWD: /usr/local/lib/python3.9/dist-packages/mopidy_iris/system.sh" >> /etc/sudoers'
mkdir -p ~/Music
sudo systemctl enable mopidy
```

mopidy_config.txt

Language: **BASH**

DOWNLOAD THE FULL CODE:
magpi.cc/mopidyconfigtxt

```
[http]
hostname = 0.0.0.0

[audio]
output = alsasink

[local]
media_dir = /home/pi/Music
sudo systemctl enable mopidy
```

05 Configure Snapcast

To enable Snapcast, go back to the web interface and select Settings. Click on the Snapcast icon, and click 'Enabled'. If it doesn't work immediately, change the 'Host' setting to the full name of your server. Once it says 'Connected', Snapcast is now broadcasting all music you play over your local network. Make sure playback works as normal, although it may take an extra second or two to start as it syncs. All that remains is to create additional 'nodes' on your whole-house audio system to accept the Snapcast stream.

06 Create additional music players

You can use any device that can run Snapcast to receive your audio. If you're using a Raspberry Pi Zero and a DAC+, configure them as per step one, then just install the snapclient package as before, but not the snapserver.

The final step is to tell the system where to get the music stream:

```
sudo nano /etc/default/snapclient
```

Add the following:

```
START_SNAPCLIENT=true
SNAPCLIENT_OPTS="--host 192.168.0.4"
```

Replace 192.160.0.4 with the actual IP address of your server. Then restart:

```
sudo systemctl restart snapclient
```

Go back to the Mopidy interface and you should see the new player as a 'Group' in Snapcast. Add as many of these as you want, and enjoy your music anywhere.

Top Tips

Auto-scan your files

Mopidy won't detect new files automatically, so run `sudo mopidyctl local scan` regularly, or even as a nightly cron job.

Audio Software for Raspberry Pi

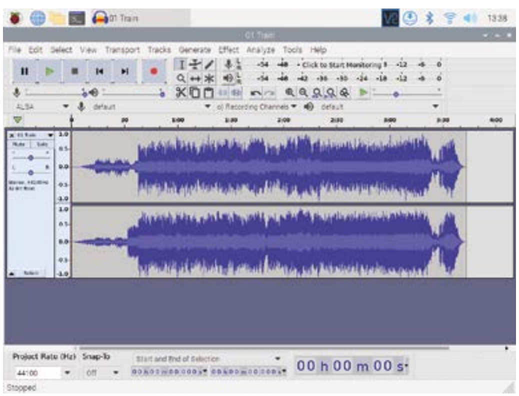
Now that you have your hardware sorted out, how do you control it? **PJ Evans** hits the wheels of steel (or the command line, in this case)

It really doesn't matter how good your Raspberry Pi audio setup is if you can't get any sound out of it. From media player software to studio-grade editing suites, there's a lot available for you to play with once you've got your headphones on. In this section, we will take a look at some of the popular audio tools available for Raspberry Pi OS. We've already covered Mopidy in the previous tutorial, so here's a guide to some of the other audio software packages available both for playback and composition.

Audacity

audacityteam.org

Audacity is a real stalwart of the open-source community. Lovingly improved over many years, from its humble beginnings, the program now reached the level of a studio-ready editing tool. Proper non-linear multitrack editing, direct recording and so many filters it will make your head spin. The plug-in architecture ensures that if there's an effect you need, chances are somebody else has already made it. Whether it's a quick import and mix down to mono, or your 20-track opus, this is the software package you need. It really is astonishing that it is completely open source with no catch.



▲ A powerful user interface and multi-track editing

ffmpeg

ffmpeg.org

This command-line application is truly the swiss-army knife of audio (and video) conversion. If all you want to to is something simple, such as convert a newly-captured WAV file to MP3, there is no quicker way to do it:

```
ffmpeg -i my_kazoo_symphony.wav out.mp3
```

That's it. It will automatically detect quality and channels and make sure everything sounds the same. It supports a huge range of audio codecs, including more complex multi-channel such as AC-3 and DTS. An essential tool for anyone who needs to process audio. Install with `sudo apt install ffmpeg`.

▲ Not pretty to look at, but behind the command line are powerful features

MuseScore

musescore.org

Something a bit different here. If you are a composer or arranger, you may be interested in a package that will help with sheet music notation. Typically, commercial packages such as Sibelius are the go-to tools, but one of the most popular notation programs in the world is open-source. MuseScore provides a wide range of features including MIDI-based input and audio playback, as well as being a fully featured sheet music editor. It can interact with several closed-source packages as well. A great asset for schools and hobbyist composers.

▼ Sheet music notation made easy with this open-source software package

Volumio

volumio.com

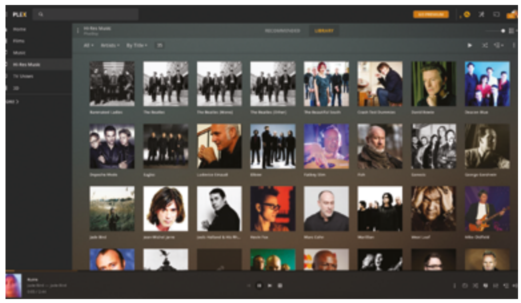
Billed as 'The Audiophile's Music Player', Volumio doesn't mess around. This is a dedicated, fully featured player squarely aimed at the high-end market. There is even a 'plug and play' OS version available right from Raspberry Pi Imager. With automatic support for a wide range of DACs (including many mentioned in this feature), Volumio will have you up and running in no time. Control is via a web interface and you can use a mobile version too, for relaxed sofa command. Its plug-in architecture allows for various music services such as Spotify or Tidal to work seamlessly.

▲ Volumio is a music player and library manager for the serious audiophile

Plex

plex.tv

Although focused on video, Plex still makes for a formidable audio player. You can run your own Plex server at home and have a single place for all your media. Plex will stream audio to any device accessing it via its web interface and many apps. Additionally, it can stream to any Google Chromecast or Sonos device with a PlexPass. Although not suitable for streaming to dedicated devices that do not support these protocols, Plex has a gorgeous interface and is friendly and intuitive to use. The ability to access your content remotely is a free feature (although some router config may be required).



▲ Access your music and videos anywhere with Plex's friendly interface

Top Tips

Simple Command-Line

Raspberry Pi OS comes with the ALSA suite of tools for control, playback and recording from the command line

LibreELEC & OSMC

An honourable mention for these two optimised operating systems that make using Kodi, the popular media centre, quick and easy. Both available from Raspberry Pi Imager.

Sonic Pi

sonic-pi.net

No feature on Raspberry Pi audio would be complete without this amazing piece of software. Code? Music? Why not both? Live Coding is a discipline that uses code to produce sound, and therefore music. Supercollider, a popular engine amongst enthusiasts, is a complex and unwieldy beast, but Sam Aaron and the Sonic Pi team have tamed it in the Sonic Pi environment. For the beginner, the detailed and accessible tutorial is what makes Sonic Pi really stand out. Once mastered, it is capable of live performance and has often been used at events. A great and fun way to learn music theory.

