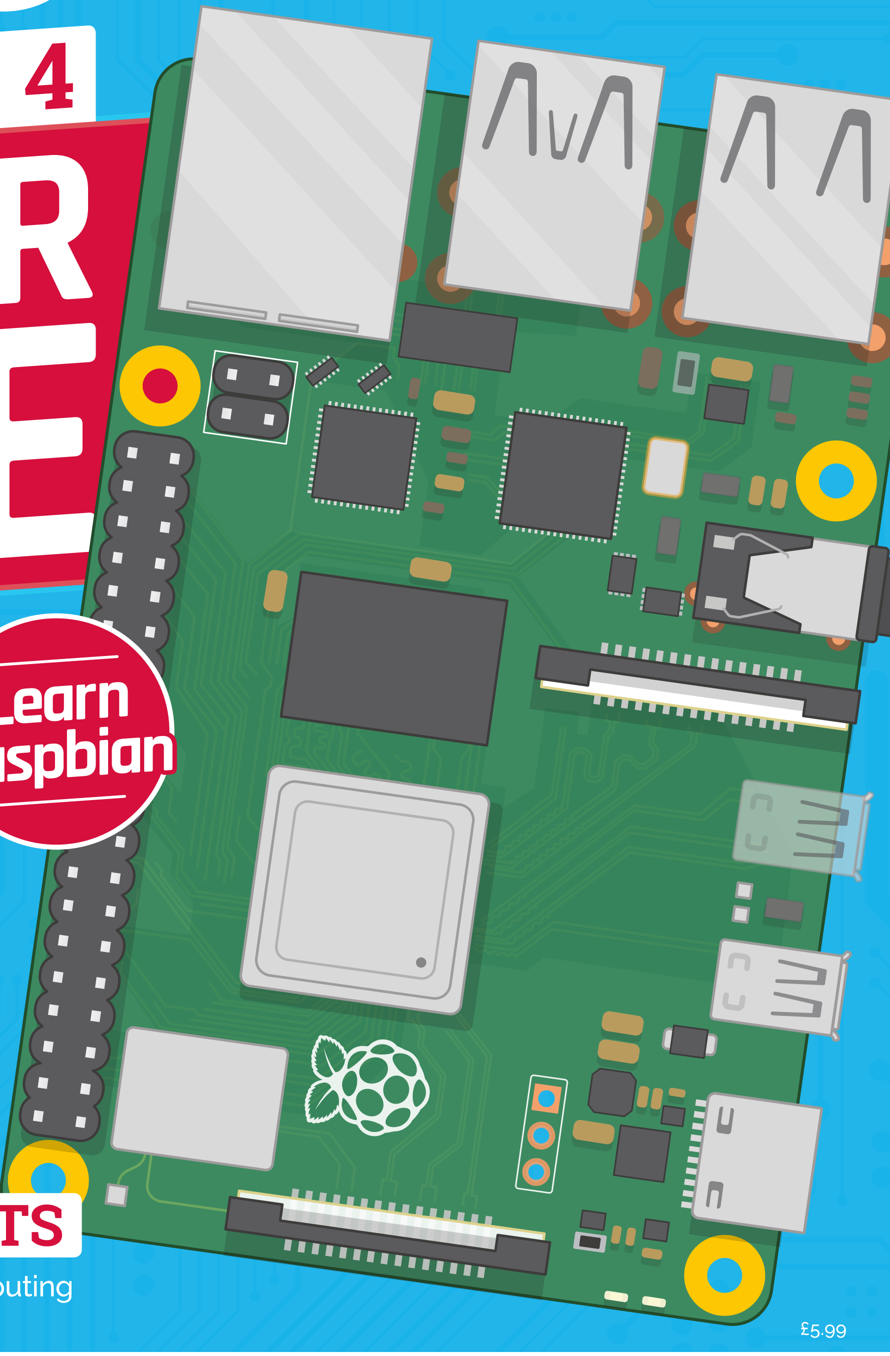
The MagPi

Issue 84 | August 2019 | magpi.cc

The official Raspberry Pi magazine

RASPBERRY PI 4 STARTER GUIDE

- Get going quickly
- Discover epic projects
- Start coding & making



Learn Raspbian

BUILD A LOW COST ROBOT

Discover robotics with everyday parts

THE BEST SUMMER PROJECTS

Fun in the sun with portable computing

£5.99

30+ PAGES OF TUTORIALS

Raspberry Pi PRESS

WIN! 1 OF 10 AMAZING JAM HATS
The MagPi
Issue 82 | June 2019 | magpi.cc
BUILD A PI KEYRING
Keep your computer with you at all times
LEARN CO
WITH SCRA
PLUS!
Build a MIDI sequencer
Model railway projects
Add NeoPixel lighting to a display
THE BEST AI LEARNING
WIN! 1 OF 10 RASPBERRY PI 4 KITS
The MagPi
Issue 83 | July 2019 | magpi.cc
Introducing Raspber
BENCHMARKS
ENGINEERING INTERVIEWS
TECH SPECS
40 PAGES OF PROJECTS
YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Issue 84 | August 2019 | magpi.cc
RASPBERRY PI 4
STARTER GUIDE
Get going quickly
Discover epic projects
Start coding & making
Learn Raspbian
BUILD A LOW COST ROBOT
Discover robotics with everyday parts
THE BEST SUMMER PROJECTS
Fun in the sun with portable computing
30+ PAGES OF TUTORIALS

WELCOME
to *The MagPi* 84

I hope you're enjoying Raspberry Pi 4. We love the programmable computer and believe it's capable of becoming the main desktop machine in the home. Starting from $35, that's a big attraction. And lots of people are discovering Raspberry Pi for the first time.

Our Raspberry Pi Starter Guide (page 24) is designed to help these newcomers with Raspberry Pi 4, and even long-standing Raspberry Pi users will learn something new.

More importantly, the whole of *The MagPi* magazine is designed to get you beyond the board and into building projects. So, if you've ever fancied making a low-cost wheeled robot, now is your chance (page 36). And, if you are more advanced with robots, we've got a tutorial on using CircuitPython to control servos and motors (page 44).

What about retro gaming? We look at how to build a space shooter with PICO-8 (page 54). Advanced makers might want to check out the Squeeze Controller (upcycled from an old hand-dynamo). This month, Mike Cook uses that controller to design an LED racing game (page 58).

And last, but definitely not least: Demolition Man is a hilarious project by PJ Evans that uses AI to detect when you say naughty words, and prints out a ticket on a thermal printer (page 50). There's lots of fun to be had with a Raspberry Pi and the right project.

The point of Raspberry Pi isn't just to build a low-cost computer. It's what you build with that computer. Make something amazing and share it with the rest of the community.

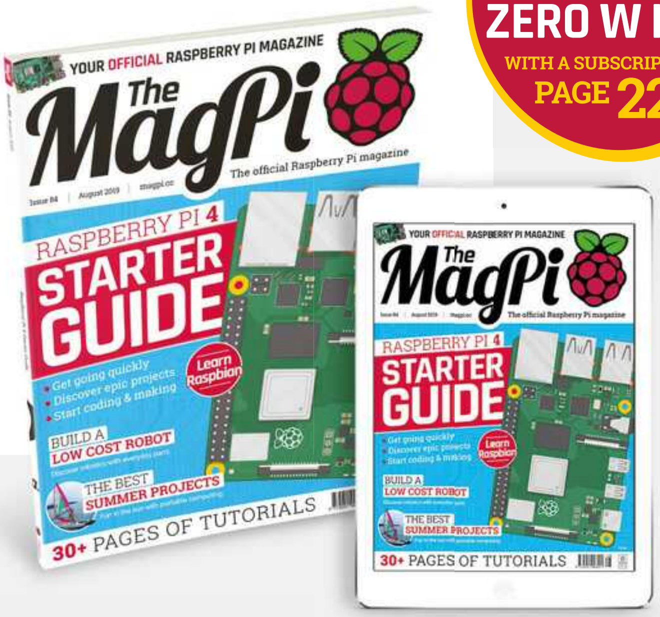
Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Editor of *The MagPi*. Tunnels from a terminal like an over-caffeinated mole. Does not play Mario Maker. Is on a rickety seaplane holiday, so may not make it back next month.

magpi.cc



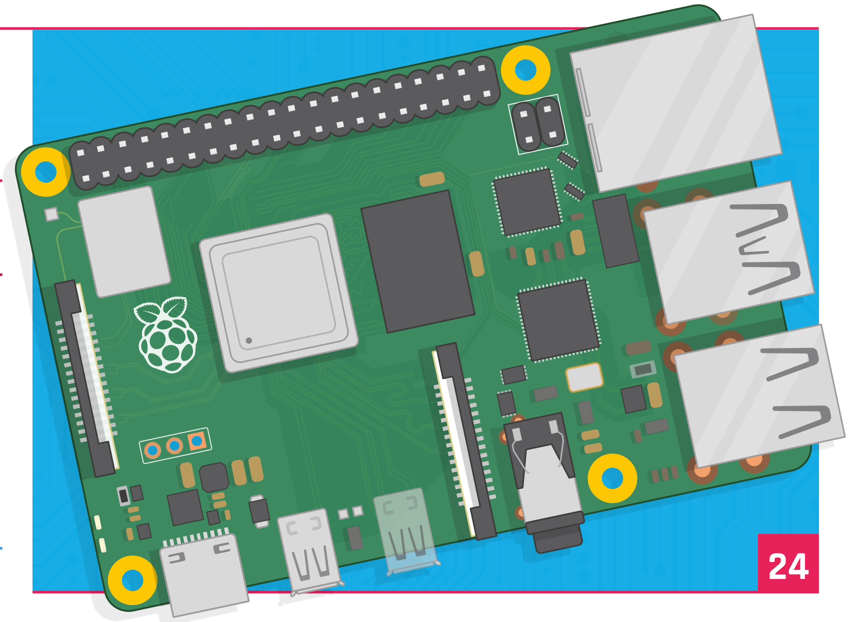
Contents

Issue 84 August 2019

Cover Feature

Regulars

Project Showcases



24

10

Smart Home Herb Garden

20

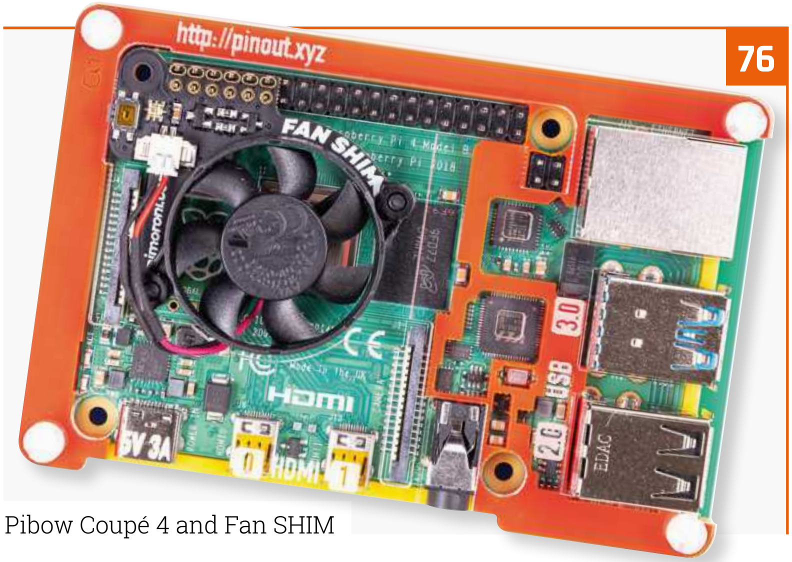
```
**** CBM BASIC V2 ****
28159 BYTES FREE
READY.
```

PIC-20

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.

Learn Lua with PICO-8

LED Racer – Pi Bakery



Pibow Coupé 4 and Fan SHIM

Anne Carlill interview

Tutorials

The Big Feature

Scorching Summer Projects

Reviews

Community

WIN ONE OF THREE **RASPBERRY PI 4** DESKTOP KITS

Entries open for Pi Wars 2020

Raspberry Pi fans get to design fiendishly challenging courses to foil ever more sophisticated robots for next year's Pi Wars. Fantastic news, thinks **Rosie Hattersley**

WELCOME TO
Pi WARS
THE RASPBERRY Pi ROBOTICS CHALLENGE COMPETITION
WARNING! WARNING! WARNING!
ROBOTS AT WORK!

