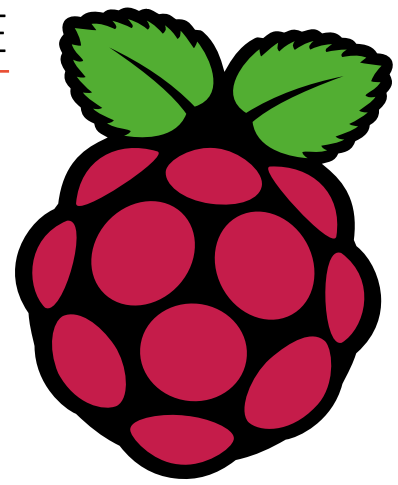




BUY IN PRINT **WORLDWIDE** MAGPI.CC/STORE

The *MagPi*



Issue 102 | February 2021 | magpi.cc

The official Raspberry Pi magazine

BUILD THE ULTIMATE MEDIA CENTRE



Games! Movies! Music!
This all-in-one does it all

14-PAGE
PICO
GUIDE

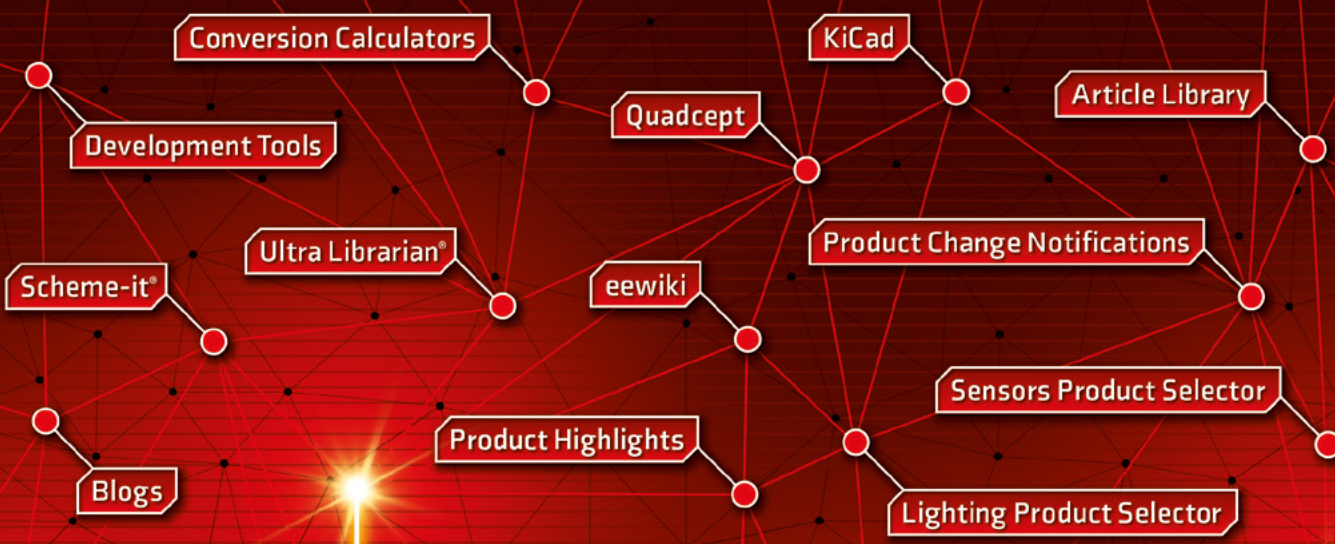
Make a
Commodore 64
retro computer

Electronics
with Raspberry
Pi 400



40 PAGES OF PROJECTS & TUTORIALS

9.6 MILLION+ PRODUCTS ONLINE | 1,200+ INDUSTRY-LEADING SUPPLIERS | 1.9 MILLION+ PRODUCTS IN STOCK



YOU INNOVATE. WE CAN HELP. [DIGIKEY.CO.UK](https://www.digikey.co.uk)



The World's Largest Selection of Electronic Components Available for Immediate Dispatch™

*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. © 2020 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

WELCOME

to *The MagPi* 102

There are times when Raspberry Pi releases something that changes everything. Raspberry Pi Pico (page 60) is exactly one of those moments.

Pico is a ‘microcontroller board’. Smaller, even, than a Raspberry Pi Zero, it is made possible thanks to a brand-new, ultra-fast, super-small, silicon chip called RP2040. Silicon designed by Raspberry Pi itself. How exciting!

I’m smitten with Pico and the possibilities it offers. I spent most of the Christmas break playing around with it. It’s genuinely revolutionary. Gareth Halfacree has written us a massive feature, and we’ve got a full tutorial and guide to getting started.

Meanwhile, Rob has built the ultimate media player with Raspberry Pi 4. This incredible project can do it all: play movies, run games, stream online video, photos, music. You name it.

On top of that, we’ve got some amazing tutorials this month. Check out KG’s Commodore 64 where they turn a Raspberry Pi 400 into a pixel-perfect retro computer (page 50), and PJ’s guide to Raspberry Pi 400 electronics with Flat HAT Hacker (page 40).

This issue is packed with Raspberry Pi and the ever-increasing of things you can make with them. 2021 is going to be an amazing year to be a digital maker. Happy new year!

Lucy Hattersley Editor



EDITOR

Lucy Hattersley

Lucy is editor of *The MagPi* and quite firmly recommends that you take out a subscription to *The MagPi* magazine right away.

@LucyHattersley





The
MagPi

HackSpace
TECHNOLOGY IN YOUR HANDS

CUSTOMPC

3 ISSUES FOR £10



FREE BOOK



magpi.cc/freebook

Subscribe to The MagPi, HackSpace magazine, or Custom PC. Your first three issues for £10, then our great value rolling subscription afterwards. Includes a free voucher for one of five fantastic books at store.rpiexpress.com/collections/latest-bookazines
UK only. Free delivery on everything.

Contents

► Issue 102 ► February 2021

Cover Feature

32 Build the ultimate media centre



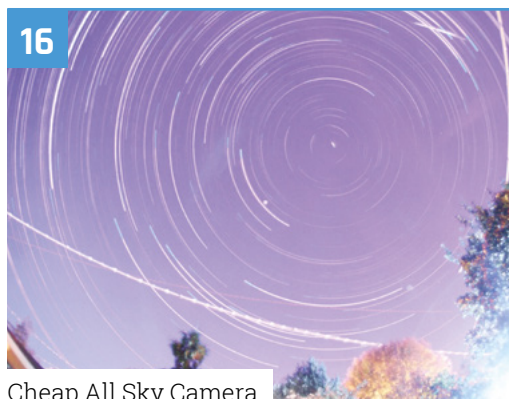
32

Regulars

- 08** World of Raspberry Pi
- 92** Your Letters
- 97** Next Month
- 98** The Final Word

Project Showcases

- 10** We Still Fax
- 16** Cheap All Sky Camera
- 18** Turing Machine Demonstrator
- 20** WeCount Traffic Sensors
- 22** Jazz Champion
- 24** Hubble Pi
- 28** RT Jam



16

Cheap All Sky Camera



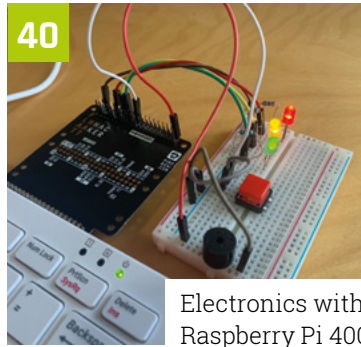
18

Turing Machine Demonstrator

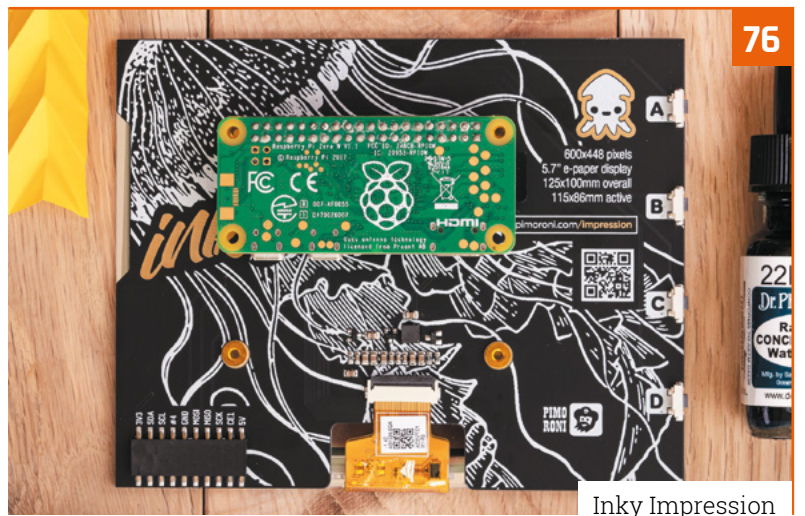
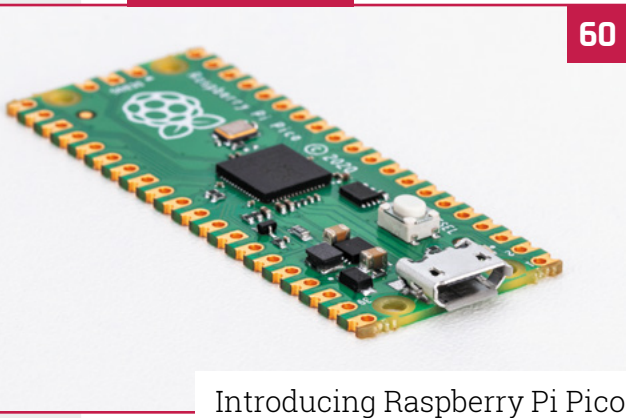
DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi (Trading) 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.

Tutorials

- 40 Electronics with Raspberry Pi 400
- 46 Create GUIs with Python – part 2
- 50 Raspberry Pi 400 C64 emulator
- 54 Add face recognition to projects
- 56 Cheap Trills – part 1



The Big Feature

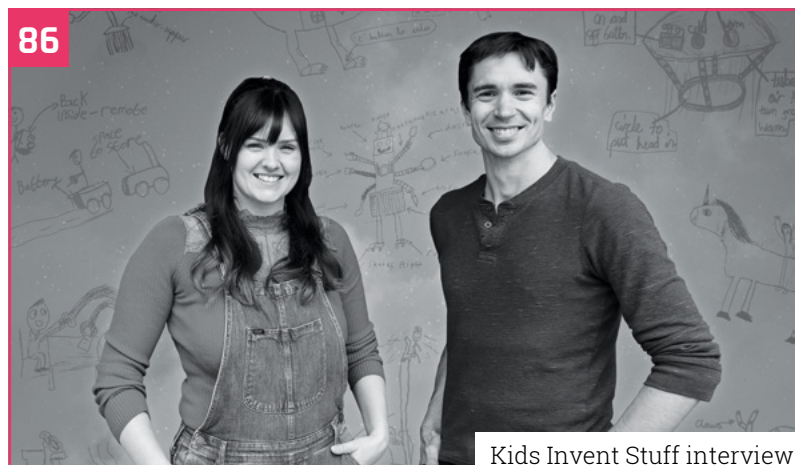


Reviews

- 76 Inky Impression
- 78 RFID HAT
- 79 NanoSound ONE
- 80 Grow Kit
- 82 10 Amazing: Day-to-day projects
- 84 Learn MicroPython

Community

- 86 Kids Invent Stuff interview
- 88 This Month in Raspberry Pi

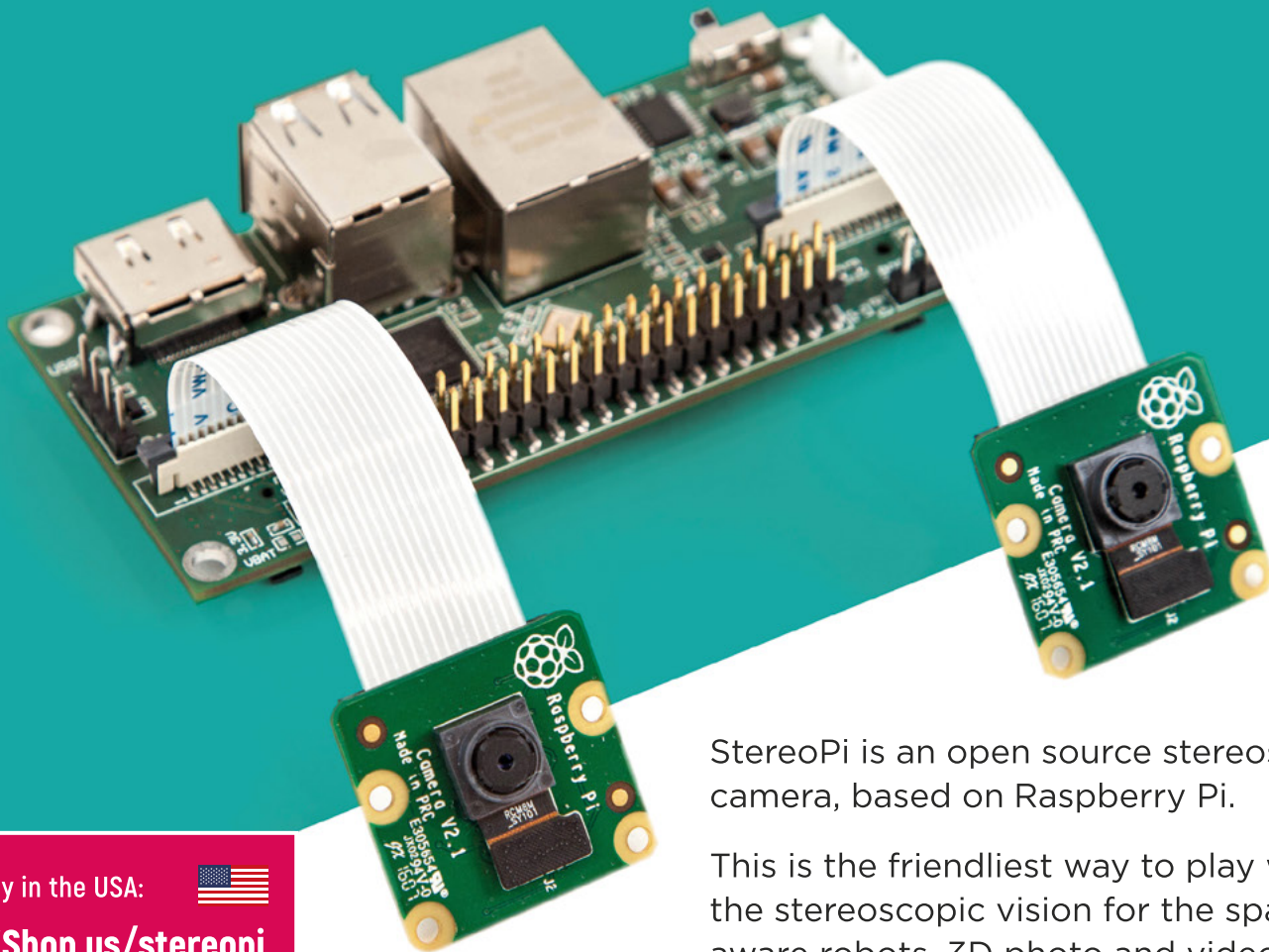


WIN
1 OF 10

RASPBERRY PI PICO

95

Do you know HOW ROBOTS SEE?



StereoPi is an open source stereoscopic camera, based on Raspberry Pi.

This is the friendliest way to play with the stereoscopic vision for the spatially aware robots, 3D photo and video!

Buy in the USA:



PiShop.us/stereopi

Buy in Canada:



BuyaPi.ca/stereopi



RASPBERRY PI INSIDE



STOCK RASPBIAN
SUPPORT



OPEN SOURCE



CROWDFUNDED
PROJECT

LinuxGizmos.com

"The StereoPi can capture, save, livestream, and process real-time stereoscopic video and images for robotics, AR/VR, computer vision, drone instrumentation, and panoramic video."

MickMake

"With it you can do things like, stream stereoscopic 3D video to YouTube, build real-time depth maps using OpenCV, create panoramics using Hugin and even a 3rd person view of real life. Cool."

Raspberry Pi Blog

"There are some excellent community efforts too, of which our current favourite is this nifty dual camera board."

Hackster News

"You can hook this up to YouTube, to Oculus Go, you can use it with OpenCV.. I cannot wait to start messing around with these because it's basically a dream come true."

Raspberry Pi releases IQaudio products

IQaudio range joining Raspberry Pi is a sound move. By **Lucy Hattersley**

Raspberry Pi is bringing four IQaudio products to market after acquiring the brand in late 2020.

“This is the first time we’ve brought third-party products into our line-up like this,” says Roger Thornton, Principal Hardware Engineer at Raspberry Pi, on Raspberry Pi’s blog (magpi.cc/iqaudioblog). “When the opportunity arose to acquire IQaudio’s brand and product line late last year, we jumped at it.”

The change means IQaudio products are listed on Raspberry Pi’s products page (magpi.cc/products) and are available from Raspberry Pi resellers, where they maintain the IQaudio brand.


IQaudio was founded in 2015 by Gordon and Sharon Garrity, together with Andrew Rankin. It was “one of the first companies to recognise the

potential of Raspberry Pi as a platform for hi-fi audio,” adds Roger.

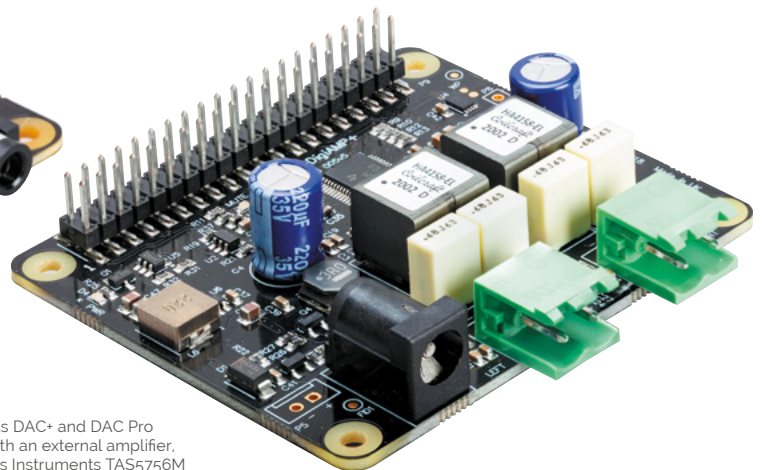
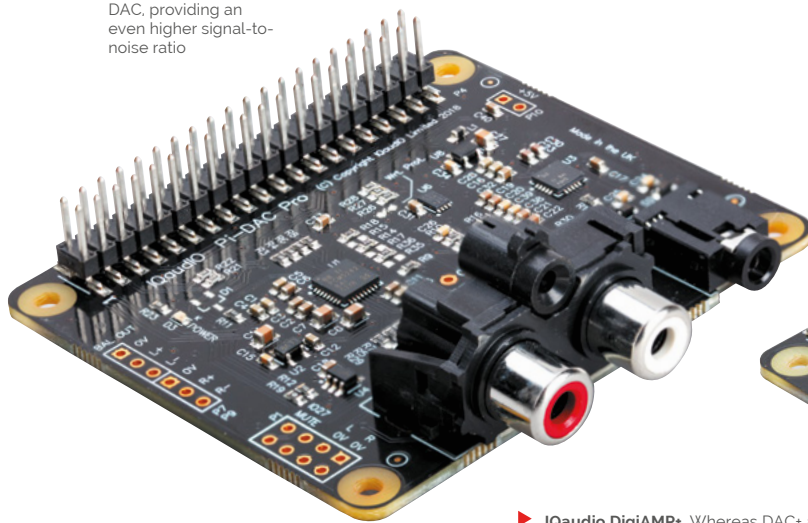
Hi-fi audio is a new market for Raspberry Pi. “We’ve never felt we had the capabilities needed to offer something distinctive,” says Roger, “leaving third parties to step in with a variety of audio I/O devices.”

“ This is the first time we’ve brought third-party products into our line-up ”

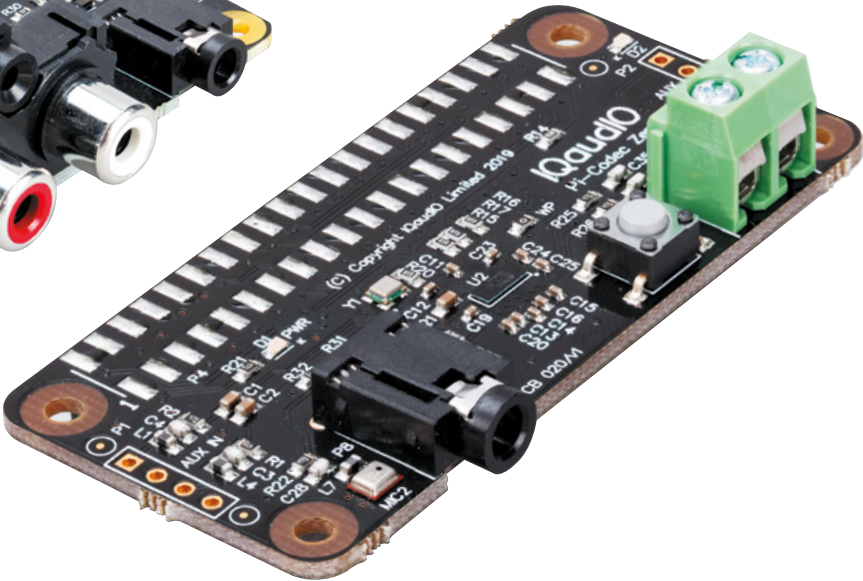
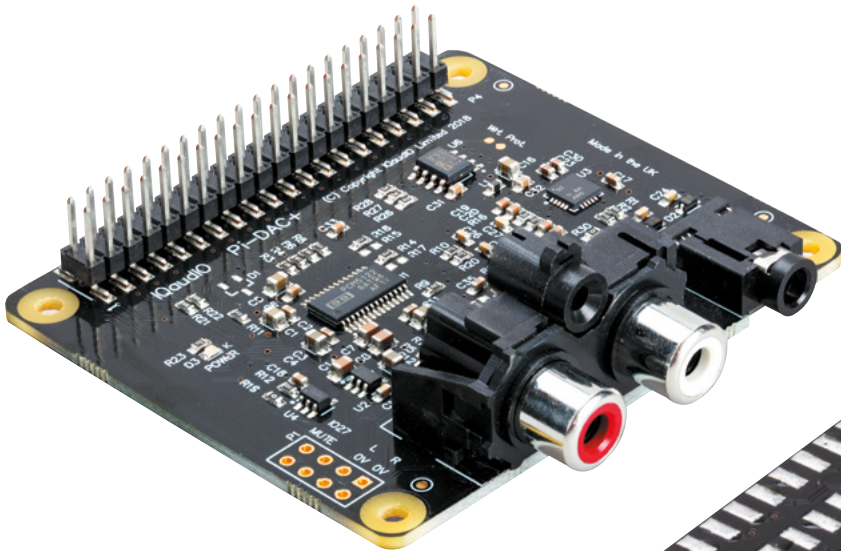
“IQaudio products are widely used by hobbyists and businesses, with in-store audio streaming being a particularly popular use case.”

The four most popular IQaudio products for Raspberry Pi – DAC+, DAC Pro, DigiAMP+, and Codec Zero – are all available to buy via Raspberry Pi Approved Resellers. 

▼ **IQaudio DAC Pro.** Priced at \$25, DAC Pro is IQaudio’s highest-fidelity audio output HAT. It supports the same audio input formats and output connectors as DAC+, but uses a Texas Instruments PCM5242 DAC, providing an even higher signal-to-noise ratio



► **IQaudio DigiAMP+.** Whereas DAC+ and DAC Pro are designed to be used with an external amplifier, DigiAMP+ integrates a Texas Instruments TAS5756M digital-input amplifier directly onto the HAT, allowing you to drive a pair of passive speakers at up to 35W per channel. Combined with a Raspberry Pi board, it’s a complete hi-fi the size of a deck of cards



- ▲ **IQaudio DAC+**. Priced at \$20, DAC+ is the lowest-cost audio output HAT, supporting 24-bit 192kHz high-resolution digital audio. It uses a Texas Instruments PCM5122 DAC to deliver stereo analogue audio to a pair of phono connectors, and also provides a dedicated headphone amplifier
- ▶ **IQaudio Codec Zero**. Codec Zero is a \$20 audio I/O HAT, designed to fit within the Raspberry Pi Zero footprint. It is built around a Dialog Semiconductor DA7212 codec and supports a range of input and output devices, from the built-in MEMS microphone to external mono electret microphones and 1.2W, 8 ohm mono speakers

REVOLUTIONIZE EDUCATION

with Raspberry Pi & PiShop!

Desktop Computer | Learning to Code | 3D Printing | Robotics

We ship into Africa
Visit our website to
find out more

PISHOP
www.pishop.co.za





We Still Fax

▲ The fax machine in the performance is more than just a fax machine

A company of global artists put on a remote performance with some upgraded retro tech, thanks to a Raspberry Pi. **Rob Zwetsloot** watches the show



MAKER
ANTS
 An experimental, new-age theatre company formed of six freelance artists from across the world.
antstheatre.com

People have been finding new ways to stay entertained in 2020 – there’s even been a resurgence in drive-in movie theatres.

The ANTS Theatre company also decided to try something new.

“Imagine immersive theatre crossed with an escape room, but in your home – and that’s We Still Fax” Paul Hernes Barnes of the ANTS explains. “We wanted to make a real, live theatre show that was offline and tactile. We Still Fax is our solution; it’s a whole new form of theatre.”

People taking part receive a special fax machine in the post – hence the name of the show – which they interact with as it ‘comes alive’ during the performance. It uses sound, light, touch, smell, smoke, and faxes throughout.

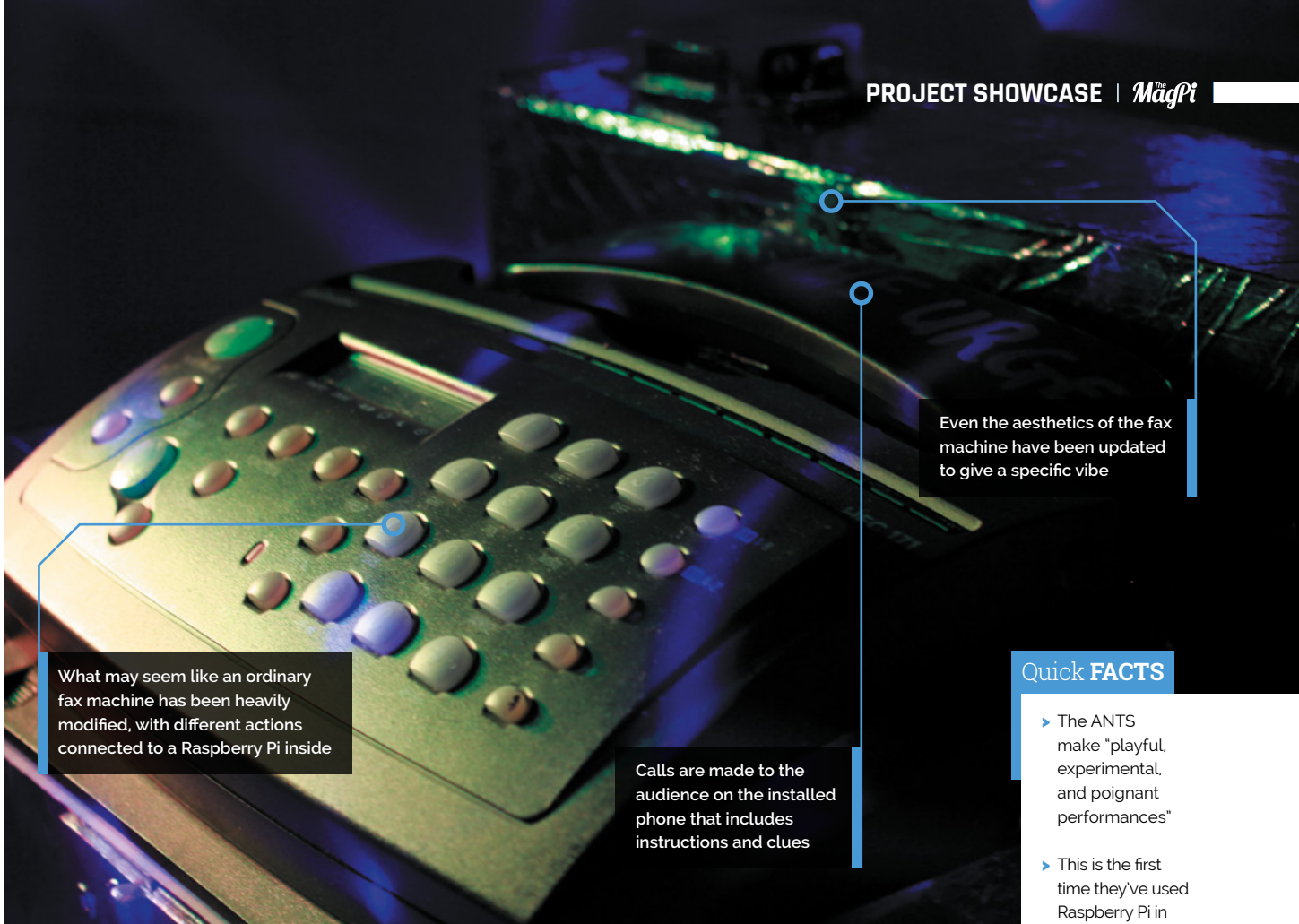
As the ANTS describe it: “You receive a mysterious machine in the post. You plug it in and something strange happens... You connect with

an alternate dimension; one in which the internet doesn’t exist and someone needs your help. To take on this incredibly important mission, you will need to crack codes, send faxes, unlock secret hatches and, when the time comes, push the big, red button. They are counting on you; their world depends on it.”

Indistinguishable from magic

Unfortunately, we have to break the illusion of the show by revealing that this interdimensional device is in fact a modified fax machine that uses a Raspberry Pi, among other things.

“The core components of the show are the fax machine, Grandstream, and Raspberry Pi,” the ANTS tell us. “In short, the Grandstream is an ATA (analogue telephone adapter) which translates phone signal into Ethernet signal and vice versa.”



What may seem like an ordinary fax machine has been heavily modified, with different actions connected to a Raspberry Pi inside

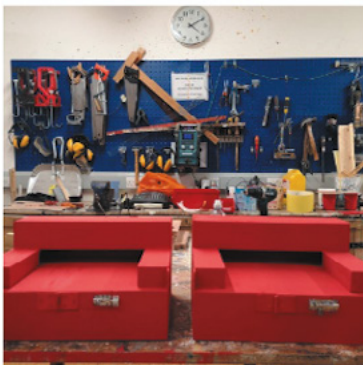
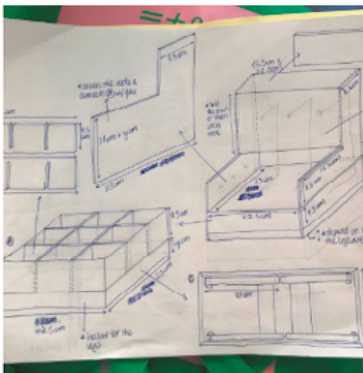
Calls are made to the audience on the installed phone that includes instructions and clues

Even the aesthetics of the fax machine have been updated to give a specific vibe

Quick FACTS

- ▶ The ANTS make "playful, experimental, and poignant performances"
- ▶ This is the first time they've used Raspberry Pi in a performance
- ▶ Every new idea they have seems to involve Raspberry Pi now
- ▶ The ANTS have been working on this since April
- ▶ The ANTS want to bring 'faxy' back

“ You receive a mysterious machine in the post. You plug it in and something strange happens ”



◀ Designing and testing has been going on throughout 2020

THE COMPLETE STEP-BY-STEP DIGITAL LEARNING BUNDLE FOR MASTERING THE RASPBERRY PI 4



POWER LEARNING
ACADEMY
www.powerlearningacademy.com

- ▶ Save hours and hours of searching the internet for tutorials
- ▶ Learn Raspberry Pi in a structured, step-by-step, how-to guide
- ▶ Start with the basics
- ▶ Progress to building complex projects
- ▶ Learn Raspberry Pi, programming and electronics
- ▶ Get help and support when YOU need it



Learn to build projects like an Amazon Alexa Clone, Motion Sensor Alarm, work with cameras and build projects that control any device in your home with your voice!

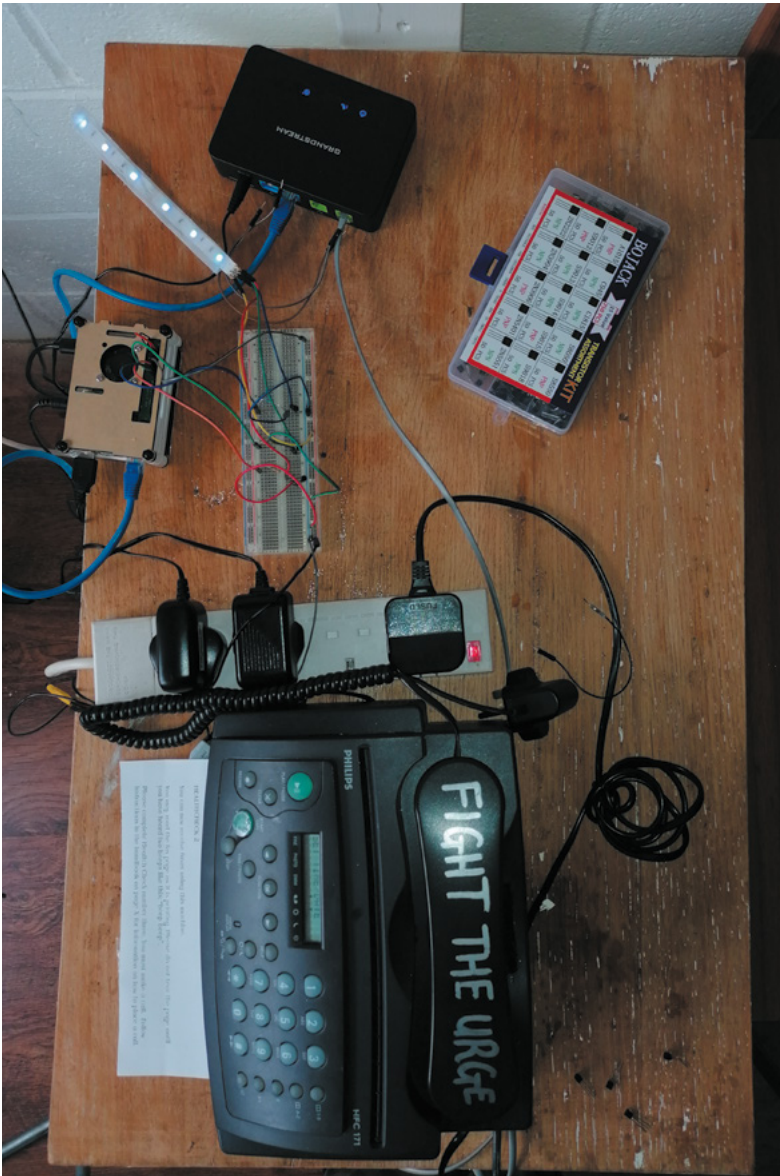
WHAT YOU GET IN THIS BUNDLE:

- ▶ 3 Comprehensive Online Video Courses
 - 21 + hours of video content
 - Ultimate Guide to Raspberry Pi: Tips, Tricks and Hacks*
 - Building Alexa Skills for Home Automation with Raspberry Pi*
 - Learning Python 3 Programming for the Absolute Beginner*
- ▶ Comprehensive starter eBook - Raspberry Pi Bootcamp
- ▶ High quality PDF downloads and resources for all projects
- ▶ Working source code for all projects with code walkthroughs
- ▶ Lifetime access, support and updates to all materials
- ▶ Get personal access to an instructor for help when you need it!
- ▶ Catered to all levels - Beginner to Advanced
- ▶ No previous experience required. All concepts taught from scratch

The most comprehensive way to learn the Raspberry Pi, build hands-on projects, learn programming and electronics!

\$1,000 value, now over 90% off, ONLY \$99 for a limited time!

GET IT TODAY FOR \$99 - LIMITED TIME OFFER
[POWERLEARNINGSTORE.COM/PI-BUNDLE](https://www.powerlearningstore.com/pi-bundle)



Audience members use the machine to make phone calls and send faxes, which are interpreted by Raspberry Pi to activate effects.

“Apart from these three, we have an LED strip which is controlled through GPIOs,” the team continue. “From these we also control the MicroFogger 2: a micro smoke machine. Finally, sound comes through speakers which are, again, connected to Raspberry Pi.”

When it comes to software, all the distribution and management of calls, sounds, lights and smoke is done in a Python script that’s constantly running in the background. “We use Asterisk, an open-source communication software, to interpret calls. Asterisk provides an in-built database, which we use to communicate between Asterisk and Python.”

▲ A sneak peek at all the different bits and pieces that will be installed into the final kit

Magic fax machine



01 As part of the show, Raspberry Pi receives a fax / phone call sent from the audience and adds the appropriate information into the database.

```

1 import os, time
2 x = "Welcome to The Mirror."
3 y = 0
4
5 while y <= len(x):
6     os.system("clear")
7     print(x[:y])
8     time.sleep(0.2)
9     y = y+1
10 time.sleep(2)
11 x = "You will stay here for as long as you can, you may
12 forfeit when you would like to."
13 y = 0
14 while y <= len(x):
15     os.system("clear")
16     print(x[:y])
17     time.sleep(0.2)
18     y = y+1
19 time.sleep(2)

```

02 The Python manager detects a change in the database and updates the current step so that the rest of the show knows what to do.



03 The Python manager launches calls, sounds, lights, smoke of the current step. These will lead the audience onto the next phone call / fax they have to make.



- ▶ The escape room inspiration is a great way to make this performance work at home
- ▶ Even test performances didn't quite go to plan, so testing was essential



Performances have been going on, with varying success. “The technical components of We Still Fax are complex and there have been a fair few issues to overcome!” the ANTS reveal. “While incredibly well-received, the initial research and development performances were patchy in terms of reliability – from the ‘perfect’ performance to one in which we had to abandon the machine for an internet version, there were a lot of learnings!”

Learning process

These sharings were crucial trials that enabled them to understand how different users would play differently – and what that would mean for the programming. “Operating the tech remotely was a significant challenge and we have developed a ‘rescue’ button that will both reboot the machine and re-send us access to Raspberry Pi via email.”