▲ Sonic Pi means code that makes music. Detailed tutorials and a huge range of sounds

Great Audio Projects

Need some inspiration? Here are some of our favourite audio-based projects built on Raspberry Pi

Raspberry Pi Zero 2 Music Player

Drew Batchelor ▶ **magpi.cc/zero2mp**

When Drew needed a new music player for a kitchen, a creative side got the better of him and he decided to design an elegant 3D-printed case for the line-out version of Pirate Audio. It runs Volumio with a modified version of the Pirate Audio plugin to allow for better control operation and a clearer display.

RFID Vintage Boombox

Jorge Miar ▶ **magpi.cc/nfcboomboxyt**

Jorge's project takes the radio concept to the next level. As well as upcycling a fantastic 1980s boombox, an NFC reader has been added so you can control what is played back with cassettes. Place the tape of choice in the player and hit the play button.

Radio Globe

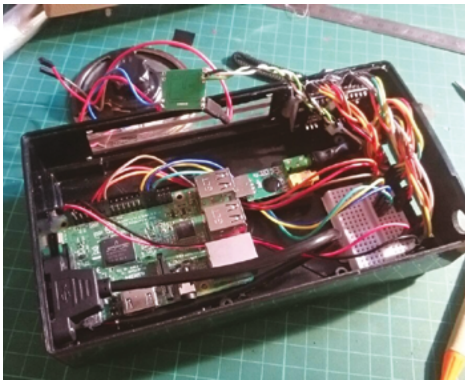
Jude Pullen ▶ **magpi.cc/radioglobe**

A beautiful concept that enables the user to explore over 2,000 radio stations from around the world based on the globe's location. This clever build plugs into Radio Garden (**radio.garden**) to provide the audio stream. Simply rotate the globe to anywhere to hear a nearby station live. Check out Jude's vlog of the build process (**magpi.cc/radioglobevlog**).

GTA Retro Radio Player

Raphaël Yancey ▶ **magpi.cc/gtaradio**

Featured before in these pages, the GTA radio player takes the radio stations featured in *GTA V* and allows the user to tune between them. The retro radio's innards are replaced with a Raspberry Pi and amplifier. A rotary encoder is read via GPIO and Python to move between stations.




THE Official
RASPBERRY PI
HANDBOOK
2022
50 Raspberry Pi
Hacks & Hints
iPod Classic
Spotify Player
BeccaCam
Digital Camera
Raspberry Pi
Pico projects
Build a
Home Assistant
Raspberry Pi
Amiga 600
Walking
Bipedal Robot
Updated!
QUICKSTART GUIDE
Start incredible projects
with Raspberry Pi
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI HANDBOOK 2022

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

CutiePi

SPECS

PROCESSOR:
BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz (Raspberry Pi Compute Module 4, Wireless, 2GB Lite (CM4102000))

DISPLAY:
8" IPS LCD (1280x800)

CONNECTIVITY:
WLAN 2.4 GHz, 5.0 GHz IEEE 802.11 b/g/n/ac, Bluetooth 5.0, BLE, 1x USB type-A, USB type-C, 1x micro HDMI 1x microSD slot

DIMENSIONS:
206(W) x 134(H) x 14(D) mm

▶ CutiePi ▶ **cutiepi.io** ▶ £176 / $229

Billed as the world's thinnest Raspberry Pi 4 tablet, **Rob Zwetsloot** checks out this Compute Module 4-powered portable computer

Raspberry Pi-based tablets are a bit of an untapped market in our opinion. While it's definitely fun to make your own, having one prebuilt and ready to go means you can immediately get to work on any number of coding projects. And even if you don't quite want it for digital making, it could make for a fun tablet. CutiePi here is a bit of a balance of both.

While one of its headline features is that it's very thin (14mm thin to be exact, thinner than a Raspberry Pi 4), it doesn't skimp on screen size with an 8-inch IPS LCD touchscreen running at 1280x800. It's bright and very responsive, the latter thanks to a Compute Module 4 being built into the tablet – in fact this is how it can be so thin while still having the power of Raspberry Pi 4.

There are some sacrifices made to the input and output ports because of this. There's only one USB port, one micro HDMI out port, and no GPIO or headphone ports. While USB ports can be extended with a hub, this adds extra space to something you'd want to be compact. A camera is installed on the rear though, much like other tablets, and you can easily access the microSD card to update the operating system from another computer.

On a final hardware note; we adore the handle on it. And it's not just a handle, it's a stand in a similar way to smart covers. You can have it propped up near vertical in landscape to use like a display with keyboard attached, or raised at a 30 degree angle to peer down on. It can also be used to prop it up in portrait orientation, and is just nice to use to carry it around with.

▶ It's very slim thanks to it using a Compute Module

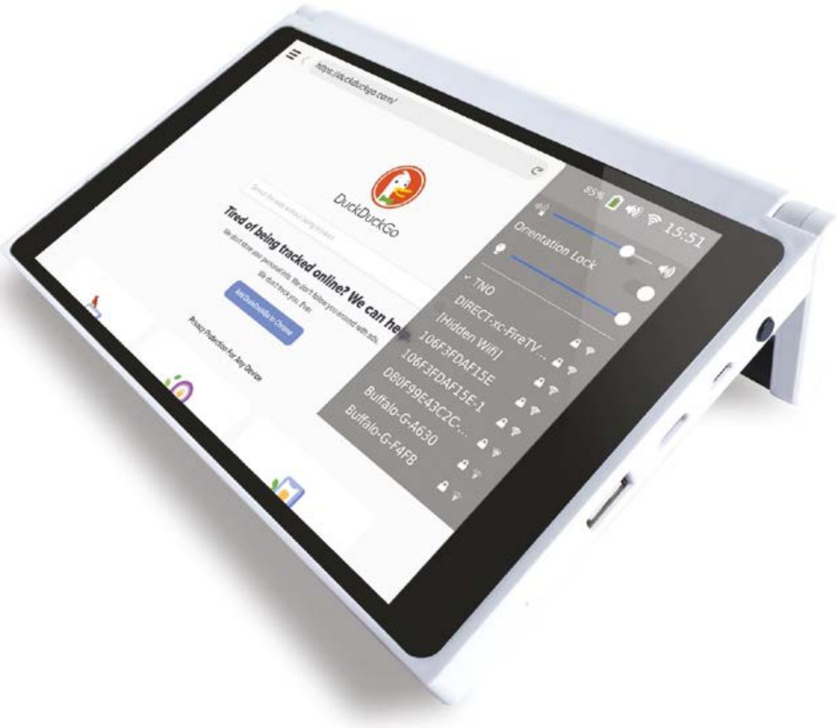
Interfacing

CutiePi uses its own custom graphical interface, known as CutiePi Shell. It's based on a browser, and allows for easier use of the tablet as a touchscreen computer – much in the way that an iPad or other tablet has its own custom display. The onscreen keyboard is very good and responsive, and the orientation of the screen will change as you move it. It's a really nice and clean experience, and has the usual trappings you'd expect, like the ability to turn off the display with a button, a lock screen, and a rotation lock.

You can also press a button and return to Raspberry Pi OS's default desktop, where it will function just like any other Raspberry Pi.