Malaysia's Team MYRA celebrate their Pi Wars 2019 win

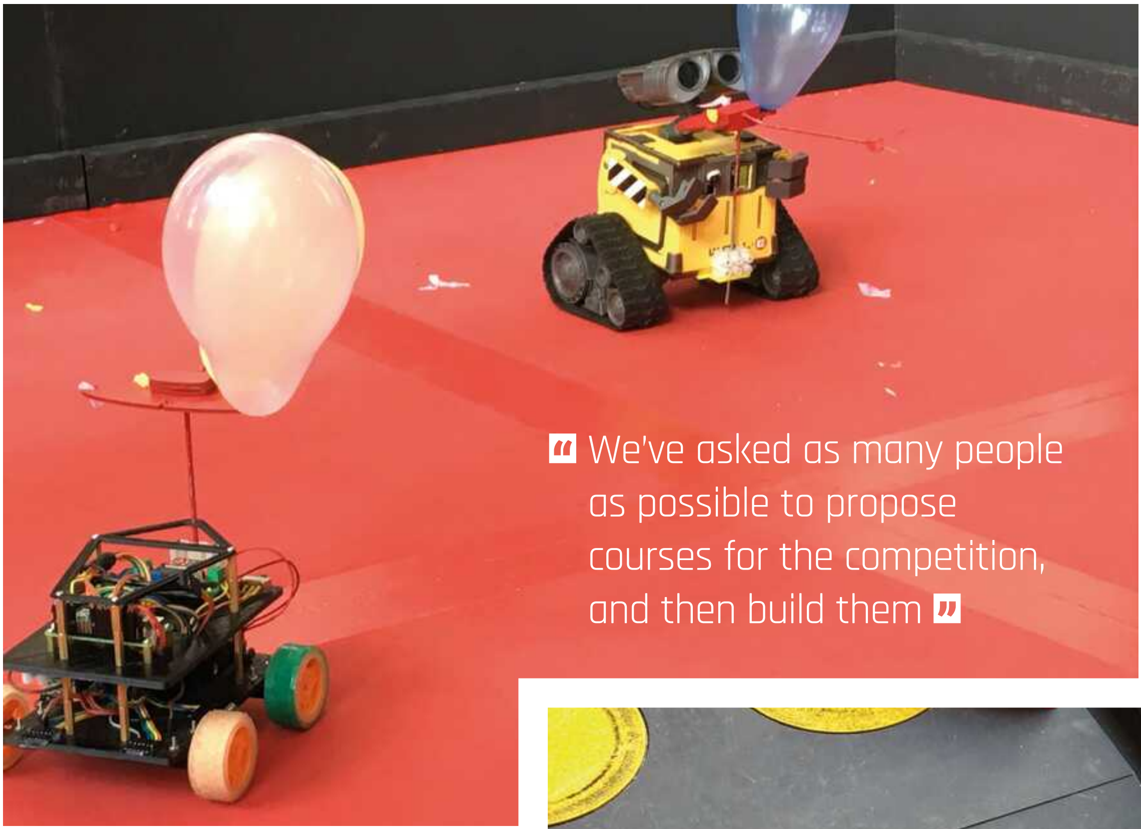
It may be the height of summer, but things are already well under way for a firm fixture on the Raspberry Pi calendar: Pi Wars 2020. The sixth Pi Wars event is due to take place in May 2020. Entries open at the start of August and must be in by 12 September.

Course favourites Pi Noon – in which robots have to burst a balloon on their opponent's robot using a spiked probe – and the Obstacle Course will return, the latter with new obstacles.

This Pi Wars is being opened up to wannabe track engineers. Most Pi Wars challenges are built by event co-organiser Tim Richardson, and he's keen to open up the course design to the rest of the community. Brilliantly, this means the rest of us get to pitch ideas for courses, and even offer our expertise in building them.

On course for success

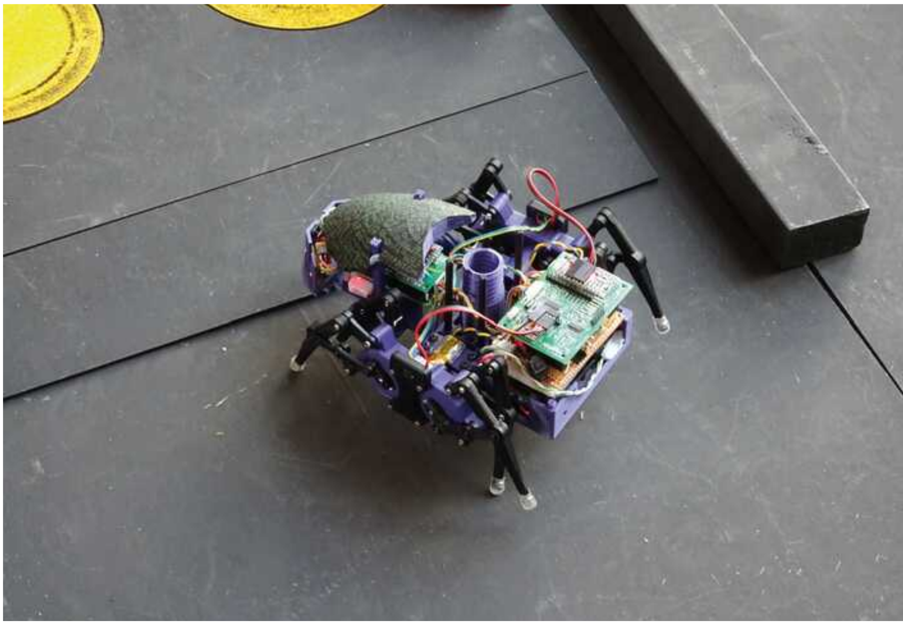
Pi Wars co-organiser Mike Horne says, "What makes 2020 special is that, for the first time, we are asking the community to contribute to the event by proposing and building courses.



> We've asked as many people as possible to propose courses for the competition, and then build them

Robots duke it out in the Pi Noon Pi corral

A walking robot tackles the Obstacle Course



At Pi Wars 2019, a special presentation was made to co-organisers Mike Horne and Tim Richardson

We've asked as many people as possible to propose courses for the competition, and then to build them."

This will also free up the indefatigable Tim to work on fiendishly clever new advances and surprises for Pi Wars 2020.

Previous events filled up very quickly, so get your skates on if you'd like to apply. There are two categories: School and Kids' Clubs teams on the Saturday, and then All Other Teams on the Sunday. To keep things competitive, teams entering are also split by levels of prior Pi Wars experience. "We've found the format works well, allowing us to have a good spread of teams," says Mike. Volunteers can also apply via the Pi Wars website: **piwars.org**.

Pi Wars 2019 featured some very clever, and cute, robots

Image Credits: Christopher Parrot, SimUK, Team MYRA, Mark Mellors, Colin Grant

OKDO Pi4 SLIDE CASE
Protective case with a smooth slide action. Available in Black, White or Clear

PWR
HDMI
HDMI
Audio

OKDO Pi4 POWER SUPPLY
Start straight from the box with a power supply that easily adapts to all international sockets.

okdo™
DESIGN THE WORLD

Smart Home Herb Garden

Growing herbs using Google's Smart Home API makes for automated flavour in your cooking. **Rob Zwetsloot** grabs a bunch

▲ Each herb can be maintained individually, so you won't overwater one plant and underwater another

MAKER

Oscar Prom

Software team lead at Deeplocal. He plans, develops, and deploys a variety of full stack and cloud software systems.

magpi.cc/pNpRxP

Warning!
Mains electricity

This project uses mains power. Be careful if you plan to recreate it

If you've ever grown herbs in your kitchen, you may have encountered some problems. Coriander flopping about everywhere. Rosemary never really regrowing. Basil growing out of control. Then you leave the house for a few days and come back to withered herbs. It's tricky! This is where something like the Smart Home Herb Garden from Oscar Prom at Deeplocal comes in handy.

"The herb garden was built for Google I/O 2019 to showcase the Smart Home API and some newly released traits on the IoT platform," Oscar explains. "We released it as a DIY project to encourage developers to use it as a jumping off point for their own Smart Home projects."

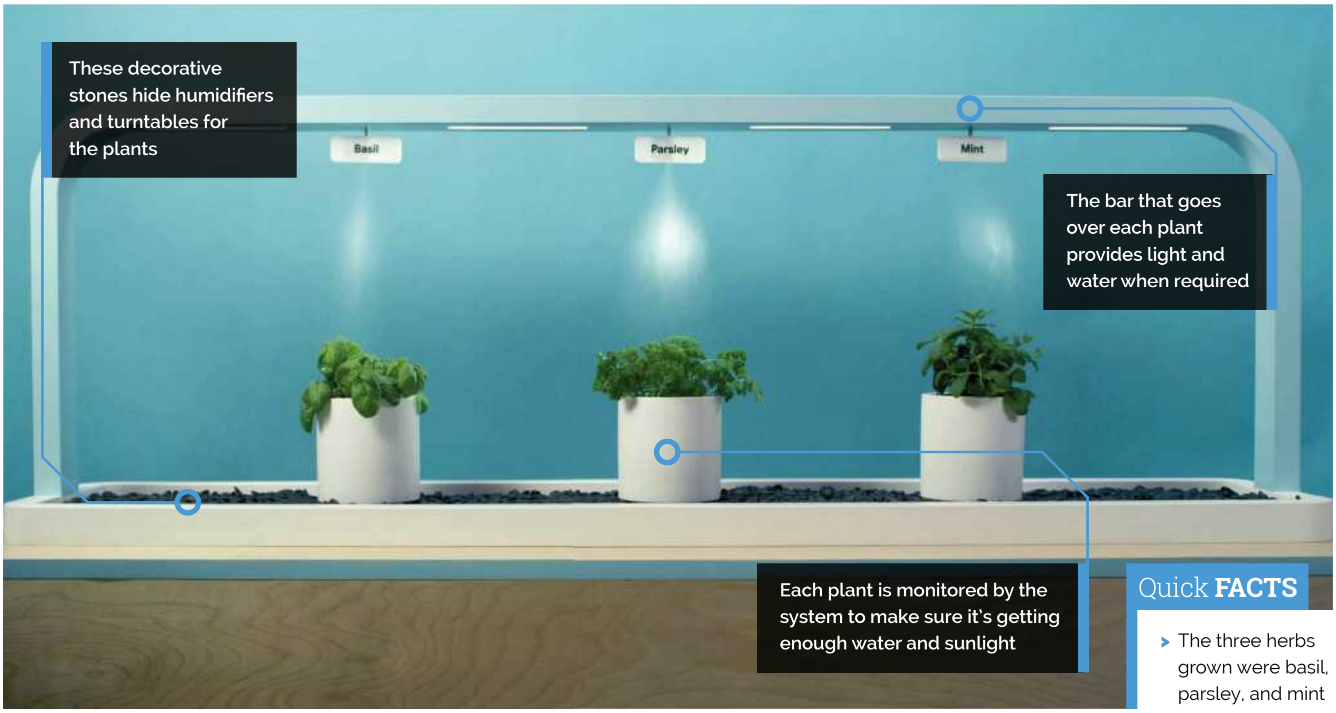
▲ Testing the mister before embedding it in a shelf is a good idea whatever project you're working on

Automated gardens are all the rage now – we've had farm robots, hydroponics, and aquaponics in *The MagPi* – so scaling it down to a small herb garden seems like a logical next step. So, when Deeplocal were asked to build a Smart Home project using voice control, it's the route they decided to take.

> "Three potted herbs sit under a beam that has lights and water misters"

Voice-activated care

The system is deceptively simple. Three potted herbs sit under a beam that has lights and water misters. There's also a humidifier on the tray that the plants sit on, and each plant can rotate to make for easier pruning and watering hard-to-reach



These decorative stones hide humidifiers and turntables for the plants

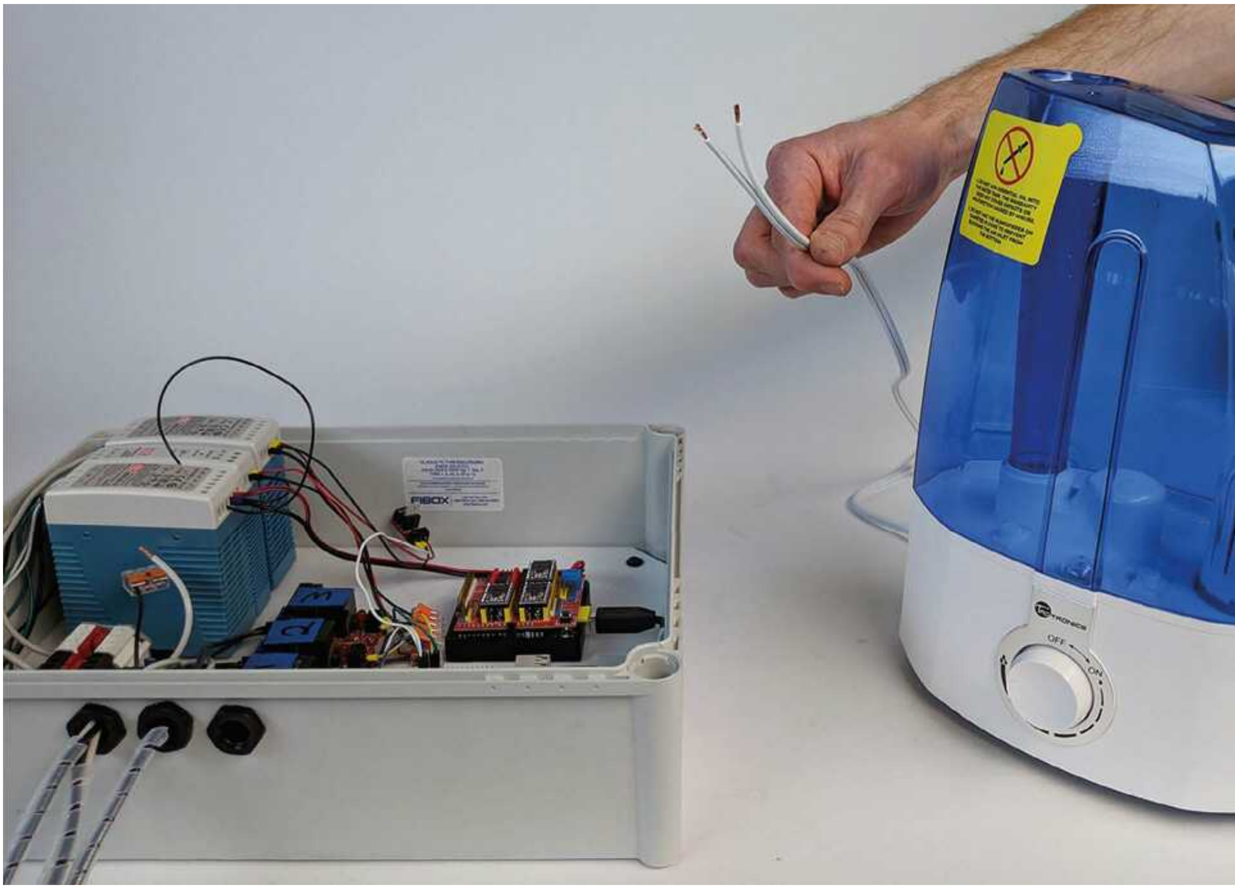
The bar that goes over each plant provides light and water when required

Each plant is monitored by the system to make sure it's getting enough water and sunlight

Quick FACTS

- The three herbs grown were basil, parsley, and mint
- The team unfortunately no longer have the garden
- It does have voice control, but is otherwise fully automated
- The system is written in JavaScript
- Find full build instructions here: **magpi.cc/fCPqeP**

The Smart Herb Garden was created for Google I/O 2019 – here it is on display there!