They also uncovered a flaw in the overall box design: “During one show, a plug located inside the casing of the fax machine fell out! We have now refined the code, design, aesthetic, timing and theatrics, as well as planted Easter eggs throughout the player’s journey! The content, design, and code are now in a good place to begin our first string of commercial performances – we can’t wait!” [W](#)

pi-top [4]

ROBOTICS KIT

Robotics & rapid prototyping with your Raspberry Pi

Power your projects with computer vision and applied AI

pi-top [4] Robotics Kit comes with electronic components such as a wide-angle camera, servos and motors, all of which plug and play with the pi-top [4] Complete or pi-top [4] DIY Edition†.

pi-top Robotics Kit with Expansion Plate
187.90 / \$199.90



Gesture Control



Obstacle Avoidance



Autonomous Driving



Object Recognition



Emotion Mapping



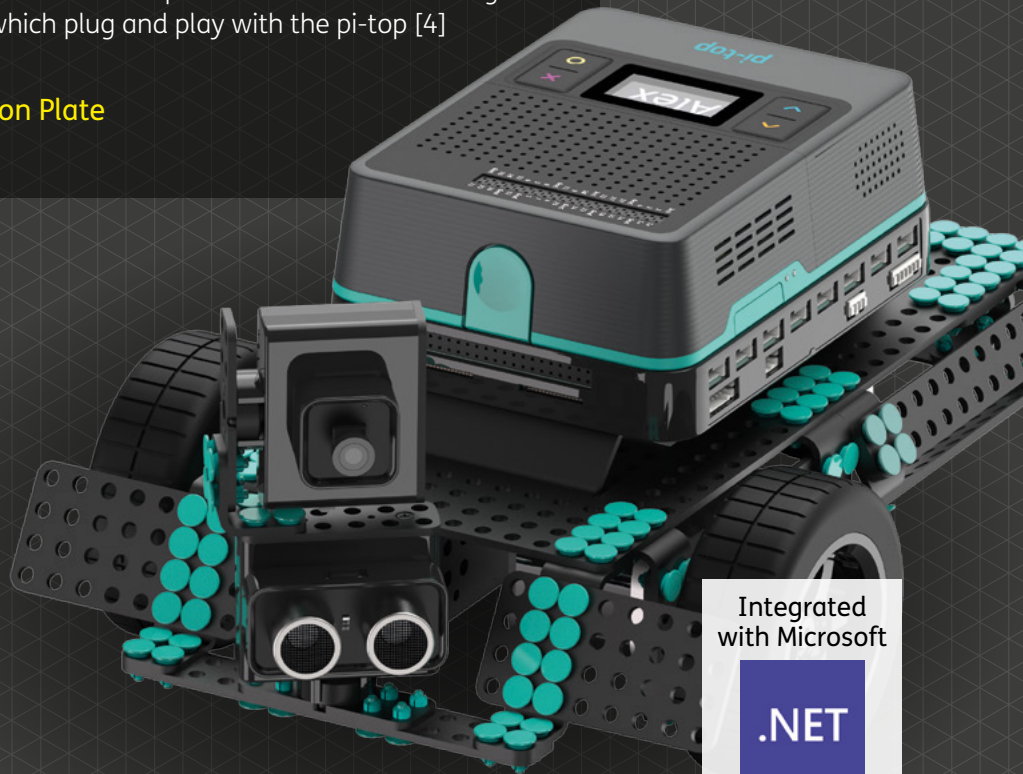
Line Recognition



Interaction



Face Tracking



Integrated
with Microsoft

.NET

pi-top.com/MagPi

Raspberry Pi is a trademark of the Raspberry Pi Foundation. †pi-top [4] and Robotics Kit with Expansion Plate sold separately.

© CEED Ltd. 2021

pi-top

Raspberry Pi made simple, robust and modular.

Cheap All Sky Camera



Jippo

MAKER

Jippo lives in Finland and as well as being interested in computing, programming, and Linux, he's a keen amateur astronomer and photographer.

magpi.cc/allskycam

▼ Meteotux Pi is easy to set up: install the software on a Raspberry Pi and run it with simple parameters. The parameters can be checked from the control line by typing `./meteotux_pi`

Keen astronomers can keep an eye on the night sky and capture meteors and satellites, as **David Crookes** explains

Meteors can be seen across the sky when it's not cloudy, but you need to be away from bright lights and looking in the right place at the right time. This is easier said than done, particularly if you also want to capture the precise moment on camera. What you need is a device that can scan a wide field of sky, and snap away whenever it picks up something bright.

The Cheap All Sky Camera certainly fits the bill. Created by Jippo – whose nickname is a reference to a late 1970s and early 1980s Finnish magazine that encouraged readers to build – it uses a Raspberry Pi 4, as well as the High Quality Camera fitted with a 180-degree CS lens. These components sit under a 20 cm acrylic dome, protecting them from the weather.

Stars in their eyes

“Before this there was a similar project that used Linux, a normal computer, a video capture card and a security camera containing a charged-coupled device,” says Jippo. “But with Raspberry Pi you

can build a really cheap and small setup with everything that's needed placed inside a dome.”

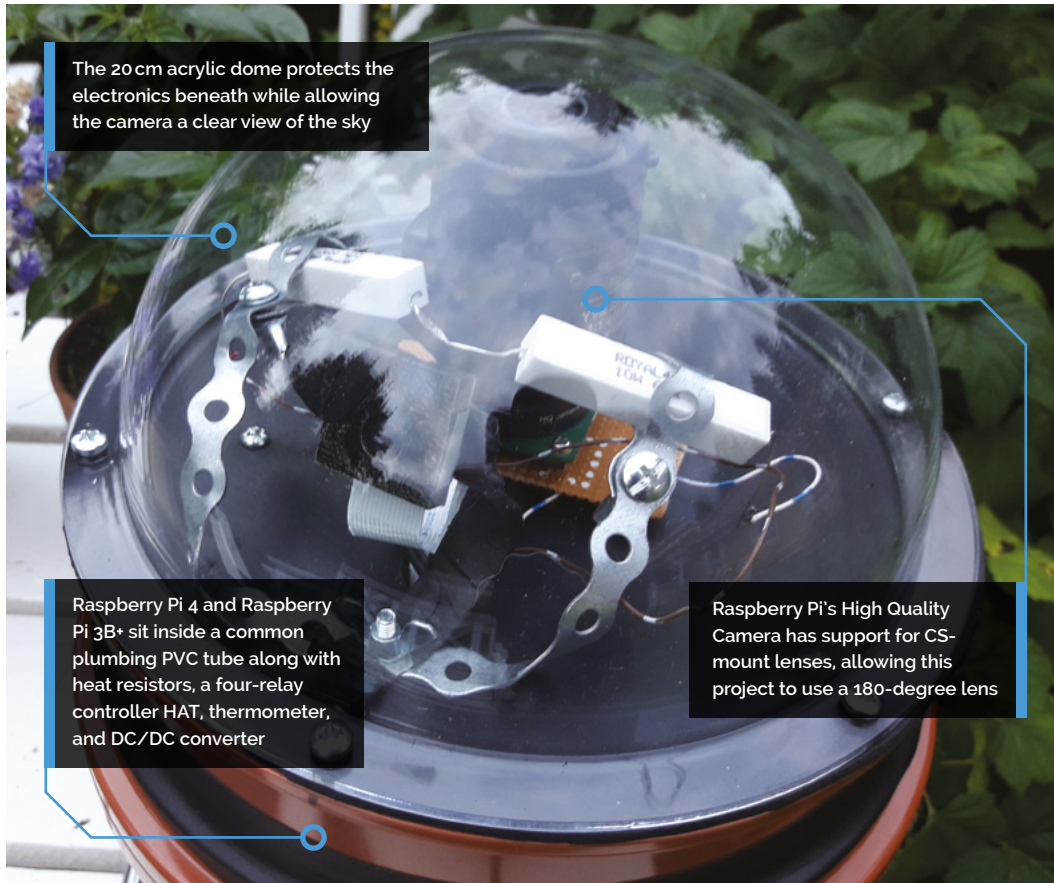
The build was inspired by the Meteotux Pi program written by Jippo's friend Jani, developed to record continuous high-resolution images during the night. “Jani and I had joined an amateur astronomy association here in Finland called URSA, and there was a group using Windows computers to capture sky events in order to calculate and identify meteors and meteor showers,” says Jippo.

“With Raspberry Pi you can build a really cheap and small setup”

“We were both using a Linux computer and we had the idea of developing a Linux program to capture those sky events. Meteotux Pi was developed after Raspberry Pi came with a hi-res [HQ Camera].” This keeps costs to a minimum, especially since Meteotux Pi will also work with older Raspberry Pi models and Camera Modules.



▲ Rather than record video, Meteotux Pi saves a series of long-exposure photos that are stacked in one image so that no frames are lost all night



The 20 cm acrylic dome protects the electronics beneath while allowing the camera a clear view of the sky

Raspberry Pi 4 and Raspberry Pi 3B+ sit inside a common plumbing PVC tube along with heat resistors, a four-relay controller HAT, thermometer, and DC/DC converter

Raspberry Pi's High Quality Camera has support for CS-mount lenses, allowing this project to use a 180-degree lens

Quick FACTS

- ▶ The project cost just \$150 to build
- ▶ It includes a 180-degree lens
- ▶ Two Raspberry Pi computers are used
- ▶ The project runs Meteotux Pi: **meteotuxpi.com**
- ▶ It takes a series of long-exposure shots

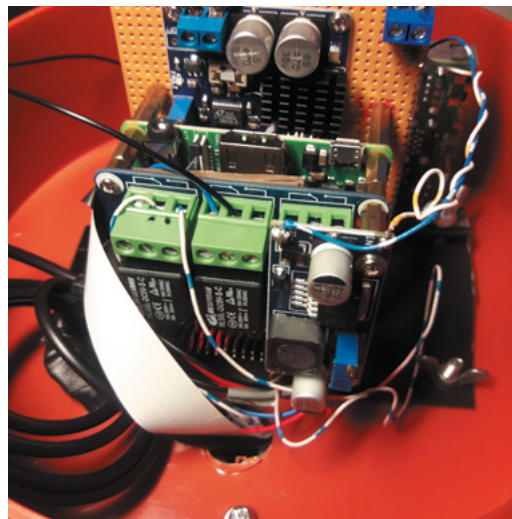
A Raspberry Pi computer runs Raspberry Pi OS Lite and it's automated to run Meteotux Pi at sunset, stopping at sunrise. "A simple Shell script runs daily and checks for these times before generating command-line options for the program and running them," Jippo explains.


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

Just warming up

To prevent the device from getting too cold, given it is going to be located outside all night, Jippo has used a second Raspberry Pi computer – a Raspberry Pi 3+ with a V2 colour camera, relay controller HAT, and temperature sensor.

"The sensor monitors the temperature inside the dome and if it gets too cold, the relay board will turn the 10W heat resistors on. This keeps the dome clear from humidity or frost and it removes raindrops a lot faster too," he reveals.



The result has been many hours of pleasure as Jippo and Jani discover the countless delights of the sky. Some of the results can be viewed on YouTube at magpi.cc/meteotuxyt. "I'm really happy with the results so far," says Jippo. 

- ◀ Jippo says the project can be compared with commercial all-sky cameras costing as much as \$10,000



Warning! Mains Electricity

This project makes use of mains power, so be extremely careful and read this advice:

magpi.cc/electricalsafety

Turing Machine Demonstrator Mark 2 (TMD-2)



Michael Gardi

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

magpi.cc/tmd2

The computations are shown on screen, with the 'tape' at the top moving left and right to read and write zeroes and ones, as determined by the table of instructions

A modern-day Turing machine that's simple to use and understand. **Nicola King** marvels at this mathematical model of computation

As many will know, Alan Turing was a pioneering mathematician who created the renowned Turing machine in 1936, and is considered by many to be the father of modern computer science. Essentially, a Turing machine mathematically models a mechanical machine that operates on a tape. Over the years, others have developed their own iterations of this iconic machine.

Ontario-based Michael Gardi is one such maker who has now created two versions of his Turing Machine Demonstrator (TMD). Acknowledging that there are some other great implementations out there, he wanted to maintain a focus on the real purpose of a Turing machine. "In my humble opinion, the complexity of these excellent and imaginative solutions often detracted from the understanding of what a Turing machine actually does," he tells us. "For TMD-1, my goal was to

demonstrate the idea of a Turing machine with as much clarity as possible. I wanted to build a machine that was simple to program and easy to understand. I was really happy with the way that TMD-1 turned out. I believe it met the stated goals of 'simple to program' and 'easy to understand'. To help accomplish those goals, the machine itself was limited to three states / three symbols, and a small ten-cell bounded tape."

The next level

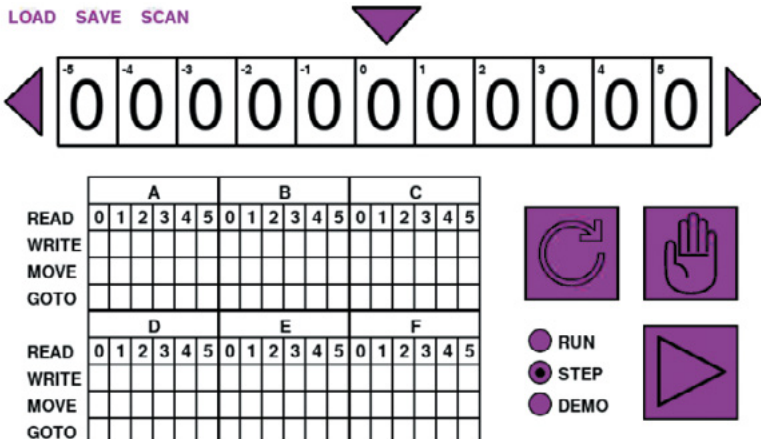
With the first model under his belt, Michael then decided to create a version with more potential depth for the Turing machine enthusiast, and TMD-2 was born. "For TMD-2 I wanted to 'up the ante'," he says. "My goal was to make a six-state / six-symbol machine with a large 100,000 cell tape. As much as possible, I tried to bring forward the simple-to-use, easy-to-understand principles from TMD-1."

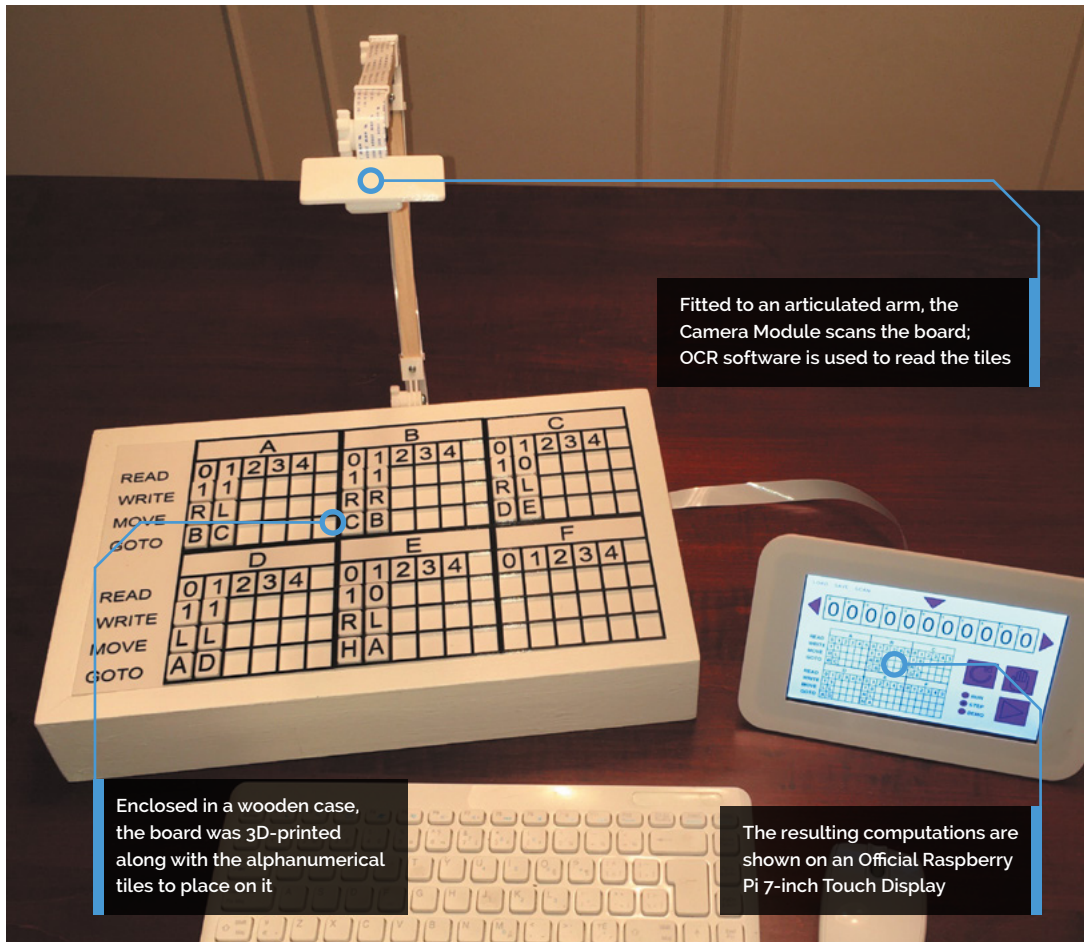
His TMD-2 makes use of a Raspberry Pi 3, an Official 7-inch Touch Display for the user interface, and a Camera Module mounted on an articulated arm above a 'State Transition Table' box. The latter can hold one of a selection of table cards 3D-printed by Michael, along with a set of alphanumerical tiles to place on them. The camera scans the current state of the machine, which is read using the Tesseract OCR (optical character recognition) library. The resulting computations are then shown on the Touch Display.

Python program

"At its heart, TMD-2 is a standalone program written in Python," says Michael. "If you just want to try the application, it will run on any computer

LOAD SAVE SCAN



**Quick FACTS**

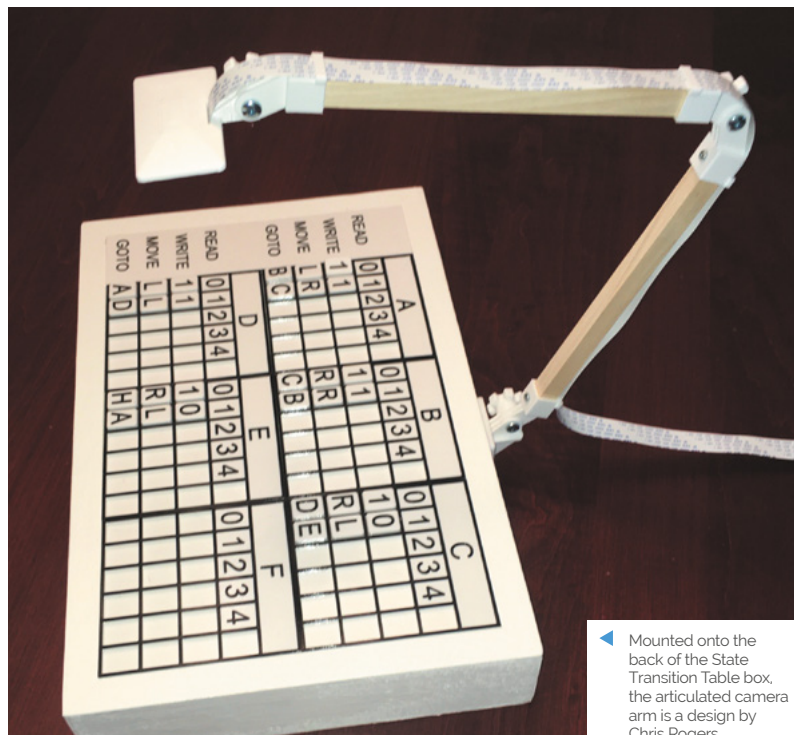
- ▶ The project involved around 50–60 hours of 3D printing..
- ▶ ..which mainly took place overnight
- ▶ TMD-2 took Michael around two months to develop
- ▶ It was his first Raspberry Pi-related development project
- ▶ Find the source code at magpi.cc/tmd2code

“ For TMD-2, I wanted to ‘up the ante’ ”

that supports Python (which is most machines). Running it on a Raspberry Pi is extremely easy, since both Python and the Pygame library it relies on are already part of the Raspbian [now Raspberry Pi OS] distribution.”

While Michael has had some great feedback on TMD-2 following his posts on Hackaday, Hackster, and Instructables, he says, “Unfortunately, with the Covid-19 restrictions here I have not been able to show these projects to friends at my local makerspace (kwartzlab.ca) which is where I would normally get the best feedback (both good and bad!). My son and daughter-in-law worked through the Quick Start Guide for TMD-2 and ‘programmed’ some of the challenge exercises. They are both avid gamers and said that it was a lot of fun, ‘like a game’.”

Michael’s TMD-2 is a truly fascinating make based on a seminal invention that, arguably, laid a solid foundation for development of the computers that we use today. What could be more inspiring?





Kris Vanherle

MAKER

Kris is the technical coordinator of the WeCount project. He works on the development of the technology used for the website, analytics, and sensors.



Enda Hayes

MAKER

Enda is a professor of Air Quality and Carbon Management at the University of the West of England. He leads the Cardiff project, and has funding for 150 sensors in total.



Dr Péter I Pápics

MAKER

Peter is a data scientist working for Transport and Mobility Leuven. He wrote large parts of the scripts which run on the Raspberry Pi computers, and API and server code.

WeCount Traffic Sensors

Citizen scientists are helping researchers to monitor traffic data in six European cities and it's thanks to Raspberry Pi, as **David Crookes** discovers

It's important for planners to understand how well a city's streets are being used. They need to know how many vehicles are being driven past a particular point, what type are they, and how fast are they going in order to learn more about traffic flow.

Rather than have someone stand on the pavement and count manually, however, planners use technology. Some expensive solutions are limited to a narrow set of locations such as highways but, in the case of the WeCount project, an affordable solution has been found with the help of Raspberry Pi.

WeCount invites members of the public to place a sensor in their homes, peering out of the window, allowing live traffic maps of neighbourhoods to be built. Engineering and emissions expert Kris Vanherle says involving citizens results in greater support for future traffic management implementation. "Data collection also becomes more affordable for local authorities and citizens can even help interpret the data," he adds.

In one instance, a spike in traffic was due to a temporary road block and this was reported by one of the citizens involved. WeCount is now being used in six European cities – Madrid, Barcelona, Dublin, Cardiff, Ljubljana, and Leuven – and the data collected is uploaded to the cloud so that it can be used within initiatives related to air pollution, safety, active travel, noise, and speed.

Creating the sensors

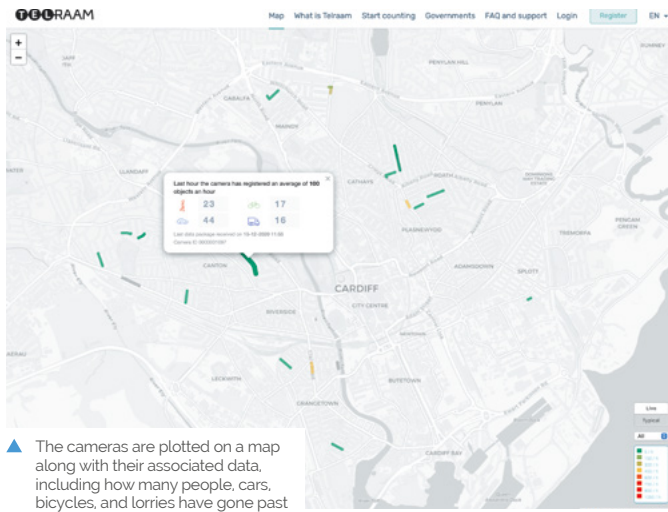
The sensors were originally built using Raspberry Pi 3B+ computers, but the production units make use of Raspberry Pi 3A+. "These already provide the required processing power and memory and we found we did not need the extra ports of its big brother after the initial development phase," explains Dr Péter I Pápics, a researcher at Transport & Mobility Leuven.

He says the project uses standard Python libraries combined with OpenCV for the image processing and object tracking algorithms. "We need to maintain the necessary frame rate, which is around 30 fps, to capture even the fastest cars over at least a few frames," he continues.

"So we are using the most simple background extraction and contour detection methods to find moving objects on each frame and then a tracking algorithm to identify and track them over their visibility period stretching across a set of consecutive frames."



▲ Each sensor involves the fitting of a Raspberry Pi Camera Module V2



A Raspberry Pi operates as a wireless hotspot and it provides a simple user interface so that camera angles can be fine-tuned. “After ten minutes, the hotspot mode is automatically disabled and Raspberry Pi attempts to connect to the local WiFi,” says Péter.

“If successful, the actual traffic monitoring script is activated and various properties of the detected objects are periodically transmitted to our services where data processing, classification, and aggregation services produce the data that’s available on our website or via the public API.”

Amassing the data

The cities involved are enthusiastic. Prof Enda Hayes, from the University of the West of England in Bristol, says he was keen to include Cardiff because it has immeasurable potential for more active travel. “The city’s inhabitants are very car-dependent and the city experiences a large influx of daily commuters, with the majority of these journeys – some 80 percent – by car,” he says.

“ We are using the most simple background extraction and contour detection methods to find moving objects ”

The sensor also allows for the counting of pedestrians and cyclists. “I delivered one to a volunteer who lived on a dead-end street and found out pedestrians and cyclists were using the street as an active travel rat-run to get to a local train station. He wanted to convert the street into a Low Traffic Neighbourhood.”

Kris says the data also allows researchers to observe how traffic behaves if lanes are closed on adjacent streets and it’s been used to increase compliance to speed limits. “WeCount empowers citizens and anyone who wants to join can do because the platform is open,” he says. **M**



Quick FACTS

- ▶ Researchers want 150 to 250 sensors in each city
- ▶ They are working with local authorities
- ▶ The exact location of a sensor is not revealed
- ▶ The project will run until December 2021
- ▶ The data can be accessed at **telraam.net**



- ◀ Kris Vanherle and colleague Giovanni Maccani work on the production of a sensor device which is housed inside an expensive casing made by TEKO

Jazz Champion

This unusual upcycling project transforms a vintage chess machine into a musical game. **Phil King** thinks, "Jazz... nice!"



Leo Neumann

MAKER

Leo is a computer scientist and audiovisual artist. His work investigates the connections between sound, light, science, and technology. Essentially, he likes to bang on things he encounters around him.

tonlicht.studio

If someone were to give you an old chess machine that didn't work, what would you do with it? Turning it into a musical game

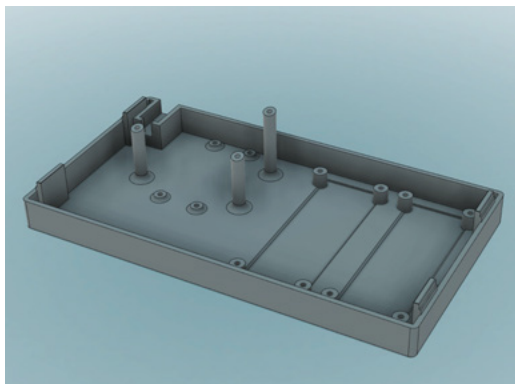
involving chord progressions probably doesn't come to mind, but that's just what jazz fan Leo Neumann has made from a 1978 Chess Champion MK1.

"The inside was completely missing with the exception of a beautiful keypad," he recalls. While he wasn't sure what to do with it at first, and it gathered dust for a couple of years, a conversation with a friend about converting trash objects into musical instruments sparked an idea. "It suddenly clicked: I would turn the chess computer into a jazz computer!"

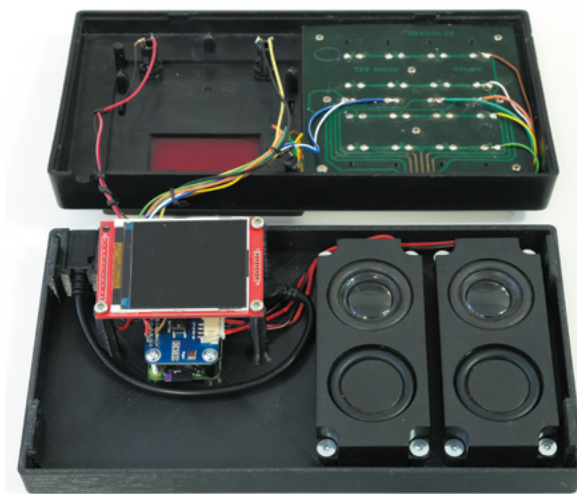
Leo noted a striking resemblance between the notation of chords and chess moves. "For example, E6 E7 means in chess that a pawn moves from square E6 to E7. In jazz it would describe E major chords extended by a sixth or seventh note," he says. "Also, I much liked the idea that playing a chess game is akin to playing a duet in a call-and-response type of scenario."

A little knight music

Leo says the musical game he created is very simple to play, involving pressing the machine's buttons to choose chords. "You enter a jazz chord, wait two seconds while the computer comes up



▶ It took a couple of attempts to get the 3D-printed base design right and align the LCD under the tinted window



▲ The inner workings include a Raspberry Pi Zero with WM8g60 Audio HAT, speakers, and 176x220 LCD

with a matching chord, and then it is your turn again. One can choose chords of any key and it can be major or minor, diminished or augmented, and possibly extended by 7th, 9th, 11th, or 13th notes."

The computer will then seek to come up with a chord that sounds as "cool as possible when played after the given user chord," he reveals. "It does so by generating 1000 random candidate chords and using a metric to assess how well they fit. The best chord is taken and the computer score is increased by the value according to the metric."

Once the user has entered their chord, the same metric is used to add points to their score. "The game has no end – one can play as long as desired, since the main point is not to 'win' by scoring more points, but to just have fun playing jazz music."

Striking the right chord

It took Leo around two months to develop the project, including replacing the chess machine's original seven-segment display with a 176x220



The machine's original seven-segment display was replaced with an LCD to show chord names and scores

Leo designed and 3D-printed a new base to house all the electronics, including a Raspberry Pi Zero

Press the button for the musical key, then add extra notes (such as a seventh) if desired

Quick FACTS

- ▶ Leo designed a new 1970s-style font for the display
- ▶ The chess machine didn't work and its base was missing
- ▶ Leo's software uses the Pyo library for sound generation
- ▶ It was developed on a PC, but runs fine on Raspberry Pi Zero
- ▶ His next musical project is called Transit: magpi.cc/transit

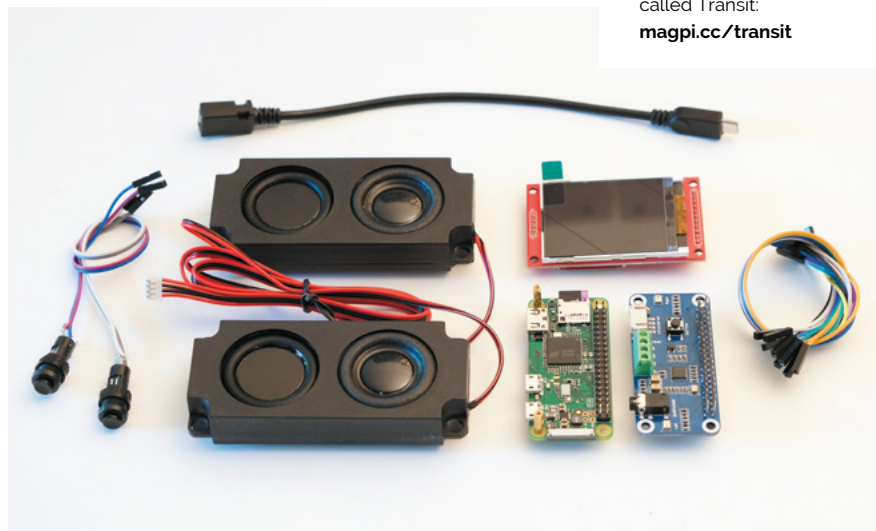
LCD to show longer chord names, and 3D-printing a new base to house a Raspberry Pi Zero, WM8960 Audio HAT, and speakers.

By the far the hardest part of creating the Python software was to design an algorithm that rates how well a new jazz chord follows a given one. After trying several musical theory approaches and finding that they didn't come up with the desired "jazzy feel", he opted to play all the common chord combinations and rate them manually – a process that took three days and was a "numbing experience."

"The ratings form the basis of the metric and I integrated some additional heuristics such that chords do not sound too dissonant, too consonant, too repetitive, nor too simple," he explains. "I just underestimated how difficult it is to translate human artistic preference into computer code and learned a lot here."

As well as being a fun game to play, Leo thinks his Jazz Champion could double as a useful aid for composing jazz music, by generating chord sequences. "It is actually easy: one can just repeatedly input the last computer-generated chord to get an entire sequence!"

“ He opted to play all the common chord combinations and rate them manually ”



▲ All the components required to convert a chess machine into a jazz music game

Hubble Pi



Santiago Rodriguez

A seminar at Berlin Humboldt University introduced 21-year-old Spaniard Santiago to coding in Python for numerical physics, and kick-started his first ever Raspberry Pi project.

magpi.cc/hubblepigitt

A 'Eureka!' moment at university led one student to retrofit his telescope with an HQ Camera sensor and aim it at the stars. **Rosie Hattersley** learns more

When physics student **Santiago Rodriguez** learned about the possibilities of Python for numerical physics, he immediately recalled a conversation with a friend about a new product from Raspberry Pi, the **High Quality Camera**.

This camera would work with Santiago's beloved, but rather small, Maksutov telescope. Having been a huge fan since childhood of all things space-related, and with plans to study astrophysics and masters and PhD level, he suddenly saw how coding might just fit in... and the idea for Hubble Pi was born.

Santiago's goal was to use a Raspberry Pi 4 and HQ Camera to take pictures with his telescope of planets, stars, and maybe even DSOs (deep-sky objects). He also decided to load his Raspberry Pi with established astronomy programs such as KStars, which can display a live map of the night sky directly on the telescope using an attached display.

Magic mirror lens

He realised that the HQ Camera's "bigger, exposed sensor would allow it to essentially use the telescope as a big mirror lens. It was quite cheap for the quality it could offer." For controlling the sensor for astrophotography, the flexible, open-source nature of Raspberry Pi appealed too.

“ He began by learning what the sensor could do from the HQ Camera documentation ”

To achieve this, Santiago developed AstroCam, a Python GUI for controlling the camera for astrophotography. "Using it, I can control ISO, shutter speed, and exposure time (with some limitations due to the digital nature of the rolling shutter)," he explains.

AstroCam can also take multiple exposures automatically, and capture RAW image files – both important capabilities for astrophotography.



◀ Astronomy programs such as KStars can be installed on Raspberry Pi to provide a map of the night sky



Hubble Pi features a Raspberry Pi 4 and a touchscreen from which image capture can be triggered. The setup can also be accessed using a remote desktop connection

The photos taken of the night sky with Raspberry Pi's HQ Camera are comparable to those captured using a DSLR camera with the telescope rig

The HQ Camera and the telescope's optics work really well together, with a readily available C-mount-to-telescope adapter connecting them seamlessly

Quick FACTS

- ▶ Santiago set up his astro camera pointing out the window
- ▶ His friends thought he was spying on his neighbours
- ▶ Pointing the telescope at the sun without covers was another howler
- ▶ But he does recommend sharing your live photos with friends
- ▶ And even recommends it as a conversation starter



▲ The Hubble Pi setup captured fantastic close-ups of the moon, including this one showing Mare Crisium

▼ Hubble Pi, named after the famous astronomer Edwin Hubble, was able to photograph distant planets, including Jupiter



“ Hubble Pi can also work wirelessly or as a standalone if needs be ”

Controls are triggered locally via a touchscreen, or through a remote desktop connection from a laptop connected via Ethernet or a wireless LAN hotspot hosted by Raspberry Pi. He was reasonably confident the project would work, having seen similar endeavours based around Raspberry Pi 3B.

Santiago had never used Python before, so he began by learning what the sensor could do from the HQ Camera documentation, after which the main challenge was coding the GUI program with Tkinter and adapting it for his astrophotography needs. “I based my AstroCam script’s main loop and the image preview code on that of a Python programmer who had developed it for a normal camera use,” he says. The source is credited in the documentation in Santiago’s GitHub repository: magpi.cc/hubblepigitt.

Although others had posted similar projects on GitHub, no one had written about using the C-mount adapter for the HQ Camera for astrophotography at the time. Fortunately,



C-mount-to-telescope adapters are fairly common, and Raspberry Pi itself could simply be mounted using a simple phone adapter.

▲ Another moon shot, showing Crater Tycho and Mare Nubium

Shooting stars

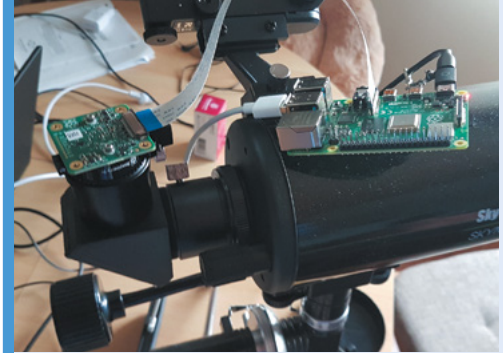
Santiago is reasonably pleased with the results from Hubble Pi. “Most good USB cameras for astrophotography start at about €200 and require a connected computer at all times. Hubble Pi can also work wirelessly or as a standalone if needs be,” he notes.

Raspberry Pi 4 and HQ Camera sensor work very well, but photos are limited by the optical limitations of his telescope, while humidity and light pollution are also factors. “Planets and stars look alright, but somewhat blurry without stacking (due to atmospheric distortion), and DSOs are very difficult to capture due to the f/14 aperture ratio from the small Maksutov (brighter/bigger optics should deliver better results here),” discloses Santiago.

“My friend’s DSLR doesn’t perform much better when attached to my telescope at prime focus,” he says. “The only real limitation from the sensor side of things is its digital rolling shutter, which means it can’t do hour-long exposures like DSLRs.” But an expensive star tracker (to compensate for the drift during the lengthy exposure) costs upwards of €300.

When finances allow, Santiago intends to develop Hubble Pi in this direction. For now, he’s pretty pleased with his first Raspberry Pi project. “This may have been the first, but it certainly won’t be the last!” **M**

Assemble your rig



- 01** You need a Raspberry Pi 3 or 4 with 2GB+, a touchscreen display, and a 32GB microSD card, plus an HQ Camera (sensor only), telescope, and 1.25-inch C-mount adapter. Power from a USB-C or a USB power bank.



- 02** Create a `wpa_supplicant.conf` file and empty `ssh` file in Raspberry Pi’s boot directory (instructions at magpi.cc/headless) so you can access Hubble Pi directly from a computer. For remote connections, use XRDP or a monitor and keyboard-mouse combination.



- 03** Connect your HQ Camera and enable the camera (see magpi.cc/picamera2), then mount the telescope and take some test shots by typing `raspistill -o Desktop/image.jpg` into a Terminal.

RT Jam

Real-time music jams with friends, with a little help from Raspberry Pi, are a great way to keep in touch. **Rosie Hattersley** hears one musician's story



Mike Vargo

California-based engineer and musician Mike previously used Raspberry Pi in a HearSee project to create an ultrasonic sound map.

magpi.cc/rtjam

Engineer and Linux enthusiast **Mike Vargo** was exploring options for how to set up real-time music jams with his friends when fellow musician **Darius Mostowfi** introduced him to Raspberry Pi. He'd already tried out several hardware and software options, but once Mike realised that Raspberry Pi OS was simply a Debian distro, things slotted into place and RT Jam began to take shape.