A lack of GPIO pins does mean you're limited in the digital making you can do either way. For pure code it's great, and far more hackable than any other tablet, however if you want to connect it to the real world it's a bit trickier.

For what it may lack in GPIO it makes up for in media playing. YouTube and other video services run great, and the speaker is decent. With a few parental controls this could be a great budget tablet for younger people wanting to explore coding.



> It's not just a handle, it's a stand in a similar way to smart covers

▲ The different orientations are great for however you want to use it

◀ The handle is sturdy and so is the entire build of the tablet

Verdict

While lacking in ports it makes up for a lot with its user-friendly design and interface. Great for younger makers.

8/10

PecanPi **Streamer**

▶ Orchard Audio ▶ **magpi.cc/streamer1** ▶ £480/$550

This black box contains some of the highest quality audio components available alongside a Raspberry Pi 3B, but is it worth it? **PJ Evans** engages his golden ears.

SPECS

DIMENSIONS: 195 x 120 x 100mm

DAC: Dual flagship Burr-Brown PCM1794

SIGNAL-TO-NOISE (RCA): 130dB (A-weighted)

DYNAMIC RANGE (RCA): 125dB

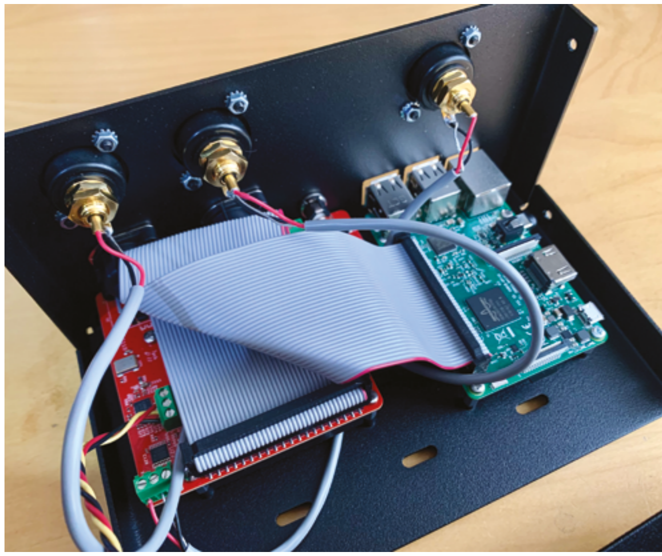
TOTAL HARMONIC DISTORTION (RCA): 0.0003%

Raspberry Pi continues to make waves in the world of high-end audio. At the very top of the pile is Orchard Audio. This company cares not for flashing lights, gimmicks or even Bluetooth, but instead is dedicated to one thing and one thing only: producing the best sound possible. Orchard's unique proposition is the use of the highest quality components, right down to the resistors. So it was with great anticipation that we received the PecanPi Streamer v3.

Minimalist

This case is minimalism taken to a new level. Looking at the rear, we see the ports from a Raspberry Pi 3B, and not only the usual phono (RCA) sockets but XLR sockets too. This unassuming box is aimed right at the professional market as well as the audiophile.

> This case is minimalism taken to a new level



▲ A familiar Raspberry Pi 3B sits next to state-of-the-art audio components

▲ A small black box with an audio monster inside

Driving the PecanPi is Volumio, a popular interface amongst audiophiles. It can accept a number of different services such as Spotify or SoundCloud. In our tests it spotted our local Plex DLNA server immediately and we were playing music without delay. Such is the dedication to pure sound that Bluetooth and wireless LAN are unavailable because the radio interference is unwanted. The PecanPi demands a wired Ethernet connection, and nothing else.

But what a sound. Even with fairly average speakers, *The Dark Side Of The Moon* encoded with FLAC gave amazing detail with a depth and warmth we'd never heard before. This will not be a disappointing product to those who care deeply about how their music sounds.

Verdict

If you can stomach the price, Orchard Audio's humble black Pi-containing box could change the way you think about sound and music.

9/10

Wireframe

Join us as we lift the lid on video games

Visit **wfmag.cc** to learn more

10 Amazing: Home automation projects

Make your home smart with Raspberry Pi

You too can live in the house of tomorrow by using Raspberry Pi to control many aspects of your home. While this issue is all about going outside, here are some things you can do in the comfort of your own home.

▲ Magic Mirror

At a glance

This classic Raspberry Pi project is the ultimate way to help you reflect upon the day's appointments, making sure you look good for whatever you're doing, and also weather/event appropriate.

magicmirror.builders

▲ Smart home door lock

No key required

This smart lock is so secure, it has three-factor authentication that includes an NFC fob, a PIN code, and a one-time code via text. It might secure your biscuits.

magpi.cc/smartlock

▸ PiHue

Clap on

Control the lights in your house just from one Raspberry Pi – and make it fancier by also controlling their colours.

magpi.cc/pihue

▲ Voice controlled coffee machine

Coffee. Black. Hot.

While *Star Trek* replicators might not exist yet, you can at least yell at your coffee machine to make you an espresso to get you going in the morning

magpi.cc/snipscoffee

▲ Home music system

Soundtrack for life

Ditch the high-end systems for something a little cheaper and more customisable, like this tutorial series from *The MagPi* regular PJ Evans.

magpi.cc/musicsystem

▲ AlarmPi

Smart wake up

The AlarmPi not only wakes you up with a buzzer, it will even read you the news and weather from a specified RSS feed

magpi.cc/alarmpi

▶ Smart doorbell

See who's there

Some smart doorbells notify you of a ring, and allow you to check who's at the door – this one lets you also have a video chat with them if you so desire.

magpi.cc/smartdoorbell

▲ Raspberry Pi refrigerator

Is it cold?

Power outages to fridges and freezers, especially if they happen without you noticing, can be devastating. Always know if and when your fridge is warming up with this sensor.

magpi.cc/refrigerator

▶ Ultimate media centre

Games, movies, photos

Save some HDMI ports by having all your visual entertainment means served via one Raspberry Pi in this ultimate home theatre.

magpi.cc/102

▲ Raspberry Pi thermostat

Warm your home

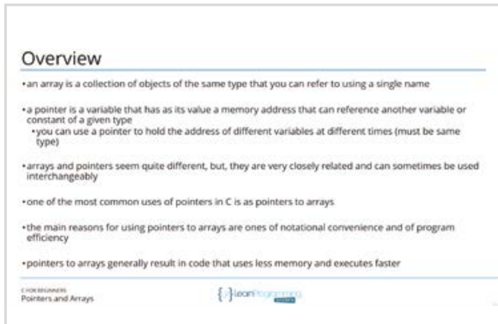
A smart thermostat can help you heat your home more efficiently, and save money in the long run if set up properly. It's also very hacky though, so be careful.