▶ A store-bought humidifier needs to be modified to work in the system

▼ The humidifiers release spooky water vapour from the rocks

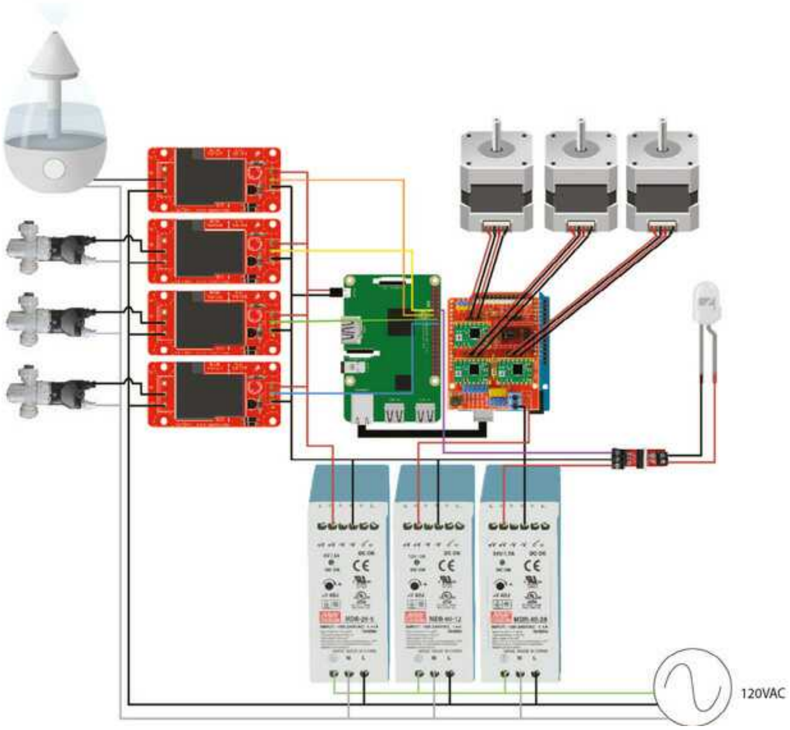
> “It automatically rotates the plants to distribute sunlight evenly”

areas. There’s even a special function that lets you ‘spotlight’ a specific plant if you want to really show off your prize parsley.

“Raspberry Pi provides a familiar and inexpensive platform to get any project off the ground,” Oscar tells us. “We needed something low-power and internet-connected that could control custom hardware, and there is no dev board that hits that sweet spot better than a Raspberry Pi.”

No growing pains

After having tried our own hand at growing herbs in the past, we had to ask about the project’s herb-growing prowess: “It’s much better than a human!” asserts Oscar. “It remembers to water the kitchen herbs without issue and automatically rotates the plants to distribute sunlight evenly. We



▲ It looks a little complicated, but it's not too bad really

◀ Don't make our mistake: make sure to prune your herbs!

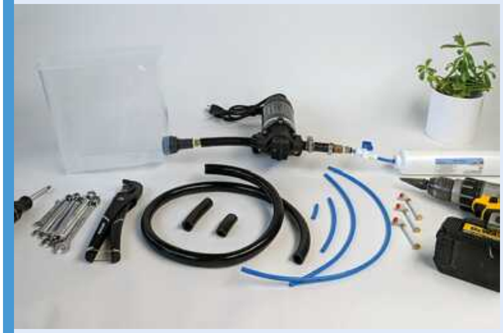
can even increase the brightness of the grow lights on our cloudy Pittsburgh days (read: often)."

This isn't Deeplocal's first rodeo with Raspberry Pi either, and it seems like the team specialise in amazing home improvement projects.

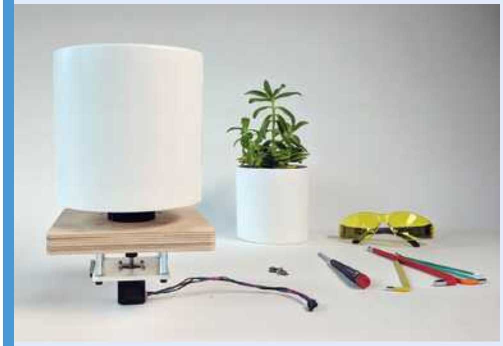
"We've built a [Raspberry Pi-powered], voice-controlled drink mixer and an all-in-one button that starts Netflix, turns off your lights, and orders takeout," Oscar says. "Not to mention, a lot of our prototypes are built on top of Raspberry Pi boards because we can iterate so quickly."

While the herbs in their garden grew well, they never got to use them in a meal. Maybe next time.

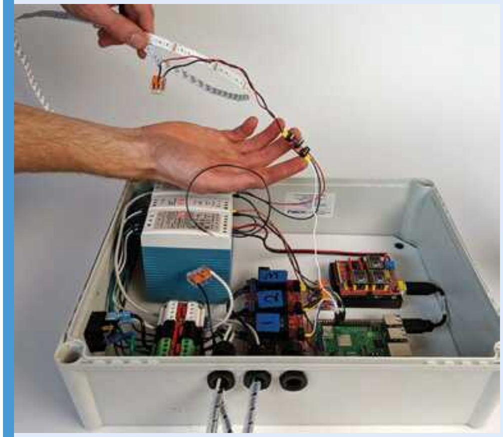
Building a smart herb garden



01 A water pump system needs to be properly put together to provide water for the herbs. The humidifier system is handled by an off-the-shelf humidifier controlled by a relay, so it doesn't use this system.



02 The pots use magnets to snap to a custom rotation device that is set up in three sections along the bottom of the herb garden. Each of the rotators is controlled by a simple electric motor, and they use some 3D-printed parts.



03 Lights are attached and can have their brightness controlled so that they give the perfect amount of light to the plants. This project uses white LEDs, although grow bulbs are more standard practice.

Marvin Go-Kart

Mark Cantrill's new mode of transport suits his go-kart-loving daughters to a tee, as **David Crookes** discovers

MAKER

Mark Cantrill

An electronics engineer, husband, and father-of-two who is usually found running the Cotswold Raspberry Jam's MicroPiNoon arena. He's known for creating the PiZ-Moto motor driver for Raspberry Pi Zero and an FPGA board for the original Raspberry Pi.

@AstroDesignsLtd

Golf and go-karting would appear to be at opposite ends of the sporting spectrum, but when Mark Cantrill swung by an ageing electric golf trolley, it set the wheels in motion for a rather unique idea.

Having originally considered using Raspberry Pi to control the trolley and eventually turn it into a Dalek or a rough-terrain garden explorer fitted with a camera, he began to ponder how it could make his life a bit easier.

"I thought I'd use it to power a go-kart and save my back from dragging my kids around the garden," he laughs. But first he had to figure how to get Raspberry Pi to control the trolley, so, after fitting a new 12 V car battery, he began to look at the device's radio-controlled mode with a view to reverse-engineering it.

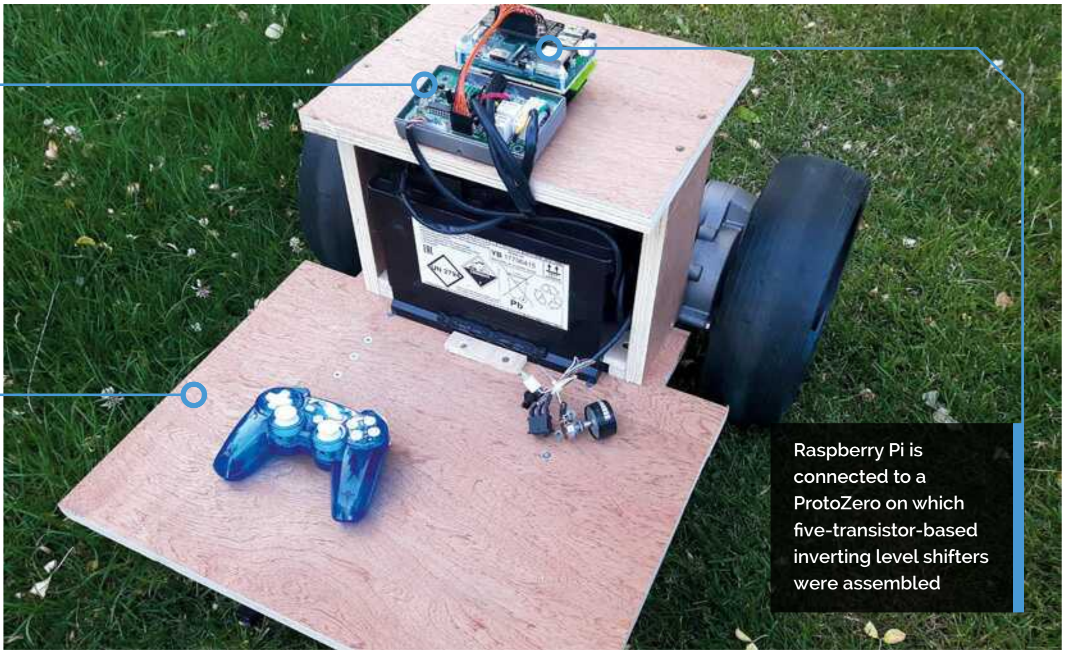
Go, go-kart

"The trolley could be made to go forwards, backwards, left, and right using a five-button remote control, and the fifth button would bring it to a stop," Mark explains. He soon noted that the radio receiver was a plug-in option, with a seven-pin connector joining it to the main speed-controller micro.

"I thought it would be possible to remove the radio receiver and replace it with Raspberry Pi," he says. "Since the motor controller on the golf trolley was 5 V, the 3.3 V outputs from Raspberry Pi needed level-shifting, which I achieved by using a 2N2222 NPN transistor. Five transistor-based inverting level shifters were quickly assembled onto a ProtoZero board."

At this point, Mark began to write the software and he was able to call upon his experience of running MicroPiNoon robots at the Cotswold Raspberry Jam. "The golf trolley had two motors, two wheels, and a front stabiliser, so it was essentially very similar to the MicroPiNoon robots, just a bit bigger.

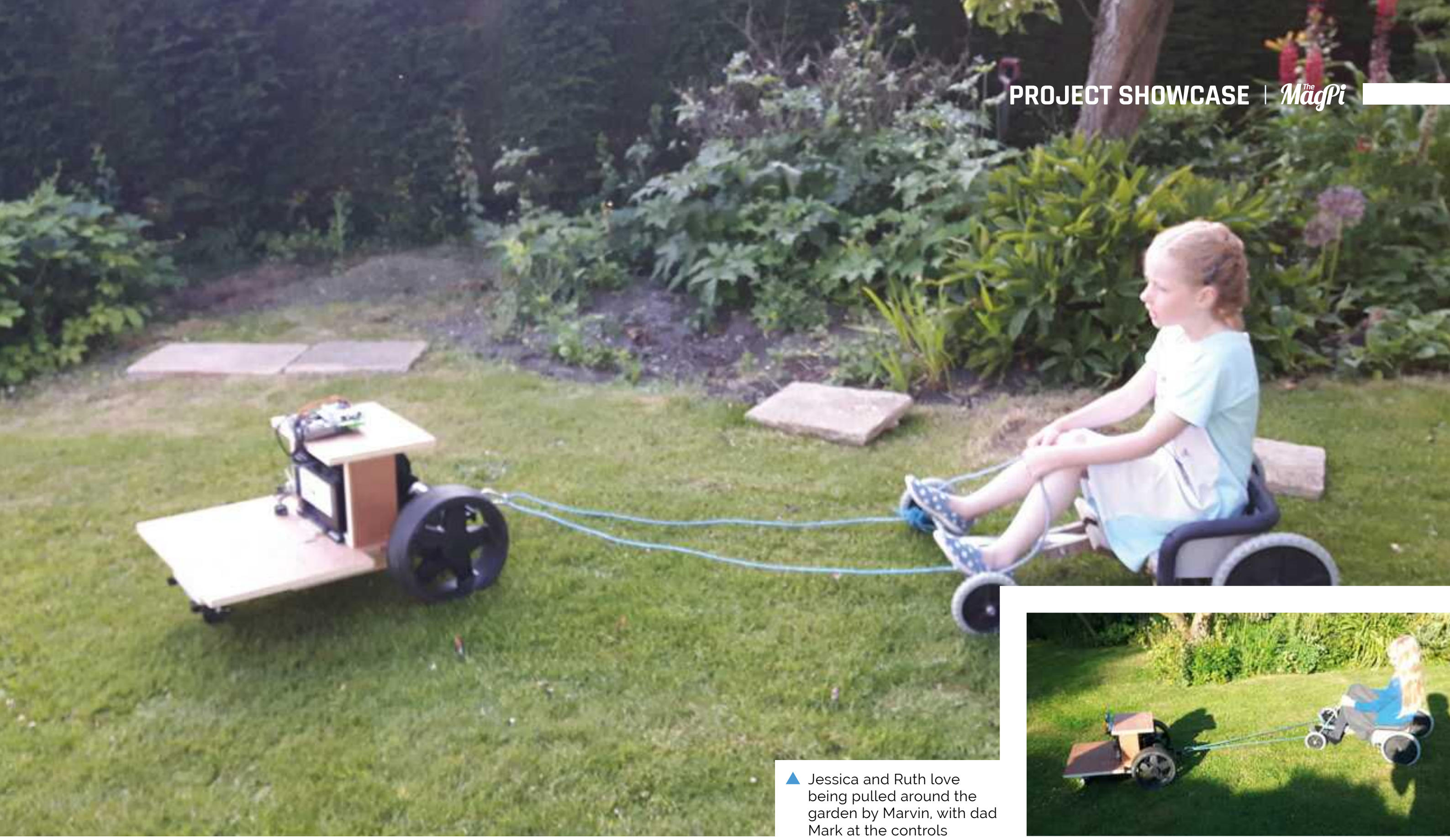
"All I needed to do was replace the functions that usually control the EduKit 3-compatible PiZ-Moto pHAT with functions to drive the 5-bit output