"I was familiar with Voice over IP (VoIP) from previous experience and decided to try to build something with latency low enough to allow you to play as a group online," says Mike. "When I saw that Raspberry Pi OS was just another Debian distro, I realised my laptop Linux code would compile and run on the Pi."

Low latency

To work properly, RT Jam would need a person-to-person latency of less than 25 milliseconds. "I was leery of a peer-to-peer product because of the many issues with network firewalls, so I decided to opt for a client/server type structure," Mike explains. However, "large variances" in sound, the input-output performance on Linux, Windows, and macOS made a software-only solution "very troublesome", he says. "Raspberry Pi gave me a turnkey solution."

Mike realised his 4GB Raspberry Pi 4 could be also used to run the broadcast server, including NGINX (nginx.com), and applauds the ability to install so many standard packages so easily. "Compiling the code on Raspberry Pi was very straightforward."

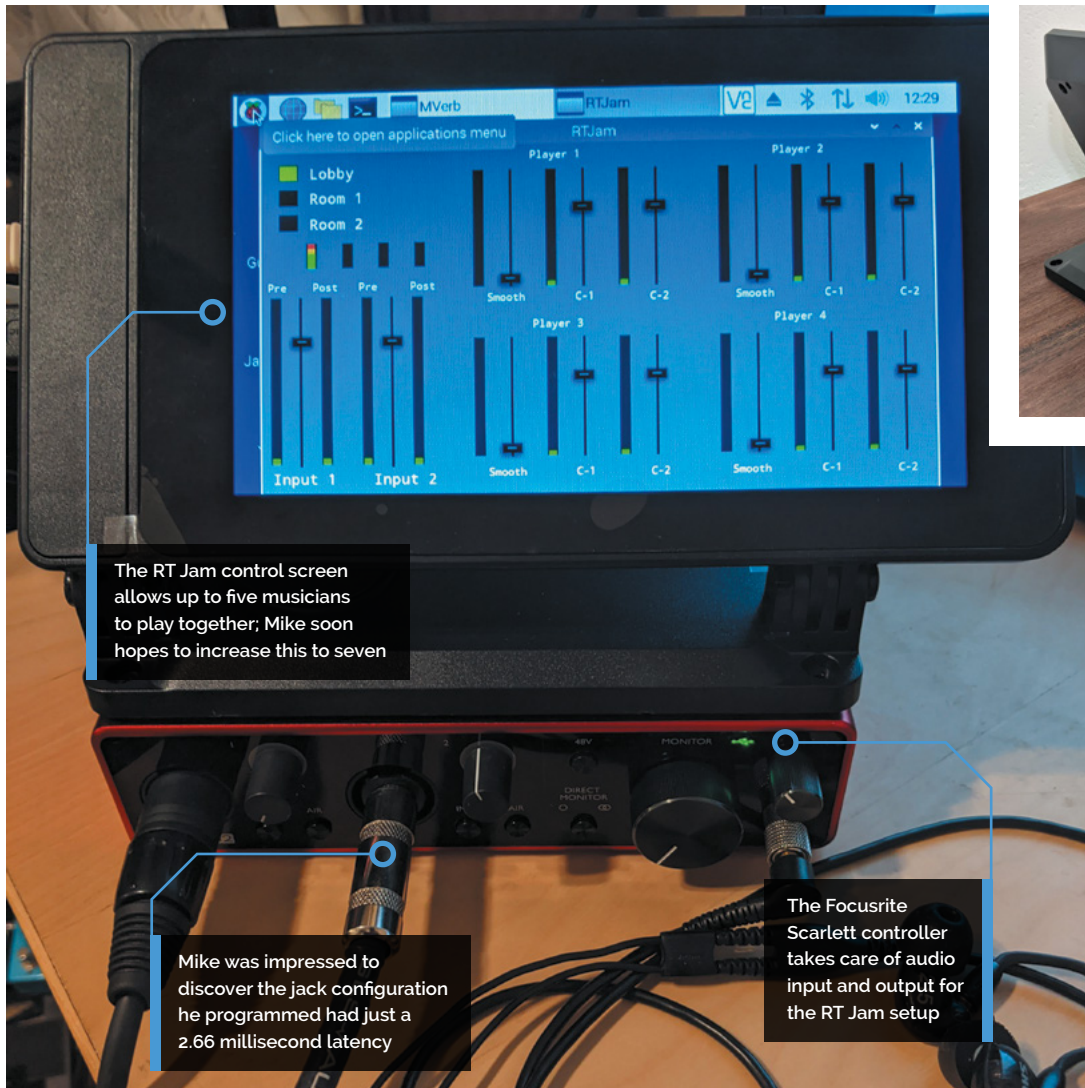
Sound and vision

Before he began working on RT Jam – he describes the "want level" as a silver lining born of necessity – Mike had tried out Raspberry Pi for a putative HearSee project (also on GitHub: magpi.cc/hearsee) in which ultrasonic range sensors on a handheld stick convert distance readings into a sound form that would allow you to 'see' with your ears. That project is on hold, but it acted as an excellent demonstration of Raspberry Pi sensors' capabilities. "Raspberry Pi worked great for hooking up the sensors and creating the sound," he says. However, low-latency audio would work with Raspberry Pi only when using a USB 2.0 device for both input and output. He chose a Focusrite Scarlett controller, which he loves for its aesthetics as well as its performance.

Mike began by using DISTRHO/DPF, a software framework for audio plug-ins, using which he was able to program all the audio I/O via jack. He had help from falkTX (magpi.cc/falktx) who had been working on this framework for some time.



▶ The Scarlett 2i2 mixer provides 24-bit audio support for the musicians involved in an online jam



- ▲ RT Jam is powered by Raspberry Pi 4, while mixing controls are accessed via a touchscreen

Quick FACTS

- ▶ Mike loves being able to jam with his friends
- ▶ Coronavirus restrictions made him consider remote options
- ▶ Although he's an engineer, Mike says he's still terrible at soldering
- ▶ Virtual jamming is "a little different than in person"...
- ▶ But listening to what other people are playing is still key!

“ Once Mike realised that Raspberry Pi OS was simply a Debian distro, things slotted into place and RT Jam began to take shape ”

The design for the VoIP elements was Mike's own. “I have done this kind of stuff before, so it was not too hard,” he says, “but I had not written any C++ code for 20 years, so I was a bit rusty. I'm sure if somebody looks at the code (it's open-sourced), they will find lots of things to laugh at!”

One of the challenges was to get Raspberry Pi's touchscreen working right and to figure out the settings for real-time audio. Once he was happy with everything, Mike simply copied it all as a microSD image.

He's keen to develop RT Jam, adding capacity for more musicians to jam along. “I also want to hook up an Icecast feed from the server, so even people who don't have the software can just listen to a live broadcast of the room feed.”



- ◀ Mike's setup uses the Linux-based Ardour Mixer MIDI app

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 W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero W Kit 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 W Starter Kit

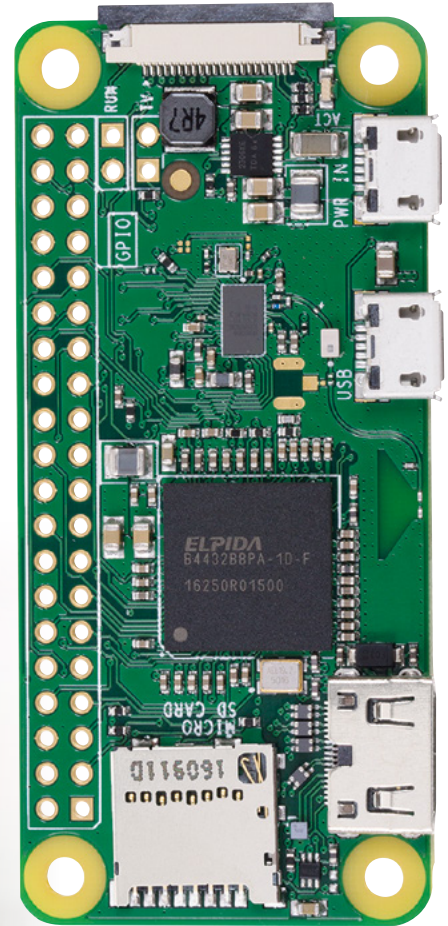
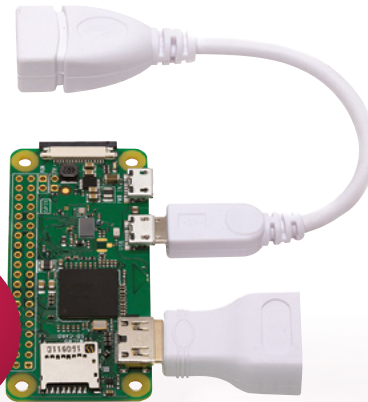
WITH YOUR FIRST 12-MONTH SUBSCRIPTION

Subscribe in print today and you'll receive all this:

- ▶ Raspberry Pi Zero W
- ▶ Raspberry Pi Zero W case with three covers
- ▶ USB and HDMI converter cables
- ▶ Camera Module connector

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

WORTH **£20**



 Buy now: magpi.cc/subscribe

SUBSCRIBE
on app stores

From **£2.29**





BUILD THE ULTIMATE MEDIA CENTRE

Create the ultimate media box using just a Raspberry Pi

Raspberry Pi is capable of many things, and we've built a lot of projects with one. Having several Raspberry Pi computers performing similar functions seemed a little redundant to us, though, which gave us a brainwave – what if we could combine multiple uses into one amazing project?

Thus we decided to see how many media services we could squeeze into just one Raspberry Pi 4. Luckily, the majority of the tools are already available – you just need to know how to combine them all. Get ready for the perfected media experience.



Parts list

Everything you need to put together your ultimate media centre

REQUIRED



▲ **Raspberry Pi 4**
£34 / \$35 | magpi.cc/raspberrypi4

▼ **Argon One M.2**
£42 / \$45 | magpi.cc/argononem2



▼ **Official Raspberry Pi 4 power supply**
£8 / \$10 | magpi.cc/psu4



▲ **Mini Remote Control**
£5 / \$6 | magpi.cc/miniremote



▲ **HDMI cable**

▼ **Game controllers**



▲ **Flic USB**
£20 / \$23 | magpi.cc/flicusb



◀ **16GB microSD card**

OPTIONAL



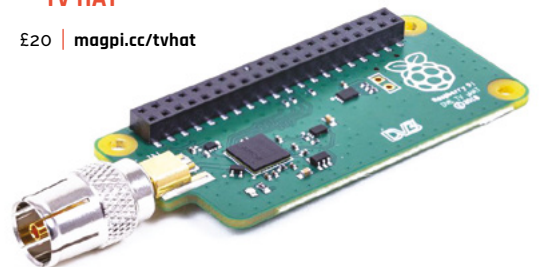
▲ **M.2 storage drive**

▶ **Ethernet cable**



▼ **Raspberry Pi TV HAT**

£20 | magpi.cc/tvhat



◀ **Wireless keyboard**



▶ **Removable USB storage**



Set up your system

Build your media PC and install RetroPie

PUT IT TOGETHER

01 Install RetroPie

We're using RetroPie as it allows you to emulate several game systems, as well as installing Kodi under it for video and music. You'll need to burn your microSD card manually for this – we advise using Raspberry Pi Imager (download it from magpi.cc/imager). It's easy to use: click 'Choose OS', select RetroPie, then choose the 'RPI 4/400' version for Raspberry 4 and 400. Then click 'Choose SD card', select your microSD card. Finally, click 'Write'.

02 Install Raspberry Pi

We're using the Argon One M.2 case for a few reasons – it's secure, it keeps your Raspberry Pi cool, it can use a Nanosound DAC if that's your thing, and it routes all the inputs to the rear of the case. Because it has all these bells and whistles, it does take an extra step to install. The instruction

manual that comes with the Argon One shows you how to add the daughterboard to Raspberry Pi and then install it into the case with some screws. Follow these instructions, and keep the power button pins in the default position.

03 Hook it up

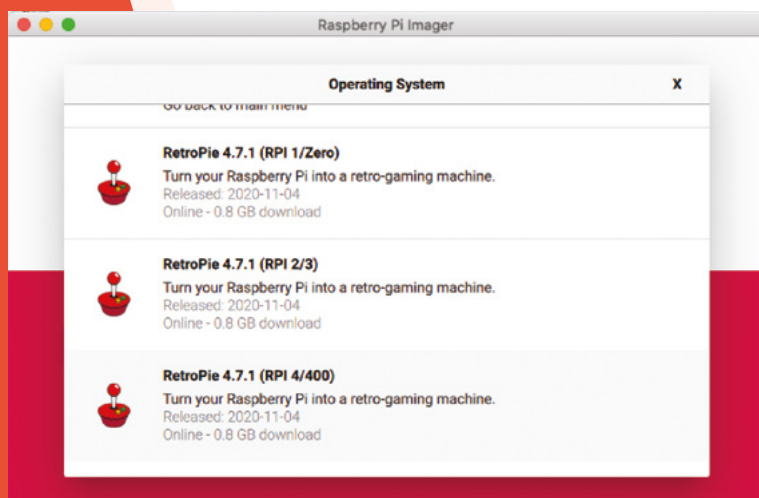
For first-time setup, we recommend connecting a USB keyboard, along with having it connected to a monitor instead of a TV for comfort – however, connecting it to your selected TV is also fine. If you also plan to use a wired game controller, connecting it now is also a good idea. The last thing to connect should be the power supply.

If you want to add an M.2 SSD drive for increased internal storage, do so now!

INITIAL SETUP

01 First boot

With a power supply plugged in and switched on, press the power button on the back of the case to boot up your Raspberry Pi. RetroPie will do some initial automated setup stuff, then ask you to configure your games controller



▲ Raspberry Pi Imager makes it easy to install RetroPie – choose the RPI 4/400 version



▲ Press the relevant buttons to configure your games controller; hold any button to skip a function



▲ Select the Configure WiFi option from the RetroPie menu and enter your network details

buttons. If you don't have a controller connected yet, you can press **F4** on the keyboard to get to the main menu. If you've run out of buttons to assign, hold down any button and it will skip the function. Keep doing this until you get to the end, and confirm 'OK' to finish.

If you're using a wired network and controllers, you can stop here! Just add ROMs over the network or via a USB stick to play games!

“ Make sure you turn the controller on just after the reboot ”

02 Wireless connection

If you don't intend to use an Ethernet connection, you can connect to your wireless network in the RetroPie menu. Select 'Configure WiFi' and it will open a text menu. Select 'Connect to WiFi' and choose your network from the list. Enter your password and hit OK – you may need to wait a moment or two, but it should then be fully connected.

To transfer ROMs over the network, go to **\\RETROPIE** in the Windows File Explorer, or **smb://retropie** on Mac, and open up the **roms** folder

Stream from Steam

RetroPie allows you to use Steam Link software to stream games from a gaming PC straight to your TV! To do this, go to the RetroPie menu and then RetroPie Setup. Go down to Manage Packages and choose Experimental Packages from it. In that list will be 'steamlink' – install it and it will then appear in the main games menu, ready for you to start streaming from a gaming PC running Steam with Remote Play on.



03 Bluetooth controllers

Press **F4** and you'll open the command line. From there, make sure Bluetooth is installed using the command:

```
sudo apt install pi-bluetooth
```

Type **exit** to return to the graphical interface. Go to the menu and select Bluetooth Configuration. Select Register and Connect Bluetooth Devices while your Bluetooth controller is in pairing mode, then select it from the screen and follow the pairing instructions. Once connected, you may need to reboot your Raspberry Pi before configuring the buttons – make sure you turn the controller on just after the reboot.

Install Argon ONE software

To install the fan control to the system, press **F4** to get to the command line, and enter the following:

```
curl https://download.argon40.com/argon1.sh | bash
```



Configure Kodi

Go beyond games to stream your own media files over a network

01 Install Kodi

Kodi is our preferred media centre software for watching video and playing music, and you can install it to RetroPie. Once again from the RetroPie menu, go to RetroPie Setup and then Manage Packages, Optional Packages. From there you'll be able to install Kodi – although it may take a while. Once installed, you'll find it in the Ports menu as an option after a reboot.

02 Setup Kodi

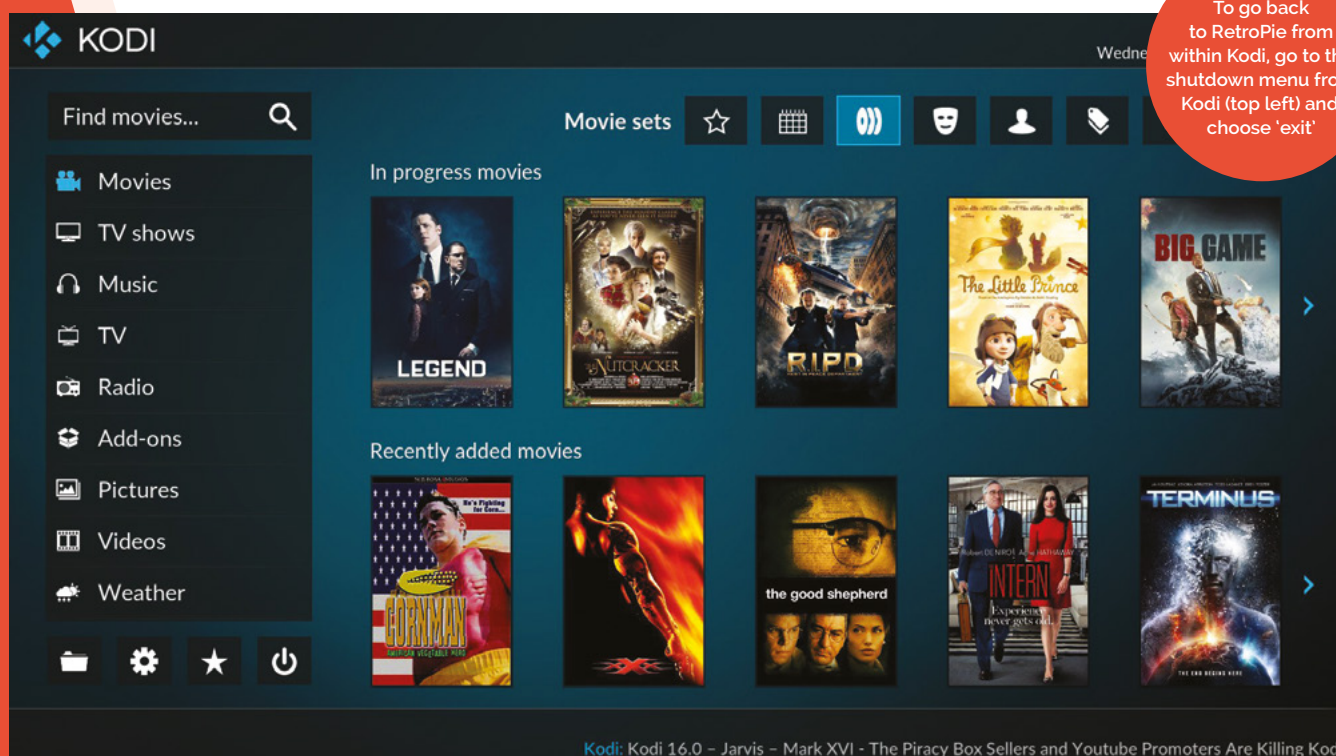
Kodi will be automatically set up after installation, however you may want to change some settings, especially if you're experiencing

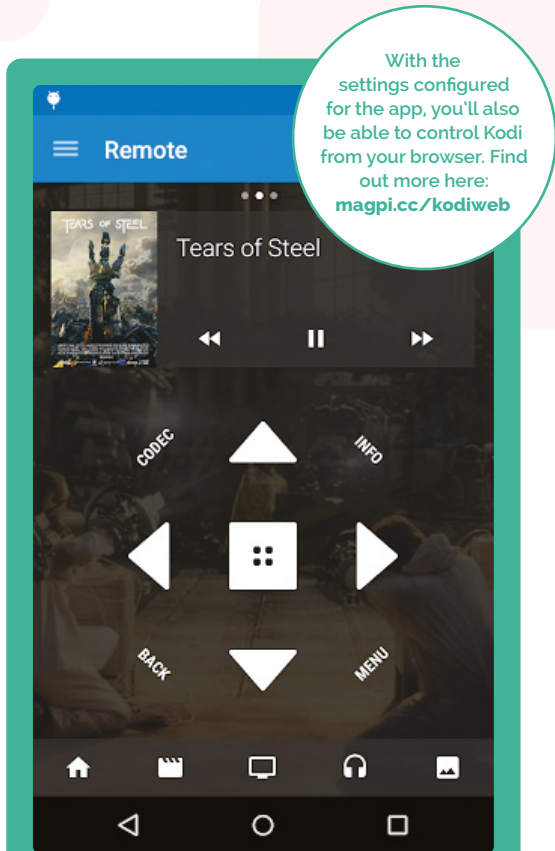
overscan issues. Find them in the settings menu, and be sure to give the name of the system an update if you want to as well so you can identify it on your network.

03 Add networked sources

The easiest way to access media (videos, music, or pictures) over your local network is to add them as media sources. Head back to the settings menu from the top of the interface. From there, go to File Manager and select Add Source. You'll be able to browse your available networked machines, or alternatively write in the SMB route if you know it.

To go back to RetroPie from within Kodi, go to the shutdown menu from Kodi (top left) and choose 'exit'





With the settings configured for the app, you'll also be able to control Kodi from your browser. Find out more here: magpi.cc/kodiweb

Smart device remotes

You can also control Kodi using Android and iOS devices, and there's an official app for it as well, which you can find here: magpi.cc/kodiremote.

While it's downloading to your desired device, you'll need to tell Kodi to allow you to use it. From the Settings menu, go to Services and find the Control option. From this list, you want to turn the following options to ON: 'Allow programs on other systems to control Kodi', 'Allow control of Kodi via HTTP', and 'Announce these services to other systems'.

Open the app and you should be able to select the name you gave your Kodi setup from the list and immediately start controlling it.

Using IR remotes

The Flirc USB dongle allows you to program specific media commands into itself from another computer before plugging it into a media centre. You may need a USB extender so it's visible, though!

The Argon ONE case does have an IR receiver, but it seems to have a couple issues with working at the moment. You can always try installing the IR software with `argonone-ir` from the command line, as there will likely be a future update for it.

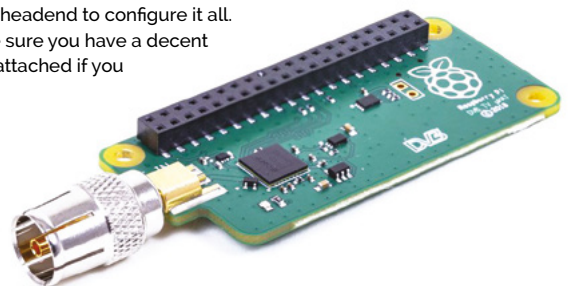
“ The first time you start Kodi, you'll have to do a basic setup ”

Watching TV

With a Raspberry Pi TV HAT, you can watch and record TV using Kodi. You can have it streaming TV via a different Raspberry Pi if you don't have an aerial socket near your home theatre, or have it as a server and viewer from the same device. We have a full guide you can find here: magpi.cc/tvthatguide.

The system uses Tvheadend to configure it all. You'll need to make sure you have a decent amount of storage attached if you plan to record TV.

▶ A Raspberry Pi TV HAT enables you to watch and record digital terrestrial TV



Essential add-ons

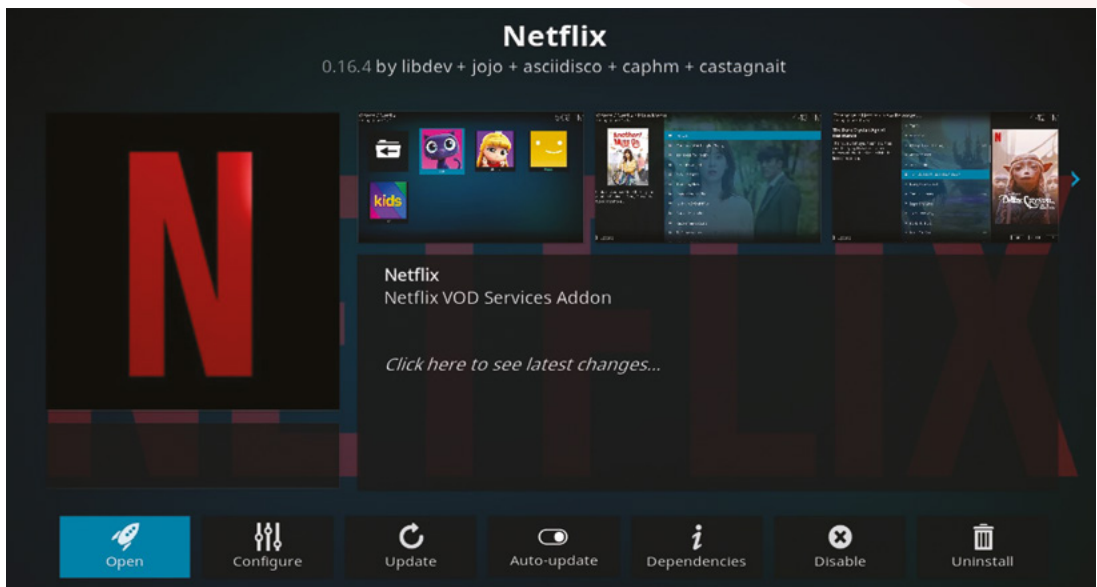
Get streaming services and TV running on Kodi for the full smart TV experience



Warning! Unofficial Add-on

Check that Netflix add-ons like this are legal in your region.

▶ Catch your favourite Netflix shows and movies on Kodi using a third-party source



INSTALL NETFLIX

01 Get the source

While Kodi doesn't have an official Netflix add-on, you can easily add a third-party one that allows easy access to the streaming service. First of all, head to Settings and go to System. Scroll down to Add-ons and turn 'Unknown sources' to on. Go back and head to the File Manager – go to Add Source and enter the following address:

<https://castagnait.github.io/repository.castagnait/>

Call it 'netflix' to make things easier.

02 Install the add-on

From the Settings menu, go to Add-ons and scroll down to 'Install from zip file'. As 'netflix' is now a source, you can select that to find a file called **repository.castagnait-x.x.x.zip** to download. Go back to Add-ons and go to 'Install

from repository', where you'll find an option for CastagnaIT Repository for Kodi. Select it, then Video Add-ons, and you'll see the Netflix add-on. Go to install it, choosing the latest version, and say OK to extra stuff that will be installed.

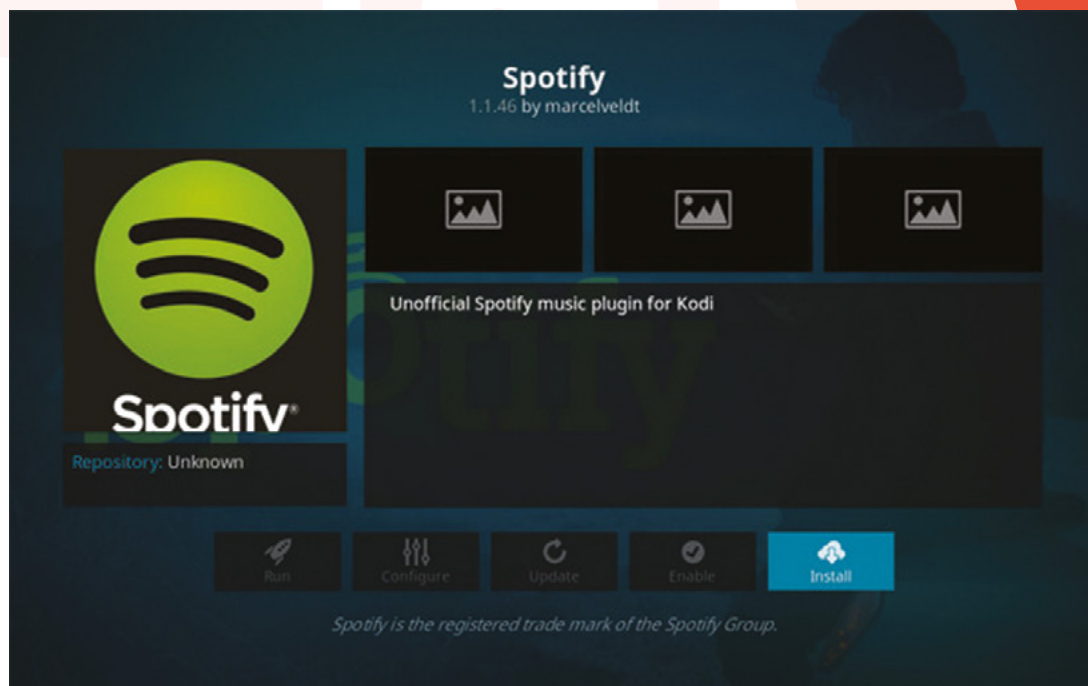
“ Go to install it, choosing the latest version, and say OK to extra stuff ”

03 Using Netflix

You may need to add extra software for Netflix if you're having issues. Exit Kodi, and press **F4** in RetroPie to access the command line. Enter the following:

```
sudo apt install python3-pip
pip3 install cryptodome
sudo apt install libnss3
```

Head back to Kodi and Netflix should launch.



▲ Listen to your favourite tunes with an unofficial Spotify add-on for Kodi

INSTALL SPOTIFY

01 Spotify source

There isn't an official Spotify app on Kodi, so you'll need to install from a third-party source like with the Netflix add-on, although this one you'll need to transfer from another computer. Head to magpi.cc/spotifykodi on a computer and download the zip file from the link under 'Install with repository'. If you're able to transfer it to your media PC over the internet, do so, or load it to a USB stick and plug it into the media box we've been building.

02 Install source and add-on

Like with Netflix, head to Settings, Add-ons, and 'Install from zip file'. Head to the folder you downloaded it to, or go through the USB storage you're transferring it with to find the zip file. Head to 'Install from repository' and go to Marcelveldt's BETA repository. Go to Music Add-ons and then install the Spotify app from there. If you're having trouble installing, you may need to disable the Netflix add-on and restart.

Install YouTube

Unlike Spotify and Netflix, this one is a bit easier – a YouTube app is available in the standard Add-ons list for Kodi! From the Add-ons menu, go to Download, Video Add-ons, and scroll down to YouTube. Install it from there and it will live in the Add-ons menu from now on. Like the other apps, you'll have to use the standard Kodi interface to navigate it.

03 Configure and use Spotify

Use the Configure option from the screen where you installed the Spotify app and enter your Spotify login details. Once that's done, head back to the home screen and you'll find Spotify under the Add-ons menu. Open it to start browsing and listening – once again, it will use the standard Kodi interface instead of the usual Spotify one.

Electronics with Raspberry Pi 400



PJ Evans

MAKER

PJ is a writer, tinkerer, and software engineer. He has broken enough LEDs to light up Blackpool illuminations.

@mrpjvans

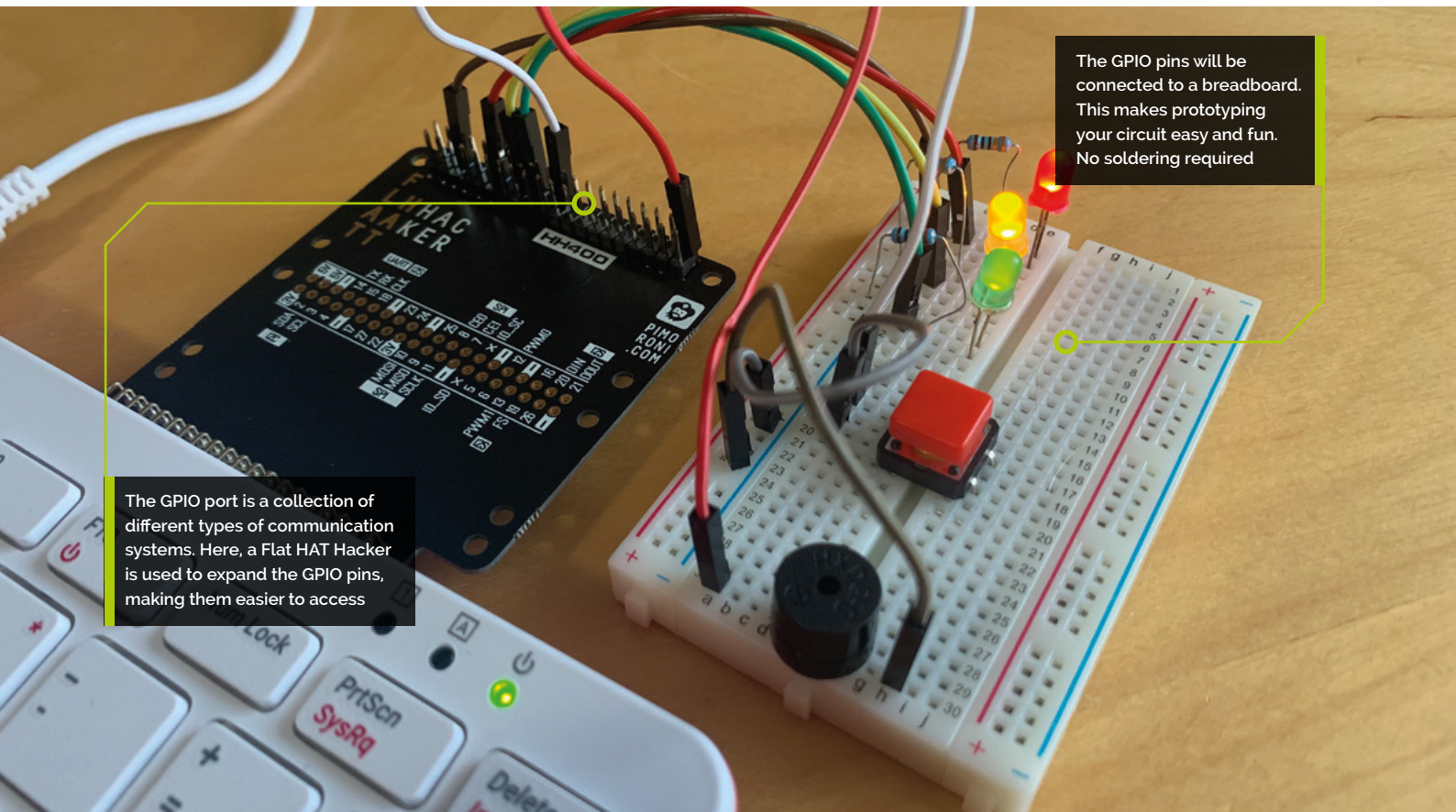
Start your electronics adventure. Build your first circuits and control them with code

You've got your new Raspberry Pi 400, admired its bright casing, and been amazed by the engineering that fitted so much power in a keyboard form factor. So now what? Well, there's one thing that separates Raspberry Pi 400 from other desktop computers: its GPIO (general-purpose input and output) port. These 40 pins on the rear of the machine allow you to easily connect all kinds of sensors and indicators directly to Raspberry Pi 400 and then control them with code. If you're excited by this but don't know where to start, this tutorial will guide you on your first steps into an amazing and rewarding world.

BREADBOARD AND CIRCUIT

01 Set up safely

Electronics is a delicate business. The components we are using are easily damaged, both physically and through being incorrectly wired up. The first thing to do is ensure you have a clean workspace. Carpets or anything that generates a lot of static should be avoided. Set up your Raspberry Pi 400 with a mouse and monitor. We're going to be coding with Thonny,

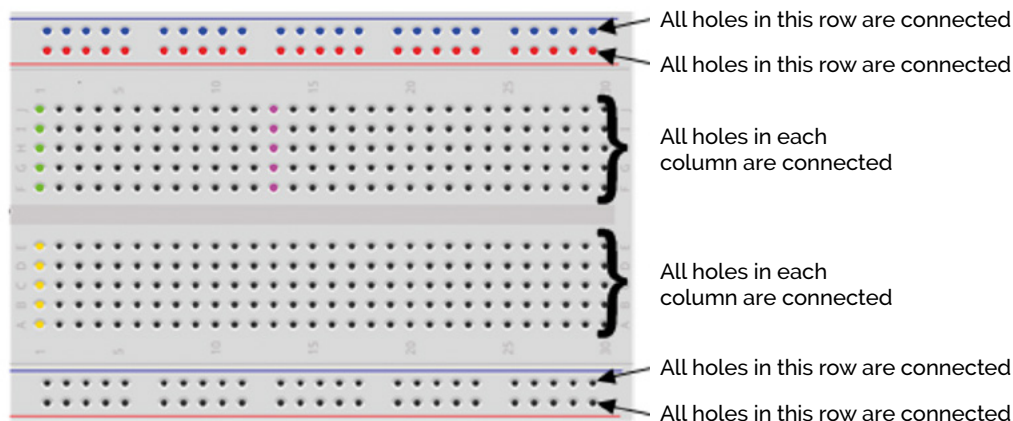


The GPIO pins will be connected to a breadboard. This makes prototyping your circuit easy and fun. No soldering required

The GPIO port is a collection of different types of communication systems. Here, a Flat HAT Hacker is used to expand the GPIO pins, making them easier to access

You'll Need

- > CamJam EduKit #1
magpi.cc/edukit
- > Flat HAT Hacker
magpi.cc/flathathacker



a simple yet powerful development tool provided with Raspberry Pi OS. You'll find it under 'Programming' in the main menu.

02 Introducing the EduKit

The CamJam EduKit #1 is a fantastic starter kit for those wanting to dip a toe into the world of building circuits. Rather than deluging you with a range of weird and wonderful components, this is a £5 selection of the absolute basics. It also comes with freely downloadable worksheets that go through a range of projects in satisfying detail. Inside you'll find a breadboard, some resistors, a buzzer, LEDs, and cables to connect everything up. If you would like a detailed list of what is included, have a look here: magpi.cc/edukitworksheets.

“ The 40 pins on the rear of the machine allow you to easily connect all kinds of sensors and indicators ”

03 Wiring up

The breadboard supplied with the EduKit is used for prototyping simple circuits. Each column of points/holes is wired together to create rails so we can use components to 'bridge' the rails. Additionally, there are four rails arranged in rows for easily supplying power. To safely connect the breadboard to your Raspberry Pi 400, the Flat HAT Hacker elegantly exposes the GPIO pins so that we can connect jumper cables between it and the breadboard. With the power off, connect the Flat HAT Hacker to the GPIO port.

04 Introducing LEDs

LEDs (light-emitting diodes) are cheap but fickle things. They must be placed the right way around and we must be careful with the amount of current they draw: too much and your LED will snuff out for good (it's not dangerous, but can be frustrating). We solve this by placing a resistor in the circuit. It's a component that reduces the current, a bit like squeezing a garden hose. We need the 330Ω (ohm) resistors. Any higher and there won't be enough current to light up the LED. Finally, have a look at the legs of the LED. The longer one is the positive side. If you put the LED in the wrong way around, it won't light up.