magpi.cc/thermostat

Learn C with Raspberry Pi

Start coding in the very popular C language with these resources. By **Phil King**

C Programming For Beginners – Master the C Language



CREATOR
Udemy

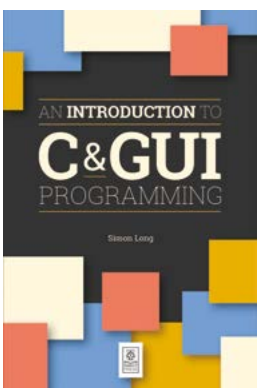
Price:
£60 / $110
magpi.cc/udemyc

From its origins in the early 1970s, C had grown to become one of the most widely used programming languages in the world. It has even spawned the spin-offs C++ and C#. A mid-level language, C is packed with useful functions and is known for its speed. The downside is that it's a little harder to learn than interpreted high-level languages such as Python.

If you're a newcomer to C, one of the best resources is this well-structured Udemy course. It normally costs £59.99, but is often available on a special offer for a lot less – £19.99 at the time of writing.

Following an overview and how to install the required software, it starts with coding a simple project with Code:Blocks and then Visual Studio Code, before moving onto key topics including variables and data types, operators, control flow, arrays, functions and, of course, pointers (which often confuse beginners). Everything is covered in great detail and you can learn at your own pace.

Reading material



AN INTRODUCTION TO C & GUI PROGRAMMING

Written by Raspberry Pi's own Simon Long, this two-in-one book comprises an introduction to learning C on Raspberry Pi, followed by a guide to creating GUIs with it.

▶ magpi.cc/guibook

C PROGRAMMING IN EASY STEPS

Aimed at beginners, this easy-to-follow guide features step-by-step tutorials that will give you a sound understanding of C and the ability to write and compile your own programs.

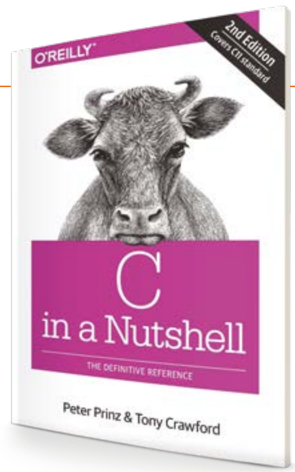
▶ magpi.cc/cineasysteps

PROGRAMMING IN C

Its 544 pages cover everything from the fundamentals to advanced topics in great detail. It also features practical examples, making it an ideal companion for an online course.

▶ magpi.cc/ckochan

C in a Nutshell, 2nd Edition



CREATOR
Peter Prinz, Tony Crawford

Price:
£56 / $70

magpi.cc/cinanutshell

For more experienced coders looking for a comprehensive reference guide to the C language, this hefty book comes highly recommended.

Its 812 pages are crammed with detailed explanations of every feature in C and its runtime library, including multithreading, type-generic macros, and library functions.

It should prove a very useful resource for those doing a C course, or anyone seeking to brush up on their knowledge. Want to understand an unfamiliar function? You'll find it here, along with an example.

The book is organised into three main sections: firstly, C language concepts and language elements, with separate chapters on types, statements, pointers, memory management, I/O, and more. Then the C standard library, and finally basic C programming tools in the GNU software collection.

learn-c.org

CREATOR
Interactive Tutorials

Price:
Free

learn-c.org

If you fancy just diving in and having a go at C coding, this free interactive site is a good place to start. There's no software to install – just follow the instructions for each example and type in your code. The Basics section guides you through fundamental topics such as variables and types, arrays, conditions, strings, loops, and functions.

Once you've got the hang of those, you can move onto an advanced set of tutorials covering aspects such as pointers, structures, dynamic allocation, recursion, linked lists, binary trees, unions, function pointers and bitmasks.

Each mini tutorial comprises some text to read, followed by an exercise where you will need to change or write code in the online editor to achieved the desired output. It's a fun way to learn.

Online courses

Study C coding online with these courses

C PROGRAMMING TUTORIAL FOR BEGINNERS

If you don't mind video learning, check this out. With a running time of nearly four hours, this YouTube tutorial from CodeCamp.org takes you through all the key aspects of the C language.

▶ **magpi.cc/codecampc**

CS50'S INTRODUCTION TO COMPUTER SCIENCE

This free course from Harvard University is an excellent starting point for beginners to programming in general. Most of the content is in C and there's a cloud-based IDE for coding.

▶ **magpi.cc/cs50**

ADVANCED C PROGRAMMING: POINTERS

Understanding pointers in C can be confusing for beginners. While this course is aimed at more experienced C coders, it offers expert tips on how to use pointers. Code examples are provided.

▶ **magpi.cc/cpointers**

Selin Ornek

A young maker who loves dogs, robots, and Star Wars, and finds ways to combine them

- Name **Selin Ornek** | Occupation **Student**
- Community role **Maker** | Url **magpi.cc/selin**

Last issue we featured the amazing Kimberlina robot, the Star Wars-inspired Raspberry Pi Pico battle bot. Its creator, Selin Ornek, loves to make robots and is the captain of her school's FRC (First Robotics Competition) team, River Robotics.

▲ Hashmet is able to recognise people by their face

"I started coding when I was eight years old and was building robots when I was ten." Selin says. "I have built six robots; ic4u and ic4u2 are guide dog robots for blind people, BB4All is a school assistant robot [whose] main goal is to prevent bullying. Both robots are voice-controlled and can interact with people. My other robots which I built just for fun are "Hashmet the Android", the one-wheeled Star Wars droid "D-0" and "Kimberlina" the battlebot, as I love androids and Star Wars."

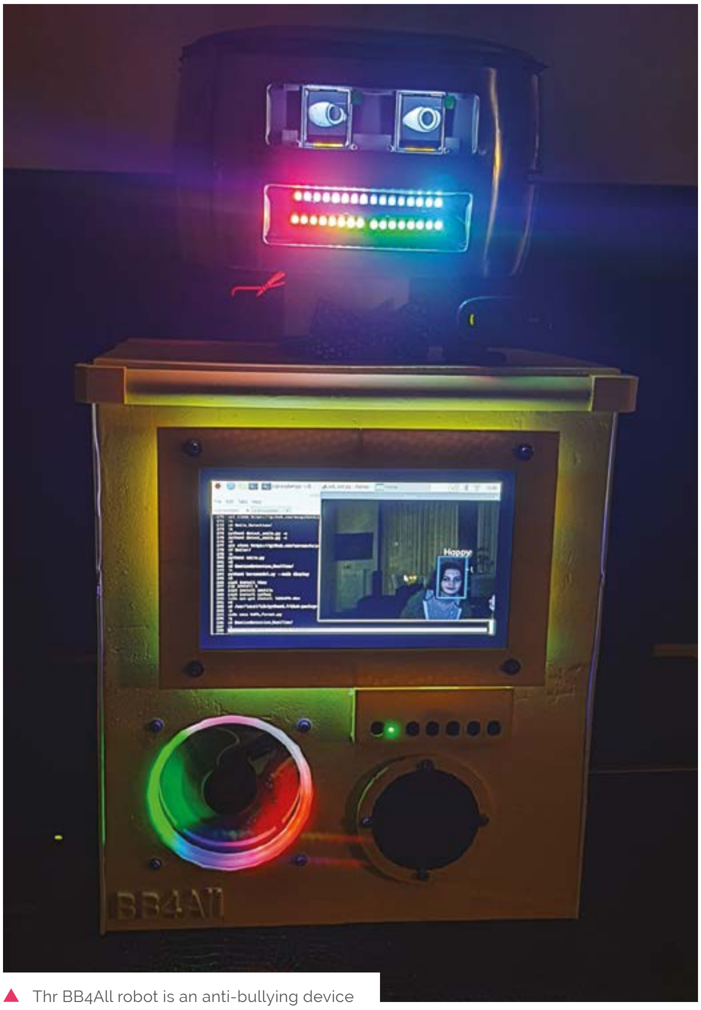
> I started coding when I was eight years old, and was building robots when I was ten

What is your history with making things?