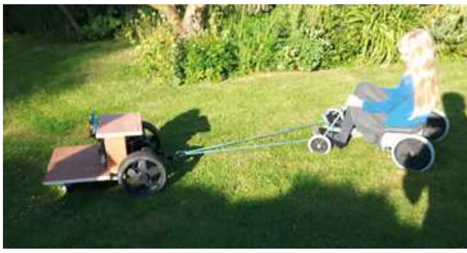


The robot features the golf trolley's original motor controller, which has a handy on-board 5 V power supply

Mark is building a new casing for the towing robot, with an extra wheel fitted beneath the base

Raspberry Pi is connected to a ProtoZero on which five-transistor-based inverting level shifters were assembled





▲ Jessica and Ruth love being pulled around the garden by Marvin, with dad Mark at the controls

> " I thought I'd save my back from dragging my kids around the garden "

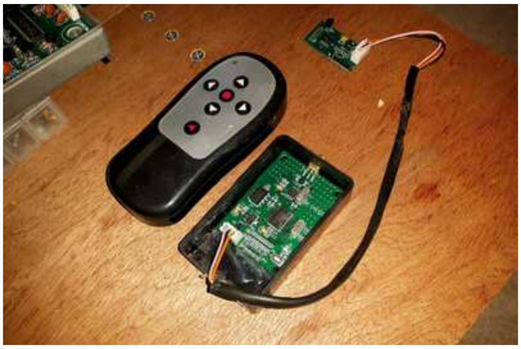
that goes to the golf trolley's motor controller via the level shifter. I was able to reuse my existing code to do this."

Be in control

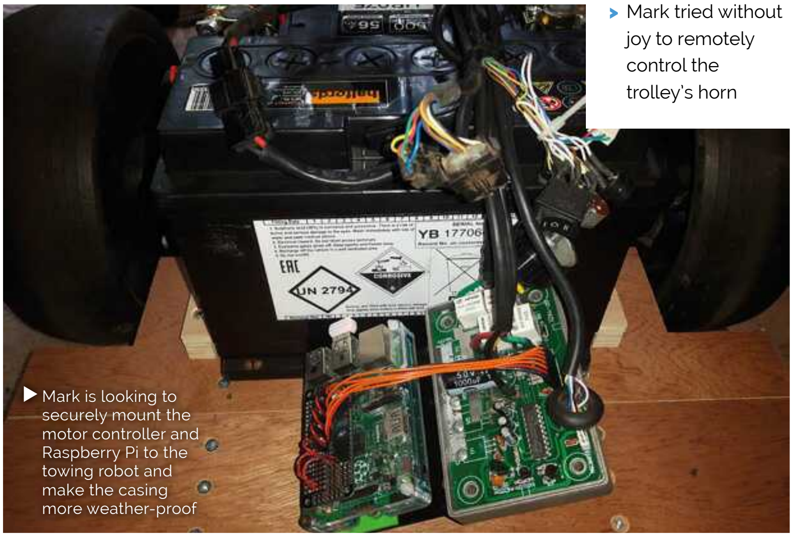
Mark took the enhanced prototype golf trolley to Raspberry Fields last year and allowed people to play around with it. At this point, the trolley wasn't attached to a go-kart, but it highlighted a particular problem: the PlayStation 3 controller Mark was using – a model made by Rock Candy – would shut down after a few minutes of no activity.

"It would then search for something to communicate with, but this wouldn't necessarily be the last thing it was talking to," Mark says. "It meant that it would sometimes find another robot – on one occasion it hooked up with a device called X-Bot, sending it crashing off a table."

The controller problem remains unresolved and Mark says he also needs to work on better acceleration controls. "I'm also working out how to get the controller to slow down gradually instead of stopping abruptly," he adds. But when everything is working well, Marvin is a joy. The robot can whizz around the garden with a go-kart in tow, to the great excitement of his daughters. "They love it," Mark concludes.



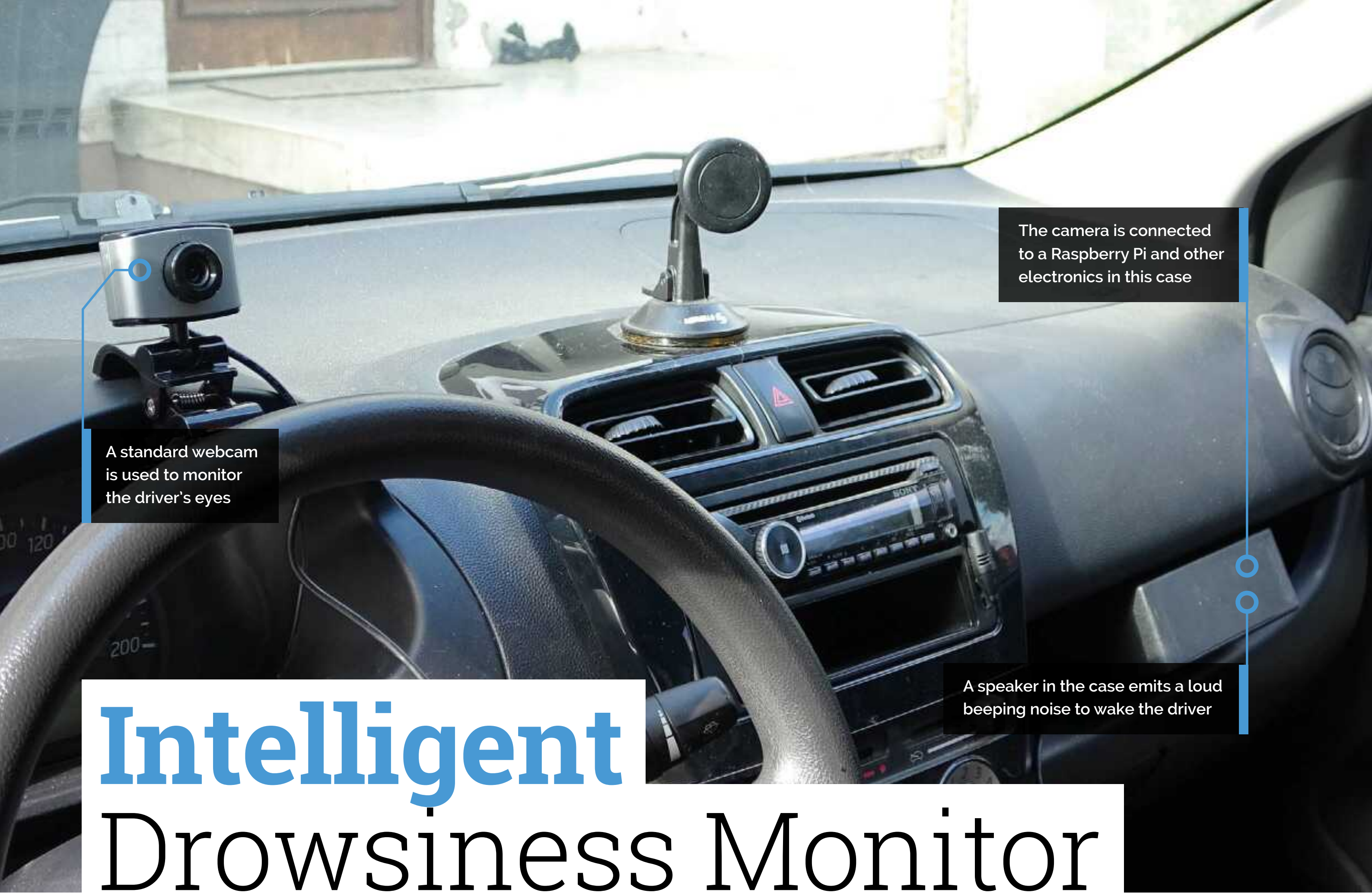
▶ The golf trolley's radio receiver was reverse-engineered so Marvin could replicate the instructions it issues using Raspberry Pi instead



▶ Mark is looking to securely mount the motor controller and Raspberry Pi to the towing robot and make the casing more weather-proof

Quick FACTS

- Marvin has a Raspberry Pi Model B+
- A PlayStation 3 gamepad provides wireless control
- The interface was built on a ProtoZero board
- The project cost about £110
- Mark tried without joy to remotely control the trolley's horn



The camera is connected to a Raspberry Pi and other electronics in this case

A standard webcam is used to monitor the driver's eyes

A speaker in the case emits a loud beeping noise to wake the driver

Intelligent Drowsiness Monitor

A clever device that detects if you are sleepy behind the wheel, and gives you a loud wake-up call. **Nicola King** finds out more

MAKER

Luis Oliver and Andre Hernandez

Two Mexico City-based makers. Luis is also a founder of **blankit.com.mx** – aiming to create a comprehensive IoT platform for healthcare applications.

magpi.cc/ckywoP

A road accident after falling asleep at the wheel prompted Luis Oliver, and his friend Andre Hernandez, to develop a driver drowsiness monitoring system that can work in any vehicle. "Luckily it was a small accident," Luis tells us, "but I realised that the next time it happened, it could be my last. So I decided to try and create a smarter solution than just a cup of coffee."

The camera can be mounted in various positions, but should point at the driver's face

Following some initial research, Luis turned his attention to creating a physical device that could detect if a driver was falling asleep. "I came to the resolution to create a system that continuously monitors my conscious state." Once he'd settled on the concept for the system, he began to research which hardware and software he was going to use. "The obvious choice was a Raspberry Pi as the main processor of my project, as it needed to be small and portable."

Wake up sleepy-head!

Over the next couple of weeks, and following a number of successful tests in front of computers and eventually in vehicles, the Intelligent Drowsiness Monitor was born, incorporating a crash notification system, but how does it work exactly? Luis and Andre describe the monitor as a kind of 'guardian angel'. Luis elaborates, "It is continuously monitoring your face at all times. If it notices that you are falling asleep, at that time it will emit a loud sound – which, we can assure you, will wake you up. However, if an accident nevertheless happens, the monitor will send an alert at that time to family or friends

Andre tests out the system by closing his eyes – note that for safety reasons, he's parked in his driveway!

> I personally see great potential for this type of technology to be included in all cars

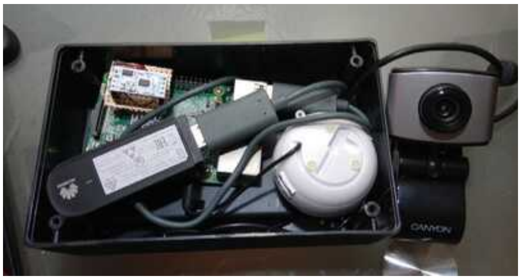
and to emergency services, if the user opts to use that functionality."

Luis says the most complicated part of the design was installing the OpenCV computer vision library (used to analyse the camera images) on Raspberry Pi: "You have to perform a complete build to make it work correctly." Details can be found on the project's Hackster.io page (**magpi.cc/ckywoP**). "After that, we proceeded to develop the algorithm to monitor drowsiness by vision and notifications in real time through GSM."

Luis and Andre had to debug the vision algorithm regularly, and tweak the sensitivity parameters of the monitor, but it was important that "this process was carried out exhaustively so that the test on the car [would] work without problems [or] errors," says Luis.

In the driving seat

While detecting driver behaviours/tiredness is not a new concept, as some luxury car manufacturers do already have similar systems in some of their vehicles, Luis and Andre think that such technology should be open to all, as Luis tells us: "I personally see great potential for this type of technology to be included in all cars, not just luxury cars – it is an economic system that can be easily installed, does not consume much energy, and takes care of you at all times."



A Raspberry Pi, powered by a battery pack, is connected to the camera, speaker, an accelerometer, and mobile data dongle

He adds, "In fact, we are in talks with manufacturers in our country… We will try to improve the prototype with the new Raspberry Pi, and thus be able to make a proper pitch, and see if they would be interested in working with us to develop a pilot or some test on their vehicles."

We certainly wish the pair luck with that endeavour. Safety on the world's roads is such an important issue, and anything that helps to improve our protection can only be a good thing.

Quick FACTS

- OpenCV is used to analyse the camera image and detect closed eyes
- If the driver's eyes are closed for three seconds, an audible alert is triggered
- A VMA204 accelerometer will detect a possible collision…
- …In which case, a crash alert is sent to family or emergency services
- The notification system uses a Soracom mobile data dongle and AWS IoT

Smart Window Fan

When you need to keep your home cool during the summer months, a smart window fan could be just the thing. **Phil King** enjoys a breath of fresh air

MAKER

Ishmael Vargas

From the Chicagoland area, Ishmael has a background in industrial electronics and electrical systems. He likes to experiment with Raspberry Pi and Arduino.

magpi.cc/kzibLu

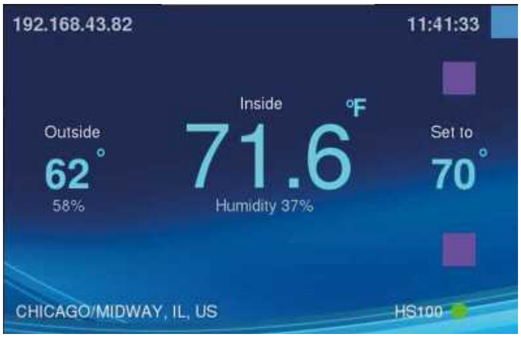
Summer days, and nights, can be uncomfortably hot and humid in the Chicago area. As the sun goes down, the outside temperature drops, but homes may remain hot. This is where a window fan comes in useful, blowing cooler air into the house. Last summer, Ishmael Vargas was using a small window fan upstairs and, after turning it on in the afternoon, he found he had to get up in the middle of the night to turn it off. "That is when I thought there must be a better way to control this fan," he recalls, "and I started putting this project together."