▲ A breadboard is an essential tool for learning electronics. Here's how it is wired up

▼ **Figure 1** Your first LED circuit. Be very careful with the wiring, especially the orientation of the LED

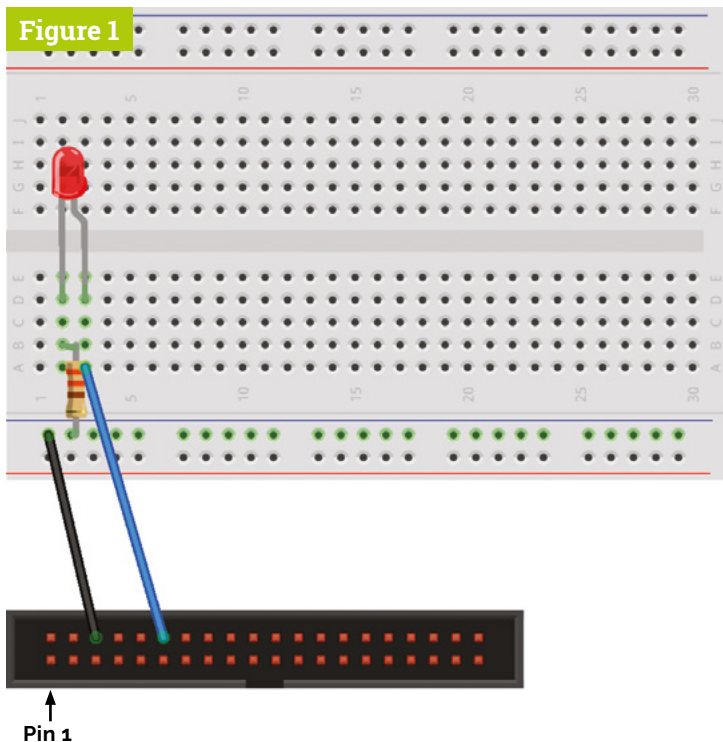
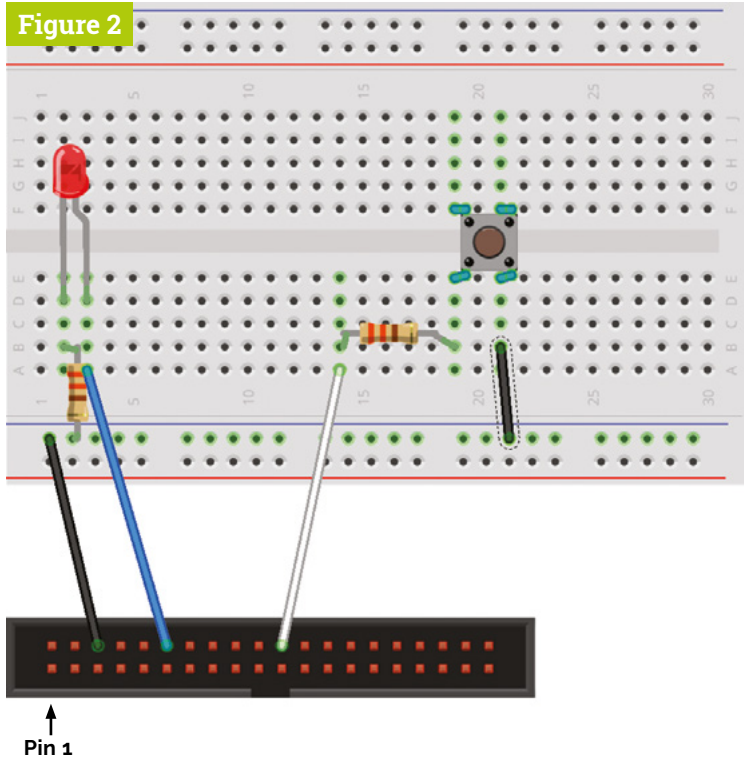


Figure 2



▲ Figure 2 Add a switch to control the LED using code

05 Power up

Study our first circuit (Figure 1) carefully. As shown, use a jumper cable to connect a GND (ground) pin on the GPIO header to a hole in the ‘-’ row of the breadboard. This forms a ‘ground rail’ that goes along the entire row so we can easily create a circuit. Now make another connection between pin GPIO 18 on the GPIO and one of the columns as shown. Later on we’ll switch the pin’s 3.3V electrical current on and off by code. To protect the LED, add a 330Ω resistor between the ground rail and a different column. Finally, bridge the two columns with the red LED, making sure the longer leg is not on the resistor side. Check and double-check your work.



Power off

Although the voltages used are low and won’t cause you any harm, a misplaced wire could be curtains for your computer. Power down before attaching, moving, or removing wires, and double-check everything before flicking the switch.

06 Just add code

Once you’re satisfied that everything is connected correctly, boot up your Raspberry Pi 400 and start Thonny from the Programming menu. Carefully enter the code in **led.py**. This uses a library (a collection of things to help you do stuff easier) called GPIOZero, which takes a lot of complexity away from talking to the GPIO. The



Pull up, or down, pins?

Not all GPIO pins are the same. By default, the first eight GPIO pins on Raspberry Pi are set to pull-up, while the rest are set to pull-down. With a pull-down pin, like GPIO 25 used for our button, it is a ‘0’ when the button is not pressed, a ‘1’ when the button is pressed, and will return to ‘0’ when released. Some circuits use a resistor connected to the button. With a resistor, it doesn’t matter if the pin is pull-up, or pull-down. If you are having trouble with a button in a circuit, try adding a resistor, or use a different GPIO pin. You can adjust the default GPIO pin behaviour using the **config.txt** settings. See magpi.cc/gpiocontrol for more information.

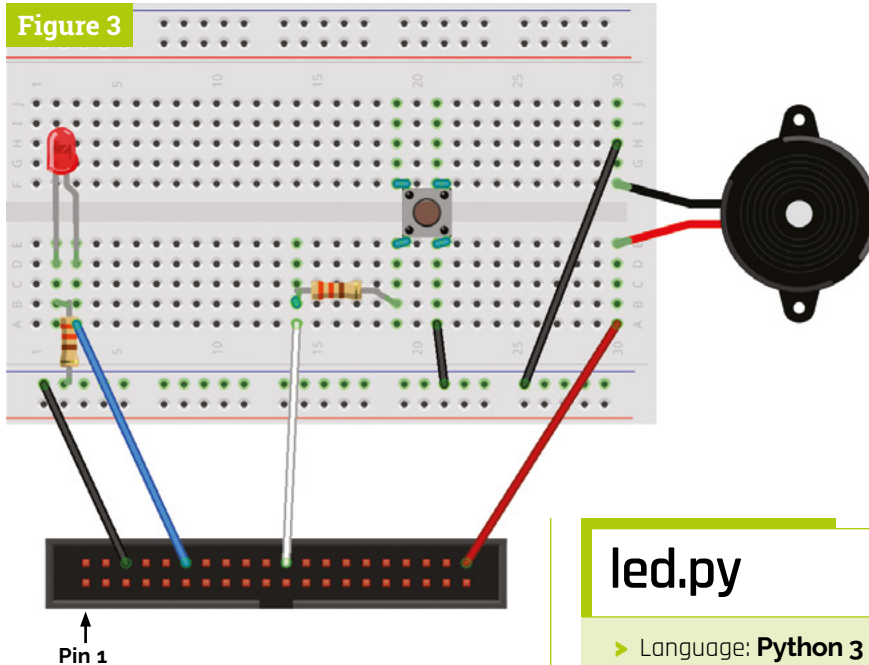
code configures pin GPIO 18 to be an output (we can send current out when we want to) and then sets it to ‘on’, which means we are sending the current. After five seconds it turns the LED off again. The **print()** commands let us know what is happening in the Shell at the bottom. Save the **led.py** file to your home directory and then run it using the Run button. Does the LED light up? If not, check your circuit and code carefully.

07 On the blink

Now you have successfully illuminated your LED, let’s do something a little more fun. Change the code to that shown in **led_blink.py**. Run it and see what happens. Blinking! Click the Stop button when you’re done. Can you change the speed of the flashes?

08 Getting some input

As well as sending output, Raspberry Pi’s GPIO header can read input from a large range of components. These can include switches,



▲ **Figure 3** Practise your Morse code with a buzzer and switch

temperature, pressure and light sensors, and much more. As you may have seen from the code, we can assign some (not all) of the GPIO pins to be either an input or output. This allows us to read in information from a sensor, make a decision in code about what to do and then, if we wish, instruct an output device. So let's start by simply switching on the LED when a button is pressed.

“ As well as sending output, Raspberry Pi's GPIO header can read input from a large range of components ”

09 Press to play

In the EduKit there is a tactile switch. Like the LED, these can bridge the columns of the breadboard, only making a connection when the button is pressed down. With your Raspberry Pi 400 off, wire everything up as shown in **Figure 2**. You should have a connection between the switch and pin GPIO 25 and also the ground

led.py

> Language: **Python 3**

DOWNLOAD THE FULL CODE:



magpi.cc/electronics400git

```
001. import time
002. from gpiozero import LED
003.
004. red = LED(18)
005.
006. print("LED on")
007. red.on()
008. time.sleep(5)
009. print("LED off")
010. red.off()
```

led_blink.py

> Language: **Python 3**

```
001. import time
002. from gpiozero import LED
003.
004. red = LED(18)
005.
006. print("Blinking the LED")
007.
008. while True:
009.     red.on()
010.     time.sleep(0.5)
011.     red.off()
012.     time.sleep(0.5)
```

Figure 4

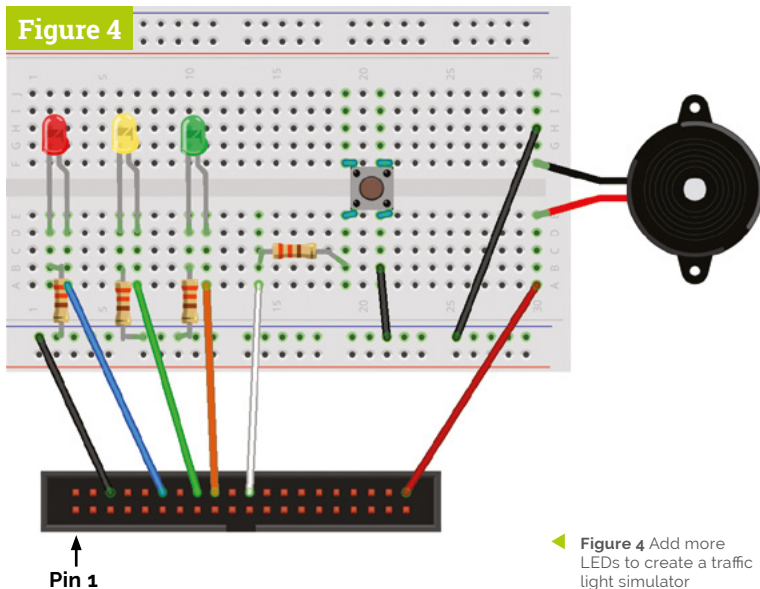


Figure 4 Add more LEDs to create a traffic light simulator

led_button.py

> Language: Python 3

```
001. from gpiozero import LED, Button
002. from signal import pause
003.
004. red = LED(18)
005. button = Button(25)
006.
007. button.when_pressed = red.on
008. button.when_released = red.off
009.
010. pause()
```

buzzer.py

> Language: Python 3

```
001. from gpiozero import Buzzer, Button
002. from signal import pause
003.
004. bz = Buzzer(21)
005. button = Button(25)
006.
007. button.when_pressed = bz.on
008. button.when_released = bz.off
009.
010. pause()
```

rail. Power up and add the `led_button.py` code: When you run it, the LED should only light up when the button is pressed. Can you change the code so the opposite is true?

10 Make some noise

Here's an example of another type of output: sound. Included in the kit is a simple buzzer. The longer leg is the positive end, just like the LED. Connect the buzzer to two separate columns on the breadboard and, using jumper cables, connect the short leg column to the ground rail and the long leg column to pin GPIO 21, as in **Figure 3**. Change your code to `buzzer.py`. Run the code and when you press the button, you should hear a satisfying buzz. Can you add to the code to make a buzz and light up the LED at the same time?

“ Now we have three LEDs, a button, and a buzzer, can you make a game? You can decide on your own rules ”

11 Traffic lights

There are plenty of GPIO pins to play with, so add two more LEDs wired to 330 Ω resistors and the ground rail, with the positive side going to pins GPIO 23 and GPIO 24 (see **Figure 4**). You now have a set of traffic lights! Using what we've learnt about switching current on and off and timing, can you create the classic traffic light sequence when the button is pressed? (Stuck? A

Top Tip

GPIO Documentation

The numbering of GPIO pins is not in numerical order. Also, pay close attention to PIN1 and PIN40 on both Raspberry Pi 400 and Flat HAT Hacker. More information on GPIO can be found here: magpi.cc/gpio.



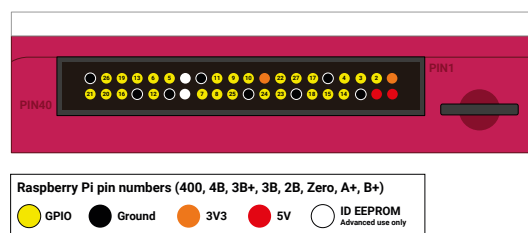
solution is here: magpi.cc/electronics40ogit). Can you get the buzzer to make a pedestrian crossing sound?

12 Now make a game

A final challenge for you before you move on with your electronics adventure. Now we have three LEDs, a button, and a buzzer, can you make a game? You can decide on your own rules. How about having to press the button when green lights up? If you get it wrong, the buzzer sounds. You can use Python's 'random' library to pick a random LED to light up (there's an example at magpi.cc/electronics40ogit). Also, you can work through the free CamJam EduKit worksheets for a closer look at how these circuits work. Put your imagination to work and have fun! 🎮

The GPIO pins on Raspberry Pi 400

In this tutorial we're using a Flat HAT Hacker to make wiring up the GPIO pins easier. However, you can connect jumper wires directly to the GPIO pins on Raspberry Pi 400 just like on any other Raspberry Pi model. Make sure to use the correct pin numbers, by checking with this pin guide and look at the PIN1 and PIN40 markings next to the GPIO pins on the back of Raspberry Pi 400. If you tilt Raspberry Pi 400 so the keyboard is facing towards you, the pins will be in the same position as the Flat HAT Hacker pins. Make sure you have Raspberry Pi 400 turned off when connecting and disconnecting wires. An interactive pinout guide can be found here: magpi.cc/pinout.



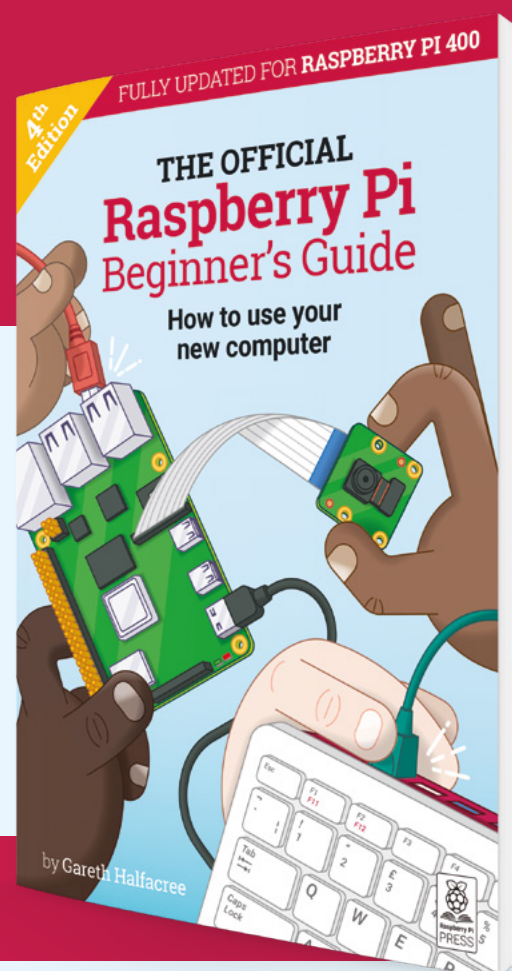
THE OFFICIAL Raspberry Pi Beginner's Guide

The only guide you need to get started with Raspberry Pi

- Learn how to set up your Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch 3 and Python languages
- Create amazing projects by connecting electronic components to Raspberry Pi's GPIO pins

Plus much, much more!

Buy online: magpi.cc/bgbook4



Create GUIs with Python: Spy name chooser

Make an interactive GUI application



Laura Sach

Laura leads the A Level team at the Raspberry Pi Foundation, creating resources for students to learn about Computer Science.

@CodeBoom



Martin O'Hanlon

Martin works in the learning team at the Raspberry Pi Foundation, where he creates online courses, projects, and learning resources.

@martinohanlon

So far in this series, you've learnt how to customise your GUI with a variety of different options. It's now time to get into the really interactive part and make a GUI application that actually responds to input from the user. Who could resist pushing a big red button to generate a super secret spy name?

Since you already know how to create an app, why not go ahead and create a basic window and add some text if you like? Here is some code to get you started, and this code also includes some comments (the lines that start with a #) to help you structure your program:

```
# Imports -----
from guizero import App, Text

# Functions -----

# App -----
app = App("TOP SECRET")

# Widgets -----
title = Text(app, "Push the red button to
find out your spy name")

# Display -----
app.display()
```

Run this code and you should see a window with the text (Figure 1).

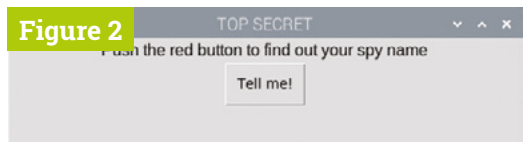
Add a button

Let's go ahead and add a button to the GUI. Add `PushButton` to your list of imports so that you can use buttons. (Be careful to use a capital B!)



Figure 1

► **Figure 1** Displaying the text in a window



▲ **Figure 2** You now have a button

Underneath the `Text` widget, but before the app displays, add a line of code to create a button.

```
button = PushButton(app, choose_name,
text="Tell me!")
```

Your code should now look like `spy1.py`. Run it and no button will appear, but you'll see an error in the Shell window:

```
NameError: name 'choose_name' is not defined
```

This is because `choose_name` is the name of a command which runs when the button is pressed. Most GUI components can have a command attached to them. For a button, attaching a command means "when the button is pressed, run this command." A GUI program works differently to other Python programs you might have written because the order in which the commands are run in the program depends entirely on the order in which the user presses the buttons, moves the sliders, ticks the boxes, or interacts with whichever other widgets you are using. The actual command is almost always the name of a function to run.

Create a function

Let's write the function `choose_name` so your button has something it can do when it is pressed.

Look at your program and find the functions section. This is where you should write all of the functions which will be attached to GUI widgets, to keep them separate from the code for displaying the widget. Add this code in the functions section:

```
def choose_name():
    print("Button was pressed")
```



Figure 3

▲ **Figure 3** Text is output to the Shell window

Your code should now look like **spy2.py**. The button will now appear (**Figure 2**). If you press the button, it may appear that nothing has happened, but if you look in your Shell or output window, you will see that some text has appeared there (**Figure 3**).

Instructing your function to first print out some dummy text is a useful way of confirming that the button is activating its command function correctly when it is pressed. You can then replace the `print` statement with the actual code for the task you would like your button to perform.

Inside your `choose_name` function, type a `#` symbol in front of the line of code that prints “Button was pressed”. Programmers call this ‘commenting out’, and what you have done here is told the computer to treat this line of code as if it were a comment, or in other words you have instructed the computer to ignore it. The benefit of commenting a line of code out instead of just deleting it is so that if you ever want to use that code again, you can easily make it part of your program again by removing the `#` symbol.

Add some names

On a new line, add a list of first names. You can choose the names in your list and there can be as many names as you like, but make sure that each name is between quotes, and the names are each separated by a comma. A collection of letters, numbers, and/or punctuation between quotation marks is called a *string*, so we say that each name must be a string.

```
first_names = ["Barbara", "Woody",
               "Tiberius", "Smokey", "Jennifer", "Ruby"]
```

Now add a list of last names as well:

```
last_names = ["Spindleshanks", "Mysterioso",
              "Dungeon", "Catseye", "Darkmeyer",
              "Flamingobreath"]
```

Now you will need to add a way of choosing a random name from each list to form your spy name. Your first job is to add a new import line in your imports section:

```
from random import choice
```

spy1.py

► Language: **Python 3**

**DOWNLOAD
THE FULL CODE:**

 magpi.cc/guizero code

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003.
004. # Functions -----
005.
006. # App -----
007. app = App("TOP SECRET")
008.
009. # Widgets -----
010. title = Text(app, "Push the red button to find out your spy name")
011. button = PushButton(app, choose_name, text="Tell me!")
012.
013. # Display -----
014. app.display()
```

spy2.py

► Language: **Python 3**

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003.
004. # Functions -----
005.
006. # App -----
007. app = App("TOP SECRET")
008.
009. # Widgets -----
010. title = Text(app, "Push the red button to find out your spy name")
011. button = PushButton(app, choose_name, text="Tell me!")
012.
013. # Display -----
014. app.display()
```

This tells the program that you would like to use a function called `choice` which chooses a random item from a list. Someone else has written the code which does this for you, and it is included with Python for you to use.

In your code for the `choose_name` function, just below your lists of names, add a line of code to choose your spy’s first name, and then concatenate it together with the last name, with a space in between. Concatenate is a fancy word that means

Big red button

At the moment, your button is not big or red! You used properties in part one of this tutorial series to change the appearance of your text on the ‘Wanted’ poster, so can you use the properties of the `PushButton` widget to change the background colour and the text size?

Note that it may not be possible to change the colour of a button on macOS, as some versions of the operating system will not allow you to do so, but you should still be able to alter the text size.

spy3.py

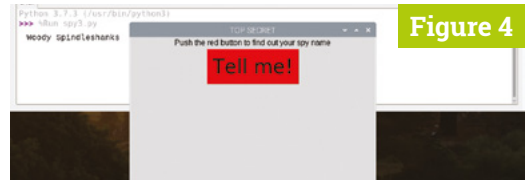
> Language: Python 3

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003. from random import choice
004.
005. # Functions -----
006. def choose_name():
007.     #print("Button was pressed")
008.     first_names = ["Barbara", "Woody", "Tiberius", "Smokey",
009.                  "Jennifer", "Ruby"]
010.     last_names = ["Spindleshanks", "Mysterioso", "Dungeon",
011.                 "Catseye", "Darkmeyer", "Flamingobreath"]
012.     spy_name = choice(first_names) + " " + choice(last_names)
013.     print(spy_name)
014.
015. # App -----
016. app = App("TOP SECRET")
017.
018. # Widgets -----
019. title = Text(app, "Push the red button to find out your spy name")
020. button = PushButton(app, choose_name, text="Tell me!")
021. button.bg = "red"
022. button.text_size = 30
023.
024. # Display -----
025. app.display()
```

03-spy-name-chooser.py

> Language: Python 3

```
001. # Imports -----
002.
003. from guizero import App, Text, PushButton
004. from random import choice
005.
006. # Functions -----
007.
008. def choose_name():
009.     #print("Button was pressed")
010.     first_names = ["Barbara", "Woody", "Tiberius", "Smokey",
011.                  "Jennifer", "Ruby"]
012.     last_names = ["Spindleshanks", "Mysterioso", "Dungeon",
013.                 "Catseye", "Darkmeyer", "Flamingobreath"]
014.     spy_name = choice(first_names) + " " + choice(last_names)
015.     #print(spy_name)
016.     name.value = spy_name
017.
018. # App -----
019. app = App("TOP SECRET")
020.
021. # Widgets -----
022. title = Text(app, "Push the red button to find out your spy name")
023. button = PushButton(app, choose_name, text="Tell me!")
024. button.bg = "red"
025. button.text_size = 30
026. name = Text(app, text="")
027.
028. # Display -----
029.
030. app.display()
```



▲ Figure 4 Outputting a spy name

‘join two strings together’ and the symbol in Python for concatenation is a plus (+).

```
spy_name = choice(first_names) + " " +
choice(last_names)
print(spy_name)
```

Your code should now resemble **spy3.py**. Save and run it. When you press the button, you should see that a randomly generated spy name appears in your console or Shell, in the same place where the original “Button was pressed” message showed up before (**Figure 4**).

Put the name in the GUI

That’s good, but wouldn’t it be nicer if the spy name appeared on the GUI? Let’s make another Text widget and use it to display the spy name.

In the widgets section, add a new Text widget which will display the spy name:

```
name = Text(app, text="")
```

When you create the widget, you don’t want it to display any text at all as the person won’t have pressed the button yet, so you can set the text to “”, which is called an ‘empty string’ and displays nothing. Inside your **choose_name** function, comment out the line of code where you print out the spy name.

Now add a new line of code at the end of the function to set the value of the name Text widget to the **spy_name** you just created. This will cause the Text widget to update itself and display the name.

```
name.value = spy_name
```

Your final code should be as in **03-spy-name-chooser.py**. Run it and press the button to see your spy name displayed proudly on the GUI (**Figure 5**).

You can press the button again if you don’t like the name you are given, and the program will randomly generate another name for you. 🎲

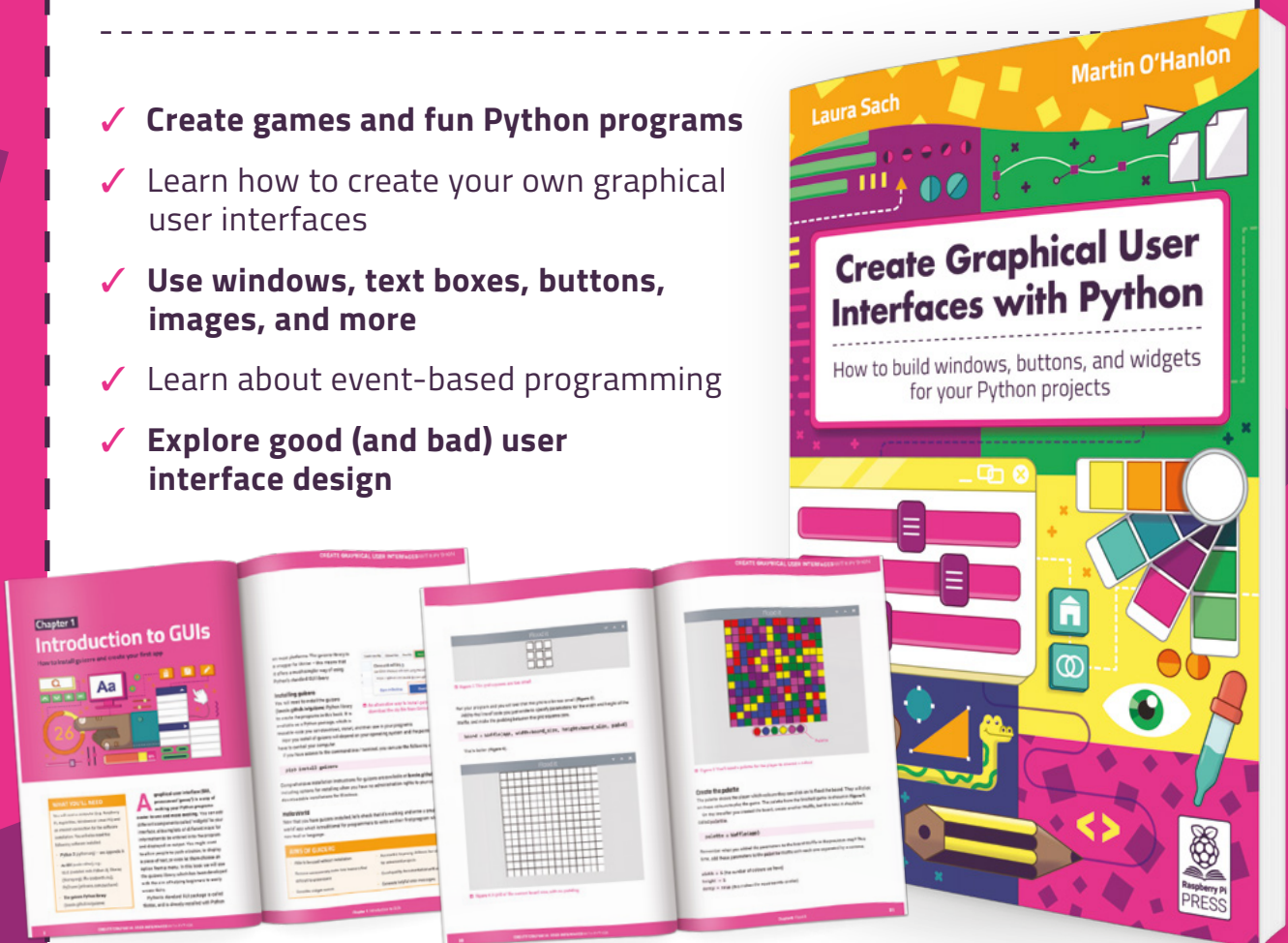


▲ Figure 5

▲ Figure 5 The finished spy name chooser

Create Graphical User Interfaces with Python

- ✓ Create games and fun Python programs
- ✓ Learn how to create your own graphical user interfaces
- ✓ Use windows, text boxes, buttons, images, and more
- ✓ Learn about event-based programming
- ✓ Explore good (and bad) user interface design



Buy online: magpi.cc/pythongui

Turn Raspberry Pi 400 into a legal C64 emulator

UK copyright law puts C64 emulation enthusiasts in a difficult position. This new VICE emulator fork replaces system files of unknown origin with official versions



K.G. Orphanides

MAKER

K.G. makes, writes about, and helps to preserve unusual gaming software and hardware.

@KGOOrphanides

If you're a fan of Commodore's classic 8-bit computers, you'll have noticed that the VICE emulator is nowhere to be found in Raspberry Pi OS's software repositories. Unlike some emulators, which use clearly authorised or reverse-engineered ROM images of the original computers' firmware, the ROMs included in VICE are ambiguously sourced, with little explicit licensing information.

That's a problem for anyone who wants to use VICE in the UK, as British copyright law is particularly stringent. Fortunately, almost all of

Commodore's firmware is legally available for free via current licence holder Cloanto.

A fork of VICE 3.4, the most recent version of the emulator that runs smoothly on Raspberry Pi, has been created. This fork is stripped of any ROMs whose licensing status is unclear. It ships with a script to integrate Cloanto's ROMs.

This tutorial stands alone to give you a copyright-compliant C64 emulator on Raspberry Pi 400, but can also build on last month's Spectrum emulation project, using the same GPIO joystick controller setup.

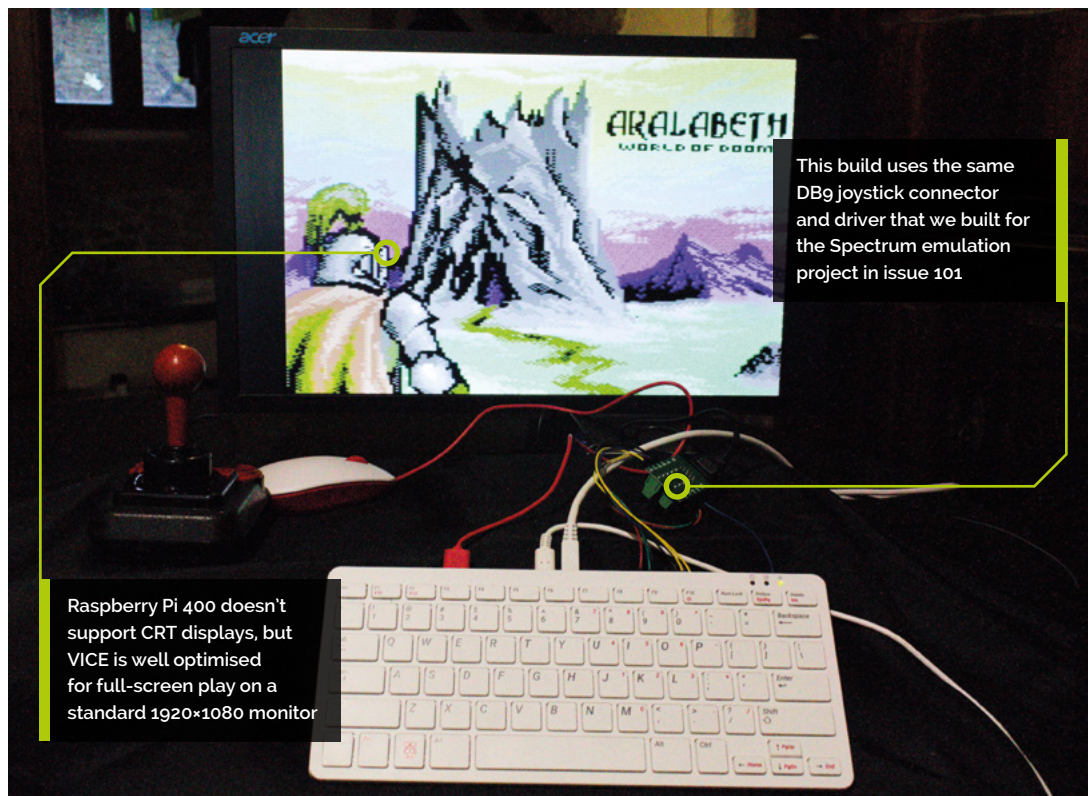
You'll Need

- > VICE 3.4 copyright compliant fork magpi.cc/vice
- > C64 Forever c64forever.com



Warning! Copyright

Only some C64 games can be legally downloaded. See our Legal ROMs page for more information. magpi.cc/legalroms



This build uses the same DB9 joystick connector and driver that we built for the Spectrum emulation project in issue 101

Raspberry Pi 400 doesn't support CRT displays, but VICE is well optimised for full-screen play on a standard 1920x1080 monitor

01 Preparation

If you followed last month's tutorial in its entirety, you'll probably want to undo the final step, which opens FUSE on boot. Delete or rename `/home/pi/.config/autostart/fuse.desktop`.

Next, we'll download Cloanto's legal Commodore ROM set, C64 Forever (c64forever.com). It's only available as a Windows MSI file. If you have a Windows PC, simply install it.

If you have an x86-based Linux or macOS system, Wine 5.0 and above (winehq.org) includes full MSI file support, so you can similarly install that, run the installer from your file manager, locate your Wine prefix directory (`~/.wine` by default), and you're good to go. Users of older versions can open a command terminal and run:

```
wine msiexec -i c64forever8.msi
```

If Raspberry Pi is your only computer, you can use Box86 (magpi.cc/box86) to run an x86 version of Wine; that's an extensive emulation project in its own right, but the TwisterOS image (magpi.cc/twisteros) comes loaded with Box86 and Wine to make this a little easier.

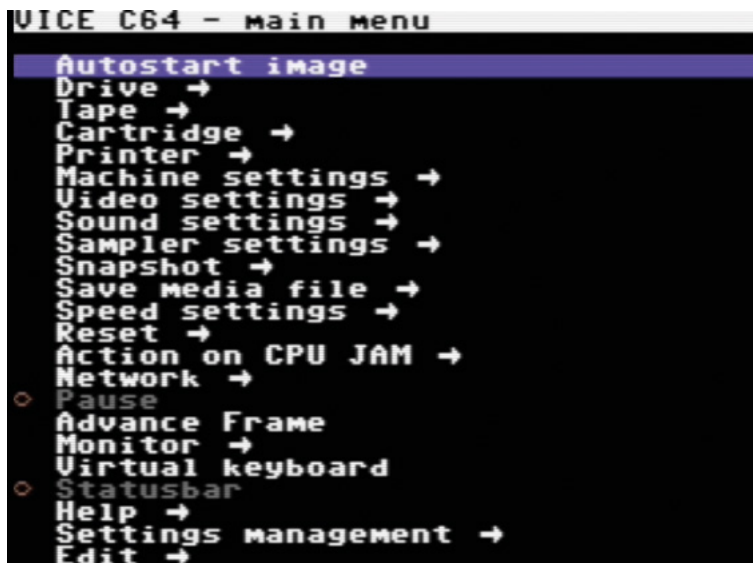
02 Locate your system ROMs

C64 Forever Express is free and includes all the ROMs you'll need for most Commodore 8-bit systems, except the PET and the C64 Direct-to-TV released by Ironstone. It also lacks Creative Micro Designs' SuperCPU ROM, but our fork uses an SCPU64-compatible ROM created by the VICE team.

After installation, you'll find the ROMs in `/users/Public/Documents/CBM Files/Shared/rom` of your Windows system drive. Copy this directory over to Raspberry Pi.



▲ With pin-sharp controls, Sarah Jane Avory's award-winning Zeta Wing is a stunning example of the new wave of commercial C64 games



A full C64 Forever Plus licence (\$15 / £11 from c64forever.com) includes extra features for Windows and Wine users, and it's also the obvious choice if you wish to support Cloanto financially.

Both versions include a collection of around 100 games that Cloanto licensed from its developers and publishers, including Jack the Nipper, Stormlord, and Auf Wiedersehen Monty. Registered users will find the games on their Windows system drive at `/users/Public/Documents/CBM Files/Games` – copy that over, too, for later use.

▲ VICE's menu is hugely comprehensive, but you'll mostly interact with its Machine, Drive, and Video settings

03 Get ready to build VICE

We'll be using a version of VICE 3.4 that includes no copyrighted firmware ROMs. It won't build or work without ROMs, so we'll replace these using a script that ships with this fork. In a Terminal, enter the following two commands:

```
sudo apt install autoconf automake build-essential byacc dos2unix flex libavcodec-dev libavformat-dev libgtk2.0-cil-dev libgtkglex11-dev libmp3lame-dev libmpeg123-dev libpcap-dev libpulse-dev libreadline-dev libswscale-dev libvte-dev libxaw7-dev subversion texi2html texinfo yasm libgtk3.0-cil-dev xa65 libsd12-dev libsd12-image-dev
```

```
git clone https://gitlab.com/mighty-owlbear/vice-3-4-copyright-compliant-uk.git
```

Now copy the `rom` directory from C64 Forever into your unpacked `vice-3-4-copyright-compliant-uk` directory.

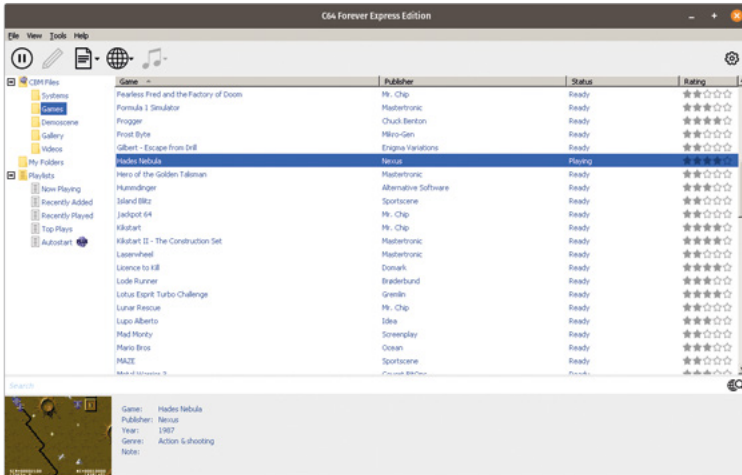
Top Tip

RTFM

The VICE manual is a chunky but vital reference. Read it online at magpi.cc/vicemanual.

▼ Physical games are still published for the C64, like this 5,25-inch floppy disk release of Dungeoneer's Akalabeth port, but you need extra hardware to use them





▲ C64 Forever is a native Windows program, but both its free Express and paid-for Plus editions include fully licensed Commodore ROMs for use with emulators

04 Build and install

The **copyrom** script supplied with this version of VICE looks for official ROM files in a directory called **rom** and copies and renames them as expected by VICE. Not all system ROMs are available; where this is the case, the script creates dummy files. If further ROMs are officially released, these can be replaced in an installed version by manually overwriting the files in **usr/local/lib/vice**.