When I was eight years old my dog Korsan passed away. I was really upset and wanted to bring him back to life. I had to do a math project for school, and I had to interview a technical professional, so I asked my dad's friend, a mechanical engineer, lots of questions then showed him a drawing of a robot that I wanted to make to bring Korsan back to life. He told me that I should learn coding and robotics. My English teacher was also using MIT Scratch in class to make the lessons more fun. I asked him how he made the games, as I loved playing *Minecraft* at that time. He encouraged me to try and make games myself using Scratch.

I also started to learn how to code from books and the internet. My parents then bought me a Mbot that I took to pieces and made into a little robot that I called "çirozbot" which means skinnybot. I used Scratch to code MBot and that is when I learned that I would be able to make more advanced projects using Arduino and C++.

The first project I built was a smart collar for dogs with Arduino. I then used Arduino in the first version of IC4U. I had decided to take IC4U to Dublin to Coolest Projects International 2018, whilst building him I



Thr BB4All robot is an anti-bullying device

IC4U2 out and about as a robotic guide dog

entered one of Coolest Projects' social media competitions, and won my first Raspberry Pi Zero. Then I won first place in the hardware category at Coolest Projects, and one of my prizes was a Raspberry Pi 3B+. After this, I started to use Raspberry Pi in all my projects.

How did you learn about Raspberry Pi?

As I got more confident coding and using more advanced products in my projects, I started looking up other options I could use. Raspberry Pi was on the top of my list as I was looking for a powerful microcontroller and I was also very eager to learn coding with Python. Unfortunately, these were quite expensive, especially as I try to pay for all the parts I use for my robots out of my allowance. So I was extremely happy to win a Raspberry Pi Zero and Raspberry Pi 3B+ and start using them.

What was the first thing you made with Raspberry Pi?

IC4U2, [which was] the second version of my robot guide dog for blind people.

I had built IC4U as when we were on holiday in France, I saw a guide dog with its visually impaired owner. I had never seen one before, as there were only a couple in Turkey at that time. It really made me think of Korsan and how sad I was when he died. I began to think that if *I* had been so upset, how would a blind person feel? Not only would they lose their best friend, but they would also lose their eyes again.

So I decided to build IC4U, my robot guide dog. The Guide Dogs UK association and the American Guide Dog association were amazing, answered all my questions, and explained just how a guide dog is trained and what they should do. I built the first version using Arduino, IC4U and I won first place in Coolest Projects International 2018 in the hardware category. I was also very kindly given a Google AIY Voice HAT and Vision Kit. As soon as I got back home, I started to build IC4U2.

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. We love to hear how the Raspberry Pi Foundation's resources are being used to teach

02. The return of the mini mp3 player, albeit one that can also play videos, games, etc...

03. The aesthetics of this robot are great – like a futuristic Pi Wars competitor

04. Cool use of computer vision by Lorraine – we wonder if it could be used for projection mapping?

05. The Switch Lite is probably one of the better handhelds to hack to be a retro console, good work

06. Yes, this is an April Fools. Yes, we fell for it. We saw it a few days after, to be fair.

07. A LEGO games console? It's happened before.

08. An exceptionally cool mirror build

09. We would happily beta test RaspberryPints

10. A cyberdeck made out of LEGO? You've piqued our interest

11. Such a big speaker powered by such a small computer

01

Amanda Haughs
@MsHaughs

Replying to @TheMagPi

TONS of new learning for me & 7th graders this month! Learning to program @Raspberry_Pi Pico, Kitronik motor driver, LCD screen so Ss can prototype their "Edu of Future" TA robot that would help Ts provide more individualized instruction & emotional support @RaspberryPi_org

02

Drew_Makes
@drew_makes

Replying to @TheMagPi

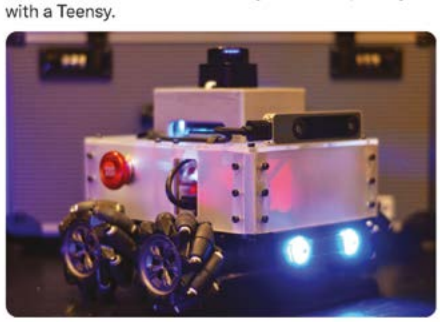
I made a little music player using a Raspberry Pi Zero 2 with a 3D printed case. Details are here: drewbatchelor.com/portfolio/pi-z... #raspberrypi

03

Aditya Kamath
@kamathsblog

Replying to @TheMagPi

Finishing up a robot project with a Pi4. It has an Arduino in there, and I'm working towards replacing it with a Teensy.

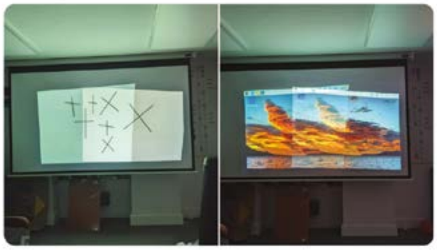


04

Lorraine Underwood
@LMcUnderwood

Replying to @TheMagPi

I need to line up several projectors for a big exciting project that's coming soon! I'm using raspberry pi cameras alongside the projectors, openCV and some cool maths to change the image so the projectors look lined up.



05

StonedEdge
@StonedEdgeMods

Replying to @TheMagPi

Finished this a while ago now, but its my CM4 in a Switch Lite shell. Collaboration project with my good friend @CNCDan

06

diyelectromusic
@diyelectromusic

Replying to @TheMagPi

Well I dug out my Raspberry Pi MT-32 Pi as part of my latest #LoFiOrchestra performance on Friday!
I'd say this was my most challenging performance to date :)
#MagPiMonday

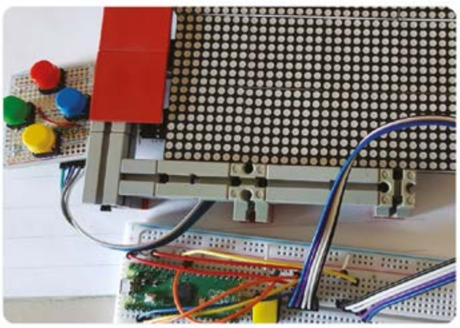
> **diyelectromusic** @diyelectromusic · Apr 1
> The #LoFiOrchestra proudly presents a performance of John Cage's 4'33".
>
> Featuring 18 @Arduino Unos, 11 Nanos, a Pro Mini and a @Raspberry_Pi this might well be the first time this has been performed using 8-bit synthesis on microcontrollers.
>
> youtube.com/watch?v=sxbi8q...