As he was already using a DHT22 temperature and humidity sensor for another project, he opted to use that, connected to a Raspberry Pi Zero running a Python program, to monitor the room temperature. This is then compared with the external temperature; if the latter is cooler, the window fan is turned on via a smart WiFi power plug (TP-Link HS100) – a much simpler method than wiring the fan up to a relay.

> If the external temperature is cooler than that inside, the window fan is turned on via a smart WiFi power plug

Weather report

To keep things simple, Ishmael opted to source the outdoor temperature from Weather.com (The Weather Channel) using the pywapi Python library, rather than wiring up an external sensor. "The temperature provided by Weather.com as compared to the temperature in my car could differentiate by one or two degrees. This is close enough for this project," he explains. "In other parts of the world or rural areas where they do not have as many weather stations, an outdoor sensor might be required."



192.168.43.82 11:41:33

Outside 62° 58%

Inside 71.6 °F Humidity 37%

Set to 70°

CHICAGO/MIDWAY, IL, US HS100

▲ Viewable via VNC on a smartphone, the program window features temperature data and control buttons

One issue he discovered was that in the early morning, the fan could end up blowing warm air into the house. "Depending on the size of the fan, the size of the room, and house materials, the inside temperature might never be as cool as outside," he says. "For example, if the temperature outside is 65°F (18°C), the temperature inside might only drop to 67°F (19.5°C) through the night. As the temperature outside starts to climb, you want to keep the fan off." This resulted in him adding an 'inhibit' mode to turn the fan off at 6am.

Remote control

Rather than having the fan program run automatically on bootup, Ishmael opted to

A smart WiFi socket is used to turn the window fan on and off

Quick FACTS

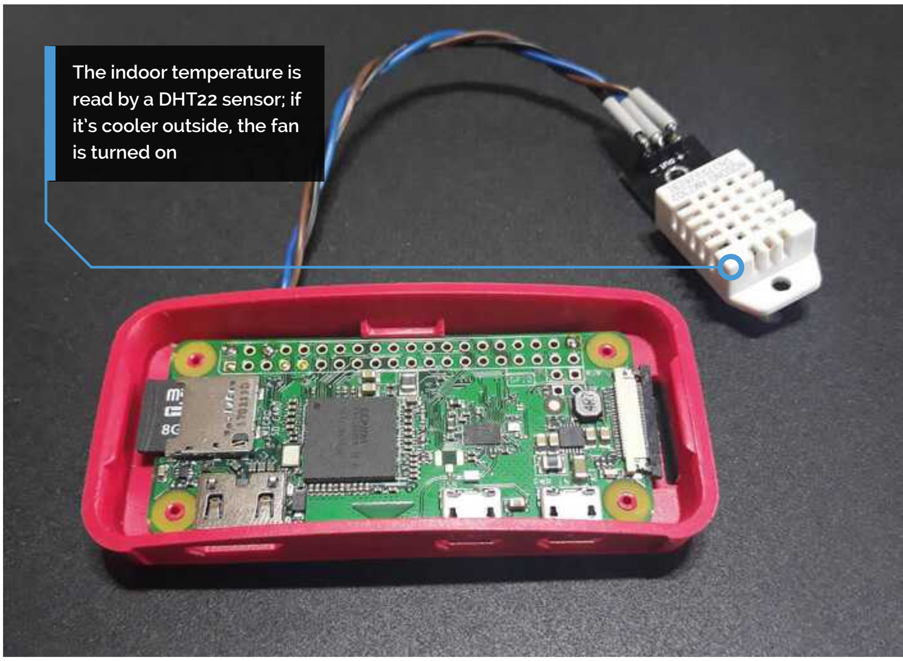
- The project took about two weeks to complete
- The DHT22 sensor is measured every eight seconds
- Weather information is retrieved every 40 seconds
- A Bash script is used to turn the smart plug on and off
- Ishmael is working on an FPV WiFi paintball turret project

start and control it manually via an Android smartphone. The latter runs the VNC Viewer app, enabling remote access to Raspberry Pi's desktop, on which there is a shortcut to start the fan application; this then displays a Pygame window with temperature information and control buttons.

"The fan application has two buttons to change the [desired temperature] set-point up or down," reveals Ishmael. "Also, the button on the upper right is to close the application and return to the desktop." His aim is to have more than one project running on his Raspberry Pi, and have a desktop shortcut for each application.

While the original project used a single fan, he has since modified it to add another. "I have been reading that two fans are required for best performance," he says. "One to blow in and another to blow out."

This certainly is a cool project, in more than one way. If you'd like to have a go at building a similar system, you can read Ishmael's Hackster guide (**magpi.cc/kzibLu**) and check out his GitHub repo (**magpi.cc/zAzGZS**) for the code.



The indoor temperature is read by a DHT22 sensor; if it's cooler outside, the fan is turned on

▲ The DHT22 sensor is connected to power, ground, and GPIO 4 pins on a Raspberry Pi Zero – a 10kΩ resistor is recommended

PIC-20

Retro enthusiast Adam Sommerfield revives broken computers by allowing their original keyboards to be used in combination with Raspberry Pi, as **David Crookes** explains

MAKER

Adam Sommerfield

Married with two young sons, Adam runs a small YouTube channel called Banana Retro, and he oversees Facebook groups dedicated to Raspberry Pi and retro emulation.

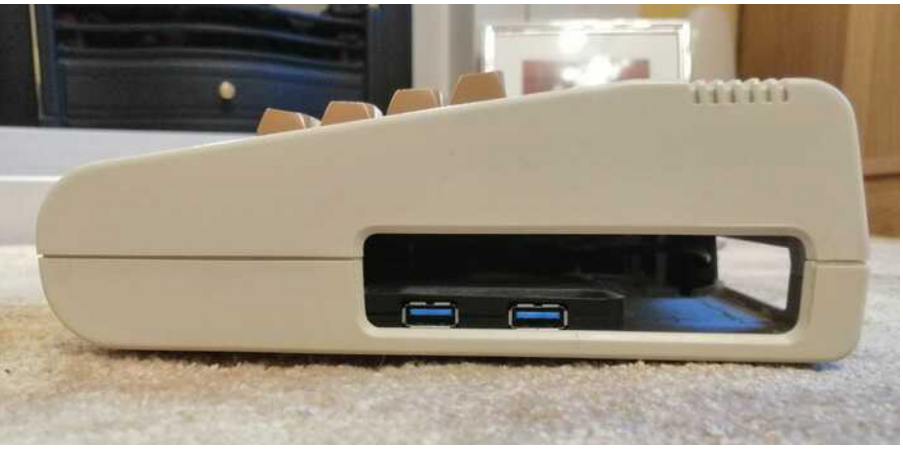
magpi.cc/soPCYo

When Adam Sommerfield was five years old, his parents bought him a Commodore VIC-20 computer. Released in 1981, it was very popular at the time – the first to sell a million units – but Adam's machine didn't actually work, so his dad took it back and swapped it for a Commodore Plus/4 instead.

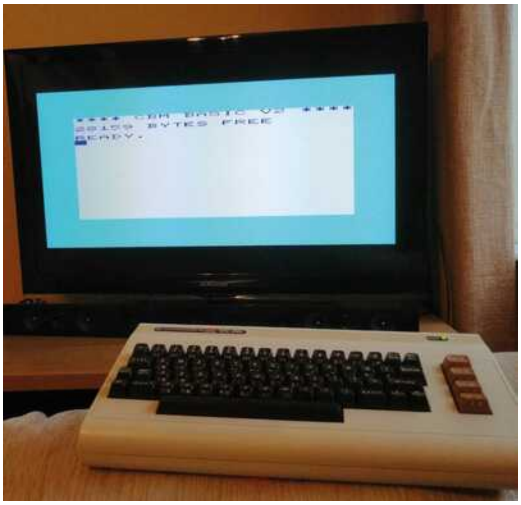
Ever since that childhood incident, Adam had wondered what that VIC-20 would have been like to use. He even had a game for it, Crazy Cavey, which he had never played. While online one day, he came across a device that would allow old machines to be used as a USB keyboard, so he hatched a plan involving a broken VIC-20 and Raspberry Pi.

Coming across an inexpensive, non-working VIC-20 on eBay sold for spares or repairs, he sought to bring it back to life by emulating the original computer. Knowing he could do this on Raspberry Pi using the emulation software Combian64, written by Carmelo Maiolino, he got to work.

"I removed all the internal parts from the VIC-20 other than the keyboard itself," recalls Adam. "I then stripped the keyboard down and gave it a good clean using an air duster and cleaning fluid. From there, it was a matter of figuring where Raspberry Pi and the necessary cables needed to go."



▼ A couple of USB ports are viewable from the side of the machine. The holes could be filled for a neater finish

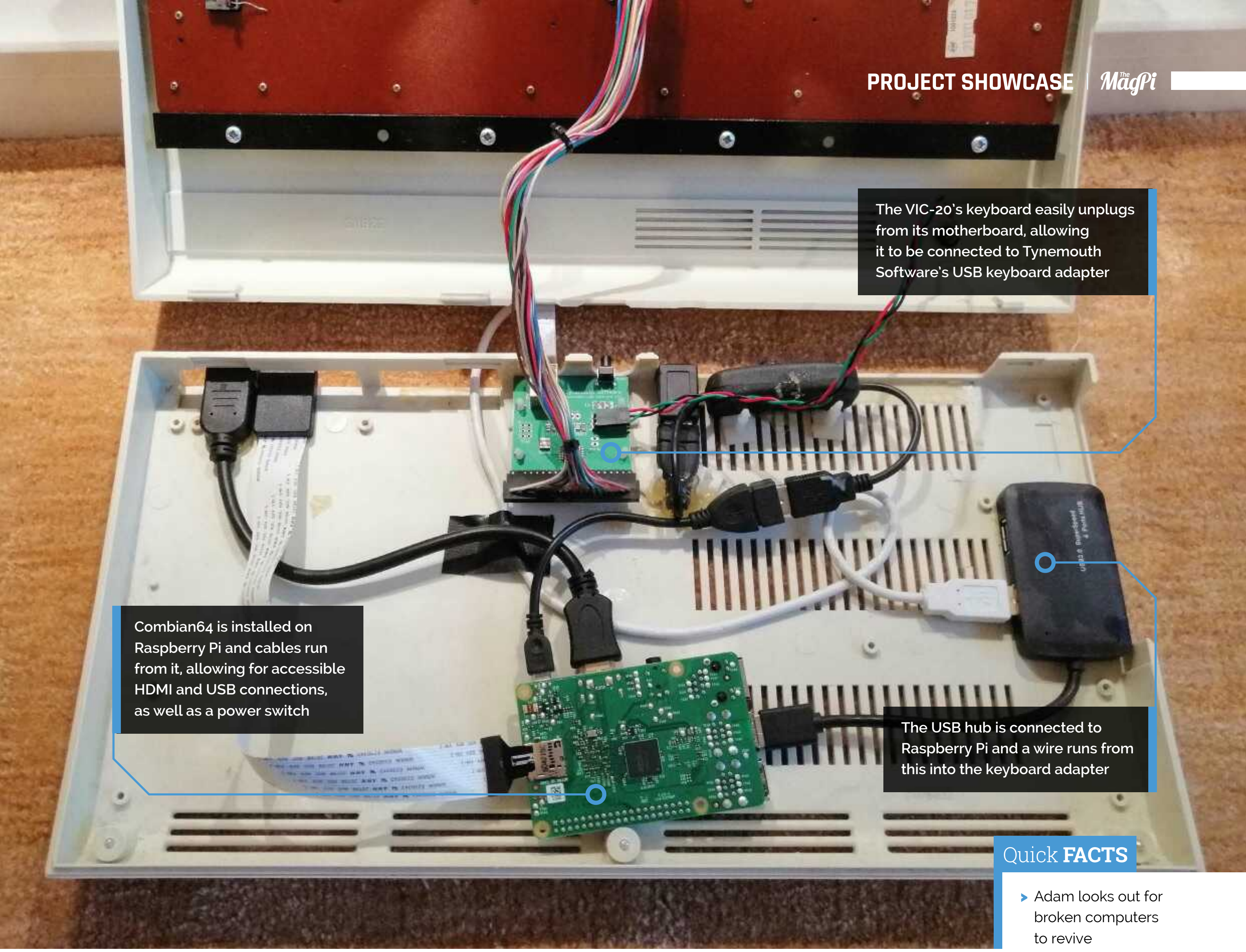


▲ When PIC-20 is connected to a monitor, it looks and behaves like an original VIC-20 from the 1980s

Converting the keyboard

A crucial part of the project was a custom USB keyboard adapter – to allow the VIC-20 keyboard to be plugged into Raspberry Pi. Adam had seen that David Cullen, of Tynemouth Computers, creates and sells a number of these device for various retro computers. "That took care of the hard work," Adam says. With the adapter attached to the VIC-20 keyboard, it was possible to connect it to Raspberry Pi. "I looked to place its ports as close to the casing's holes as possible, but these were spread across the back and side of the casing so I used short male-to-female leads instead, giving access to HDMI, USB power, and the microSD card slot," he explains. "I also used a four-port USB hub to allow components such as joysticks to be inserted."