```
cd vice-3-4-copyright-compliant-uk
./copyrom.sh
./autogen.sh
./configure --enable-sdlui2 --without-oss --enable-ethernet --disable-catweasel --without-pulse --enable-x64
make -j $(nproc)
sudo make install
```

Note that you have to build VICE **--without-pulse**, as above, for working audio. If you'd rather install VICE to a user directory, you can add **--prefix=/home/pi/viceinstall_3.4_clean** and run **make install** as a standard user, instead of using **sudo**.

▶ Cinemaware's Defender of the Crown is one of a couple of games licensed for exclusive distribution by Kryoflux as a demonstration of its imaging system's handling of copy-protected C64 disks



05 Test VICE

Although VICE includes several different emulators, we're primarily interested in the C64. Two different C64 emulators are supplied: **x64sc** is more accurate, while the older **x64** emulator – which we specifically built using the **--enable-x64** parameter above – is less CPU intensive, making it the best choice for Raspberry Pi.

In the Terminal, we'll create a directory for your C64 game collection, download Kryoflux's authorised C64 release of Cinemaware's Defender of the Crown, and use it to test the emulator.

```
cd
mkdir C64 && cd C64
wget http://www.kryoflux.com/download/DEFENDEROFTHECROWN.zip
x64
```

VICE should open with a blue window and an invitation to press **F12**. Do so, and marvel at the sheer number of configurable options available. Fortunately, we'll only need a few of them. But let's start by loading that game...

Press **ENTER** on 'Autostart image', browse to **DEFENDEROFTHECROWN.zip**, press **ENTER**, go down to the PRG file called **"!V-MAX!** and press **ENTER** to load it.

06 Configure inputs

The GPIO joystick setup we built last month works perfectly with VICE 3.4, as will any USB joystick or joypad, but you'll notice that it initially only works in the menus.

Press **F12** to open the menu. Go to Machine settings > Joystick settings > Joystick device 2. Press **ENTER** to select it, scroll down to Joystick at the bottom, and press **ENTER** again.

Press left on your keyboard or joystick to go back up one level to the Joystick settings menu. Go down to 'Joystick 2 mapping' and select it. Select each of Up, Down, Left, Right, and Fire in turn, and tap the joystick button you want within the five-second configuration period to assign it.

If you accidentally assign the wrong button to one of these,



Published by Protovision, Lasse Öörni's MW Ultra is one of many modern C64 games to get both physical and digital releases

use the keyboard to navigate back and reset it. Press **ESC** to return to your game.

Note that most, but not all, single-player C64 games default to using Joystick 2.

07 Tweaking VICE

VICE x64 works smoothly on Raspberry Pi 400, but you'll want to delve into its settings menu to get the most out of it.

First, go to Video settings > Size settings and enable Fullscreen. To provide a bit more screen real estate, go to Video settings > VICII border mode and select None. Test your setup again with your favourite game or demo.

Getting rid of the border on a 1920×1080 display may result in some graphical overspill to the right-hand side on some programs that use unusual parallax scrolling tricks, but it's a very minor and generally invisible issue.

Once satisfied with your configuration, go to Settings management > Save current settings.

08 Homebrew haven

VICE supports vanilla PRG program files, D64 disk images, CRT cartridge images, and T64 and TAP tape images, among others. As usual in these tutorials, we largely recommend modern C64 software, which is often distributed digitally, as the easiest way to get top-notch free and commercial games for your emulated retro system. You can find an itch.io C64 collection at magpi.cc/itchc64.

Specialist publishers such as Psytronik (psytronik.itch.io), Double Sided Games

(doublesidedgames.com), Protovision (protovision.itch.io), and BitmapSoft (bitmapsoft.co.uk) also continue to put out spectacular commercial games for the C64.

Meanwhile, passion projects have seen authorised ports of games from other systems, such as Dungeoneer's C64 version of Richard 'Lord British' Garriott's Akalabeth (magpi.cc/akalabeth) and Double Sided's release of L'Abbaye Des Morts (magpi.cc/desmorts).

09 Using C64 Forever's RP9 files

If you wish to use the games that came with C64 Forever, you'll have to rename them from the default RP9 extension. On Raspberry Pi OS, you can use the rename command-line tool for this. Drop the entire zip files into your C64 games directory, open a Terminal in the directory, and enter these commands:

```
sudo apt install rename unzip
rename -v 's/\.rp9/\.zip/' *.rp9
```

Once renamed, you'll find that the zip files contain either D64 disk images or T64 tape images. Extracting them makes it easier to distinguish one format from the other.

```
unzip -o \*.zip
rm *xml *.txt *.png *.zip
```

Although many of the game files supplied are cracked versions, Cloanto makes it clear that licences to distribute the titles have been obtained from their original developers. Licensing details are provided in the Cloanto EULA. [\[M\]](#)

Top Tip

Demoscene conversion

Demos show off the C64's sound and graphics capabilities. Try The Elder Scrollers from magpi.cc/c64scrolls.

Add face recognition to your projects with Raspberry Pi

Buttons? Where's the fun in that? Why not trigger an event just by looking at a camera?



PJ Evans

[@MrPJEvans](#)

PJ Evans is a developer and wrangler of the Milton Keynes Raspberry Jam. He runs a LoRa gateway, which is probably the nearest he'll get to his own radio breakfast show.

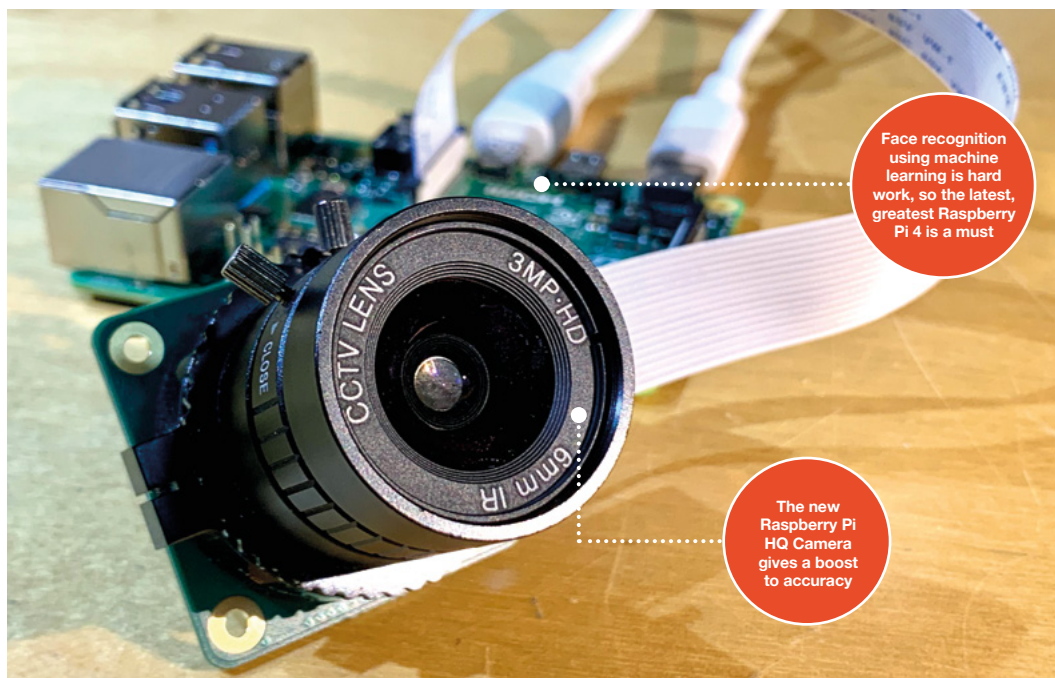
It's hard to comprehend how far machine learning has come in the past few years.

You can now use a sub-£50 computer to reliably recognise someone's face with surprising accuracy. Although this kind of computing power is normally out of reach of microcontrollers, adding a Raspberry Pi computer to your project with the new High Quality Camera opens up a range of possibilities. From simple alerting applications ('Mum's arrived home!'), to dynamically adjusting settings based on the person

using the project, there's a lot of fun to be had. Here's a beginner's guide to getting face recognition up and running.

STEP ONE PREPARE YOUR RASPBERRY PI

For face recognition to work well, we're going to need some horsepower, so we recommend a minimum of Raspberry Pi 3B+, ideally a Raspberry Pi 4. The extra memory will make all the difference. To keep as much resource as possible available for our project, we've gone for a Raspberry Pi OS Lite installation



YOU'LL NEED

- ◆ **Raspberry Pi 4B**
hsmag.cc/RaspPi4
- ◆ **Raspberry Pi High Quality Camera**
hsmag.cc/RaspPiHQ

with no desktop. Make sure you're on the network, have set a new password, enabled SSH if you need to, and updated everything with `sudo apt -y update` && `sudo apt -y full-upgrade`. Finally, go into settings by running `sudo raspi-config` and enable the camera in 'Interfacing Options'.

STEP TWO ATTACH THE CAMERA

This project will work well with the original Raspberry Pi Camera Module, but the new official HQ Camera will give you much better results. Be sure to connect the camera to your Raspberry Pi 4 with the power off. Connect the ribbon cable as instructed in hsmag.cc/HQCameraGetStarted. Once installed, boot up your Raspberry Pi 4 and test the camera is working. From the command line, run the following:

```
raspivid -o test.h264 -t 10000
```

This will record ten seconds of video to your microSD card. If you have an HDMI cable plugged in, you'll see what the camera can see in real-time. Take some time to make sure the focus is correct before proceeding.

STEP THREE INSTALL DEPENDENCIES

The facial recognition library we are using is one that has been maintained for many years by Adam Geitgey (github.com/ageitgey/face_recognition).

“

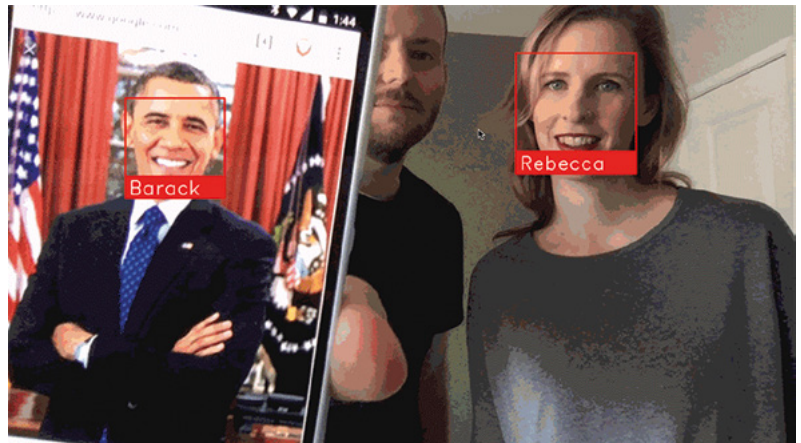
These libraries can also identify 'generic' faces, meaning it can detect whether a person is there or not

”

It contains many examples, including Python 3 bindings to make it really simple to build your own facial recognition applications. What is not so easy is the number of dependencies that need to be installed first. There are way too many to list here, and you probably won't want to type them out, so head over to hsmag.cc/FacialRec so that you can cut and paste the commands. This step will take a while to complete on a Raspberry Pi 4, and significantly longer on a Model 3 or earlier.

STEP FOUR INSTALL THE LIBRARIES

Now that we have everything in place, we can install Adam's applications and Python bindings with a simple, single command:



```
sudo pip3 install face_recognition
```

Once installed, there are some examples we can download to try everything out.

```
cd
git clone --single-branch https://github.com/ageitgey/face_recognition.git
```

In this repository is a range of examples showing the different ways the software can be used, including live video recognition. Feel free to explore and remix.

STEP FIVE EXAMPLE IMAGES

The examples come with a training image of Barack Obama. To run the example:

```
cd ./face_recognition/examples
python3 facerec_on_raspberry_pi.py
```

On your smartphone, find an image of Obama using your favourite search engine and point it at the camera. Providing focus and light are good, you will see:

"I see someone named Barack Obama!"

If you see a message saying it can't recognise the face, then try a different image or try to improve the lighting if you can. Also, check the focus for the camera and make sure the distance between the image and camera is correct.

STEP SIX TRAINING TIME

The final step is to start recognising your own faces. Create a directory and, in it, place some good-quality passport-style photos of yourself or those you want to recognise. You can then edit the `facerec_on_raspberry_pi.py` script to use those files instead. You've now got a robust prototype of face recognition. This is just the beginning. These libraries can also identify 'generic' faces, meaning it can detect whether a person is there or not, and identify features such as the eyes, nose, and mouth. There's a world of possibilities available, starting with these simple scripts. Have fun. □

Above ♦
Who are you? What even is a name? Can a computer decide your identity?

QUICK TIP

Don't worry if you don't have a Raspberry Pi High Quality Camera – you can use the original or any compatible USB camera.

CODE

hsmag.cc/GitHubFR



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

Cheap Trills for all

Use Trill sensors to touch your life, and projects, thanks to Raspberry Pi



Mike Cook

MAKER

Veteran magazine author from the old days, writer of the *Body Build* series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

The Trill boards began as a Kickstarter campaign from Bela, an offshoot of a research team in the Centre for Digital Music at Queen Mary University of London.

They started out with a highly optimised sound generating board for the BeagleBone Black board, and then developed the Trill sensors to accompany that system. However, they soon realised these little boards were useful for all makers, so let's meet the Trills. This month we will look at communicating with the Trills, and then look into projects using them in the following instalments. They are normally designed to be attached to the Arduino class of boards, and also they can be used on a Raspberry Pi computer with a C++ program. But here, we will show you how to use Python with the sensors.

01 What's a Trill

Here at the Bakery, we can remember when Trill was a big brand budgerigar box of seeds, with constant TV advertising and the slogan 'Trill makes your budgies bounce with health'. It also has the musical meaning of a rapid repeat of alternating notes. But here it is a touch sensor designed for adding to a computer to input tactile information. They come in a variety of shapes: bar, ring, square, hexagon, and a DIY craft version. They can detect not only touch, but also multiple touches, and where on the tile they were made.

02 Python






It would be good if Trill sensors could be incorporated into a Python program, but there is no Python library to do this, which is why we wrote one. Bela had released an open-source C++ class to read them, so 'all' we had to do was to convert it into a Python class. This sounds a lot easier

to do than it was, but we eventually produced a very close replacement to the original C++ class. This allows us to write programs that use the Trill boards with the minimum of complexity, and in a simpler way than the original C++ class.

03 Connector mix

Each Trill board contains a powerful Cypress CY8C20XX6A/S microcontroller, preprogrammed to act as a touch sensor for the shape of board it is on. Even the source code for the microcontroller's firmware is open-source. Each board, except the Trill craft, comes with Grove-style I2C connectors that finish in four individual male DuPont connectors. The Trill Craft board has a different connector to the other boards; this uses a right-angled row of pins, and the best way of connecting to that is to use a short length of four-way ribbon connector with sockets on both ends.

Figure 1

| | Trill Type | Default Address |
|--------|---------------------------------------------------------------------------------------|-----------------|
| Bar |  | 0x20 |
| Square |  | 0x28 |
| Craft |  | 0x30 |
| Ring |  | 0x38 |
| Hex |  | 0x40 |

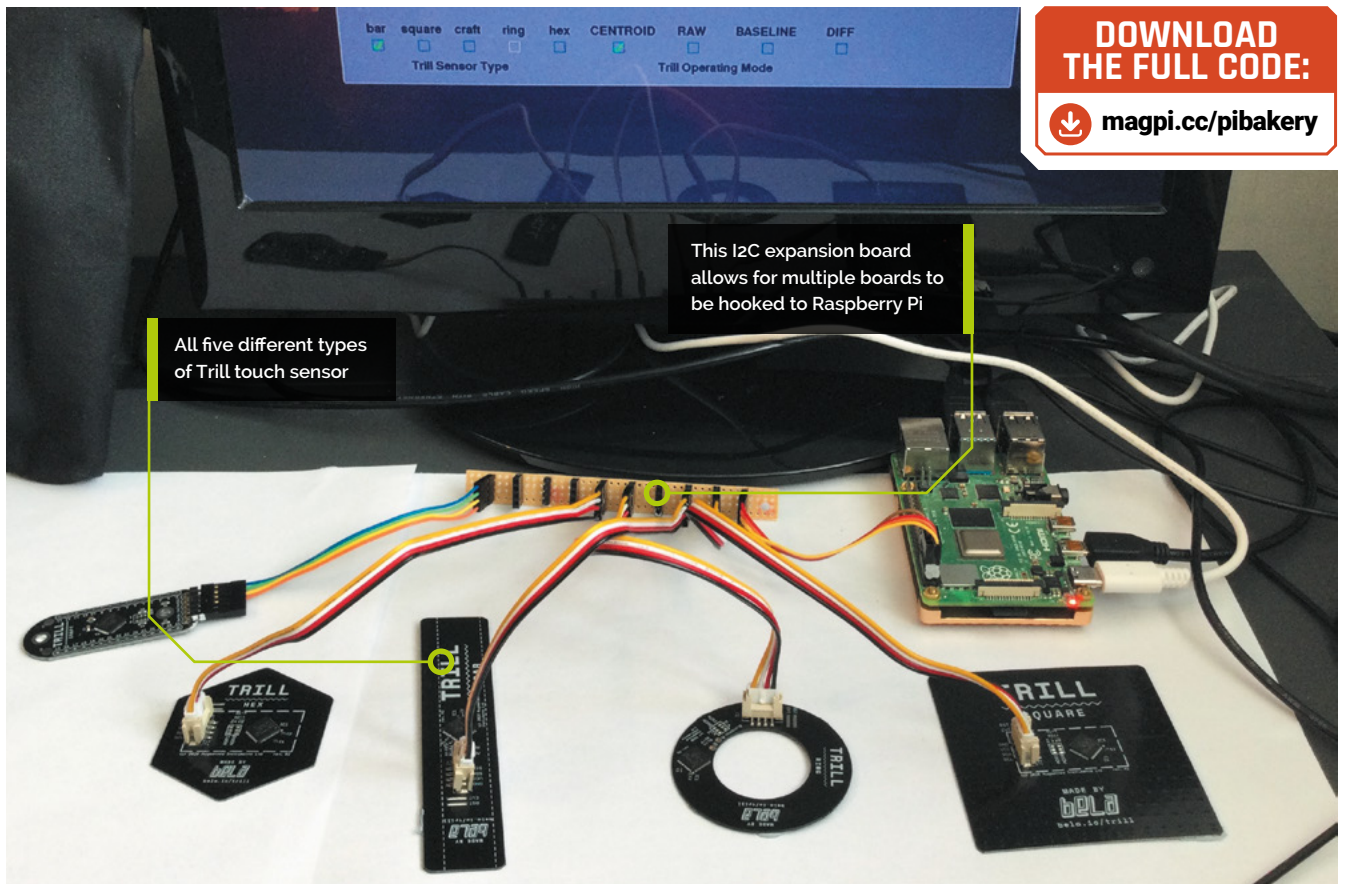
▲ **Figure 1** Default address for each Trill touch sensor type

You'll Need

- ▶ One or more Trill sensors magpi.cc/trill
- ▶ Make your own I2C expander or buy magpi.cc/trillhub

DOWNLOAD THE FULL CODE:

magpi.cc/pibakery



All five different types of Trill touch sensor

This I2C expansion board allows for multiple boards to be hooked to Raspberry Pi

04 The interface

All Trill boards communicate over the I2C bus and have fixed address ranges, for each board type, in blocks of eight. So for five board types, that is 40 boards you could attach. **Figure 1** shows the default address for each type of board, and **Figure 2** shows what solder links to make to the back of the board to change the default address. There could be nine of any one Trill type, but that reduces the maximum number of the next type to just seven. If that is not enough for you, then you can use the little utilised I2C bus 0, for another 39 Trills.

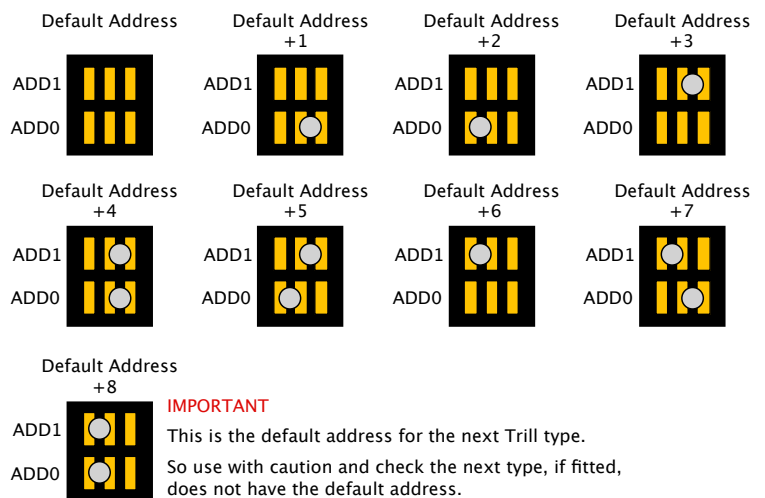
05 Connecting the Trills

The Trill boards need to be connected to the I2C bus. While Bela does sell a Hub to allow connection to ten boards, we decided to make our own. We made a nine-way I2C expander board, limited only by the longest length of stripboard we had to hand. We used a four-way DuPont pin header on the left, for connection to the GPIO pins, followed by a mixture of four-way DuPont pins or sockets (**Figure 3**, overleaf). Note that if you are using I2C bus 0, you will also need two 1K8 (1.8 kΩ) pull-up resistors on this board, mounted end on.

06 Connectors

Figure 4 (overleaf) shows a photograph of our I2C expander board. Note how you can't see the far end of each connector. This has a white mark to show you what way round to plug them. We used typing correction fluid because it dries quickly.

Figure 2



▲ **Figure 2** Changing the default I2C address on a Trill sensor

Figure 3

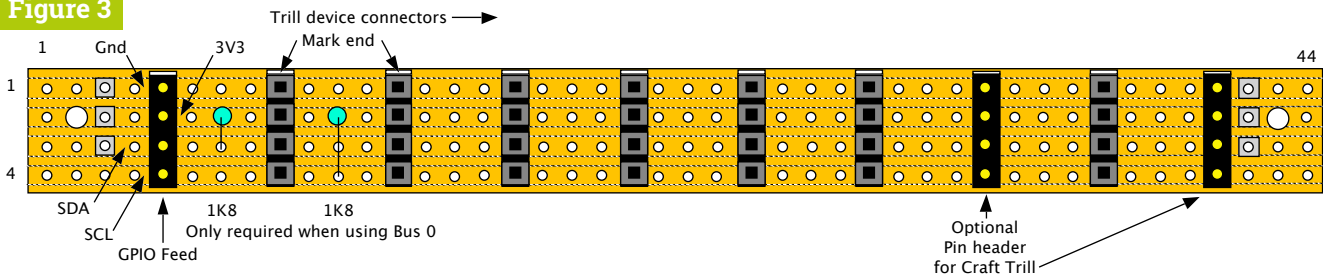
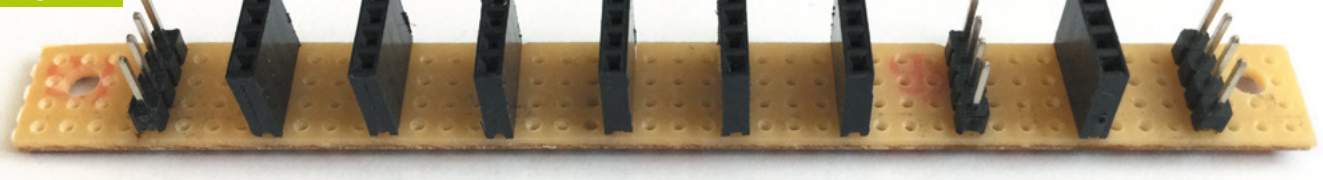


Figure 4



▲ Figure 3 Physical layout of the I2C expander board

▲ Figure 4 Photograph of I2C expander board

While you can plug the individual wires from the Grove connector into the sockets, you get a much more reliable connection if you glue them together into one strip (Figure 5). Plug them into a socket as the glue dries to make sure they are aligned correctly. Mark the connector with the black wire white, and match this to the white mark on the expander board.

07 Connection to the GPIO

All that is left to do is to connect the I2C expander board to the Raspberry Pi computer's GPIO pins. Figure 6 shows this for using the normal I2C bus 1. We used an eight-way two-row connector and made a lead to connect to the pins at the left-hand side of the expander board. If you want to use I2C bus 0, Figure 7, then we recommend using a 32 by two-row connector over the GPIO pins, for ease of use. You could use individual socketed flying leads, but you would have to check the connections carefully every time you wanted to attach your I2C expander board.

08 Testing the sensors

On our GitHub page (magpi.cc/pibakery) you will find all the software. The class file is called **trill_lippy** and should be in the same directory as the program using it. As a first test, attach one Trill sensor, boot up, and load in the **trill_testy** file. Then uncomment one line that corresponds to your sensor shape. This creates an instance of the class using bus 1 at the default address, for that shaped board. Run this, then look at the numbers streaming out: see how they change as you touch the sensor. If you don't see this then check your wiring.

09 Understanding the modes

Each Trill can be used in a number of different modes, of which CENTROID is the most useful as this returns things like the number of touches detected, as well as their size and position. However, you can also access the data, under the hood, that makes up those values. The RAW mode, as its name implies, is the direct reading from each capacitive sensor. The BASELINE is the background reading from each sensor with no touches, and the DIFF mode is the difference between the raw data subtracted from the baseline reading.

10 Visualising the data

In order to help you visualise the data, we have written a program called **graphic_trillion** which lets you see the touch data on the background of the board's shape. It first scans all possible Trill addresses on I2C bus 1 and compiles a list of all sensors found. Then the boxes for selecting that sensor type are enabled. Clicking on a sensor will select that type. If you click on a sensor and that sensor is already being displayed, but more than one sensor of that type is connected, then the next address of that sensor type will be selected.



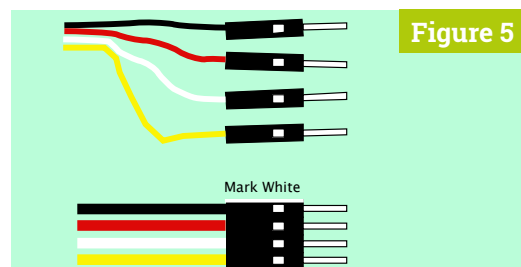
Warning!
I2C devices are not hot-pluggable

Never make any changes in connections to the GPIO pins while power is applied to your Raspberry Pi.

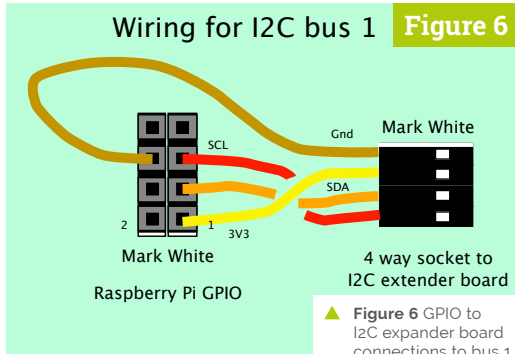
Top Tip

I2C bus driver

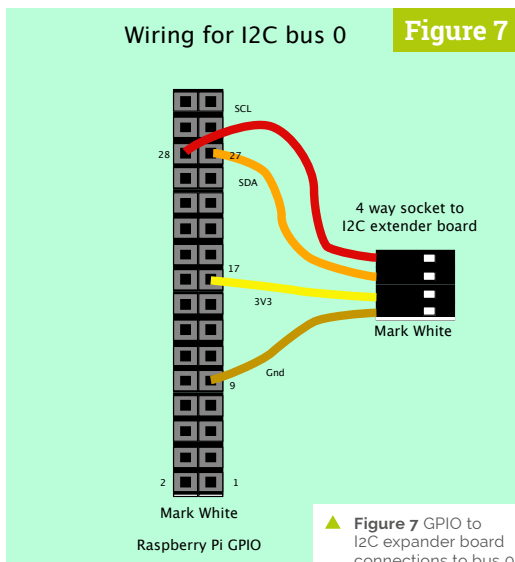
The class library uses the **simbus2** driver. If not installed already, you need to type **pip3 install simbus2** in a Terminal window to install it.



▲ Figure 5 Gluing floating connectors together



▲ Figure 6 GPIO to I2C expander board connections to bus 1



▲ Figure 7 GPIO to I2C expander board connections to bus 0

11 Options

There are a number of options that can be tweaked to obtain more reliable or faster readings from the Trill sensor. **Figure 8** shows the screen for a square type, and you can see the touch. On the right-hand side, there are a number of options for setting up the sensors. These are best adjusted while looking at the readings in a data mode.

“ The BASELINE is the background reading from each sensor with no touches ”

Then go back to the CENTROID mode to see what effects it has on the touch information. **Figure 9** shows the Trill craft in DIFF mode. Note here the individual sensors have been highlighted with a colour corresponding to each data reading.

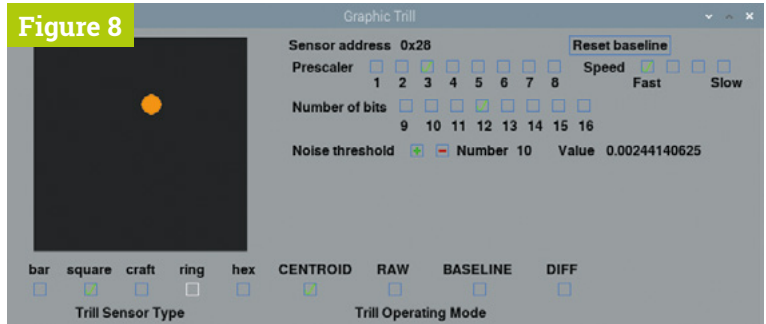


Figure 8

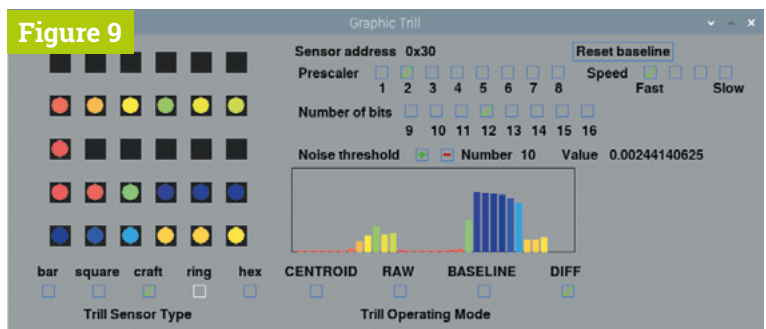


Figure 9

12 Python vs C++

We did a number of things differently in the Python class library to the original C++ library, mainly in the area of device discovery. Our take is that when you are doing a project, you know what sensors you have and what their address is. So we simplified the number of ways you could discover what devices were connected if you ever needed to do that. If an address of 256 and a Trill type of 'none' is used, then the instance would not be fully initialised and verified, so you would have to do that 'by hand'. The `trill_scanty` program illustrates this process.

13 Finally

Note that when using a two-dimensional sensor, like the square and the hex, it should be considered as a single touch sensor only. While it can detect multiple touches, it does so by working out the number of horizontal and vertical touches detected; there can be a different number of these, and it can't pair them up. However, more advanced tracking algorithms could be devised for things like gesture recognition.

So, now you have enough information to start using the Trill touch sensors. Next month we will see how to mount these sensors into a project, and look at some of the things you can do with them. 📄

▲ Figure 8 A single touch on the Square Trill sensor in the CENTROID mode

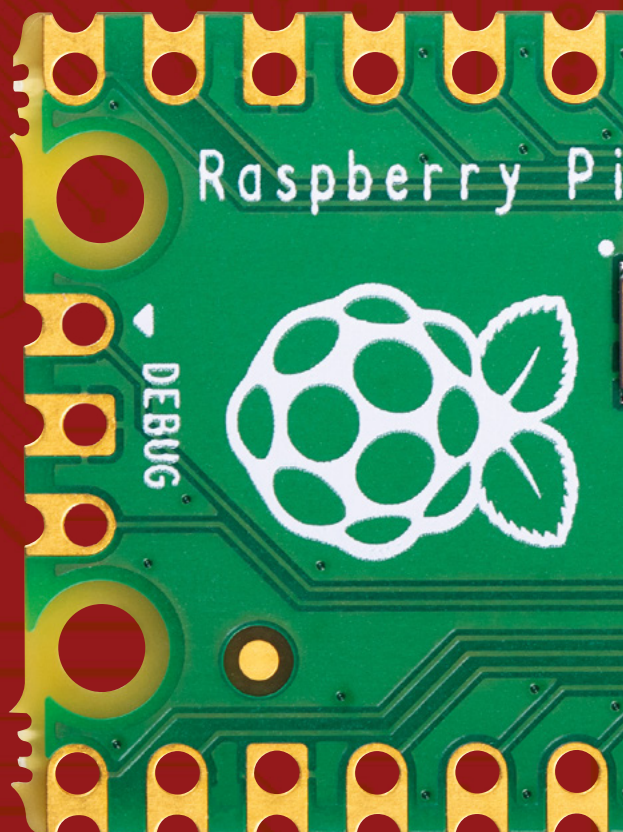
▲ Figure 9 A double touch for a Craft Trill sensor in the DIFF mode

Top Tip

Sources

You can also get Trill sensors from online retailers like Pimoroni, which might offer a faster service to your locality or, if you are buying something else from them, save on postage. See magpi.cc/trillsellers.

INTRODUCING RASPBERRY PI PICO



Take a dive into Raspberry Pi Pico and its brand-new custom-built RP2040 microcontroller. By **Gareth Halfacree**

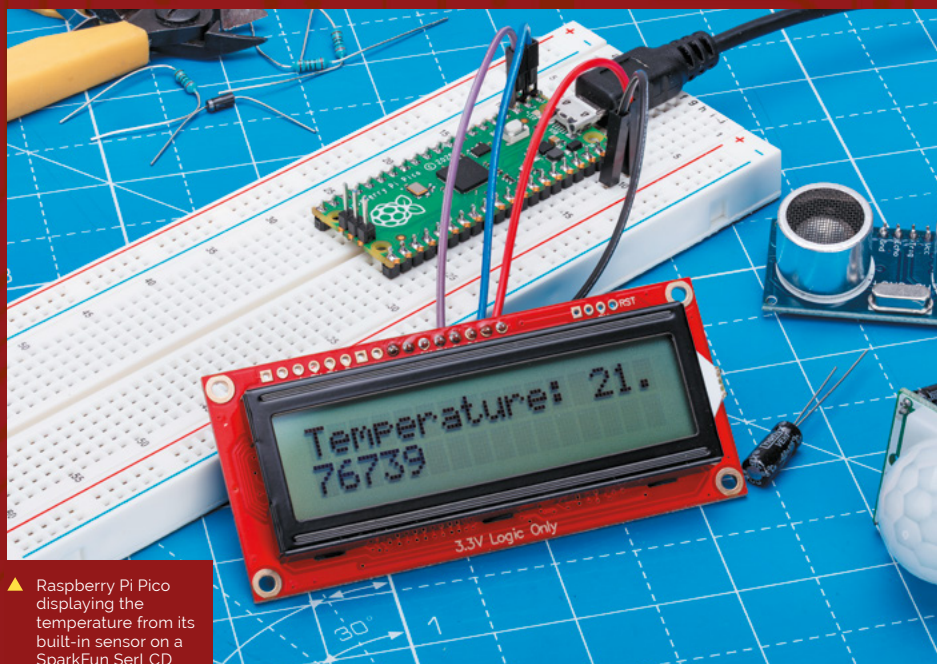
Raspberry Pi Pico is a microcontroller board built on silicon designed by Raspberry Pi.

Microcontrollers are computers stripped back to their bare essentials. You don't use monitors or keyboards, but program them to take their input from, and send their output to the input/output pins. Using these programmable connections, you can light lights, make noises, send text to screens, and much more.

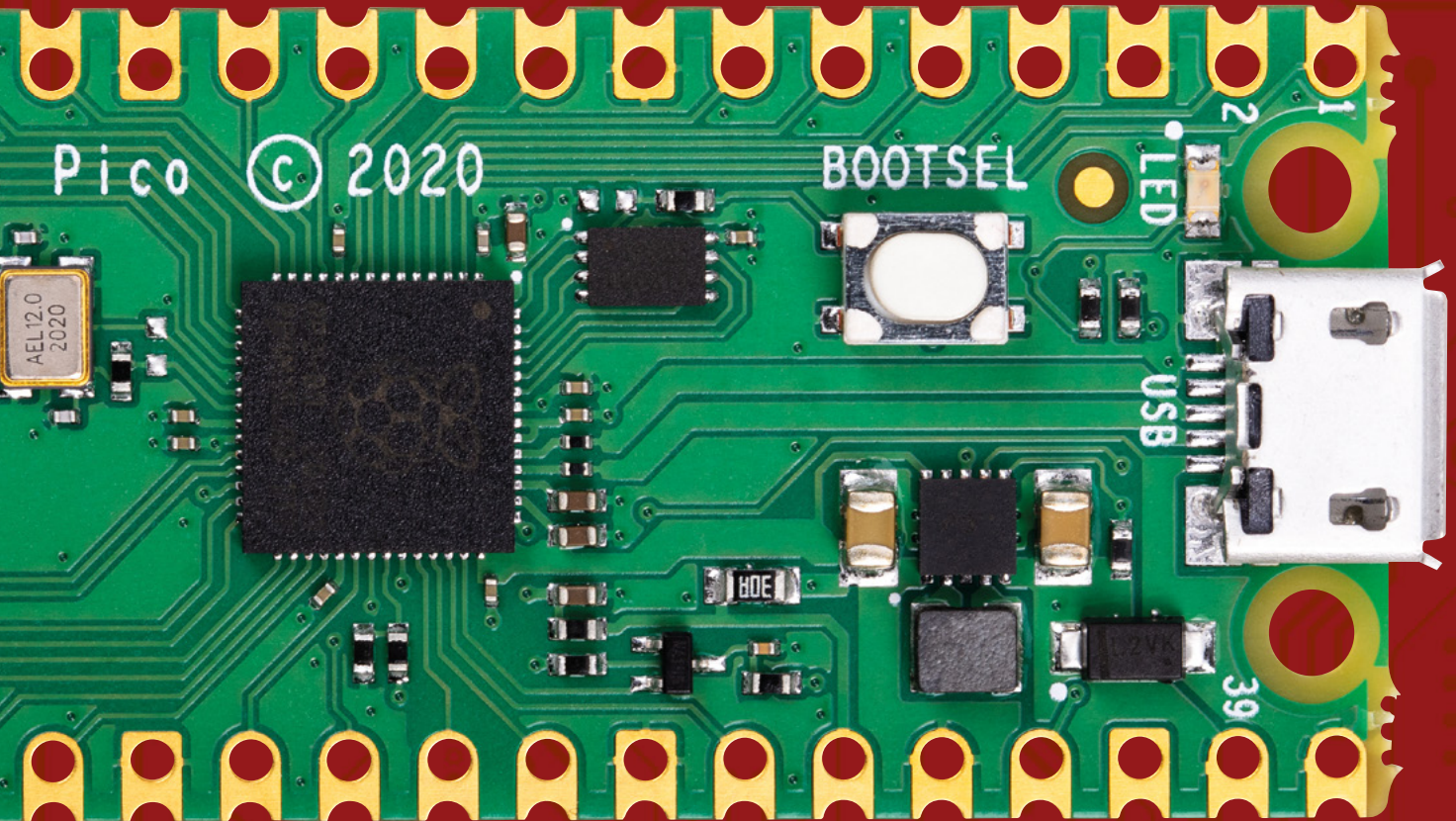
The ultra-light, ultra-small Raspberry Pi Pico microcontroller board is ideal for embedding inside digital projects. Raspberry Pi Pico represents two major firsts for Raspberry Pi: it's the first microcontroller development board from Raspberry Pi; it's also the first device to use a silicon chip designed by Raspberry Pi's in-house Application-Specific Integrated Circuit (ASIC) team.

Raspberry Pi Pico is a development board built around this powerful yet low-cost RP2040 microcontroller.

Like Raspberry Pi computers, Raspberry Pi Pico features a pin header with 40 connections,



▲ Raspberry Pi Pico displaying the temperature from its built-in sensor on a SparkFun SerLCD

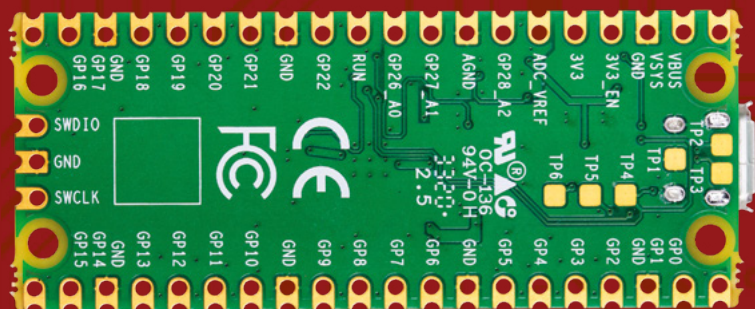


along with a new debug connection enabling you to analyse your programs directly from another computer (typically by connecting it directly to the GPIO pins on a Raspberry Pi).

▣ Raspberry Pi Pico represents the start of a new era for Raspberry Pi ▣

Pico is an incredibly interesting new device from Raspberry Pi. It offers a wealth of connectivity for external hardware – and enough processing power to handle complex tasks. All this in a compact board which costs less than a cup of coffee. You can pick up a Pico from just \$4 / £3.60, or free on the latest edition of HackSpace magazine (see ‘Get your free Pico’).

Built with everyone from absolute beginners to professional engineers in mind, Raspberry Pi Pico represents the start of a new era for Raspberry Pi. We can’t wait to see what you all make with it.



Get your free Pico

The best place to get Raspberry Pi Pico at launch is free with *The MagPi*'s sister publication, HackSpace magazine issue #39. You can buy a copy of HackSpace magazine online from our Raspberry Pi Press store (magpi.cc/store) or WHSmith and other newsagents. Raspberry Pi Pico is also sold by Raspberry Pi resellers around the globe. For a full list, see the Raspberry Pi Pico product page on Raspberry Pi's website (magpi.cc/pico).

GET TO KNOW RASPBERRY PI PICO

Take a tour of Raspberry Pi Pico and the RP2040 chip which drives it

Raspberry Pi Pico is a brand new, low-cost, yet highly flexible development board designed around a custom-built RP2040 microcontroller chip designed by Raspberry Pi.

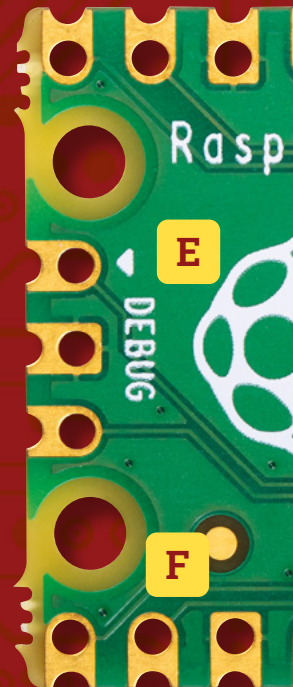
Raspberry Pi Pico – ‘Pico’ for short – features a dual-core Cortex-M0+ processor (the most energy-efficient Arm processor available), 264kB of SRAM, 2MB of flash storage, USB 1.1 with device and host support, and a wide range of flexible I/O options.

The castellated pin headers ensure Pico is equally at home on a breadboard for experimentation as it is soldered onto a circuit

board and driving a finished product. The high-performance processor cores coupled with RAM and storage give it impressive flexibility.

A real highlight comes in the form of Programmable Input/Output (PIO) capabilities: bridging the gap between software and hardware, Pico’s PIO allows developers to define new hardware features in software – expanding its capabilities beyond any fixed-function device.

Pico is set to prove itself not just an impressive new tool for Raspberry Pi users, but a must-have gadget for anyone investigating physical computing projects.



Specifications

- ▶ RP2040 microcontroller chip designed by Raspberry Pi in the United Kingdom
- ▶ Dual-core ARM Cortex-M0+ processor, flexible clock running up to 133MHz
- ▶ 264kB of SRAM, and 2MB of on-board flash storage
- ▶ Castellated module allows soldering direct to carrier boards
- ▶ USB 1.1 Host and Device support
- ▶ Low-power sleep and dormant modes
- ▶ Drag & drop programming using mass storage over USB
- ▶ 26 multifunction GPIO pins
- ▶ 2x SPI, 2x I2C, 2x UART, 3x 12-bit ADC, 16x controllable PWM channels
- ▶ Accurate clock and timer on-chip
- ▶ Temperature sensor
- ▶ Fast floating-point libraries in ROM
- ▶ 8x Programmable IO (PIO) state machines for custom peripheral support

C | BOOTSEL

Hold the BOOTSEL (boot select) button when powering up Pico to put it into USB Mass Storage Mode. From here, you can drag-and-drop programs, created with C or MicroPython, into the RPI-RP2 mounted drive. Pico runs the program as soon as it is switched on (without BOOTSEL held down)



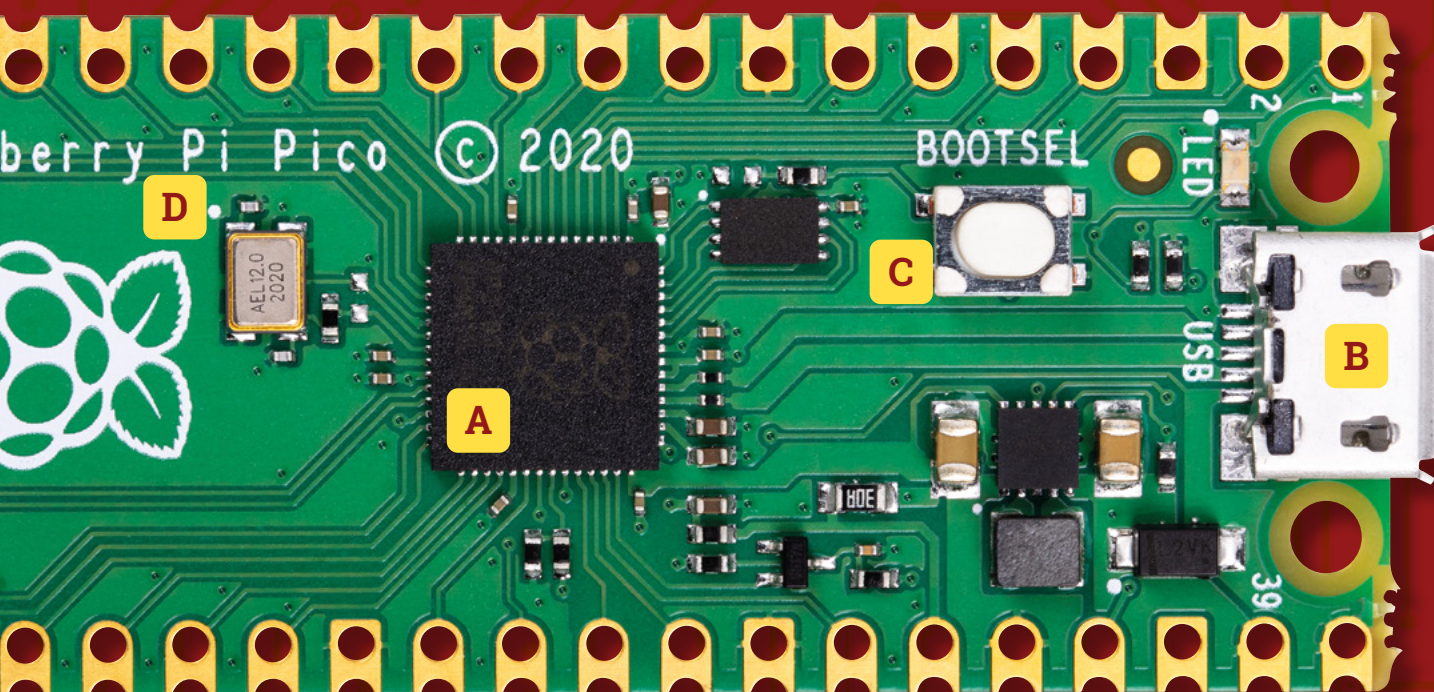
A | MICROCONTROLLER

RP2040 is a custom-built dual-core microcontroller, designed in-house at Raspberry Pi



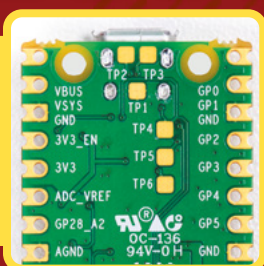
B | USB

A micro USB port provides power and data, letting you communicate with and program Raspberry Pi Pico (by dragging and dropping a file)



D | LABELLING

Silkscreen labelling on the top provides orientation for the 40 pins, while a full pinout is printed on the rear (pictured)



E | DEBUGGING

A Serial Wire Debug (SWD) header provides hardware debugging capabilities, letting you quickly track down problems in your programs



F | PINS

Raspberry Pi Pico's pins are castellated, allowing pin headers to be fitted for breadboard use or the entire board to be soldered as a flat module



SAY HELLO TO RP2040

What is a microcontroller? Get to know the technology inside Raspberry Pi's home-made silicon

RP2040 is a low-cost microcontroller device, with the same focus on quality, cost, and simplicity that characterises the 'big' Raspberry Pi. Microcontrollers interact with the hardware of a board much like an application processor does in a larger device.

Application processors like the Broadcom BCM2711 used in Raspberry Pi 4 are designed to run multiple programs under an operating system, like Raspberry Pi OS. These programs access external hardware through interfaces provided by the operating system.

In contrast, microcontrollers like RP2040 interact directly with external hardware and typically run a single program from the moment you turn them on.

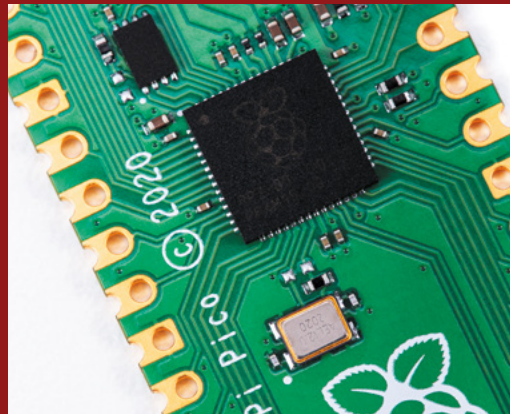
Just as Raspberry Pi is an accessible computer, RP2040 is an accessible microcontroller, containing almost everything makers need to embed it inside a product.

RP2040 is supported by both C/C++ and MicroPython cross-platform development environments, including easy access to runtime debugging. It has a built-in UF2 bootloader enabling programs to be loaded by drag-and-drop. The built-in USB can act as both device and host. Meanwhile, floating-point routines are baked into the chip for ultra-fast performance. It has two symmetric processor cores and high internal

bandwidth, making it useful for signal processing and video applications. The chip has a relatively large amount of internal RAM but uses external flash storage, allowing you to choose how much memory you need.

“ Microcontrollers are an exciting new area for Raspberry Pi fans to explore ”

Microcontrollers are an exciting new area for Raspberry Pi fans to explore. See the RP2040 data sheet for more information (magpi.cc/2040data).

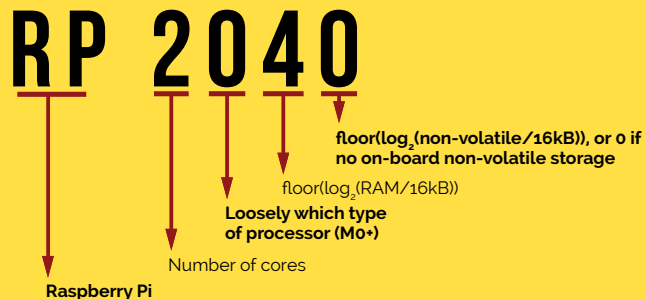


- ◀ The heart of Raspberry Pi Pico, RP2040 is a microcontroller with some unique functionality
- ▶ The significance of the letters and digits in RP2040's name

Behind the name 2040

The post-fix numeral on RP2040 comes from the following:

1. Number of processor cores (2)
2. Loosely which type of processor (M0+)
3. The amount of RAM, from the function $\text{floor}(\log_2(\text{RAM} / 16\text{kB}))$; in this case it's 256kB
4. The amount of non-volatile storage, from the function $\text{floor}(\log_2(\text{non-volatile} / 16\text{kB}))$, or 0 if no on-board non-volatile storage



PROGRAMMING RASPBERRY PI PICO

Raspberry Pi Pico is designed for everyone, from complete beginners to engineers

Pico is programmed using either C/C++ or MicroPython and there is IDE support for Visual Studio Code and Eclipse. Adding a program to Pico is as easy as dragging and dropping a file while Raspberry Pi Pico is in boot mode.

MicroPython is an implementation of the Python programming language that is already popular among Raspberry Pi users. MicroPython is built specifically for microcontrollers like the RP2040 that powers Raspberry Pi Pico.

MicroPython offers the same friendly syntax as Python. It allows for full control over Raspberry Pi Pico's various features, including its Programmable Input/Output (PIO) functionality.

Programs written for other MicroPython-compatible microcontroller boards will work on Raspberry Pi Pico, and vice versa – sometimes needing minor modification for different features between boards – giving Raspberry Pi Pico a

healthy library of projects and tutorials beyond those developed by Raspberry Pi itself.

Meanwhile, the C/C++ SDK is fine-tuned to RP2040 and has all the headers, libraries, and build systems necessary to write programs in C, C++, or assembly language. Additionally, the C/C++ SDK provides higher-level libraries for dealing with timers, USB, synchronisation, and multicore programming, along with additional high-level functionality built using PIO such as audio.

Beginners looking to get started with the MicroPython port should start with the Raspberry Pi Pico Python SDK documentation and be sure to pick up a copy of *Getting Started with MicroPython on Raspberry Pi Pico* and read the tutorial on the following pages.

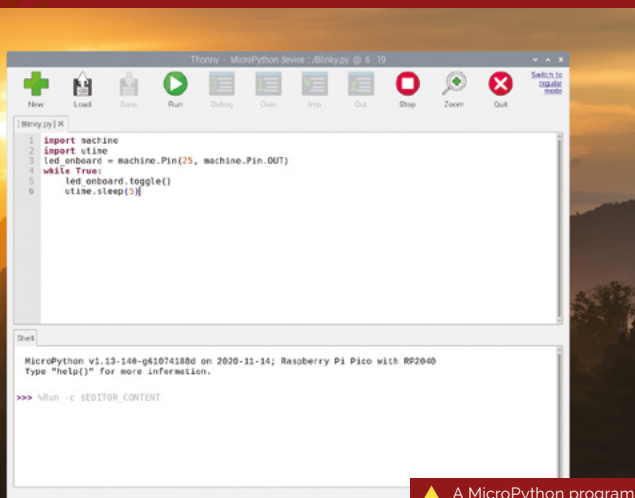
Makers looking to explore the C/C++ SDK should download the Pico C/C++ SDK documentation (magpi.cc/picocsdk).

Raspberry Pi data sheets

Make sure to read, and bookmark, these new Raspberry Pi Pico and 2040 data sheets.

- ▶ Pico Python SDK
magpi.cc/picopythonsdk
- ▶ Pico C/C++ SDK
magpi.cc/picocsdk
- ▶ Raspberry Pi Pico data sheet
magpi.cc/picodatasheet
- ▶ RP2040 data sheet
magpi.cc/2040datasheet

Pico Python SDK
A MicroPython environment
for the RP2040 microcontroller



▲ A MicroPython program that blinks Raspberry Pi Pico's LED on and off

JAMES ADAMS AND NICK FRANCIS ON RASPBERRY PI PICO



James Adams

James Adams, Chief Operating Officer at Raspberry Pi



Nick Francis

Nick Francis, Senior Engineering Manager at Raspberry Pi

What goes into making Raspberry Pi's first in-house microcontroller and development board

"It's a flexible product and platform," says Nick Francis, Senior Engineering Manager at Raspberry Pi, when discussing the work the Application-Specific Integrated Circuit (ASIC) team put into designing RP2040, the microcontroller at the heart of Raspberry Pi Pico.

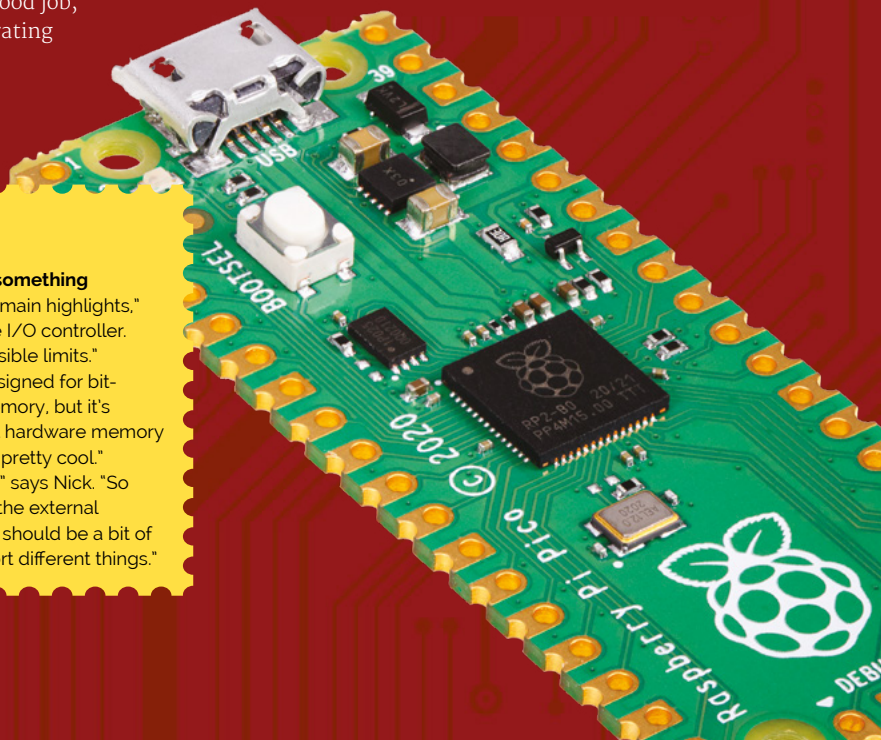
It would have been easy to have said well, let's do a purely educational microcontroller "quite low-level, quite limited performance," he tells us. "But we've done the high-performance thing without forgetting about making it easy to use for beginners. To do that at this price point is really good."

"I think we've done a pretty good job," agrees James Adams, Chief Operating Officer at Raspberry Pi. "We've obviously tossed around a lot of different ideas about what we could include along the way,

and we've iterated quite a lot and got down to a good set of features."

A board and chip

"The idea is it's [Pico] a component in itself," says James. "The intent was to expose as many of the I/O (input/output) pins for users as possible, and expose them in the DIP-like (Dual Inline Package) form factor, so you can use Raspberry Pi Pico as you might use an old 40-pin DIP chip. Now, Pico is 2.54 millimetres or 0.1 inch pitch wider than a 'standard' 40-pin DIP, so not exactly the same, but still very similar.



Understanding PIO

Programmable Input/Output (PIO) is a key feature of RP2040 and something we're very excited about. "The Programmable IO block is one of the main highlights," says Nick. "It's a very flexible multi-configuration, multi-state-machine I/O controller. Essentially, we can interface to anything you can think of – within sensible limits."

"Maybe the way to describe it is: it's a custom processor, a core designed for bit-banging input/output [I/O]," James adds. "It's got instructions and memory, but it's highly specialised to do I/O bit-banging. It has FIFOs – first in, first out hardware memory buffers – to stuff data in, shuffle data in and out, and stuff like that. It's pretty cool."

"It gets whoever's going to use the device closer to their hardware," says Nick. "So you get to really, really think about the bits and how you're talking to the external devices. The list of interfaces it will support will grow over time, and it should be a bit of interesting fun for people to try and hook them up and make it support different things."

“After the first prototype, I changed the pins to be castellated so you can solder it down as a module, without needing to put any headers in. Which is, yes, another nod to using it as a component.”

Getting the price right

“One of the things that we’re very excited about is the price,” says James. “We’re able to make these available cheap as chips – for less than the price of a cup of coffee.”

“It’s extremely low-cost,” Nick agrees. “One of the driving requirements right at the start was to build a very low-cost chip, but which also had good performance. Typically, you’d expect a microcontroller with this specification to be more expensive, or one at this price to have a lower specification. We tried to push the performance and keep the cost down.”

“ We’re able to make these available cheap as chips ”

Raspberry Pi Pico also fits nicely into the Raspberry Pi ecosystem: “Most people are doing a lot of the software development for this, the SDK (software development kit) and all the rest of it, on Raspberry Pi 4 or Raspberry Pi 400,” James explains. “That’s our primary platform of choice. Of course, we’ll make it work on everything else as well. I would hope that will be as easy to use as any other microcontroller platform out there.”



▲ A reel of Raspberry Pi Pico microcontroller boards ready for distribution

Eben Upton on RP2040

“RP2040 is an exciting development for Raspberry Pi because it’s Raspberry Pi people making silicon,” says Eben Upton, CEO and co-founder of Raspberry Pi. “I don’t think other people bring their A-game to making microcontrollers; this team really brought its A-game. I think it’s just beautiful.”

“What does Raspberry Pi do? Well, we make products which are high performance, which are cost-effective, and which are implemented with insanely high levels of engineering attention to detail – and this is that. This is that ethos, in the microcontroller space. And that couldn’t have been done with anyone else’s silicon.”

PROGRAM RASPBERRY PI PICO WITH MICROPYTHON

Take your first steps with Raspberry Pi Pico and learn to program your new microcontroller. **By Gareth Halfacree**

Raspberry Pi Pico is set up, by default, for use with the C/C++ Software Development Kit (SDK). The C/C++ SDK is an extremely flexible and powerful way to interact with your Raspberry Pi Pico. However, there's a more beginner-friendly method: MicroPython, a port of the Python programming language designed specifically for microcontrollers.

In this tutorial we're going to switch the Pico firmware from C/C++ to MicroPython and create our first program, which flashes the LED on the board.

In the next tutorial we will demonstrate how to attach pins and wire up your Pico so it's ready for you to explore a wealth of electronic projects.

01 Open in boot mode

Take your Raspberry Pi Pico and a micro USB to USB-A cable, and connect the small micro USB

end of Pico. Hold down the small button on your Raspberry Pi Pico marked 'BOOTSEL' and plug the larger USB-A cable end into your computer (we are using a Raspberry Pi).

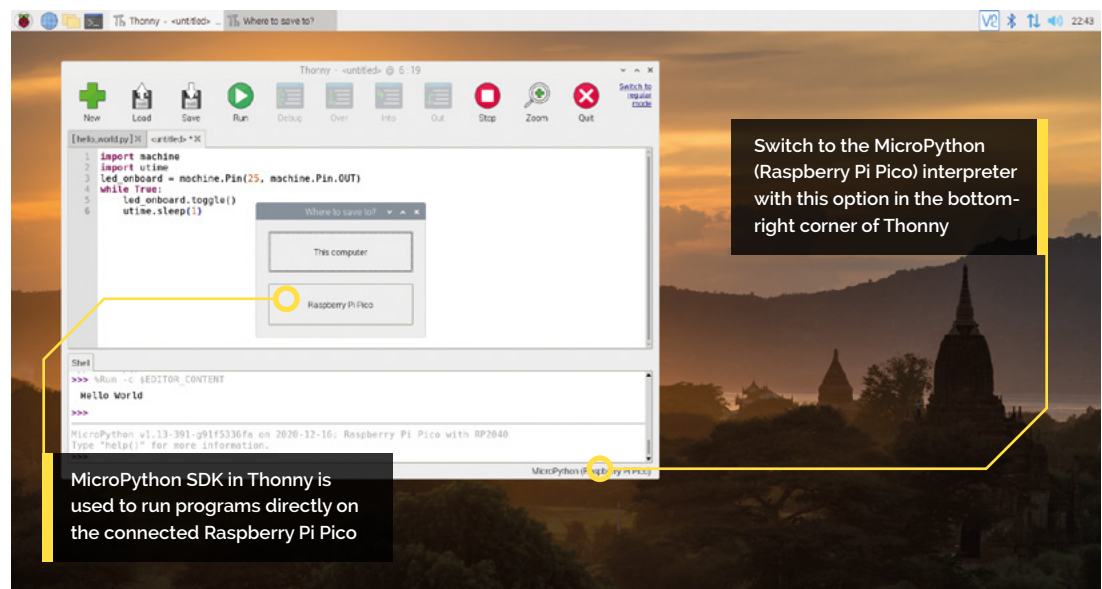
Wait a few seconds, then let go of the BOOTSEL button. You will see your computer mount a removable drive. Click OK in the 'Removable medium is inserted' window to open Raspberry Pi Pico's on-board storage.

02 Flash the MicroPython firmware

Double-click the **INDEX.HTM** file displayed in Pico's mounted storage. Your browser will open and display the 'Welcome to your Raspberry Pi Pico' webpage. Choose the 'Getting started with MicroPython' tab, and click 'Download UF2 file' to

You'll Need

- ▶ Raspberry Pi computer
magpi.cc/products
- ▶ Raspberry Pi Pico
magpi.cc/pico
- ▶ MicroPython SDK
magpi.cc/picopythonsdk
- ▶ Internet connection



MicroPython SDK in Thonny is used to run programs directly on the connected Raspberry Pi Pico

Switch to the MicroPython (Raspberry Pi Pico) interpreter with this option in the bottom-right corner of Thonny

download the MicroPython firmware. It's a small file, so it'll only take a few seconds.

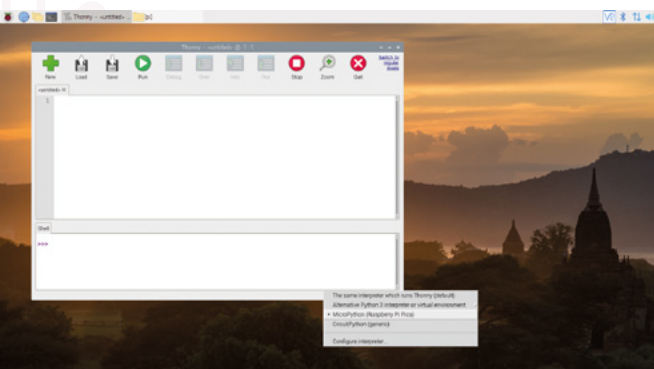
Open File Manager and locate the **micropython-16-DEC-2020.uf2** file in the **Downloads** folder (the file name may have been updated with a later date). Drag-and-drop the UF2 file to the Raspberry Pi Pico's removable drive (named 'RPI-RP2'). After a few seconds, the drive will disappear as the new MicroPython firmware is recognised and installed.

03 Switching the back end

The best way to program in MicroPython on your Raspberry Pi Pico is with the Thonny Python IDE (integrated development environment). Open the Raspberry Pi menu and choose Programming > Thonny Python IDE.

Thonny is normally used to write programs that run on the same computer you're using Thonny on; to switch to writing programs on your Raspberry Pi Pico, you'll need to choose a new Python interpreter. Look at the bottom-right of the Thonny window for the word 'Python' followed by a version number: that's your current interpreter.

Click 'Python' and look through the list that appears for 'MicroPython (Raspberry Pi Pico)' – or, if you're running an older version of Thonny, 'MicroPython (generic)'.



04 Hello World!

Writing a program for your Raspberry Pi Pico is a lot like writing a program for your Raspberry Pi. You can type commands in the Shell at the bottom of the window to have them immediately executed, or you can write a program in the main part of the window to run on-demand.

Click in the Shell area, next to the `>>>>` symbols, and type:

```
print("Hello, World!")
```

blinky_led.py

> Language: **MicroPython**

DOWNLOAD THE FULL CODE:


 magpi.cc/github