> Show this thread

07

Pierre Poliakoff
@ON4LPF

Replying to @TheMagPi

Trying to build a retro game console with a Raspberry Pi Pico (resolution 24x32)



08

Cellar Nerd
@CellarNerd

Replying to @TheMagPi

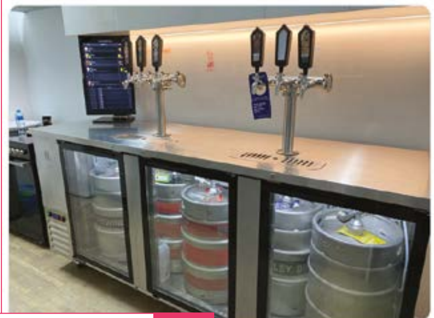
Finished my #jameswebbspacetelescope model, powered by a Pi2 and published a video detailing the project:
youtu.be/piM59_2R2Wo

09

DeKay
@Tooth_DeKay

Replying to @TheMagPi

Just been putting the finishing touches on the RaspberryPints beer tap monitoring system for the office bar. Uses a Raspberry Pi 4 & an Arduino connected to flow sensors in the beer line. All nicely displayed on a monitor.



10

Penk Chen
@penk

Replying to @TheMagPi

Not my build, but this Lego Pi deck by Floyd is just too good not to share

> **Penk Chen** @penk · Apr 4
> Check out Project Orinoco, an amazing Pi deck with a Lego case built by Floyd
>
> (my inner child is screaming rn :-)
>
> floydhayes.com/orinoco

11

Kevin Just
@juztKevin87

Replying to @TheMagPi

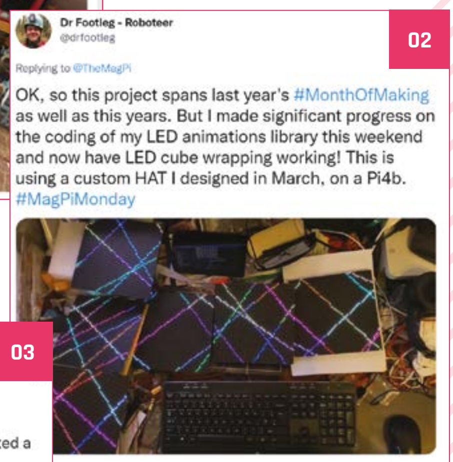
Finally finished my arcade speaker box design, printed it and during testing (playing some retropi games), I figured it would be nice to have a pot (a @pimoroni breakout) that allows adjusting the volume. Added that to the design as well, wrote a script for that and voilà 🕹

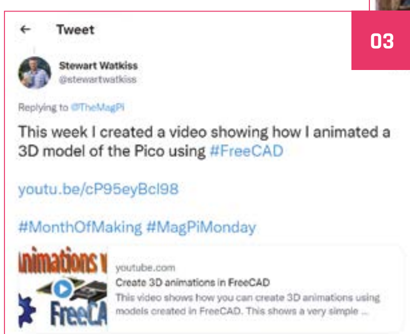
#MonthOfMaking

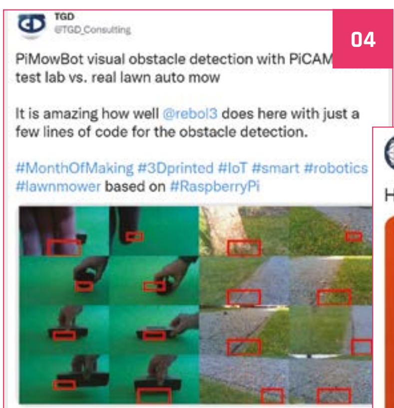
In March, we make

Over the month of March, you folks in the community engaged in the communal building activity, #MonthOfMaking. Here are some of the cool things that were shared with us over March.

Didn't finish your project? No worries. Keep going, and show it to us on a #MagPiMonday.







01. We love Alex's enthusiasm for bringing making to Puerto Rico every month of the year

02. Just because you haven't finished a project, that doesn't mean you can't continue. Footleg's cool LED animations keep on going

03. Creating 3D models of RP2040 is a cool thing for making videos and helping with 3D print designing

04. An important part of the machine learning in the PiMowBot

05. Macro keypads are very cool, and very useful, even with just six buttons

06. This cool rabbit robot (rabot?) will be coming to Pimoroni…

Crowdfund this!

Raspberry Pi projects you can crowdfund this month

PiSquare

A solution to the problem of using multiple HATs with your Raspberry Pi, PiSquare attaches to the HAT, plugs into power, then communicates wirelessly with the Raspberry Pi board to facilitate the wearing of as many HATs as you could possibly need.

▶ **kck.st/3NHZILm**

Vilros Reveal

This cool looking case allows you to add an M.2 SSD to your Raspberry Pi, and also comes with a little CPU fan and RGB lights so that is looks like your main rig. You still won't be able to play *Apex Legends* on it though.

▶ **kck.st/3ImoFrU**

Best of the rest!

Other amazing things we saw this month

AMBILIGHT PROJECT

› **magpi.cc/ambihyper**

Apparently this took some time to make, but the result of this HyperHDR ambilight project is absolutely stunning.

COMPUTE MODULE EV CHARGER

› **magpi.cc/evcharger**

This Raspberry Pi in the wild helps charge electric vehicles! Compute Modules get everywhere, it seems.

SUN TRACKER

› **magpi.cc/suntracker**

A custom case, Raspberry Pi Pico, and an e-ink screen? This battery-powered sun tracker looks and sounds very cool and fairly simple to make.

Your Letters

Default user

Do you know if there is a way to change the user name on Raspberry Pi? I don't want to use pi for my specific set up.

Lea via Facebook

If you're starting your project from scratch, we have good news: you won't need to change it anymore! The latest version of Raspberry Pi OS allows you to set a username during install so you can have whatever you wish.

You can change the username on older versions of Raspberry Pi OS by doing the following in the terminal:

sudo passwd root

To set a password for the root user, a kind of Linux user that can control everything.

logout

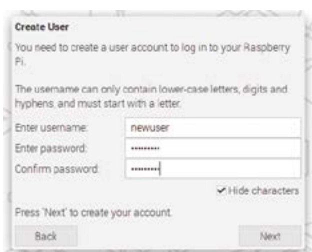
Then login as root using the password you set. You can then change the pi user with:

usermod -l [newuser] pi

With your desired username replacing [newuser]. Finally, update the home folder with:

usermod -m -d /home/[newuser] [newuser]

Log back in as usual, and you're done!



Raspberry Pi's new setup wizard allows you to set both the username and password at install

Check out the time machine radio project in issue 116 of *The MagPi*!