The USB adapter, USB hub, and Raspberry Pi were fixed with nylon supports and fasteners to act as legs, giving space between them and the base of the case. Some of the cables were also secured using hot glue. "This can get a bit of a mixed



The VIC-20's keyboard easily unplugs from its motherboard, allowing it to be connected to Tynemouth Software's USB keyboard adapter

Combian64 is installed on Raspberry Pi and cables run from it, allowing for accessible HDMI and USB connections, as well as a power switch

The USB hub is connected to Raspberry Pi and a wire runs from this into the keyboard adapter

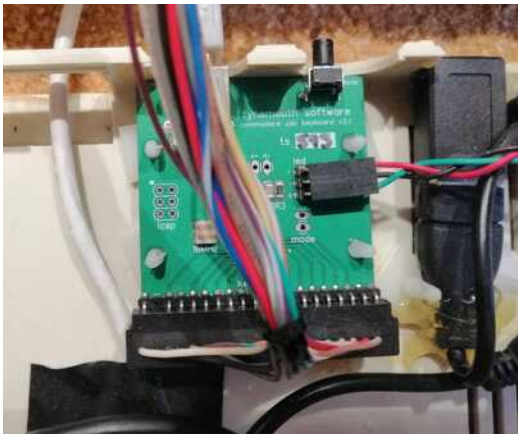
> " I like to think I've saved a micro from heading to the skip "

reaction sometimes, but if you're careful then it can work just fine."

Installing the software

With everything fitted neatly inside the case, it was a matter of closing up and booting. "Having Combian64 installed on Raspberry Pi helps to recreate the original VIC-20 as closely as possible," Adam says. The build also required no programming. "By using Combian64, it's possible to build a Commodore computer that gives near-instant boot. There is no on-screen boot text either so, if you didn't know better, it could be the original thing."

The VIC-20 is not his only retro project. Using other emulators and different USB keyboard adapters, he's converted a Commodore 64 and Commodore 16, as well as an Apple Macintosh, Apple Lisa, Acorn Electron, Sinclair QL, and ZX81. He's even worked on using Raspberry Pi to revive a broken Commodore Plus/4 – the machine that, inadvertently, became his first computer. "I like to think I've saved a micro from heading to the skip."



▲ The USB keyboard adapter also allows a retro computer to be used as a regular PC keyboard: a mode button converts keys, so the Commodore key becomes the Windows key, for instance

Quick FACTS

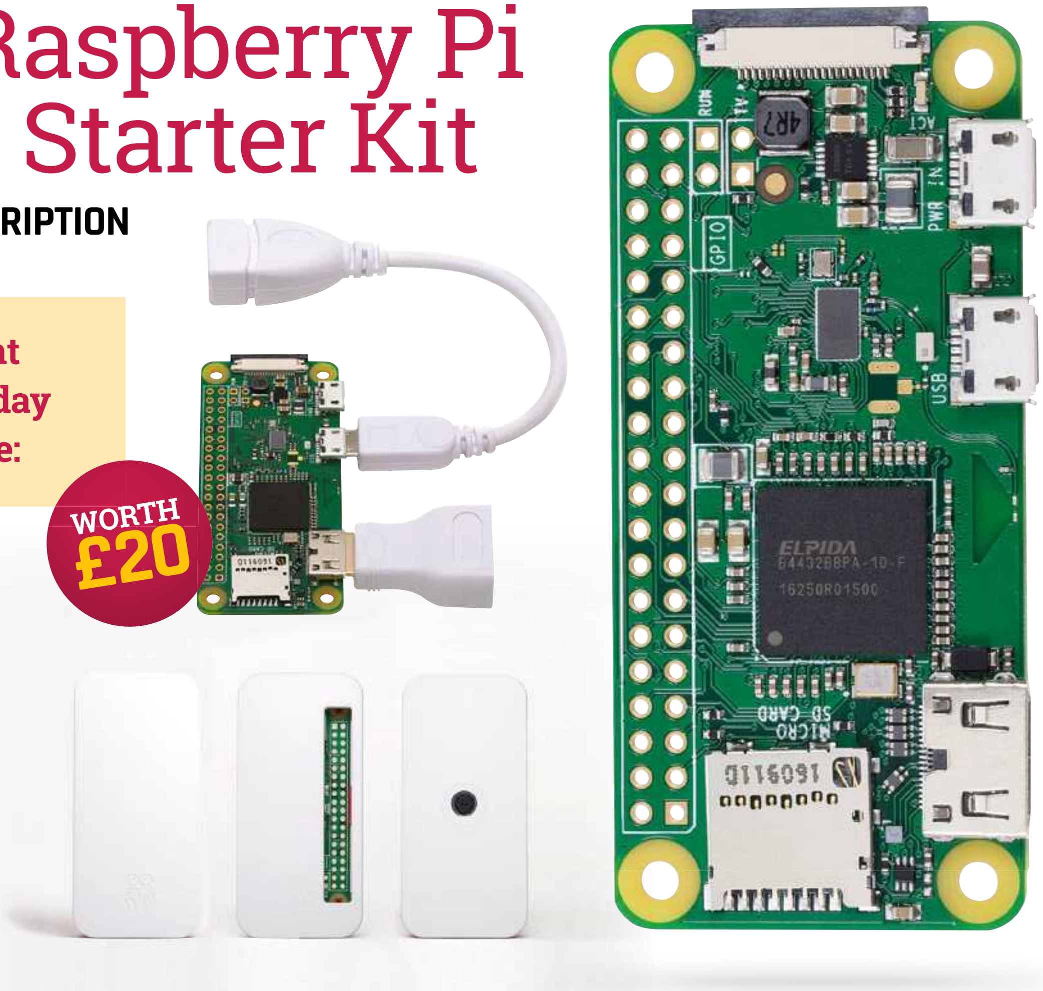
- Adam looks out for broken computers to revive
- Raspberry Pi is used to emulate old systems
- A retro conversion can cost around £145 overall
- The USB keyboard adapter kit costs $60: **magpi.cc/cepafK**
- Adam gives away any spare parts he strips

JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

Subscribe in print for 12 months today and you'll receive:



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

Offer subject to change or withdrawal at any time

Buy now: magpi.cc/subscribe

WIN! 1 OF 10 RASPBERRY PI 4 KITS
The MagPi
Introducing Raspberry Pi 4
BENCHMARKS
ENGINEERING INTERVIEWS
TECH SPECS
Your Ultimate Starter Guide
40 PAGES OF PROJECTS & GUIDES

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

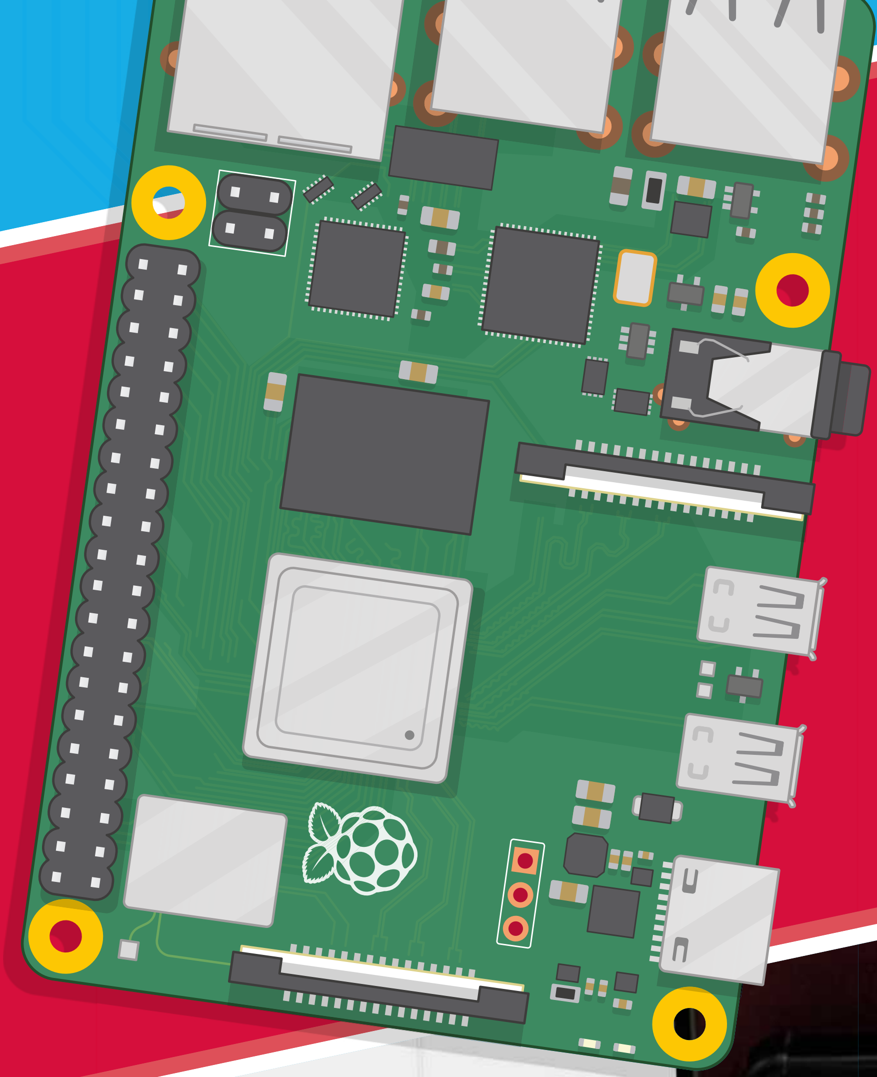
RASPBERRY PI 4 STARTER GUIDE

Get to know your brand new Raspberry Pi 4

By **Sean McManus**

It's been a month since Raspberry Pi 4 came out and we've been seeing a lot of people on social media get stuck in with their new incredible computer. We've also seen a lot of new Raspberry Pi users finally take the plunge and get their first Raspberry Pi, and to those people we say: welcome!

Some of you may still be getting your heads around your new Raspberry Pi, so we're here to help with our Starter Guide that should teach you some of the basics (and beyond!) of how to use it. Grab a microSD card and find a spare monitor, because it's time to have some fun.



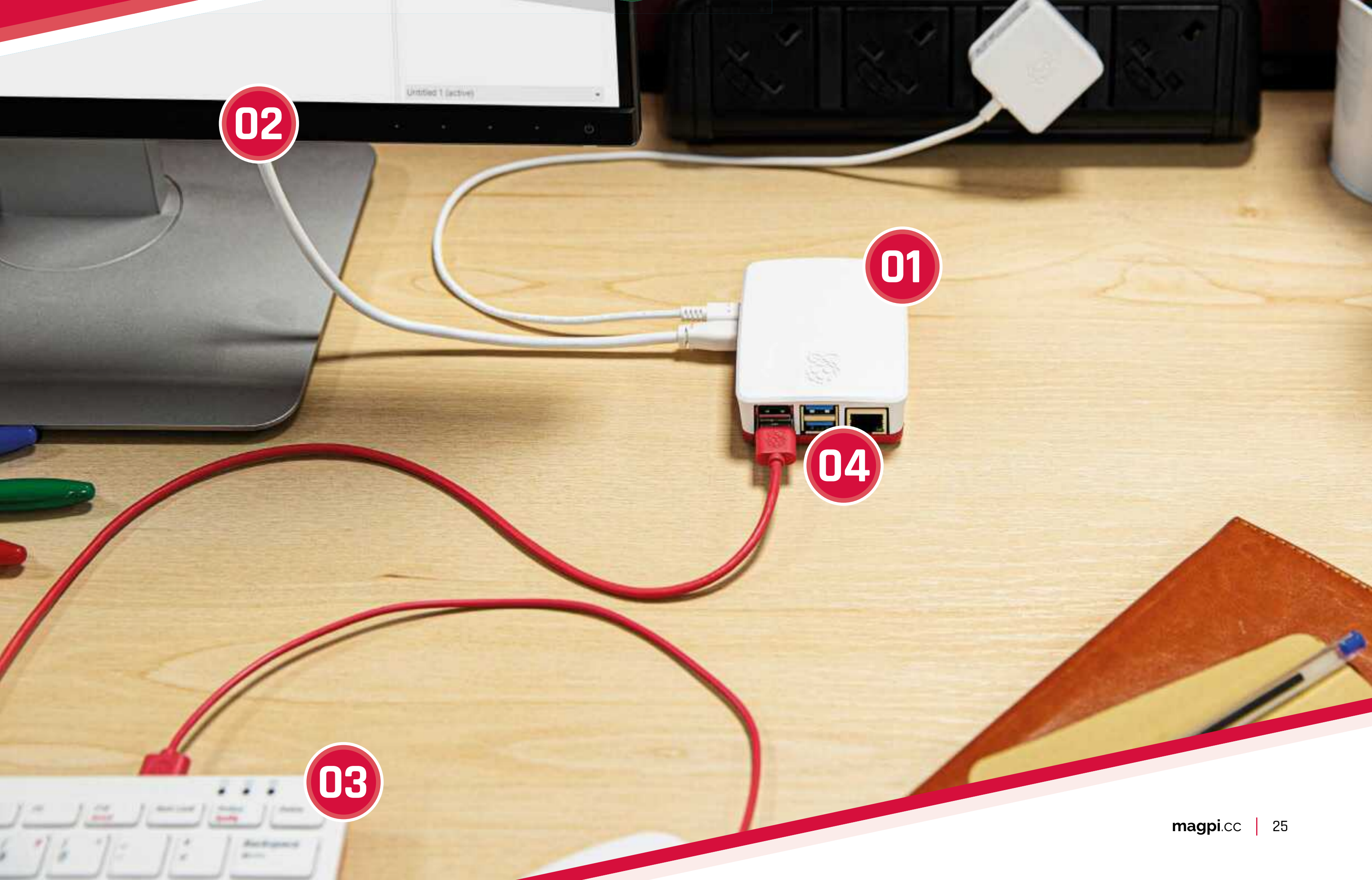
Basic setup

01 The heart of your new computing experience: Raspberry Pi 4. Find out the full, amazing specs here: **magpi.cc/benchmarks**