```
001. import machine
002. import utime
003. led_onboard = machine.Pin(25, machine.Pin.OUT)
004. while True:
005.     led_onboard.toggle()
006.     utime.sleep(1)
```

When you press **ENTER** at the end of the line, you'll see your Raspberry Pi Pico respond. Try typing the same line again, but in the main part of the Thonny window – then click the Run icon. You'll be asked whether you want to save your program to 'This computer' or 'Raspberry Pi Pico'. Click on 'Raspberry Pi Pico', give your program the name **hello_world.py**, then click OK to save and run your first program.

05 Blinky LED

While Raspberry Pi Pico can run Python programs like the one above, its true power comes from interfacing with external hardware like buttons and LEDs. You can start programming a physical computing project without any extra hardware, too, thanks to an on-board LED (assigned to the non-broken-out GP25 pin).

Click the New icon and type in the **blinky_led.py** code. Click Run, save the program to your Raspberry Pi Pico, and watch the LED on Raspberry Pi Pico: it will turn on for one second, then off for one second, then repeat. 

Top Tip

Update Thonny

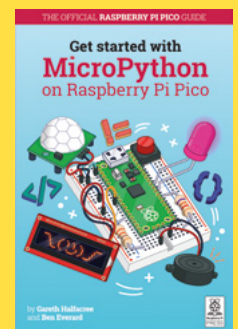
If you don't see MicroPython (Raspberry Pi Pico) in the interpreter list, you'll need to update Thonny. Open a Terminal window and type:
sudo apt update
&& sudo apt full-upgrade -y

◀ Switch the interpreter to MicroPython (Raspberry Pi Pico) in Thonny using the drop-down menu

Get Started with MicroPython on Raspberry Pi Pico

For more physical computing projects to try on your Raspberry Pi Pico, grab a copy of the new book, *Get Started with MicroPython on Raspberry Pi Pico*. As well as learning how to use Raspberry Pi Pico's pins as inputs and outputs, you'll build a simple game, measure temperatures, save and load data to your Pico's file system, and even make a burglar alarm for your room.

Get Started with MicroPython on Raspberry Pi Pico is available now from magpi.cc/picobook.



SOLDER HEADERS TO RASPBERRY PI PICO

Attach headers to Raspberry Pi Pico and get ready to explore electronics. **by Gareth Halfacree**



Warning!
Hot solder!

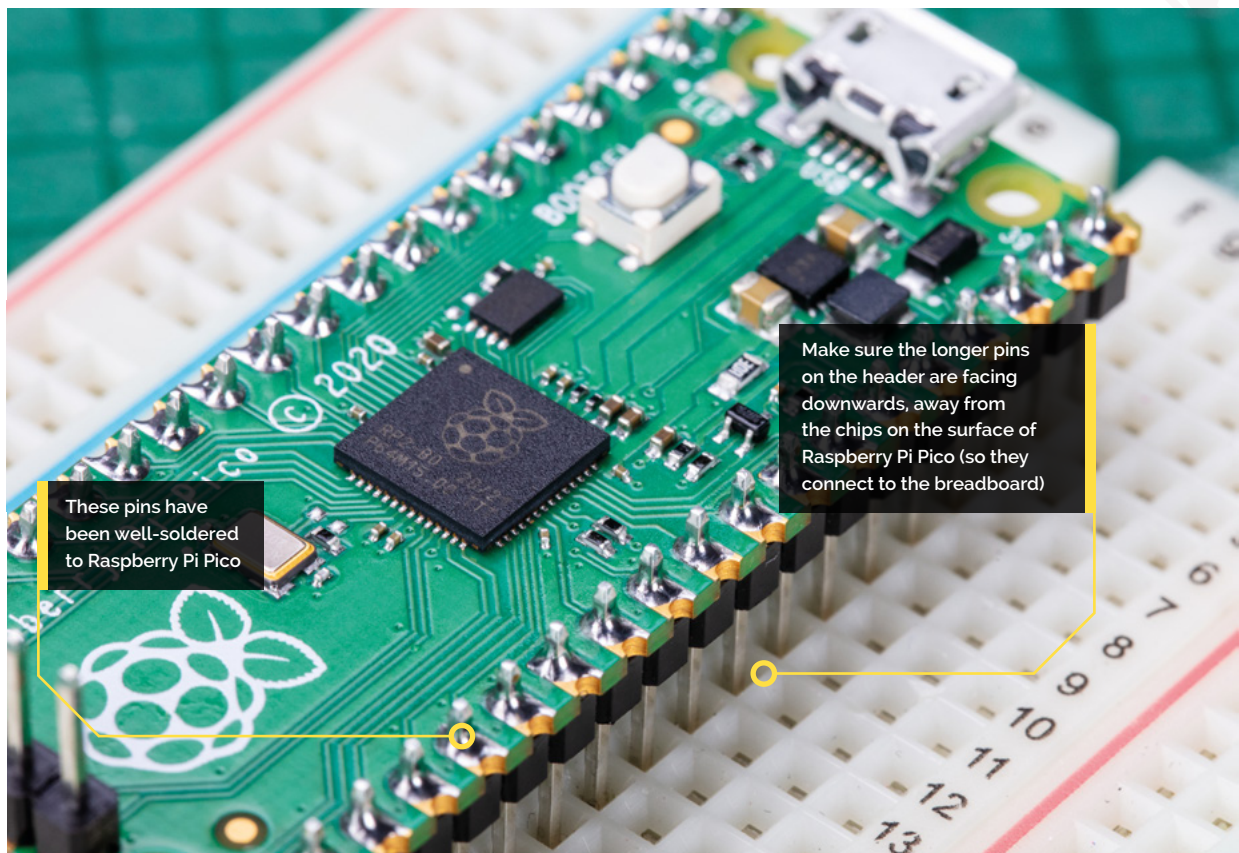
Soldering irons get very hot, and stay hot for a long time after they're unplugged. Make sure that you put the iron in the stand when you're not using it and don't touch the metal parts – even after it's unplugged.
magpi.cc/soldering

When you unpack Raspberry Pi Pico, you'll notice that it is completely flat: there are no metal pins sticking out from the sides, like you'd find on the GPIO header of Raspberry Pi. This is in case you wanted to use the castellations (bumpy edges) to attach Pico to another circuit board, or to directly solder wires.

The easiest way to use Pico, though, is to attach it to a breadboard – and for that, you'll need to

attach pin headers. You'll need a soldering iron with a stand, some solder, a cleaning sponge, Raspberry Pi Pico, and two 20-pin 2.54 mm male header strips. If you already have a solderless breadboard, you can use it to make the soldering process easier.

Sometimes 2.54 mm headers are provided in strips longer than 20 pins. If yours are longer, just count 20 pins in from one end and look at



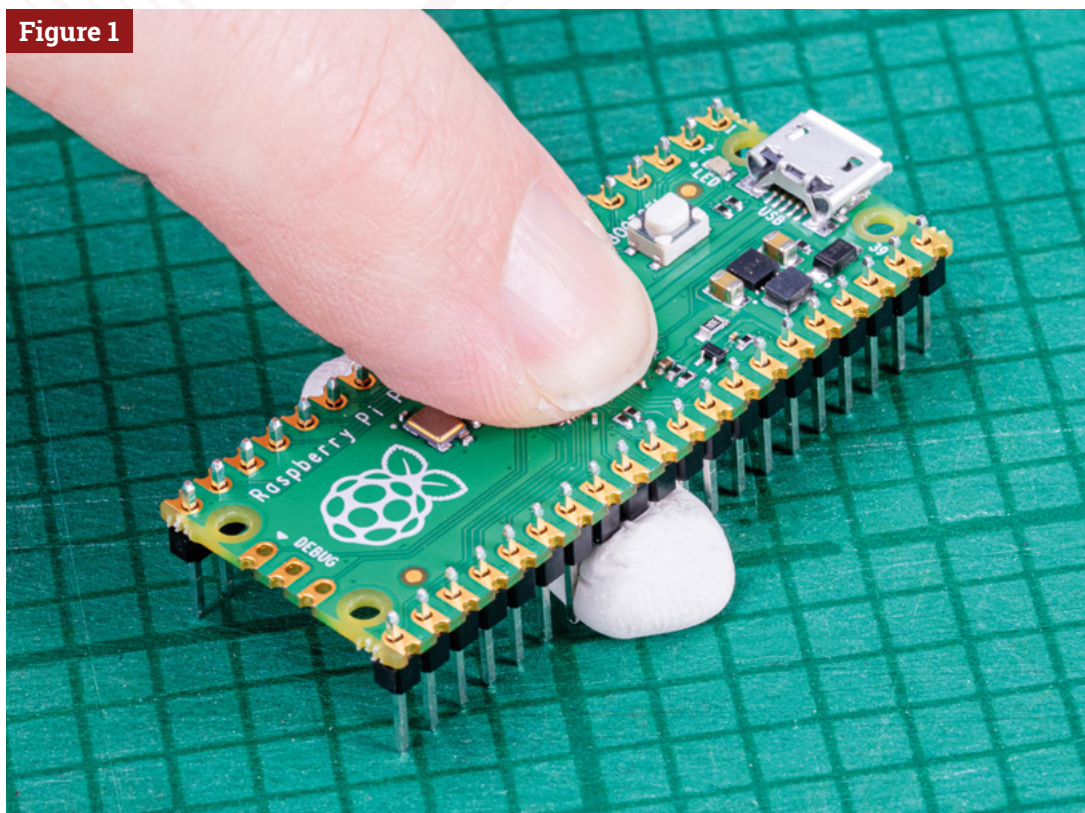
These pins have been well-soldered to Raspberry Pi Pico

Make sure the longer pins on the header are facing downwards, away from the chips on the surface of Raspberry Pi Pico (so they connect to the breadboard)

You'll Need

- ▶ Raspberry Pi Pico
magpi.cc/pico
- ▶ 2 × 20-pin headers
magpi.cc/headers
- ▶ Soldering iron and solder
magpi.cc/solderingiron

Figure 1



◀ **Figure 1** You can hold the headers in place with sticky putty before soldering

Top Tip 👍

Take it slow

Soldering is a great skill to learn, but it does take practice. Read the directions that follow carefully and in full before even turning your soldering iron on, and remember to take things slowly and carefully. Avoid using too much solder, too: it's easy to add more to a joint with too little solder, but can be harder to take excess solder away – especially if it's splashed over to other parts of your Pico.

the plastic between the 20th and 21st pins: you'll see it has a small indentation at either side. This is a break point: put your thumbnails in the indentation with the headers in both your left and right hands and bend the strip. It will break cleanly, leaving you with a strip of exactly 20 pins. If the remaining header strip is longer than 20 pins, do the same again so you have two 20-pin strips.

Turn Raspberry Pi Pico upside-down, so you can see the silkscreen pin numbers and test points on the bottom. Take one of the two header strips and push it gently into the pin holes on the left-

“ Take one of the two header strips and push it gently into the pin holes ”

hand side of your Pico. Make sure that it's properly inserted in the holes, and not just resting in the castellations, and that all 20 pins are in place, then take the other header and insert it into the right-hand side. When you've finished, the plastic blocks on the pins should be pushed up against your Pico's circuit board.

Pinch your Pico at the sides to hold both the circuit board and the two pin headers. Don't let go, or the headers will fall out! If you don't have a breadboard yet, you'll need some way to hold

the headers in place while you're soldering – and don't use your fingers, or you'll burn them. You can hold the headers in place with small alligator clips, or a small blob of Blu Tack or other sticky putty (**Figure 1**). Solder one pin, then check the alignment: if the pins are at an angle, melt the solder as you carefully adjust them to get everything lined up.

Use a breadboard

If you have a breadboard, simply turn Raspberry Pi Pico upside down – remembering to keep the headers pinched – and push both the headers

▼ **Figure 2** Alternatively, use a breadboard to hold the headers in place for soldering

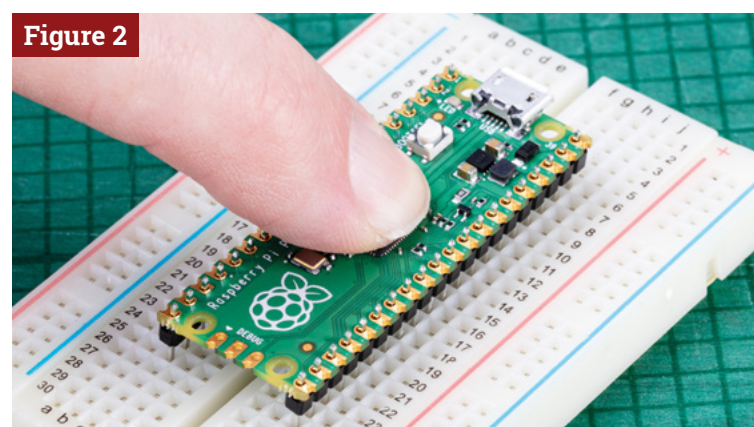


Figure 2

and your Pico into the holes on the breadboard. Keep pushing until your Pico is lying flat, with the plastic blocks on the pin headers sandwiched between your Pico and your breadboard (Figure 2).

Look at the top of your Pico: you'll see a small length of each pin is sticking up out of the pin holes. This is the part you're going to solder – which means heating up both the pins and the pads on Pico and melting a small amount of a special metal, solder, onto them.

Put your soldering iron in its stand, making sure the metal tip isn't resting up against anything, and plug it in. It will take a few minutes for the tip of the iron to get hot; while you're waiting, unroll a small length of solder – about twice as long as your index finger. You should be able to break the solder by pulling and twisting it; it's a very soft metal.

If your soldering stand has a cleaning sponge, take the sponge to the sink and put a little bit of cold water on it so it softens. Squeeze the excess water out of the sponge, so it's damp but not dripping, and put it back on the stand. If you're using a cleaner made of coiled brass wire, you don't need any water.

Top Tip

Four corners first

Solder the four corner pins first. Take your time, don't rush, and remember that mistakes can always be fixed.

Start to solder

Pick up your soldering iron by the handle, making sure to keep the cable from catching on anything as you move it around. Hold it like a pencil, but make sure your fingers only ever touch the plastic or rubber handle area: the metal parts, even the shaft ahead of the actual iron tip, will be extremely hot and can burn you very quickly.

“ Putting a blob of solder on the tip is known as tinning the iron ”

Before you begin soldering, clean the iron's tip: brush it along your sponge or coiled wire cleaner. Take your length of solder, holding it at one end, and push the other end onto the tip of your iron: it should quickly melt into a blob. If it doesn't, leave your soldering iron to heat up for longer – or try giving the tip another clean.

Putting a blob of solder on the tip is known as tinning the iron. The flux in the solder helps to

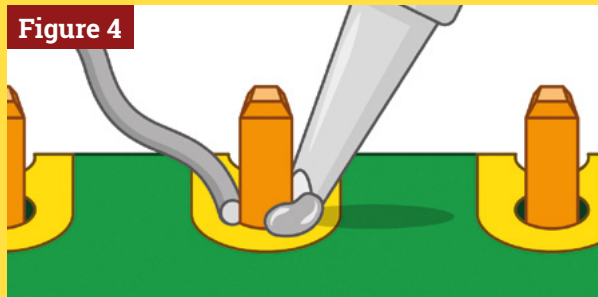
How to solder correctly

Figure 3



▲ Figure 3 Heat the pin and pad

Figure 4



▲ Figure 4 Add a little solder

Figure 5



▲ Figure 5 Now remove the iron

Figure 6



▲ Figure 6 A well-soldered pin

burn off any dirt still on the end of the iron, and gets it ready. Wipe the iron on your sponge or cleaning wire again to clean off the excess solder; the tip should be left looking shiny and clean.

Put the iron back in the stand, where it should always be unless you're actively using it, and move your Pico so it's in front of you. Pick up the iron in one hand and the solder in the other. Press the tip of the iron against the pin closest to you, so that it's touching both the vertical metal pin and the gold-coloured pad on your Pico at the same time (**Figure 3**).

It's important that the pin and the pad are both heated up, so keep your iron pressed against both while you count to three. When you've reached three, still keeping the iron in place, press the end of your length of solder gently against both the pin and pad but on the opposite side to your iron tip (**Figure 4**). Just like when you tinned the tip, the solder should melt quickly and begin to flow.

The solder will flow around the pin and the pad, but no further: that's because Pico's circuit board is coated in a layer called solder resist which keeps the solder where it needs to be. Make sure not to use too much solder: a little goes a long way.


Remove solder first

Pull the remaining part of your solder away from the joint, making sure to keep the iron in place. If you pull the iron away first, the solder will harden and you won't be able to remove the piece in your hand; if that happens, just put the iron back in place to melt it again. Once the molten solder has spread around the pin and pad (**Figure 5**), which should only take a second or so, remove the soldering iron. Congratulations: you've soldered your first pin!

Clean the tip of your iron on your sponge or brass wire, and put it back in the stand. Pick up your Pico and look at your solder joint: it should fill the pad and rise up to meet the pin smoothly, looking a little like a volcano shape with the pin filling in the hole where the lava would be, as shown in **Figure 6**.

Once you're happy with the first pin, repeat the process for all 40 pins on your Pico – leaving the three-pin 'DEBUG' header at the bottom empty. Remember to clean your iron's tip regularly during your soldering, too, and if you find things are getting difficult, melt some solder on it to re-tin the tip. Make sure to keep refreshing your length of solder, too: if it's too short and your fingers are too close to the soldering iron's tip, you can easily burn yourself.

When you're finished, and you've checked all the pins for good solder joints and to make sure they're not bridged to any nearby pins, clean and tin the iron's tip one last time before putting it back in the stand and unplugging it. Make sure to let the iron cool before you put it away: soldering irons can stay hot enough to burn you for a long time after they've been unplugged!

Finally, make sure to wash your hands – and celebrate your new skill as a soldering supremo! 

Soldering issues

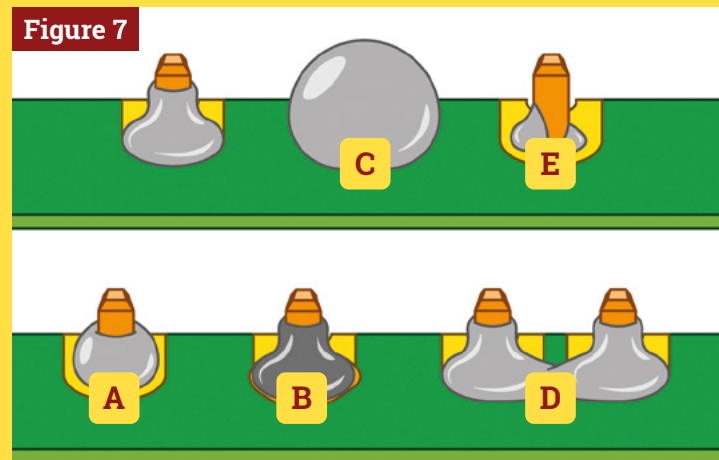
If the solder is sticking to the pin but not sticking to the copper pad, as in example **A** in **Figure 7**, then the pad wasn't heated up enough. Don't worry, it's easily fixed: take your soldering iron and place it where the pad and pin meet, making sure that it's pressing against both this time. After a few seconds, the solder should reflow and make a good joint.

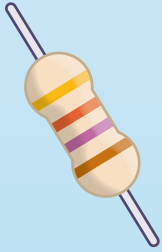
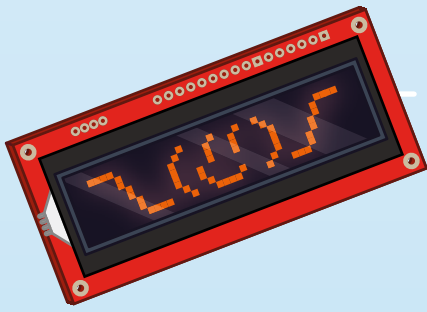
On the other hand, if the solder is too hot, it won't flow well and you'll get an overheated joint with some burnt flux (example **B**). This can be removed with a bit of careful scraping with the tip of a knife, or a toothbrush and a little isopropyl alcohol.

If the solder is entirely covering the pin, as in example **C**, you used too much. That's not necessarily going to cause a problem, though it doesn't look very attractive: so long as none of the solder is touching any of the pins around it, it should still work. If it is touching other pins (as in example **D**), you've created a bridge which will cause a short circuit.

Again, bridges are easy to fix. First, try reflowing the solder on the joint you were making; if that doesn't work, put your iron against the pin and pad at the other side of the bridge to flow some of it into the joint there. If there's far too much solder still, you'll need to remove the excess before you can use your Pico: you can buy desoldering braid, which you press against the molten solder to suck the excess up, or a desoldering pump to physically suck the molten solder up.

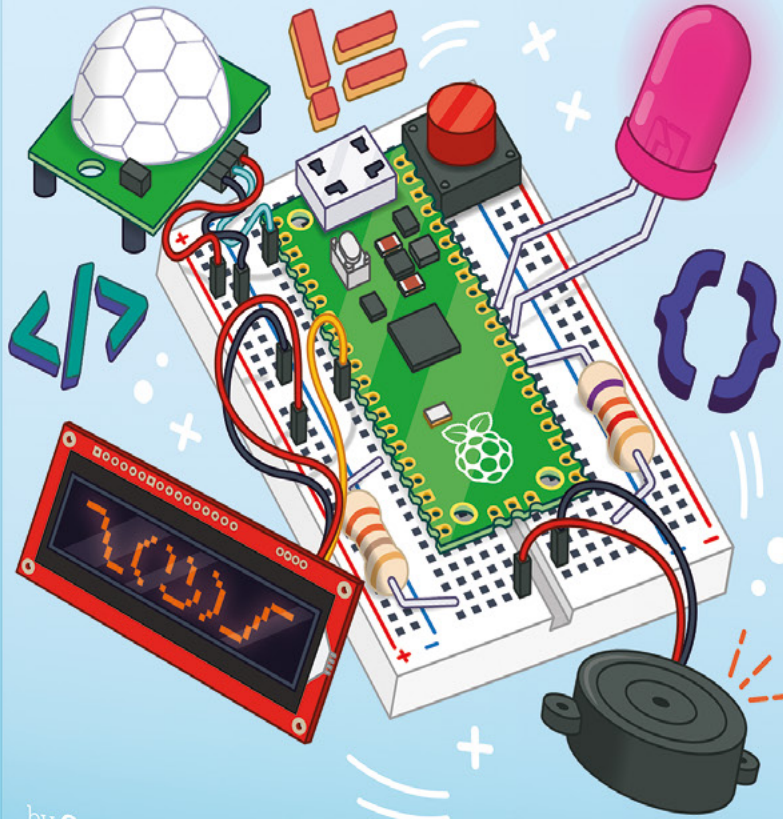
Another common mistake is too little solder: if you can still see copper pad, or there's a gap between the pin and the pad which isn't filled in with solder, you used too little (example **E**). Put the iron back on the pin and pad, count to three, and add a little more solder. Too little is always easier to fix than too much, so remember to take it easy with the solder!



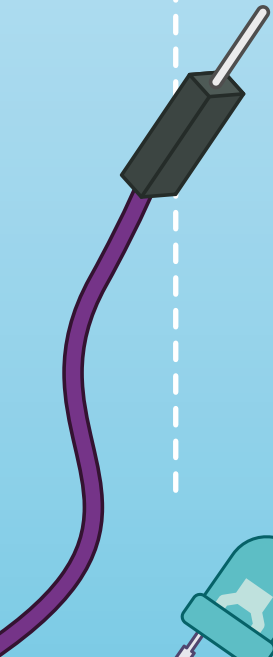
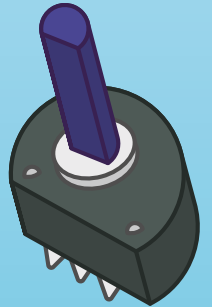
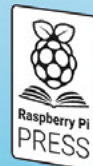


THE OFFICIAL RASPBERRY PI PICO GUIDE

Get started with MicroPython on Raspberry Pi Pico

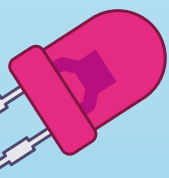


by Gareth Halfacree
and Ben Everard





Get started with MicroPython on Raspberry Pi Pico



Learn how to use your new Raspberry Pi Pico microcontroller board and program it using MicroPython. Connect hardware to make your Pico interact with the world around it. Create your own electro-mechanical projects, whether for fun or to make your life easier.



- Set up your Raspberry Pi Pico and start using it
- Start writing programs using MicroPython
- Control and sense electronic components
- Discover how to use Pico's unique Programmable IO



Available now: magpi.cc/picobook

Inky Impression

SPECS

DISPLAY:

5.7-inch EPD,
600×448 pixels,
ACeP seven-
colour, >170°
viewing angle

FEATURES:

4 × tactile
buttons,
breakout header
including
I2C, SPI, 3V3,
5V, ground

DIMENSIONS:

125.4 × 99.5 ×
9mm

USABLE DISPLAY AREA:

114.9×85.8mm

► Pimoroni ► magpi.cc/impression ► £66 / \$75

Create your own portable art gallery with this extra-large e-ink display featuring full colour. By **Phil King**

The Inky Impression is a super-size e-ink display that gives a great first impression as soon as you take it out of the box. Ours came with a beautiful impressionist painting by Lilla Cabot Perry as the default image.

Its 5.7-inch display is much bigger than Pimoroni's previous e-ink boards, the Inky wHAT and pHAT. Indeed, it's considerably larger than a full-size Raspberry Pi board, for which an extra female header (to boost its height) and metal standoffs are supplied. Care is needed to hold it by the edges when mounting, so as not to push down on the glass panel. Alternatively, it fits flush on a Raspberry Pi Zero without the extra header, for a reduced overall depth – great for putting it in a wooden frame for wall mounting.

If using a Raspberry Pi 400, you'll need a GPIO extender cable.

The slimline board even features some breakout GPIO pins on the underside, including I2C and SPI, along with four tactile buttons on its left edge – these could be handy for switching between images or sets of them in a headless picture frame setup.

Now in full colour

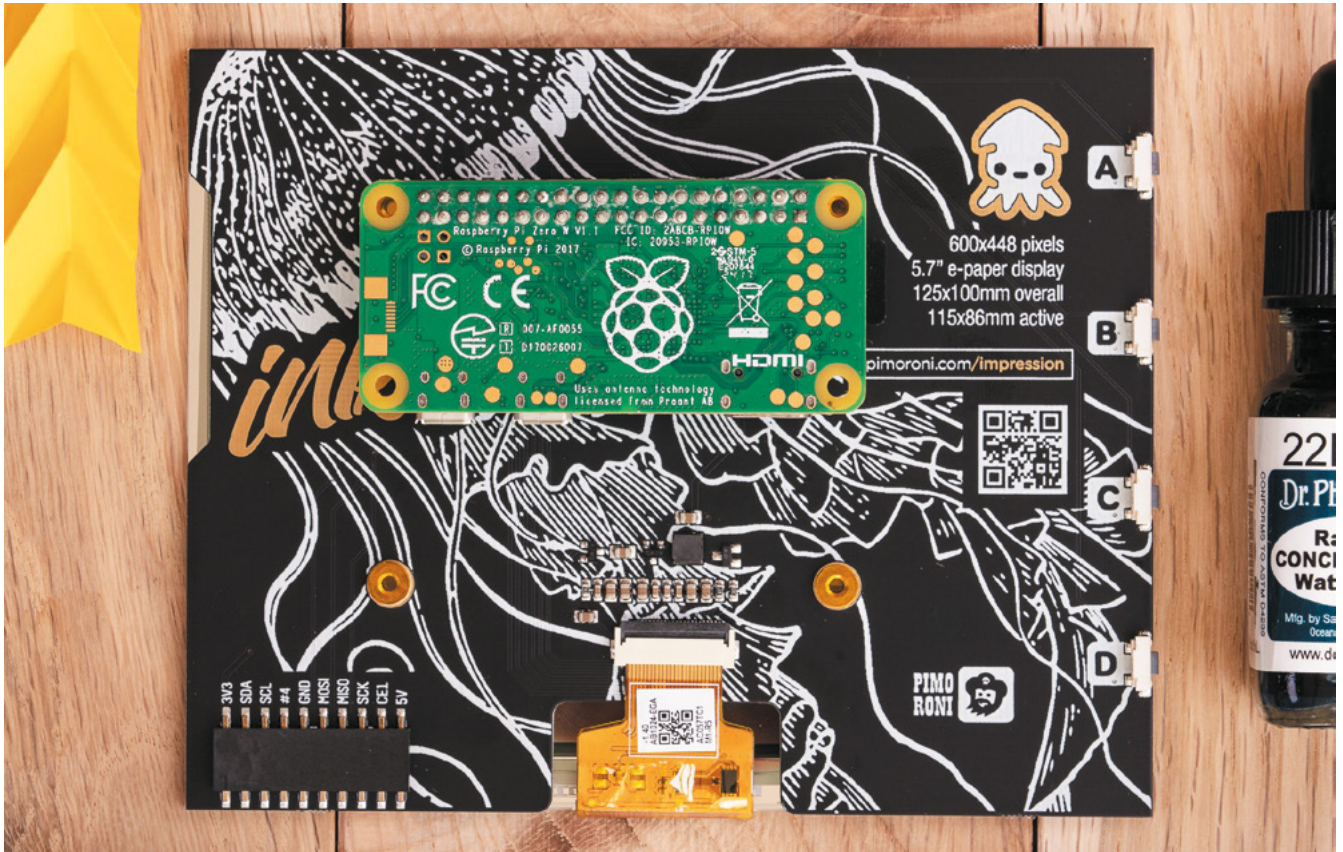
In addition to its size, the other major plus point about the Inky Impression is that it now uses ACeP (Advanced Color ePaper) to deliver seven colours: black, white, red, green, blue, yellow, and orange. This means it can display standard RGB images without the hassle of having to alter the colour mode to a special indexed palette in an image editor, as is required on the Inky wHAT and pHAT. So it's far easier to download or transfer images to use straight away; they just need to be 600×448 pixels, so may need scaling and/or cropping. Most major file formats are supported.

“ It can display standard RGB images without having to alter the colour mode ”

We found that simpler images, without too much detail, tended to work best. In particular, the Inky Impression is ideal for showing pop art, comic-book panels, and pixel art. Most standard photos will look fine on it, although you may find that the colours are a little washed out compared to the original image – this can be alleviated somewhat by altering the contrast and colour saturation of the photo.



▲ This impressive 5.7-inch e-ink screen can display full-colour images



As with all e-ink displays, it takes a few seconds to change to a new image (so it's no use for animations or videos), with a fair bit of flashing as it refreshes the different colour layers. There is also the possibility of some 'ghosting' as strong elements from the previous image may sometimes be noticeable – the best way to solve this is to show an all-white image between them.

Code examples

To use the Inky Impression, you'll need to download the Inky Python library (as used by previous boards), either using a one-line installer or cloning/downloading the GitHub repo (magpi.cc/inkygit) – the latter may be more up



- ▲ A Raspberry Pi Zero fits neatly on the rear, or the HAT can be mounted on a full-size model with the extra female header and standoffs
- ◀ A size comparison with the Inky PHAT and WHAT boards; the Inky Impression (rear) is considerably larger

to date. This includes a '7color' folder of Inky Impression code examples to get you started and demonstrate some of its use-cases.

Along with code to display a selected image, there's a nice example for a simple HTML webpage, which offers a convenient way of showing styled text on the screen. There's also a neat graphing example that plots coloured circles for data, illustrating how the Inky Impression could be useful for showing sensor or weather data in colourful charts. If using battery/portable power, a big advantage over a conventional display is the extremely low energy drain: it only draws power when changing images, which otherwise stay on screen even with the power disconnected. **W**

Verdict

While its lengthy screen refresh time means it's not suitable for every project, it's ideal for use as a digital art/picture frame or for displaying data in graphical form.

8/10

RFID HAT

SPECS

OPERATING SPECS:

Voltage: 5V; frequency: 125kHz; interface: UART; baud rate: 9600bps

I/O:

40-pin GPIO header, 0.91-inch OLED display, buzzer, card indication LED, power LED

DIMENSIONS:

55×66 mm, R 1.5 mm mounting holes

► SB Components ► magpi.cc/rfidhat ► £25 / \$34

Add tap-and-track technology to your projects with this RFID reader.

By **Lucy Hattersley**

R RFID (radio frequency identification) technology has slowly surrounded us in the last few years. Tapping to pay for things, to open doors, take and return books from the library, and to access equipment has become the norm. Many things we do now involve a tap of some sort.

The RFID HAT from SB Components comes with an RFID Module reader and two RFID tags (a plastic card and a key fob). On the HAT is also a small 0.91-inch OLED display, buzzer, and LEDs for power and card detection. There is also a GPIO extension header, enabling you to hook up the RFID HAT components to further items.

“ We’ve seen many Raspberry Pi projects make great use of RFID technology ”

The HAT (Hardware Attached on Top) runs a UART (Universal Asynchronous Receiver/Transmitter) in the 125GHz frequency range. You can pick up extra RFID tags, cards, and stickers for very little cost. Each RFID tag contains a unique identification number, which is detected when the tag is placed next to the RFID reader on the HAT.

RFID projects

We’ve seen many Raspberry Pi projects make great use of RFID technology. Perhaps the most famous is Museum in a Box (magpi.cc/miab). This project enables people to pick up objects and tap them to a Raspberry Pi-controlled speaker and hear an audio response. We’ve also seen a few projects that play Spotify albums when a model of the physical media is brought next to

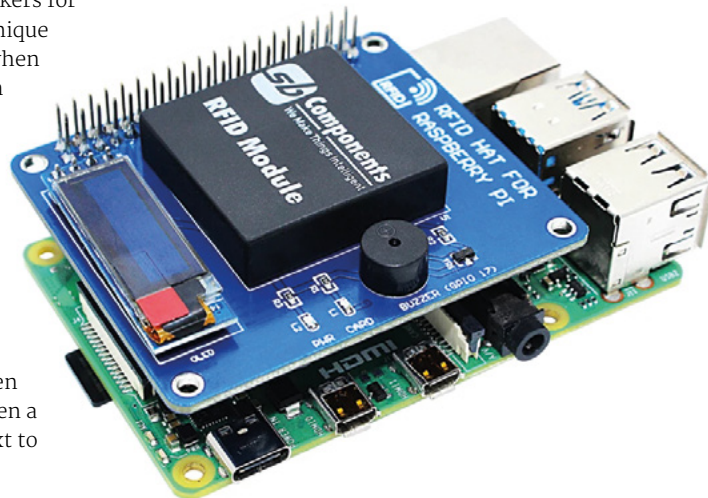
the RFID reader. In addition, you can use RFID for more standard projects such as ID badging, access control, and tracking of items.

Thanks to HAT technology, the installation is taken care of with the built-in EEPROM. You need to activate I2C and Serial Interface in Raspberry Pi OS, but it took all of five minutes to get everything up and running. There are some instructions in the store page, but we found it clearer to follow the instructions on the GitHub page (magpi.cc/sbrfidgit).

Once set up, there are three sample scripts: one which displays the RFID ID on the command line, one which displays it on the OLED screen and rings the buzzer, and a third that displays the LED on the board when an RFID tag is detected.

We couldn’t find API documentation, but reading the sample scripts was easy enough. Still, we would prefer to see more detail in the documentation. We had fun playing around with the RFID HAT, and think it’d be a great way to implement RFID technology into your Raspberry Pi project. **M**

▼ SB Components’ RFID HAT connects neatly on top of a Raspberry Pi Model B



Verdict

A fantastic HAT with good-quality components, and the OLED display is a nice touch. The documentation could be better, but we found everything easy to set up and understand.

8/10

NanoSound ONE Hi-Fi DAC Case

► Nanomesher ► magpi.cc/nanosound1 ► £59 / \$80

Combine the best of both worlds with a melding of the excellent NanoSound DACs and the Argon ONE case. **Rob Zwetsloot** grabs his listenin' ears

It's no secret that we're big fans of the Argon ONE cases – they're fantastic for keeping your Raspberry Pi cool, they look great, and they have neat features like GPIO pins under a magnetic lid and all the ports rearranged to the rear. We're also a fan of the NanoSound DACs, so we were quite intrigued when we found out about a melding of the two into one product: NanoSound ONE.

“ You have your choice of bottom part for the case ”

On paper it's a simple change – the board attached to the roof of the case that provides a fan and heatsink is replaced with one that provides a DAC and heatsink. Because of this, the ports on the rear are slightly changed, going from two HDMI ports and a 3.5 mm jack to one micro HDMI port and two RCA jacks.

This does mean you're losing the active cooling. However, passive cooling of the heatsink (which is the entire case) is definitely good enough.

Snug fit

Installing a Raspberry Pi 4 is simple – you add the little breakout board that plugs into HDMI 0 and the 3.5 mm jack on Raspberry Pi, and then connect the GPIO pins with the GPIO header on the top of the case. The breakout board is a bit of a snug fit, so don't be worried if it looks like it's not quite spaced out correctly. It is!

You have your choice of bottom part for the case. The standard piece just encloses the whole thing and provides an SD card port, while there's also the M.2 SSD storage bay option which we reviewed last month.

The whole process is very quick, and we were able to experience some excellent sounds from the DAC part in no time, which uses the same tech as the NanoSound DAC 2. You're paying a little bit of a premium for the DAC and case together (only about \$5), but you are getting a custom version of one of our favourite cases – so it's definitely worth it. 🎧



SPECS

CHIP:

Texas Instruments PCM5122

SAMPLE RATE:

192 kHz

RESOLUTION:

24-bit

SIGNAL-TO-NOISE RATIO:

112 dB

AUDIO OUTPUT:

RCA

◀ Control it with a smartphone remote or the included remote

Verdict

A great case with great sound at a reasonable price. This could be the one for folks who need the best audio out of a media centre.

9/10

Grow Kit

SPECS

DISPLAY:

IPS LCD, 0.96-inch, 160x80 pixels

INPUT/ OUTPUT:

Four tactile buttons, piezo buzzer, LTR-559 light and proximity sensor

CONNECTORS:

3 x JST SH 3P connectors for sensors, 3 x Picoblade 2P-compatible connectors with MOSFETs, breakout header

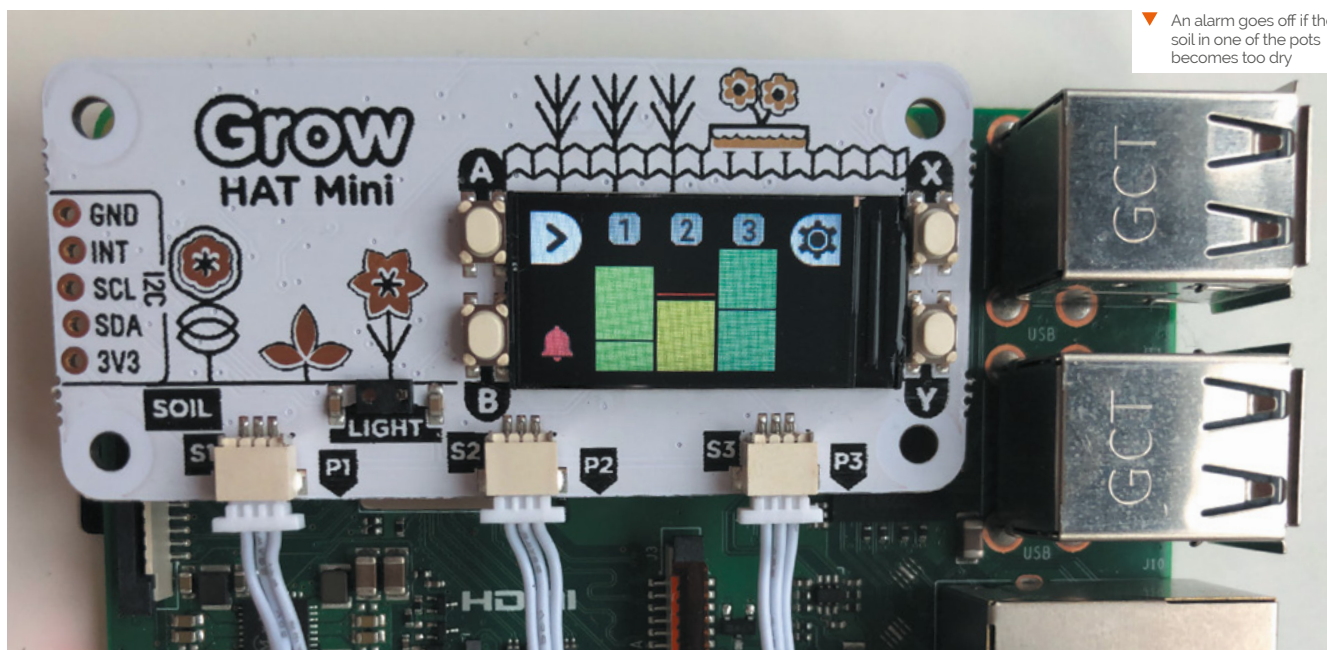
► Pimoroni ► magpi.cc/growkit ► From £30 / \$34

If your gardening efforts need a helping hand, Pimoroni's got a HAT-based kit for that. **Rosie Hattersley** samples some herbs

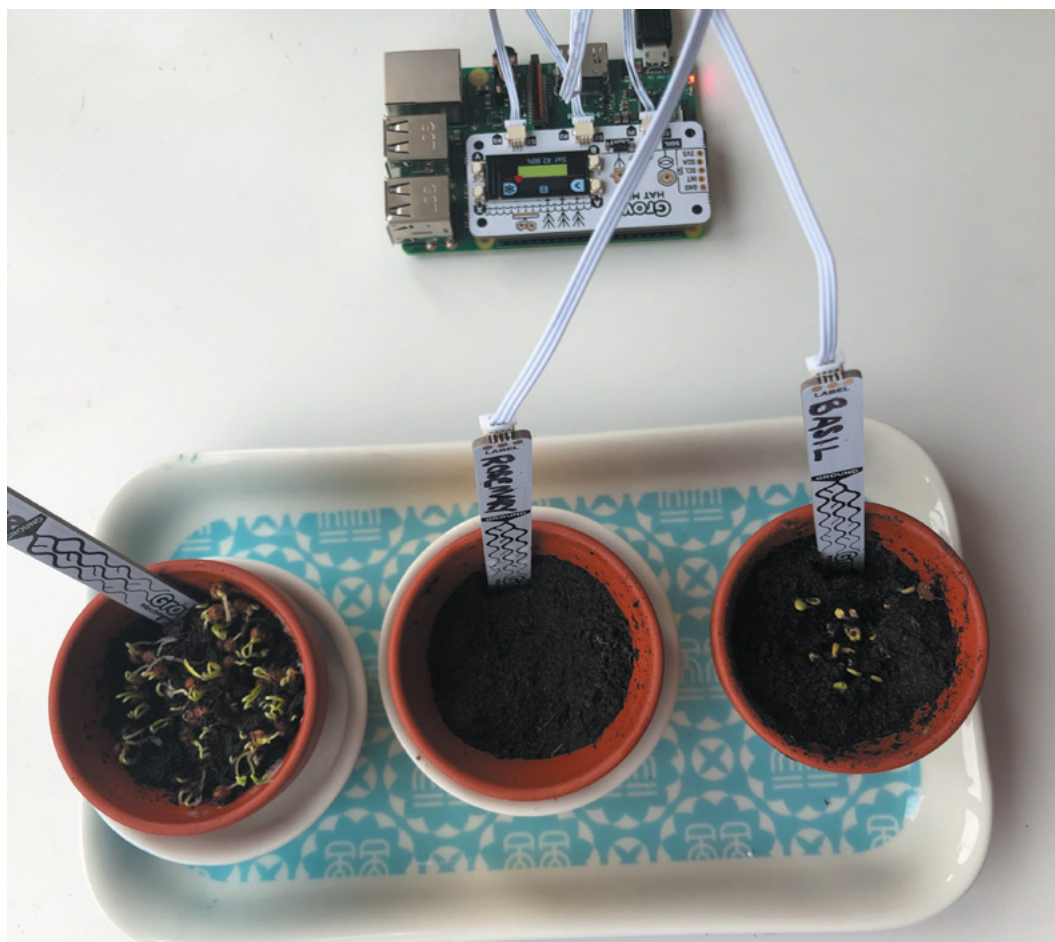
As the days begin to lengthen, it's a great time to start planting seeds and encouraging new green shoots. Cold nights and heating can play havoc with the germination process, foxing both plant and gardener, so a sensor to help keep track of soil moisture can be a real boon. The Grow HAT Mini in this kit provides a straightforward means of monitoring the moisture levels of up to three plants at once. It can either be used with your existing plants, in which case you will only need the standard £30 Grow Kit, which includes three moisture sensor sticks and cables. Alternatively, there are herb and chilli-growing kits, costing an additional £9.90 (\$10.30) each.



▲ If you add too much water, the indicator turns bright blue



▼ An alarm goes off if the soil in one of the pots becomes too dry



◀ The moisture sensors have space to label each plant

We tested the Grow Herbs kit, reasoning that the selection of basil, rosemary, and coriander would require quite diverse growing conditions and need different watering routines.

The Grow HAT Mini is sized for use with Raspberry Pi Zero, but works with any model with a 40-pin GPIO header. Setup involves updating Raspberry Pi OS (as with any software install) and entering the install code (pimoroni.com/seeds) into a Terminal, then rebooting. Once set up, the whole caboodle can be run headless (without a monitor) from a USB power supply.

“ A sensor to help keep track of soil moisture can be a real boon ”

If you've got the herb or chilli kit, you need to soak the Cocopress 'soil' pellets in water to rehydrate them before planting the seeds. Label the three seed sensor sticks and gently attach them to the respective plugs on the HAT – rather fiddly as the plugs are tiny – and then insert them into

the soil. A tiny (0.96-inch, 160×80 pixels) IPS LCD screen on the HAT immediately indicates whether each pot's soil is wet enough. Blue means too wet, while yellow or amber indicates it's too dry.

Soily something wrong

Should the saturation level of any of the plants being monitored fall below the threshold, the Grow sensor issues a sonic alert – three short beeps at roughly three-second intervals – and a blue bell icon appears on the Grow HAT's display. To deactivate the alarm, just press the cream-coloured button adjacent to the bell. If you don't water the plant within a few minutes, the alarm will sound again.

If the plant in question thrives in a dry environment or, conversely, prefers to be pretty damp, you can adjust the saturation level at which the Grow HAT triggers a warning. Pressing the A button on the HAT cycles through the percentage moisture level of each pot, and also takes you to a Settings menu where you can fine-tune the saturation levels for each plant so the alert doesn't constantly trigger for your desert-loving aloe vera. [📖](#)