Time travel

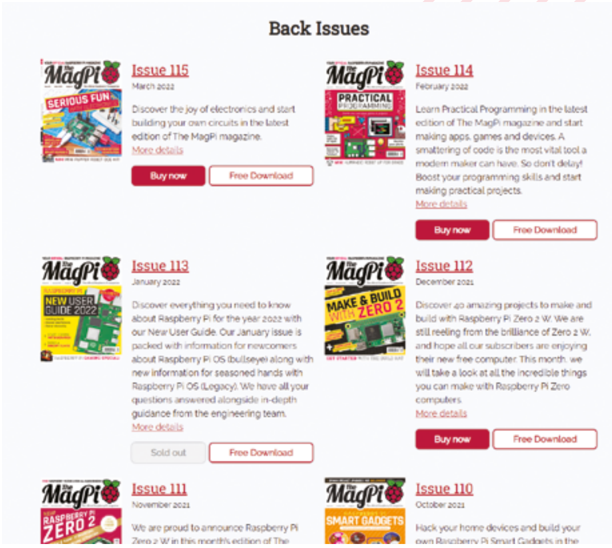
My issue 115, March 2022 on the cover in the lower right-hand corner has this: Build a Time Machine Radio, and yet nowhere in the issue is there information about building any time machine radio. I then downloaded issue 115 and there is something different on the cover in the lower right hand corner. What happened to information about Building a Time Machine Radio?

Tim via email

Due to the way magazines are produced, we send our cover off to print before we send the rest of the magazine. It was accidentally promoted on the cover, and did not make it into that issue. With the digital edition, we're able to make updates that we can't on the print cover so we tweaked it for that.

In the case of the Time Machine Radio, it did end up going into issue 116 (**magpi.cc/116**), and is a wonderful project.

All our back issues are available for free as PDFs from our website: **magpi.cc/issues**



PDF cache

I have downloaded the latest issue of *The MagPi* magazine (Issue 116) and the cover image is corrupted. The text "Hidden Hacks" cannot be read clearly, as it is partly covered in yellow. While viewing via Microsoft Edge, asimilar corruption happens on page 32.

This corruption is worse when the file is viewed via Edge but is still present in Mozilla Firefox, whereas via Sumatra PDF it is almost correct but has dots in the text "Hacks".

Re-downloading the file does nothing to improve matters.

Richard via email

A few days after he sent this email, we advised Richard to try downloading the PDF again, which he'd already done and it had worked. It is always worth clearing your cache if you can, or trying the download in another browser. Make sure to try a different PDF viewer as well, browser ones don't always work as expected.

If the problem persists, do contact us and we can usually see if there's a problem and send you the PDF directly.

Contact us!

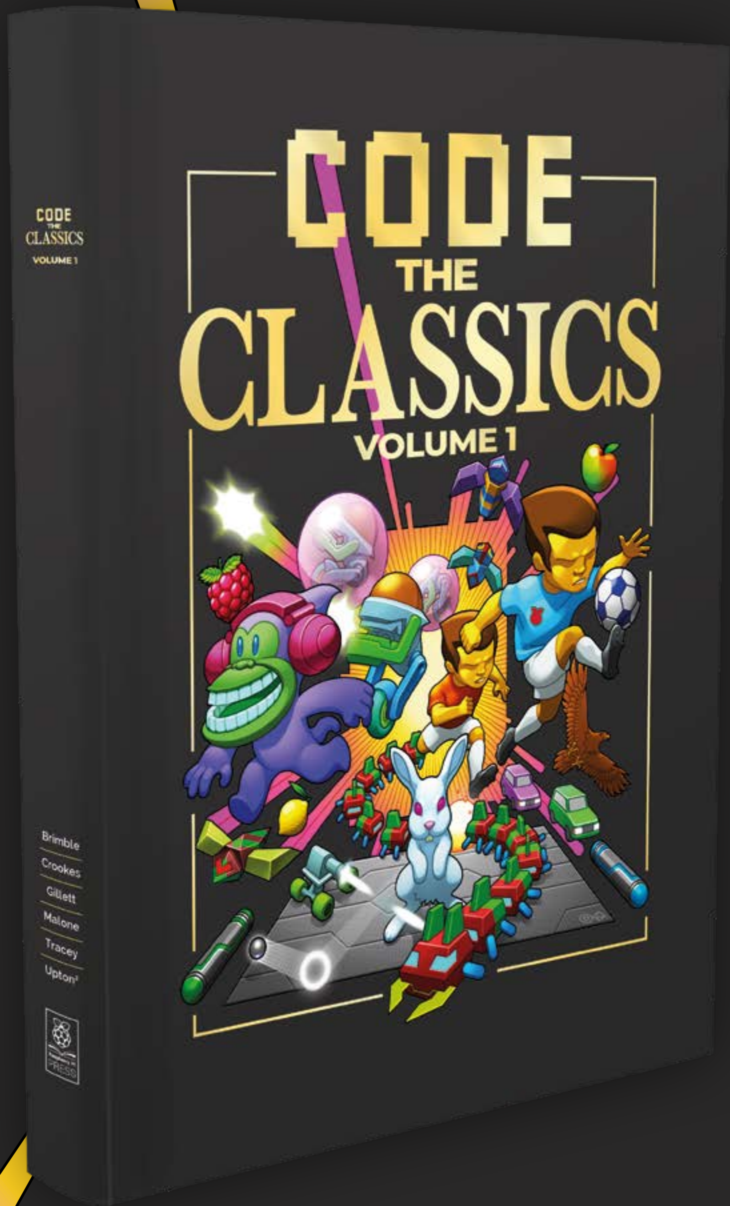
- Twitter **@TheMagPi**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.com**
- Online **forums.raspberrypi.com**

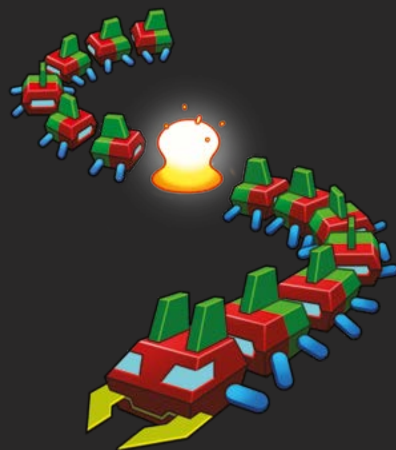
3 ISSUES FOR £5

Subscribe by phone:
01293 312193

Subscribe online:
magpi.cc/subscribe

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





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

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.

Available now: **magpi.cc/store**

WIN 1 OF 5 ARGON EON

IN ASSOCIATION WITH ARGON40

The Argon EON allows you to add four hard drives to one Raspberry Pi very easily for a huge and customisable NAS. This month, we have five to give away.