02 Power up to two 4K monitors with Raspberry Pi 4's dual micro-HDMI ports

03 With a keyboard and mouse, you can easily use it as a desktop computer

04 The USB 3.0 ports allow for high-speed file transfers



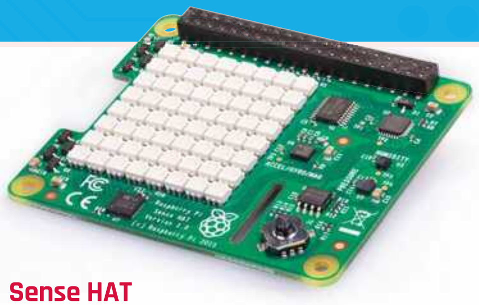
FIRST STEPS WITH RASPBERRY PI

Now you've set up your Raspberry Pi, discover some of its accessories and explore the Raspbian operating system. **Sean McManus** is your guide

Raspberry Pi Case

Protect your Raspberry Pi from spills and dust with a chic case. Official Raspberry Pi cases come in red/white and grey/black, but there are plenty of cases available from other companies too. If you'll be building electronics projects, look for a case that gives you easy access to the GPIO.

magpi.cc/YNvYfF



Sense HAT

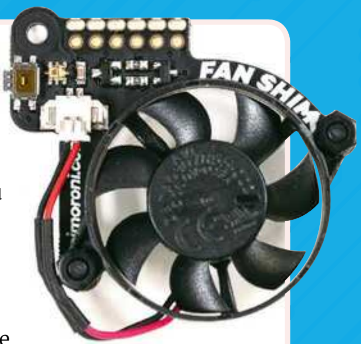
HATs (short for Hardware Attached on Top) are accessories that plug onto a Raspberry Pi's GPIO pins. The Sense HAT includes a colourful 8×8 grid of LEDs and a five-button joystick. It's packed with sensors: gyroscope, accelerometer, barometric pressure sensor, magnetometer, thermometer, and humidity sensor. It's a portable science lab!

magpi.cc/sense-hat

Raspberry Pi Camera Module V2

You can plug a Raspberry Pi Camera Module into a dedicated connector on Raspberry Pi and it has an 8-megapixel sensor. It works with Raspberry Pi 1, 2, 3, and 4. Shoot a movie in high definition, get close to nature with a bird-box cam, or secure your home.

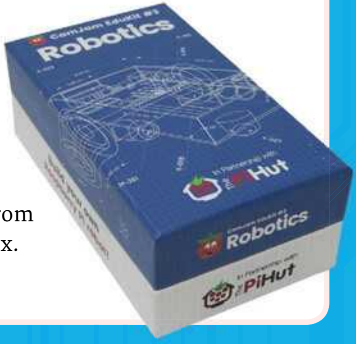
magpi.cc/camera



Fan SHIM

Raspberry Pi 4 is a bit more power hungry than earlier models. If you find it runs hot for your application, try using the Fan SHIM. It provides a software-controllable fan to cool a Raspberry Pi. You can use it together with HATs, by fitting a booster header to lift them above the fan. Read more in our review on page 76.

magpi.cc/qZYBWd



CamJam EduKit #3

With its easy control of electronic circuits, Raspberry Pi is ideal for robots. This kit contains everything needed to build your first robot, including a motor controller board, sensors, and wheels. Make a chassis from Lego, 3D-print one, or even use the box.

magpi.cc/RhpjZh

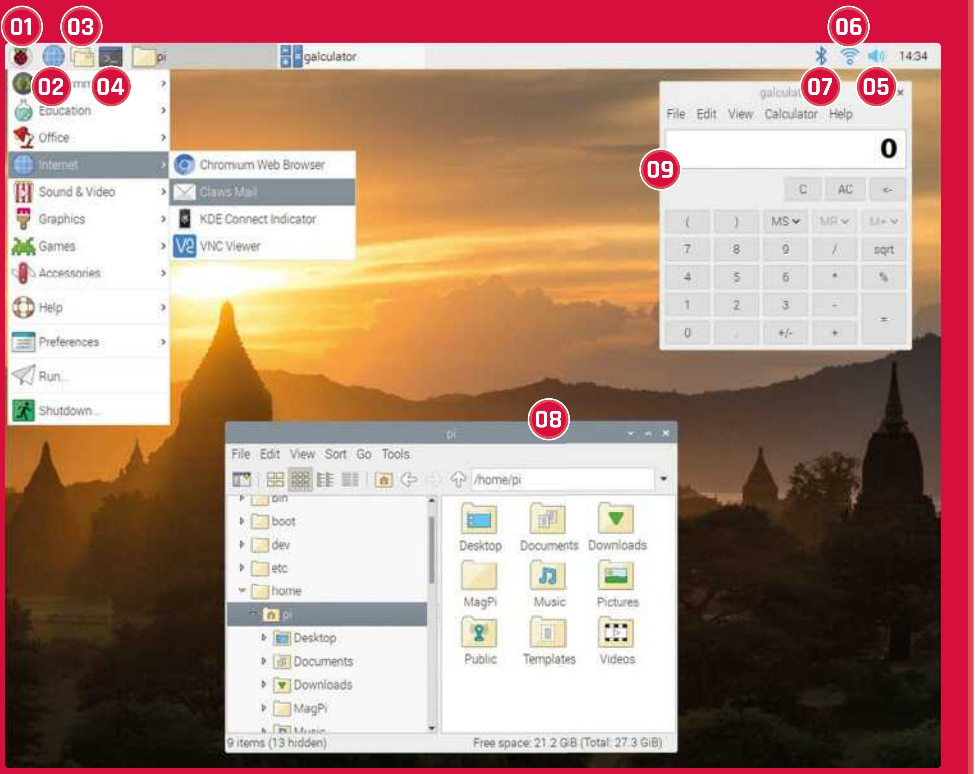
Meet Raspbian with Desktop

01 Click the Raspberry Pi logo to open the applications menu. This is where you'll find the software that's pre-installed in Raspbian, and anything else you add later.

02 Click the globe for quick access to the Chromium web browser.

03 Click the folders to open the File Manager. You can use it to find, move, copy, and delete files on your storage devices. Why not explore the Linux file system?

04 Find the Terminal here. It gives you powerful tools for managing your files and devices, and the command line is often the quickest way to get things done.



05 Volume control. Right-click to select audio output.

06 WiFi options. Turn WiFi on and off, and switch networks here. If the icon is blue like this, you're connected. You can also hover the mouse pointer over the icon to see your Raspberry Pi's IP address.

07 Manage Bluetooth connections. You can use Bluetooth devices such as keyboards and mice to wirelessly control your Raspberry Pi device.

08 File Manager. Use the hierarchical browser on the left or the Go menu to find devices connected to Raspberry Pi. You should store your files in the **/home/pi** folder.

09 With basic, scientific, and paper modes, this calculator is handy. Find it in the Accessories section of the applications menu.

Raspberry Pi Configuration

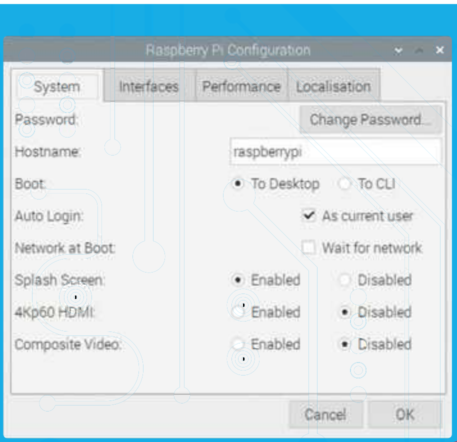
Need to adjust some settings? The Raspberry Pi Configuration tool is in the Preferences section of the applications menu. In its System tab, you can change your password, adjust display options, and set Raspberry Pi to boot to the command-line interface (CLI) instead of the desktop.

In the Interfaces tab, you can enable connections, including remote GPIO access and the camera.

To adjust the amount of memory for the GPU or to set older Raspberry Pi models to run faster (overclocking), visit the Performance tab.

You can change the time zone, keyboard, and other geographic options in the Localisation tab.

There are separate options in Preferences for configuring the appearance, audio, main menu, mouse and keyboard, and screen.



Password:
Hostname: raspberrypi
Boot: ● To Desktop ○ To CLI
Auto Login: ☑ As current user
Network at Boot: □ Wait for network
Splash Screen: ● Enabled ○ Disabled
4Kp60 HDMI: ○ Enabled ● Disabled
Composite Video: ○ Enabled ● Disabled

AWESOME APPLICATIONS

Discover some of the software that comes with Raspbian, and find out how to install more

Raspbian comes with a selection of pre-installed software (which depends on which version you install), so you can start working, learning, and making things straight away. Ranging from productivity suites to games, you'll find a well-curated collection of software is just a click away, in the applications menu. There are lots more packages to browse and install, too.

LibreOffice Writer | OFFICE

No computer is complete without a word processor. LibreOffice Writer has all the font and formatting options you would expect, and has basic compatibility with Microsoft Word.

LibreOffice Calc | OFFICE

Work out your budgets with this spreadsheet package. If you're familiar with Microsoft Excel, you'll feel at home here. LibreOffice Calc can open and use typical Excel files.

LibreOffice Impress | OFFICE

If you're presenting to a room of people, don't panic: Impress has your back. Use it to craft and display your slide deck. It's largely compatible with Microsoft PowerPoint.

Chromium | INTERNET

Chromium is the open-source version of Google Chrome. The default search engine in Raspbian is Duck Duck Go, which promises not to track you online.

Claws Mail | INTERNET

Send and receive email. The setup wizard helps you add your email account, and the streamlined interface shows you your mailboxes, message list, and a preview of the selected message.

VLC | SOUND & VIDEO

Listen to music while you're coding using VLC, a fully featured media player for music and video. It can play digital files, streams, and physical media such as CDs and DVDs.

Minecraft Pi | GAMES

We call it a game, but it's more a way of life. With Minecraft Pi, you can build things in Creative mode, and write programs to change the (game) world.

Python Games | GAMES

These tea-break games are fun, but they're also great Python demos. Find the code in the **/usr/share/python_games** folder, and open it in a Python editor to see how it works.

SmartSim | PROGRAMMING

Experiment with designing and testing digital logic circuits. You can develop custom components, and then incorporate them into other circuits. Download examples and read tutorials at **smartsim.org.uk**.

Sense HAT Emulator
PROGRAMMING

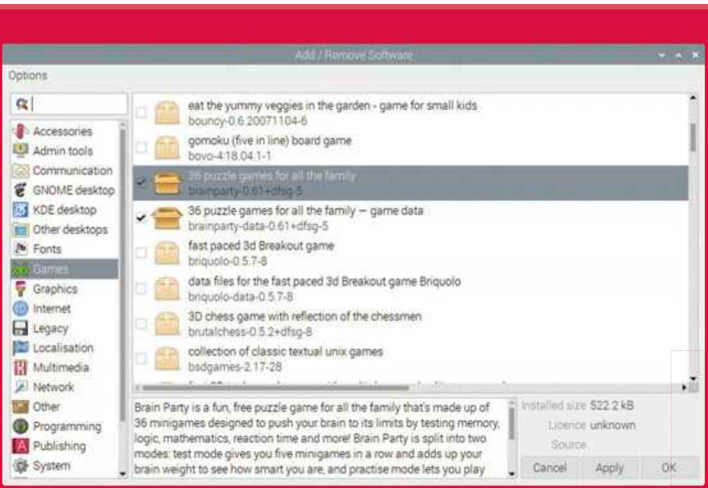
This emulator features on-screen controls to simulate temperature, pressure, humidity, and device position changes. A great way to try out the Sense HAT before you buy one!

Updating your software

Linux software comes in packages, which are compressed archives that you can download. A package manager is used to find and install them, including any other software they need to work.

Use Add/Remove Software to update your packages. From the menu, choose Preferences > Add / Remove Software. Click on Options and Check for Updates.

You can also update from Terminal. In Raspbian, the package manager is called APT. Updating all the software from the Terminal is a two-step process. First, enter `sudo apt-get update` in the Terminal to update the cache of available software. Then enter `sudo apt-get upgrade` to update the software installed on Raspberry Pi. You'll be told what changes will be made and asked to confirm by typing **Y** and pressing **ENTER**.



Add / Remove Software

Options

Accessories
Admin tools
Communication
GNOME desktop
KDE desktop
Other desktops
Fonts
Games
Graphics
Internet
Legacy
Localisation
Multimedia
Network
Other
Programming
Publishing
System

- □ eat the yummy veggies in the garden - game for small kids
  bouncy-0.6.20071104-6
- □ gomoku (five in line) board game
  bovo-4.18.04.1-1
- ☑ 36 puzzle games for all the family
  brainparty-0.61+dfsg-5
- ☑ 36 puzzle games for all the family – game data
  brainparty-data-0.61+dfsg-5
- □ fast paced 3d Breakout game
  briquolo-0.5.7-8
- □ data files for the fast paced 3d Breakout game Briquolo
  briquolo-data-0.5.7-8
- □ 3D chess game with reflection of the chessmen
  brutalchess-0.5.2+dfsg-8
- □ collection of classic textual unix games
  bsdgames-2.17-28

Brain Party is a fun, free puzzle game for all the family that's made up of 36 minigames designed to push your brain to its limits by testing memory, logic, mathematics, reaction time and more! Brain Party is split into two modes: test mode gives you five minigames in a row and adds up your brain weight to see how smart you are, and practise mode lets you play

Installed size 522.2 kB
Licence unknown
Source

Installing software

The easiest way to manage software is to use the Add / Remove Software tool in the Preferences part of the applications menu. It provides a visual front-end for the package manager.