Verdict

We were very impressed with the kit's ease of use, aided by a foolproof online installation guide. There were far too many seeds for the size of the pots that came with the herb and chilli kits, but that simply means you can plant successive sets of seeds.

8/10

10 Amazing:

Day-to-day life projects

Make life a bit easier with these Raspberry Pi projects

Raspberry Pi has long been popular in the home automation community. With years and years of ideas, projects, and advancements, a lot has been discovered and created for Raspberry Pi. Here are some of the projects that we think might help make life just that little bit better. [M](#)



**Alert!
High voltage**

Some of these projects use mains electricity. Follow safety regulations if you plan to replicate them.

magpi.cc/electricalsafety



▲ LunzPi

Radio alarm clock

This retro-looking alarm clock is anything but – it uses a Spotify playlist and can control some light bulbs as well. Make sure Sonny & Cher are on the playlist, though.

magpi.cc/lunzpi



▲ Raspberry Pi Bus Schedule

Be on time

Missing the bus can be a royal pain. Creating your own little info screen with the bus schedule for your local stop is a great way to make sure you don't unnecessarily leg it out the door in the morning.

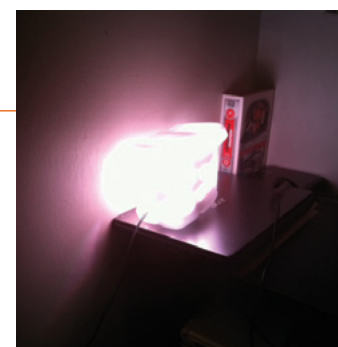
magpi.cc/bussched

▶ LED Light Sunrise Clock

Don't be SAD

Waking with the sun has some science behind it that experts say means you wake up better. Or something. Test out the theory on yourself with this cool project.

magpi.cc/sunriseclock



◀ PiHue

Automated lights

Controlling your lights is a classic home automation task, and this project allows you to control one of the mass-produced smart light bulb standards using a Raspberry Pi.

magpi.cc/pihue



▲ Touchscreen thermostat

Automated heating

Automatically adjusting the heating in your home using smart devices and very precise rules can be easily achieved using a bit of a hack with a Raspberry Pi.

magpi.cc/thermostat



▲ DAKboard

Info screen

Don't quite want something as large as a magic mirror, but love the idea of a smart info board in your home? Try out the DAKboard idea – you can even make it look like a picture or small window.

magpi.cc/dakboard



▲ BOSEBerry Pi

Internet radio circa 2011

We recently featured this upcycling build in the magazine, where David Hunt took a decade old iPod dock, gutted it, and installed a Raspberry Pi in there for a cleaner-looking, customisable, internet radio.

magpi.cc/boseberrypi

◀ MagicMirror

Fair and informed

Make sure you look good and know what's up today with this excellent AR project that is almost a rite of passage for many Raspberry Pi makers.

magicmirror.builders



▲ Smart coffee machine

Morning brown dispenser

There's a ritual to making (fancy) coffee that some folks love. Other people just need a hit of caffeine in the morning to get themselves going. This makes it easier and quicker to get a nice cup of coffee to help wake up.

magpi.cc/smartcoffee

▼ RadioGlobe

International radio locator

One of our favourite projects from 2020, this toy globe has been heavily modified so that a Raspberry Pi knows where a reticule is pointing anywhere in the world and will play radio from that location. It's very cool.

magpi.cc/radioglobe



Learn MicroPython with Raspberry Pi

Get the most from Raspberry Pi Pico by getting to grips with MicroPython. By **Lucy Hattersley**

Get Started with MicroPython on Raspberry Pi Pico

CREATOR

Raspberry Pi Press

Price:
£10 / \$14

magpi.cc/picobook

MicroPython is one of the two languages used to control the new Raspberry Pi Pico microcontroller board (the other being C/C++). Based upon Python 3, it will be a vital tool for controlling your brand new Pico.

Get Started with MicroPython on Raspberry Pi Pico, by Raspberry Pi Press, is our primary knowledge resource for those looking to use Pico. Written by Gareth Halfacree and HackSpace magazine editor Ben Everard, the book walks

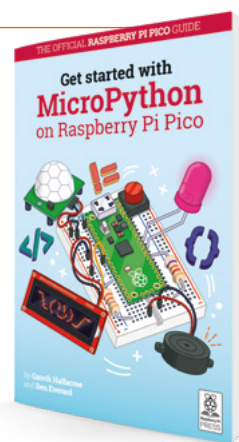
you through setting up the board and learning physical computing techniques.

After setup, each chapter covers a tutorial for building a range of starter projects: traffic lights, reaction game, burglar alarm, temperature gauge, data logger. The final chapter explores using the I2C and SPI digital communication protocols to display data on an LCD screen.

While the primary focus is on electronics and integrating Raspberry Pi Pico into real-

world projects, each chapter focuses on an aspect of MicroPython and you'll learn a lot as you go.

This book is the best place to start with Raspberry Pi Pico. Make sure you pick up a copy.



Pico Documentation

Keep these vital documents to hand

MICROPYTHON DOCUMENTATION

Bookmark the MicroPython documentation website for a complete reference guide to libraries and modules.
docs.micropython.org

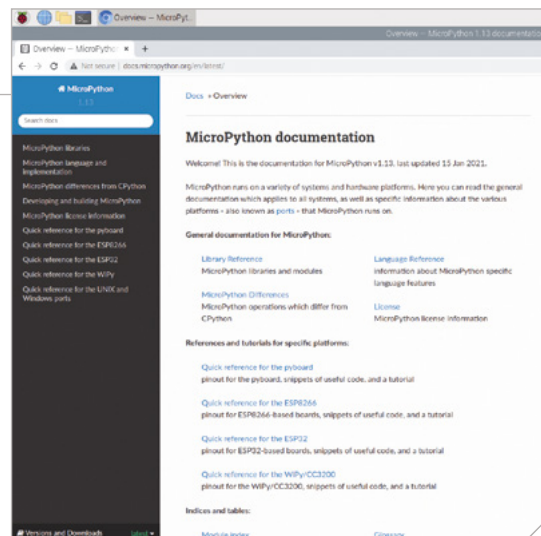
RASPBERRY PI DATA SHEETS

Data sheets, including SDK information for MicroPython and

C/C++, are found on Raspberry Pi's data sheets section.
datasheets.raspberrypi.org

PICO PYTHON SDK DOCUMENTATION

Download the Pico Python SDK documentation PDF for a complete guide to working with MicroPython on Raspberry Pi Pico.
magpi.cc/picopythonsdk



Programming with MicroPython

CREATOR

O'Reilly

Price:
£40 / \$50[magpi.cc/
pwmmicropython](http://magpi.cc/pwmmicropython)

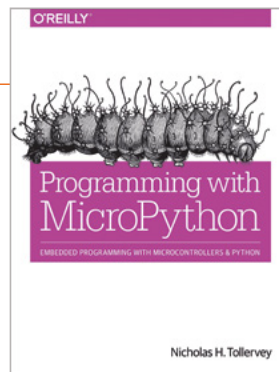
As you move beyond Raspberry Pi Pico and into a broader MicroPython experience, you should take a look at Nicholas Tollervey's *Programming with MicroPython*.

It explores a variety of devices that run MicroPython, from PyBoard to Adafruit's Circuit Playground Express and ESP8266/ESP32 boards.

The book has a detailed overview of MicroPython and how various embedded technologies can be used to make devices. It covers input

and output, GPIO, various sensor technologies, networking, sound and feedback, and works up to building a wheeled robot.

You can read chapter one online (magpi.cc/pwmmicropythonc1) to get a feel for the text. While it doesn't cover Pico, the book offers a thorough grounding in the MicroPython language.



Raspberry Pi Pico website

CREATOR

Raspberry Pi

Price:
Freepico.raspberrypi.org

There is a wealth of information and resources available on Raspberry Pi's official Pico website. You will probably come across this website during the MicroPython

setup process (as you flash Pico to work with MicroPython rather than the default C/C++ SDK).

Make sure you bookmark it and come back. It has links to the MicroPython wiki and forum, as well as links to much of the documentation listed here.

There are also project examples to offer inspiration for the kinds of things you can do with Raspberry Pi Pico (and you can expect a lot more of this in future editions of *The MagPi*). 



Websites

Learn MicroPython with these websites



REALPYTHON

An Intro to Programming Hardware in Python explores using Python to control microcontrollers and hardware devices.

realpython.com/micropython

UDEMY

Udemy has a complete range of video tutorials based around MicroPython. Most are based around the PyBoard now, and we expect Pico courses to be available shortly.

udemy.com/topic/micropython

MICROPYTHON FORUM

Learning from others is a great way to gain knowledge. The MicroPython community is a great place to meet like-minded learners.

forum.micropython.org



Kids Invent Stuff

Ruth and Shawn bring kids' invention ideas to life on their YouTube channel

- Name **Ruth Amos and Shawn Brown** | ➤ Occupation **YouTubers**
- Community role **Makers** | ➤ Website **magpi.cc/kidsinventstuff**

We all had outlandish/ amazing ideas as kids that unfortunately we did not have the skills or resources to carry out. Perhaps sometimes for the better. The team of Ruth Amos and Shawn Brown might be able to help these kids, though, with Kids Invent Stuff.

“Kids Invent Stuff is a YouTube channel which brings to life kids’ invention ideas, created to

inspire primary aged children around engineering and STEM,” they tell us. “Young people are encouraged to submit their own ideas for inventions to solve novel and entertaining technical challenges. The most creative and entertaining inventions are then showcased as part of Kids Invent Stuff YouTube episodes. The chosen idea is then built and demonstrated on film, with often hilarious consequences;

we’ve built furry electric dog cars, popcorn-firing doorbells, and alarm clocks that wake you up by dropping pool balls on your head. These have all been inventions designed by children that we have brought to life.”

Ruth is a self-taught engineer and inventor. In 2006 she was Young Engineer for Britain with her GCSE product ‘The StairSteady’, which is now sold internationally. She is a keen welder and maker and co-founded #GirlsWithDrills with Kisha Bradley.

Shawn Brown is an engineer, artist, and designer. As a multidisciplinary engineer and artist, he has worked on a myriad of projects; from building custom electric vehicles, to bringing to life the 12 m tall ‘Man Engine’ – the UK’s largest mechanical puppet. Shawn has both a master’s degree in Energy Engineering and a master’s in Fine Art.

How has Raspberry Pi helped some of your builds?

We have used Raspberry Pi in a number of builds. We

▼ The spy foam camera seems a lot more conspicuous than we imagined



worked with Estefannie, from Estefannie Explains It All, to hack our rainbow unicorn that poops jellybeans (invented by eight-year-old Tru, magpi.cc/unicornpoop) to be able to poop jellybeans when someone tweeted it: magpi.cc/unicorndemo.

This year, we built nine-year-old Nathan's foam-firing spy camera, which was built around a Raspberry Pi. You can see it in action here: magpi.cc/foamspydemo.



“ A YouTube channel which brings to life kids' invention ideas ”

What has the response been to your videos?

People seem to love seeing kids' invention ideas come to life and we've had some fantastic comments from parents and teachers saying that their children and students have

been inspired by the channel; in fact, over 95% of those submitting ideas have said that our channel and challenge had helped improved their child's understanding of engineering.

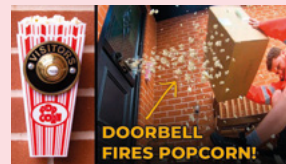
The project was set up because research showed that it was really important to inspire and keep engaged young people, particularly girls, into STEM while still at primary school. Kids make amazing problem solvers and we wanted to showcase and celebrate that on the channel. **M**



▲ It's a whole new world out in the harbour on a floating rug

Invented Stuff

Popcorn-Firing Doorbell



► magpi.cc/popcorndoorbell

Giant Rube Goldberg Machine



► magpi.cc/rubegoldberg

Zander's Robot Shopping Trolley



► magpi.cc/robbottrolley

▲ The tea train will arrive, whatever type of snow is on the line

This Month in Raspberry Pi

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. Don't panic, you'll scare the robots
02. Balancing robots are just magic
03. PiMowBot continues to be developed and is very cool
04. Some very classic 3D graphics test here
05. This is not the only project we've seen that gets mad at 2020, but it is the cutest
06. We're looking forward to seeing the Pi Wars 2021 robots in action
07. We like the idea of using e-paper as a photo frame!
08. It's cool to see the advanced features of the HQ Camera made accessible in this way
09. Jitesh's robot experiments are always great to see
10. We love the aesthetics on this
11. We like the look of this robot!



TGD @TGD_Consulting 03

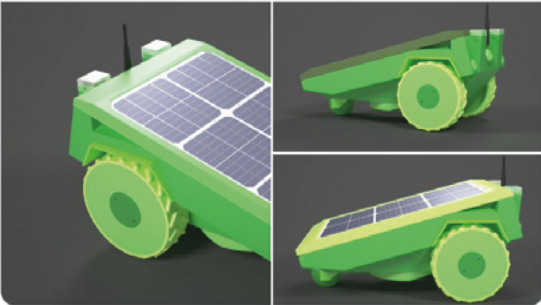
Replying to @TheMagPi

Tim brainstormed on a solar panel alternative, since the current panel has reached its end of life & is no longer made in the same dimensions. And I finished an online tutorial on building and setting up a PiMowBot.

TGD @TGD_Consulting · Dec 12, 2020

Eine passende Alternative für das bisherige 20W mono Solarpanel ist gefunden. Das Design des PiMowBots verändert sich dadurch nur unwesentlich bei gleicher PV-Leistung. Demnächst mehr Infos...

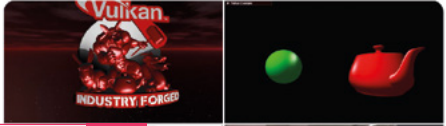
#Make #DIY #3Dprinted #IoT #smart #robotics #lawnmower based on #RaspberryPi.



JP Atkinson @JP_Atkinson 04

Replying to @TheMagPi

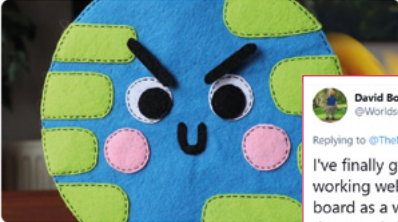

messing about with the V3DV Vulkan Mesa driver for RPI4



8 Bits and a Byte @8bitsandaByte 05

Replying to @TheMagPi

Goodmorning!
We built an internet controllable robot world that smashes 2020 with 3D printed fire! Extra bonus: it's voiced by one of our favourite makers ever!
@hannahmakes

David Booth @Worlds640 08

Replying to @TheMagPi

I've finally got the @Raspberry_Pi HQ cam module working well with the @ArdcamOfficial CSI to USB board as a webcam. Settings tweaked to increase gain, decrease brightness and increase saturation to give a MUCH better picture quality.

Brian Cortell @CannonFodder 06

Replying to @TheMagPi

Finally putting together my robot for #PiWars 2021. Meet 'Testbed' based on the Tiny 4WD chassis. @adafruit Braincraft hat for meat bag duties & Motor Bonnet for driving the motors. A small banana for scale.




Wear a mask to protect & respect others @PaulClark_Moz 07

Replying to @TheMagPi

A 9.7 inch e-paper photo thingy with a Pi 3A



New mount designed and 3D printed tool



John Chinner @JohnChinner 11

Replying to @TheMagPi

Building myself an @ExoMy_Rover !

Jitesh Saini @jiteshsaini10 09

Replying to @TheMagPi

Used a smart phone as a compass sensor to achieve precise direction control of my @Raspberry_Pi robot. The compass works through a browser, thanks to #javascript APIs. Try the apps created for this project using your smart phone with robot simulator here helloworld.co.in/article/compas...



Rob @delineator 10

Replying to @TheMagPi

I released an update to "pitfalls" - a #MicroTonal music scale explorer library for the #Monome #Norns shield - pictured here running on a #RaspberryPi 3B+




Last of the Christmas projects

Pack away your festive builds for another year

We were unable to fit all of the amazing projects from December until now. Because we love Christmas projects, we're putting them in our February issue to make sure they get noticed. [M](#)

01. Games and Christmas? We can't imagine a more iconic duo
02. Two great projects that go great together
03. We feel that the Shelf Elf does not need a power-up
04. This project is very important, don't let anyone tell you otherwise
05. We love this idea! What other public decorations are controlled by Raspberry Pi?
06. It took us a moment to realise how big this tree was – that's a Sense HAT on top!

01


Stewart Watkins @stewartwatkins
Replying to @TheMagPi

A Christmas Project using the Raspberry Pi SnowPi RGB from @ryantec

It's a game which can be played using a USB controller.

Can you react fast enough when the nose or eyes light up?

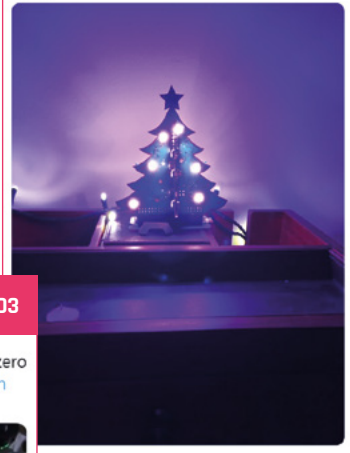
Includes a version that doesn't need SnowPi - just Pygame Zero



02

Scott Young @thebotmakes
Replying to @TheMagPi

Now that the Christmas decorations are up I've worked on a @cheerlights controlled mini Christmas tree, with a pi zero W getting the latest cheerlights colour and syncing up with all the other cheerlights across the world.



03

Martin Parker @Mr_MartinParker
Replying to @TheMagPi

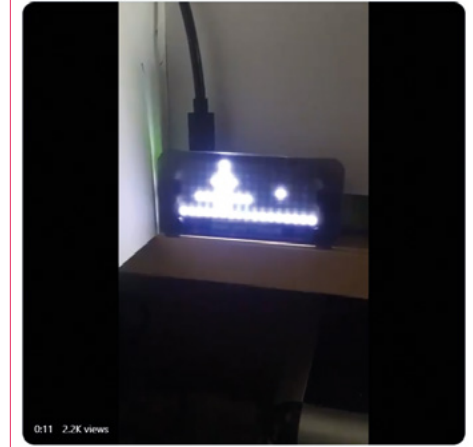
I made an #IronElfOnTheShelf using a #raspberrypi zero w, an old #sky box playback spinner, a bit of #python and using conrtab to set it running every hour.



04

Scott Young @thebotmakes
Replying to @TheMagPi

With a 4 year old this is the most important project - a sleeps countdown to Christmas 🤖. For the snowball each flake has its own physics! Pi zero W again to get the date and work the display, been a very Christmas/Pi heavy week!



05

Young in Cheshire @youngincheshire
Replying to @TheMagPi

Sending messages with lights: town Christmas tree lit with almost 2000 LEDs controlled by @Raspberry_Pi showing text messages in Morse Code @LightsMessage #EveryLightAMessage



06

ROY @royhair
Replying to @TheMagPi

Played with a Christmas Tree window decoration, including a Sensehat as a colour-changing star / message display. It is networked but I haven't connected it to the Internet yet. I might get it to display tweet messages (filtered of course!)



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

Air Monitoring HAT

Real-Time PM Sensor For **Raspberry Pi**

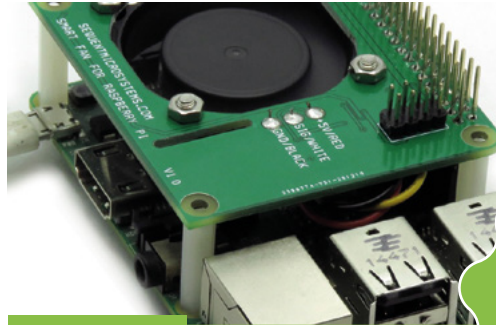
- PMSA003 Sensor
- Standard Raspberry Pi 40 Pin GPIO
- OLED Display 0.91"
- UART(serial) interface



Air Monitoring HAT

An open-source portable sensor that can measure air quality. In many cities this can be quite important. It's already been funded, so give it a look to see if it's the kind of sensor you'd like to use.

▶ kck.st/35u0i10



Smart Fan

Active cooling solutions are quite popular for Raspberry Pi 4, and this smart fan is yet another. It also allows other HATs to be added on top, and away from Raspberry Pi. It's apparently whisper quiet.

▶ kck.st/3odDAeB

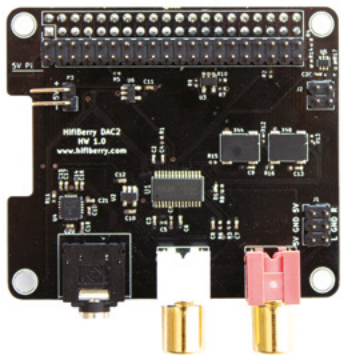
CROWDFUNDING A PROJECT?

If you've launched a Raspberry Pi-related project, let us know!
magpi@raspberrypi.com

www.hifiberry.com

PROFESSIONAL OR HIGH END?

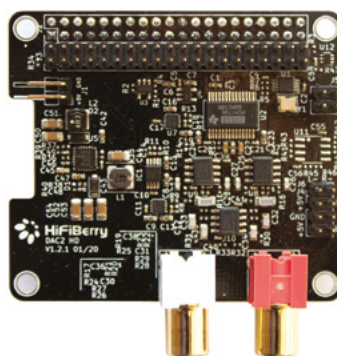
Pick the HiFiBerry DAC for Raspberry Pi, that fits your needs



HiFiBerry DAC2 Pro

- Optimised layout for best audio quality
- Headphone output
- Can be equipped with an additional DSP board

DSP add-on board adds digital signal processing capabilities

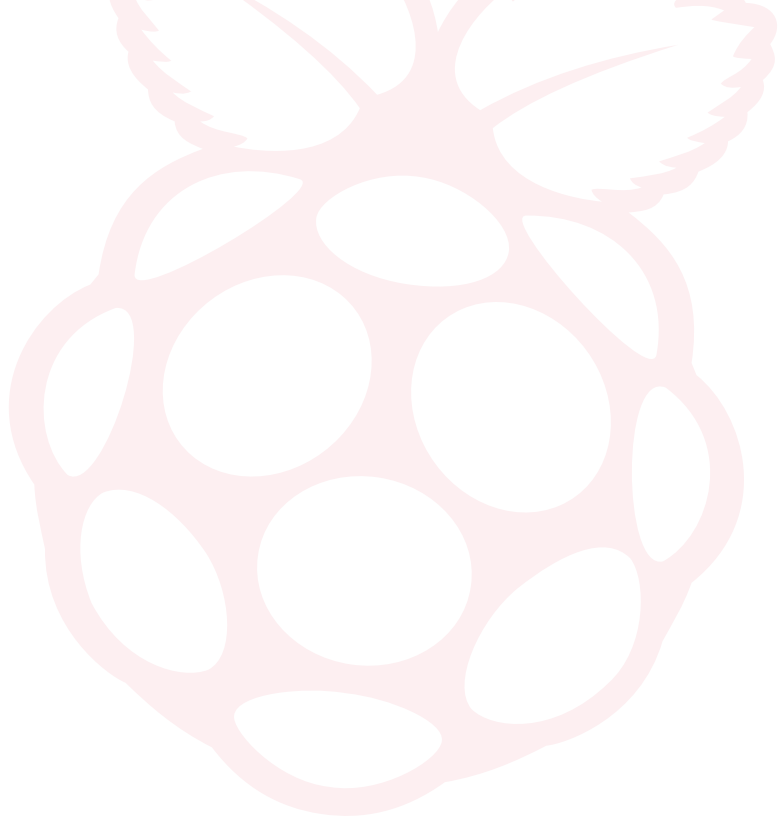
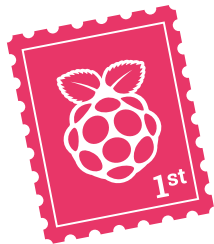


HiFiBerry DAC2 HD

- Higher dynamic range
- Lower noise floor
- Lower distortion
- Can be equipped with an additional DSP board



Your Letters



Pico in MagPi

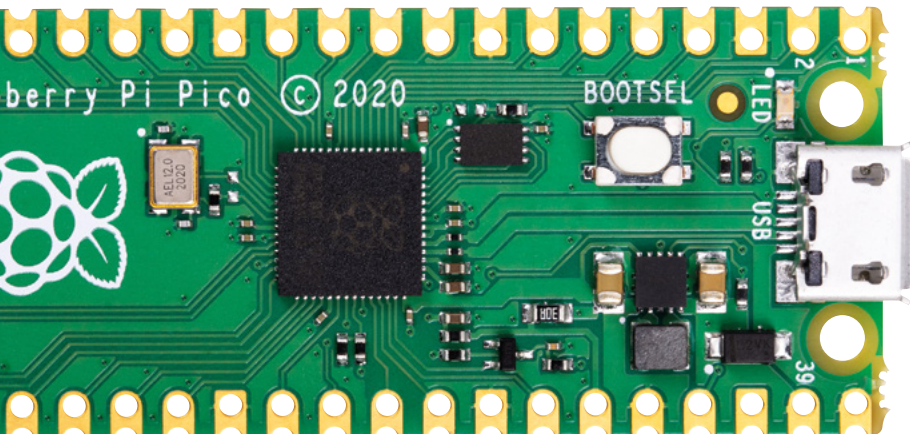
I was very surprised to see that HackSpace magazine was released with a new type of Raspberry Pi on the cover, much like you did with Raspberry Pi Zero ages ago. Can we expect to see Pico on the cover of *The MagPi*? Will you cover it like you do Raspberry Pi, or do you not do microcontrollers?

David via email

Unfortunately not this time. HackSpace magazine is one of our Raspberry Pi Press siblings, so they get the honour as they launched the same day.

As for coverage, we'll absolutely be doing stuff with Raspberry Pi Pico. We don't really shy away from microcontrollers, and we definitely love to talk about Raspberry Pi stuff. We're hoping to have many tutorials for Pico in the coming issues and years to come!

▼ Read more about Raspberry Pi Pico on page 60!



▼ The weekly live streams cover a wide range of computing topics

Digital Making at Home

Join young people from all over the world on the Raspberry Pi Foundation's weekly live stream with a chat, code together, hear from cool people, and see amazing digital making projects from kids just like you.

Subscribe on YouTube to get notifications when we go live, or watch on Facebook, Twitter, and YouTube. And don't forget to get us on your watch list!

OFFICIAL MAKING FROM HOME

Storytelling with technology

Wednesday, 20 January @ 7pm GMT

Your next stream

Join us to get inspired about animation and film-making. Watch and chat live later!

Learning from home

Are there any resources you recommend for children learning computing from home? Our youngest has a Raspberry Pi 400 and is eager to learn more about it!

Wendy via Facebook

The Raspberry Pi Foundation offers a lot of ways to help teach from home if you need to, or even if you just have an eager young maker who wants to do something at the weekend. This great blog entry from Raspberry Pi Foundation CEO Philip Colligan lists all the ways you can help young folk with digital making: magpi.cc/learnhomeblog.

This includes stuff like the Teach Computing Curriculum which has comprehensive lessons plans for all key stages, as well as Isaac Computer Science for A levels. There are also the weekly Digital Making from Home streams, our sister magazine Hello World, and the Raspberry Pi Projects site.

There's a lot to get stuck in to!

Auf Deutsch

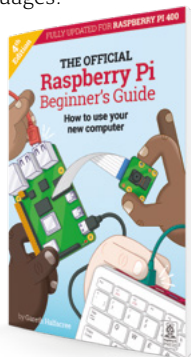
I've just contributed to get the 4th Edition of *The Official Raspberry Pi Beginner's Guide* in English. But, actually, I am looking for the German version. Is it published only as a book or is it also available as a PDF file? Please let me know.

Tim via email

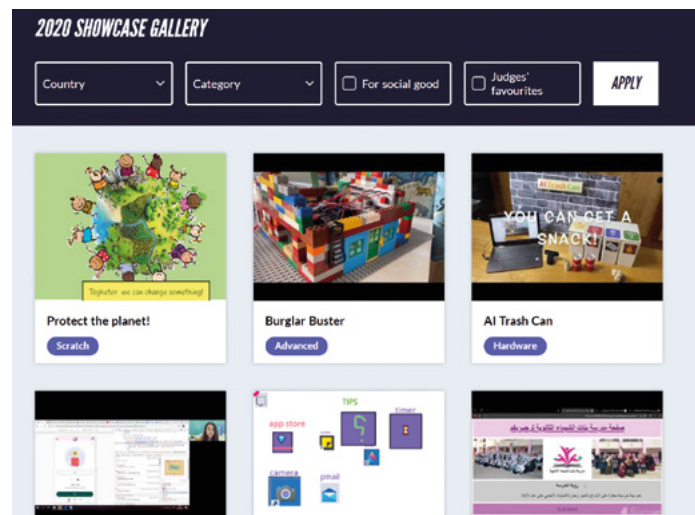
If we've published something via Raspberry Pi Press, then chances are you can get it as a free download! In the case of *The Official Raspberry Pi Beginner's Guide 4th Edition*, where there are multiple translations, you can find a drop-down menu for the language on the product page (which is here: magpi.cc/bguid4). It will even have the description in that language as well!

In addition, you can buy a physical copy of the *Beginner's Guide* in all the available languages.

▶ We think this is the best way to get started with a Raspberry Pi!



▼ Submit your project to be in the online showcase gallery!



Coollest Projects

Will there be any Coollest Projects events in 2021? I'd love to enter a project for it as I have some amazing ideas I'd like to try out!

Pari via Twitter

It was recently announced that there will be another online showcase in 2021 for Coollest Projects! While no date for the actual event has been confirmed yet, you'll be able to register for it, starting on 1 February. All projects will be shown in the online showcase gallery online, so there's competition for space! It's also completely free to enter. Check out more info here: magpi.cc/cp2021.

Contact us!

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

HackSpace

TECHNOLOGY IN YOUR HANDS

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



SUBSCRIBE AND
SAVE UP TO
35%
on the cover price



ISSUE **#39**

OUT NOW

hsmag.cc

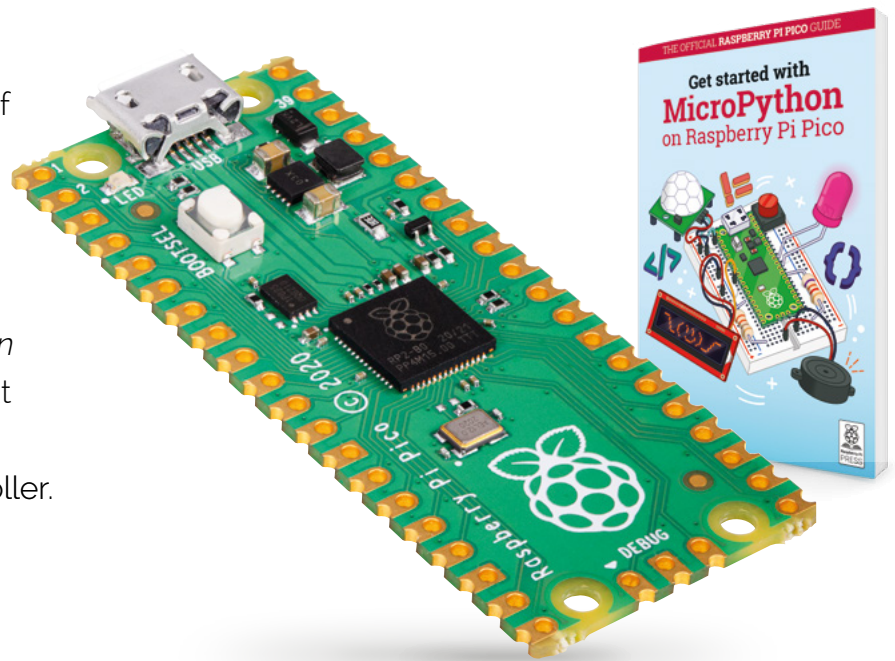


WIN ONE OF TEN

RASPBERRY PI PICO BOARDS

AND GET STARTED WITH MICROPYTHON BOOKS!

To celebrate the release of the amazing Raspberry Pi Pico board, we're giving ten away with a copy of the official *Get Started with MicroPython on Raspberry Pi Pico* book that will help you get stuck in with your new microcontroller.



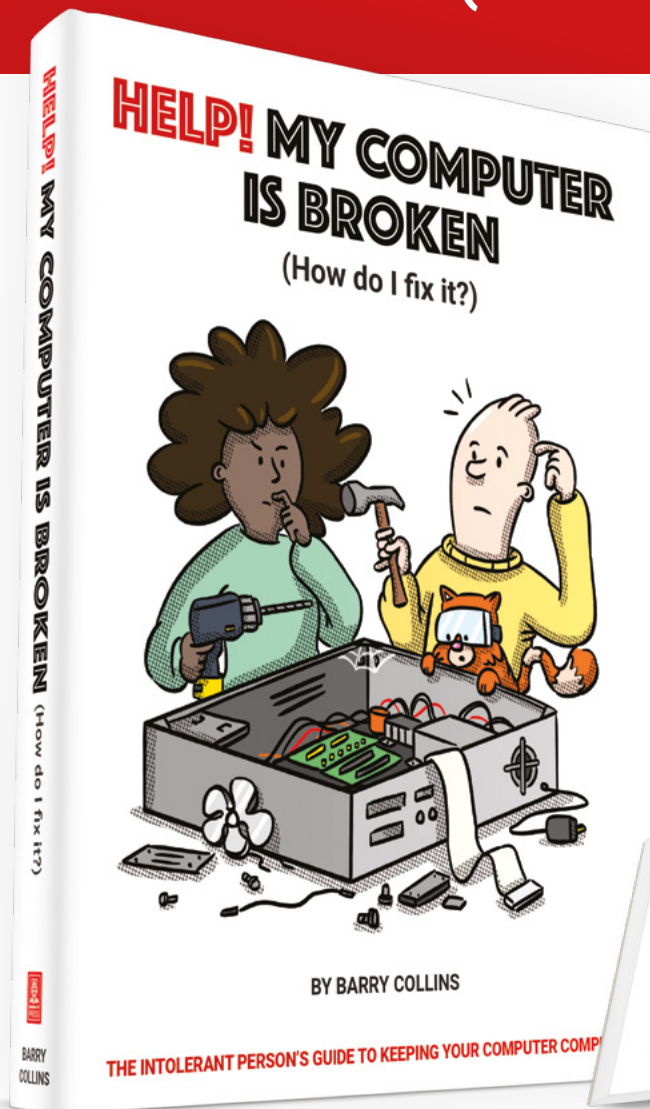
Head here to enter: magpi.cc/win | Learn more: magpi.cc/pico

Terms & Conditions

Competition opens on **27 January 2021** and closes on **25 February 2021**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, 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

#MONTHOFMAKING

GET READY TO JOIN IN WITH
OUR ANNUAL MAKER EVENT

THE MAGPI #103
ON SALE 25 FEBRUARY

Plus!

Home automation
with Raspberry Pi

Start building an
arcade machine

Control traffic
lights with
Raspberry Pi Pico

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 Editors

Phil King and Nicola King

ADVERTISING

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

DESIGN

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Sam Ribbits, James Legg

Illustrator

Sam Alder

CONTRIBUTORS

Alasdair Allan, Mike Cook, David Crookes, PJ Evans, Gareth Halfacree, Martin O'Hanlon, Rosemary Hattersley, Nicola King, KG Orphanides, Laura Sach

PUBLISHING

Publishing Director

Russell Barnes
russell@raspberrypi.com

Director of Communications

Liz Upton

CEO

Eben Upton

DISTRIBUTION

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

SUBSCRIPTIONS

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



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

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

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





There and back again with silicon

Keeping secrets and writing documentation. By **Alasdair Allan**

There is a new thing in the world, and I had a ringside seat for its creation. For me, it started just over a year ago with a phone call from Eben Upton. One week later I was sitting in a meeting room at Raspberry Pi Towers in Cambridge, my head tilted to one side while Eben scribbled on a whiteboard and waved his hands around.

Eben had just told me that Raspberry Pi was designing its own silicon, and he was talking about the chip that would eventually be

telescopes, I became interested in machine learning, and what later became known as ‘big data’. From there, I spent time investigating the ‘data exhaust’ and data living outside the cloud inside embedded and distributed devices, and as a consequence did a lot of work on mobile systems. Which led me to do some of the thinking, and work, on what’s now known as the Internet of Things. Which meant I had recently spent a lot of time writing and talking about embedded hardware.

I was there to help support putting a Coral Edge TPU onto Raspberry Pi was a particularly good wheeze.

Instead, I was going to spend the next year metaphorically locked in a room building a documentation toolchain around – and of course writing about – a totally secret product.

I couldn’t talk about it in public, and I talk about things in public a lot. Only the fact that almost everyone else spent the next year locked indoors kept too many questions being asked.

I’m rather pleased with my first year at Raspberry Pi, and with how my work on RP2040 and Raspberry Pi Pico turned out. Much like a Raspberry Pi is an accessible computer that gives you everything you need to learn to write a program, RP2040 is an accessible chip with everything you need to learn to build a product. It’s going to bring a big change to the microcontroller market. [M](#)

“ I gradually drifted sideways into playing with the toys ”

known as RP2040. Eben started out by drawing the bus fabric, which isn’t where you normally start when you talk about a new chip, but it turned out RP2040 was a rather unusual chip.

I get bored easily. I started my career doing research into the high-energy physics of collision shocks in the accretion discs surrounding white dwarf stars, but I gradually drifted sideways into playing with the toys. After spending some time working with agent-based systems to solve scheduling problems for robotic

Eben was looking for someone to make sure the documentation around Raspberry Pi Pico, and the silicon itself, was going to measure up. I took the job.

Rumour mill

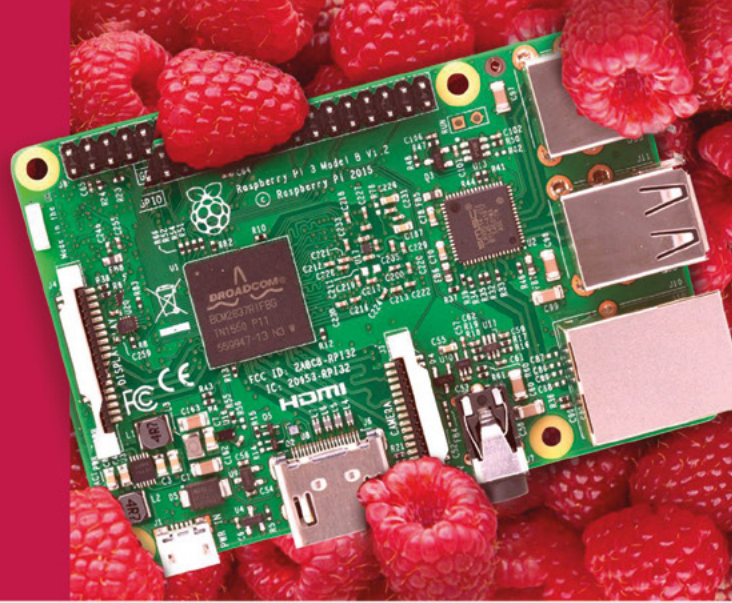
I had spent the previous six months benchmarking machine learning (ML) inferencing on embedded hardware, and a lot of time writing and talking about the trendy new world of TinyML. The rumours of what I would be doing for Raspberry Pi started flying on social media. The idea that

AUTHOR Alasdair Allan

Alasdair Allan is a scientist, author, hacker, maker, and journalist. He writes, speaks, and builds. He accidentally ended up as a privacy advocate, and is now responsible for documentation at Raspberry Pi.

alasdairallan.com

American Raspberry Pi Shop



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

Partner and official reseller for top Pi brands:



and many
others!

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

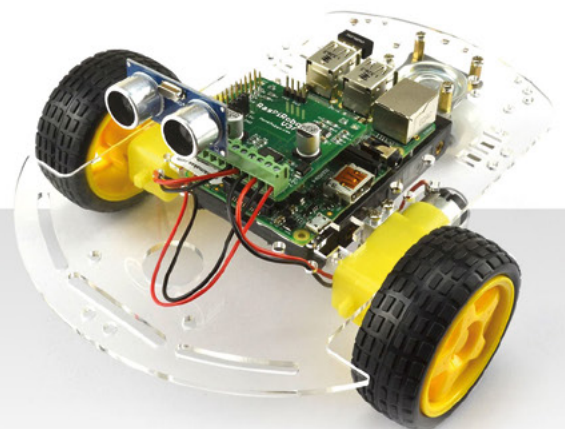
USA



Canada



BuyaPi.ca



Raspberry Pi

APPROVED RESELLER



CHECK OUT THE NEW OKDO PROJECT HUB!

Find out more at www.okdo.com/projecthub



Find inspiring projects created by our customers and engineers that show you how to build and programme cool projects and how to get started with some of your favourite boards!

Find out more about our full product range at:

www.okdo.com