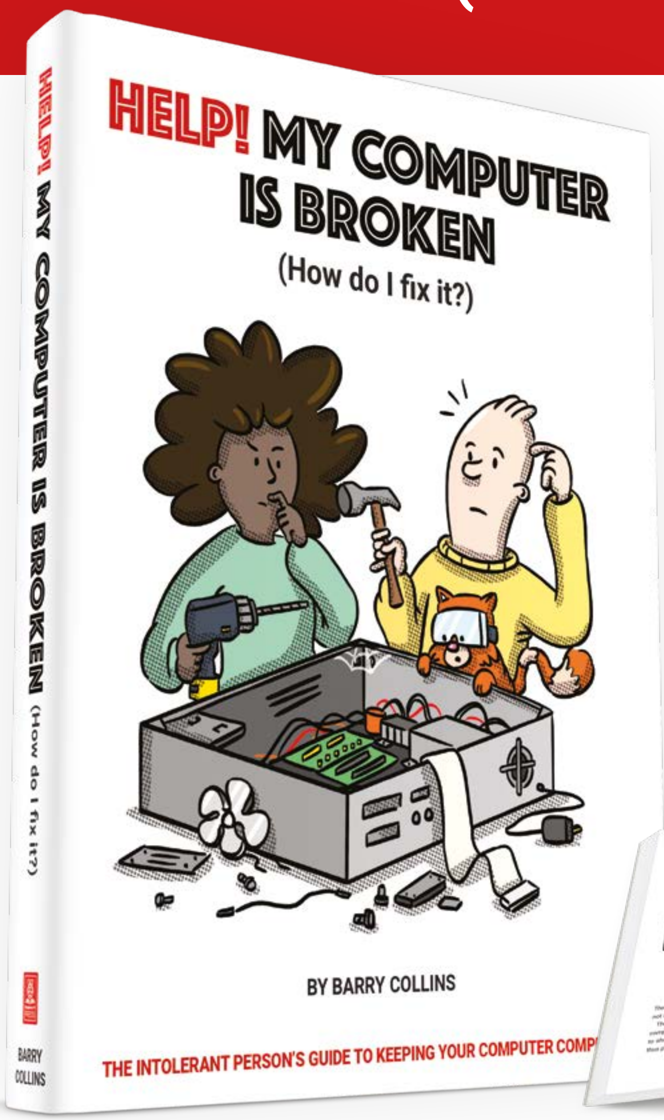
Head here to enter: magpi.cc/win | **Learn more:** argon40.com

Terms & Conditions

Competition opens on **27 April 2022** and closes on **26 May 2022**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

HELP! MY COMPUTER IS BROKEN

(How do I fix it?)



Help! My Computer Is Broken takes the most common computer problems and tells you how to fix them. It's as simple as that! If you've ever wondered why your laptop won't turn on, you can't get a WiFi connection, your printer isn't printing, or why everything is so slow – well, this is your book…

BUY ONLINE: magpi.cc/helpbook

Photography Special

THE MAGPI #118 ON SALE 26 MAY

Plus!

Raspberry Pi art & design projects

Monitor air quality

An EMP-proof Raspberry Pi

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
Ian Evenden

ADVERTISING

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

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Olivia Mitchell, Sam Ribbits, Ty Logan

Illustrator
Sam Alder

CONTRIBUTORS

David Crookes, PJ Evans, Ian Evenden, Rosemary Hattersley, Jo Hinchliffe, Phil King, Simon Long, Stephen Smith, Eben Upton

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

recycle

CC BY NC SA

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).

Raspberry Pi PRESS

ISSN: 2051-9982.

History lesson

Knowing our place in time. By **Lucy Hattersley**

Raspberry Pi is a defining point in the history of computing, I believe. There is a famous photograph of the ELLIOT computer being delivered to the City Treasurer's Department in Bethel Street, Norwich in 1957 (**magpi.cc/norwichcomputer**). When we first put a free Raspberry Pi Zero on the front of *The MagPi* magazine, one reader took a photograph of it outside the same building 58 years later. Having so much computing power, light enough to be attached to the front of a magazine, truly is a modern marvel.

Print magazines are important for legacy preservation, as well as being a great monthly joy to read. Every copy of The MagPi magazine is filed with the British Library (**magpi.cc/legaldeposit**). The legal requirement for publishers to file copies of publications has existed in English law since 1662.

In 60 years' time, when people are looking back at the history of computing and Raspberry Pi's involvement during this period, I like to think that *The MagPi* magazine is one of the places that they will look.

We're hoping to spend a little more time looking at the history of computing here in *The MagPi*. In 2021 Tim Danton wrote a book for our sister publication *Wireframe* called *The Computers that Made Britain* (**magpi.cc/ctmb**). It's a wonderful book and I fully recommend picking up a copy. It's packed with the stories of all the amazing people who made the computers that led to Raspberry Pi.

> " Print magazines are important for legacy preservation "

Meanwhile, we have been looking at the practicalities of using Raspberry Pi to explore computing concepts, covering both classic computers and ultra-modern coding techniques, in our *Retro Gaming with Raspberry Pi* (**magpi.cc/retrogaming**).

Moving forward, I'm hoping to combine the two, looking at some of the stories from those classic computers and emulating them with Raspberry Pi computers.

We've also been making a concerted effort to go both low and high-end in recent editions of *The MagPi* magazine. Thanks to those readers who noticed and wrote in with compliments. The idea is to welcome newcomers with introductory projects and basic computing techniques, while simultaneously having more detailed projects and programming techniques for those of us who are further along our computing journey. Moving forward we hope to go to higher coding peaks and further back in time while remaining friendly and at the cutting edge. Exciting stuff!

Of course, there are problems with computing in the here and now. While it is vastly more accessible these days than back in the 1950s, it's fair to say that getting hold of a Raspberry Pi computer is currently a bit of a challenge.

I'm glad Eben Upton spoke out again on these issues (see *Production and supply chain update*, page 8) to give Raspberry Pi fans an indication of where things are at. Hopefully, as the global supply chain eases, we will be able to pick up as many Raspberry Pi computers as we want, whenever we want.

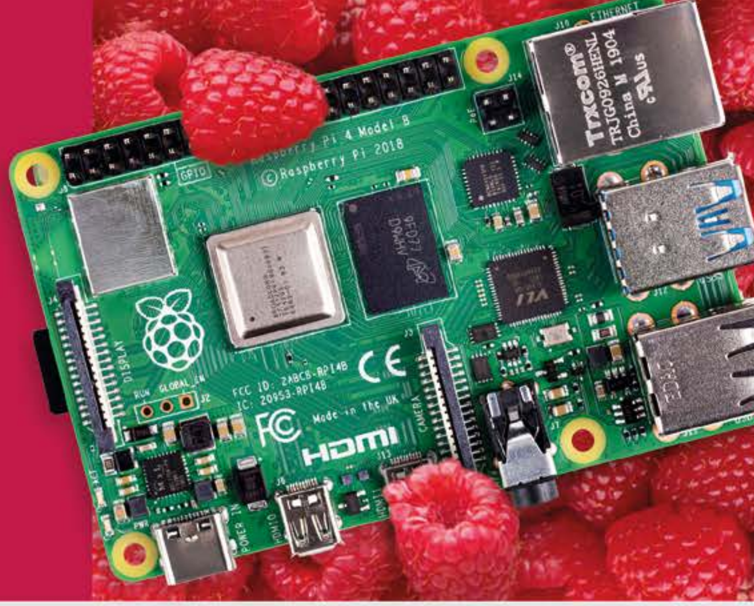
AUTHOR

Lucy Hattersley

Lucy has been making magazines for a while now and still loves the smell of freshly printed mags.

magpi.cc

American **Raspberry Pi** Shop



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

Partner and official reseller for top Pi brands:

adafruit · sparkfun ELECTRONICS · PIMORONI · HiFiBerry · and many others!

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

USA PiShop.us

Canada PiShop.ca

Raspberry Pi APPROVED RESELLER

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

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

Available at these great Pi stores:

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

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