You can click a category on the left to browse applications, or enter a keyword in the search box in the top-left to look for a particular application. Choose the applications you want to install by ticking the box beside them. Some packages require other packages to work properly, but the tool will take care of that for you. To remove an application, untick its box. When you've made your choices, click OK to install or remove your software.

Why not install these?

Fritzing | PROGRAMMING

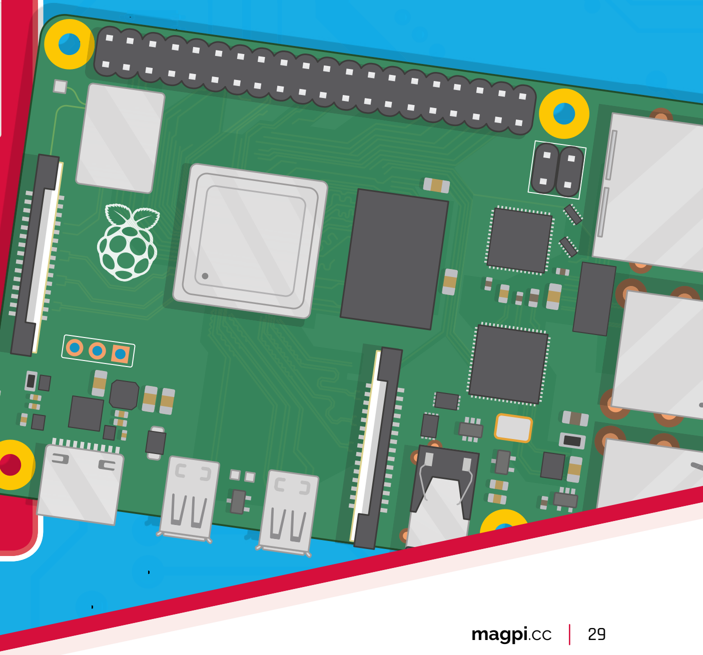
Design and document your electronics circuits based on Raspberry Pi with this design tool. We use it to make circuit diagrams in *The MagPi*.

GNU Image Manipulation Program (GIMP) | GRAPHICS

GIMP is a powerful image editor. You can use it to create digital art, but its best feature is probably the clone tool, which enables you to retouch holiday photos.

Mathematica PROGRAMMING

Good for more than just maths, Mathematica uses the Wolfram language, which has data and intelligence built in. Install it from Recommended Software, in the Preferences section of the applications menu.



GET CONNECTED

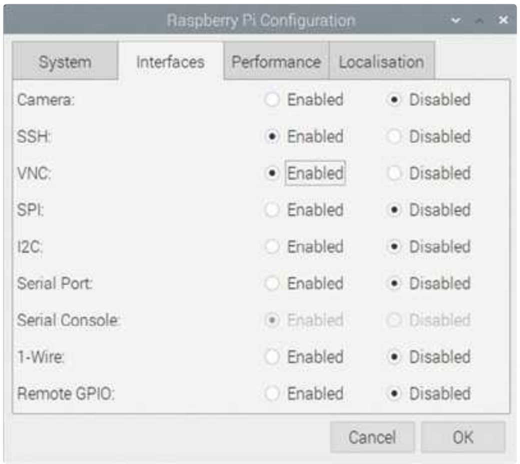
If your Raspberry Pi is in a tree photographing nature, you don't want to climb up there just to update it. Log in remotely and take control

There are two technologies you can use to connect to your Raspberry Pi: Secure Shell (SSH) and Virtual Network Computing (VNC). Before you can use them, they need to be switched on in the Raspberry Pi Configuration settings. To do so, click the Interfaces tab, then enable SSH and/or VNC.

Both approaches require you to know the IP address of your Raspberry Pi device. To find out, click the Terminal icon on the taskbar, and enter `ifconfig` at the prompt. It will show you all your network connections. You're looking for an IP address, which will be four numbers with a dot between them, like 198.51.100.0. You'll find it beside 'inet', in the details for wlan0 if you're using WiFi or in the eth0 summary if you're plugged in to the network.

Using SSH

SSH enables you to use the command line on your Raspberry Pi remotely, so it's very handy for installing software and fixing configuration files. You can't use SSH to run any applications that need the graphical desktop, though.



Raspberry Pi Configuration

System | Interfaces | Performance | Localisation

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

Cancel | OK

▲ Enable SSH and VNC before you try to use them

The software for SSH is pre-installed on Linux, macOS, and Windows 10. Start by opening the command line. On a Mac, find it by typing 'Terminal' into the Spotlight search. On Windows 10, Use **⊞+R** to open the Run dialog box, then type in **cmd**.

Once you're in the command line, enter `ssh pi@198.51.100.0`, but replace the numbers with the IP address of your own Raspberry Pi device. The 'pi' bit is your username.

> " SSH enables you to use the command line on your Raspberry Pi remotely, so it's very handy "

The first time you connect to a device using SSH, you'll see a warning that shows the ECDSA key for the device you're trying to connect to. You can (if you wish) validate that this is correct by using `ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub` in the Raspberry Pi Terminal. But it's usually OK to just type in **yes** and then press **ENTER** to confirm you want to connect.

You'll be asked to enter the password for your Raspberry Pi device. You won't see the cursor move while you do this, so type on regardless. When you press **ENTER**, you're in! You'll see a Linux welcome message, the date and time of your last login, and then the Raspbian command prompt, waiting for instructions. When you've finished, enter `exit` to leave the SSH session.

If you're using an earlier version of Windows, download PuTTY from **putty.org**. Enter your Raspberry Pi's IP address in the Host Name box, and click Open. Again, you'll see a warning if this is your first connection, which you can safely dismiss. Log in as **pi**, and enter your password. Raspberry Pi is now at your command!

Tip!

See our SSH tutorial on page 40 for more detailed information on using SSH to remotely control a Raspberry Pi.

Using VNC Viewer to manage Raspberry Pi from an iPad

Using VNC

Virtual Network Computing (VNC) enables you to remotely access the Raspberry Pi desktop, so you can manage files and run software using it. Some people use VNC to share their keyboard, mouse, and monitor between a PC and Raspberry Pi.

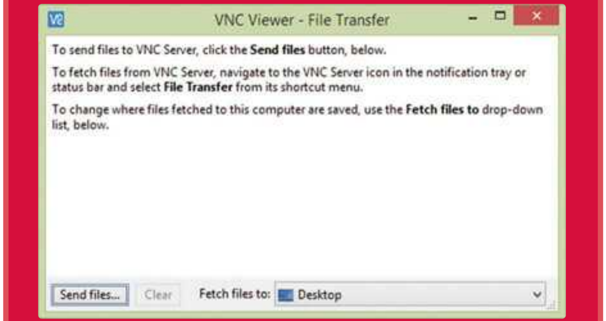
Raspbian includes VNC Server, which runs automatically in the background if you've enabled it in your settings. You'll need to download and install VNC Viewer (**magpi.cc/FuGnye**) on the device you want to use to control Raspberry Pi. VNC Viewer is available for Windows, macOS, and Linux. There are also Android and iOS apps, so you can use VNC to control Raspberry Pi from a mobile device, although it's rather fiddly without a real mouse and keyboard. Optionally, by creating a RealVNC account and registering your Raspberry Pi, you can then access it from anywhere in the world using VNC Viewer.

When you start VNC Viewer, it'll ask you to enter a VNC Server address. This is the IP address you noted when you ran `ifconfig` on your Raspberry Pi.

The first time you connect to a device, VNC Viewer warns you that it has no record of connecting to this device before, and shows you the device's signature and identification catchphrase. To verify you're connecting to the correct device, click the VNC icon on the right of the taskbar in Raspbian to see your device's details.

VNC Viewer prompts you to enter the username and password for the device you're connecting to. You'll then see your Raspbian desktop in the VNC Viewer window. You can now use your keyboard and mouse (or touchscreen on a mobile device) to control the Raspbian desktop, including using programs installed on Raspberry Pi.

Top Tips for VNC



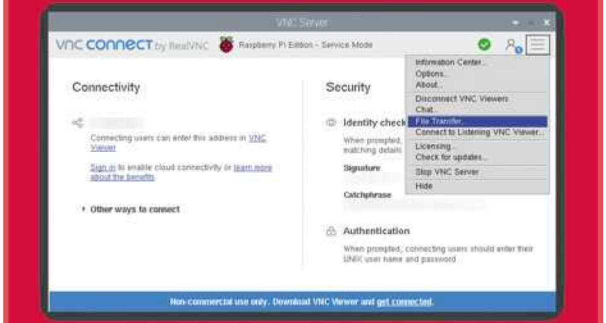
VNC Viewer - File Transfer

To send files to VNC Server, click the **Send files** button, below.

To fetch files from VNC Server, navigate to the VNC Server icon in the notification tray or status bar and select **File Transfer** from its shortcut menu.

To change where files fetched to this computer are saved, use the **Fetch files to** drop-down list, below.

▲ At the top of VNC Viewer in the middle is a tiny toolbar. The File Transfer option enables you to easily copy files to Raspberry Pi.



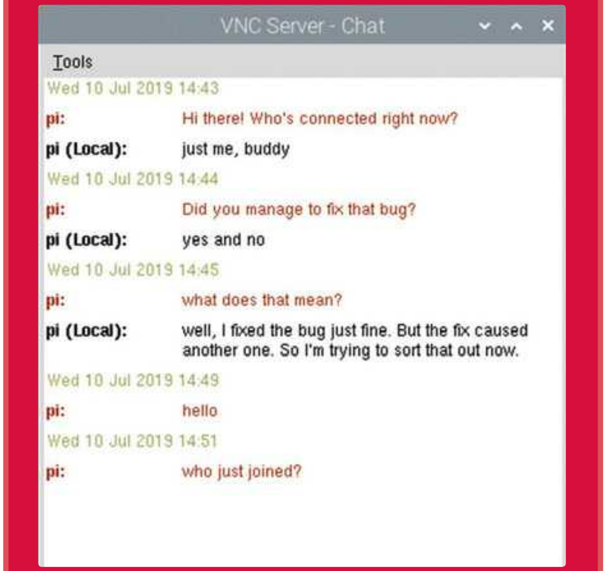
Connecting users can enter this address in VNC Viewer

Sign in to enable cloud connectivity or learn more about the benefits

When prompted, connecting users should enter their UNIX user name and password

Non-commercial use only. Download VNC Viewer and get connected.

▲ Send files from Raspberry Pi using the File Transfer option in VNC Server in Raspbian. Use the three-bars menu (top-right) to find it.



VNC Server - Chat

```
Wed 10 Jul 2019 14:43
pi:            Hi there! Who's connected right now?
pi (Local):    just me, buddy
Wed 10 Jul 2019 14:44
pi:            Did you manage to fix that bug?
pi (Local):    yes and no
Wed 10 Jul 2019 14:45
pi:            what does that mean?
pi (Local):    well, I fixed the bug just fine. But the fix caused
               another one. So I'm trying to sort that out now.
Wed 10 Jul 2019 14:49
pi:            hello
Wed 10 Jul 2019 14:51
pi:            who just joined?
```

▲ Use the toolbar in VNC Viewer to start a chat session with anyone using the server (Raspberry Pi device) and anyone else connected to it remotely.

CODE TO JOY

Raspberry Pi comes with everything you need to start programming

If you've never programmed before, you're in for a treat. Raspbian comes with several integrated development environments (IDEs) you can use to write your own programs, supporting languages that are friendly to use, and fun to tinker with. If you're a veteran coder, you'll find powerful IDEs are included to help you be productive, too.

Scratch 2

LANGUAGE: SCRATCH

With its friendly drag-and-drop commands and a library of sprites and sound effects, Scratch makes it simple to get started with coding. By minimising typing, and guiding you to sensible block combinations, it helps you to avoid common mistakes. Don't be fooled by its accessibility, though: it's a fully-fledged programming language, with plenty of potential.

Thonny

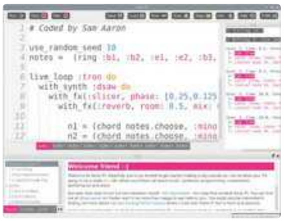
LANGUAGE: PYTHON

Python is one of the most popular languages on Raspberry Pi, and Thonny is our favourite way to edit it. In a single window it shows you your code, the shell, and your data (variables), so you can easily see what's going on and fix any bugs that creep in.

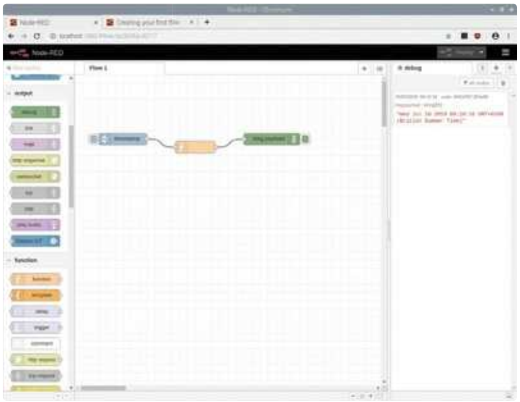
Sonic Pi

LANGUAGE: SONIC PI

Learn how to compose and perform music with code! Sonic Pi incorporates synths, samples, and effects that you can control by writing programs using a language based on Ruby. You can develop and modify programs while the music plays, to perform live concerts or improvise at home.



```
# Coded by Sam Aaron

use_random_seed 10
notes = (ring :b1, :b2, :e1, :e2, :b3,

live_loop :tron do
  with_synth :dsaw do
    with_fx(:slicer, phase: [0.25,0.125
      with_fx(:reverb, room: 0.5, mix:

        n1 = (chord notes.choose, :mino
        n2 = (chord notes.choose, :mino
```



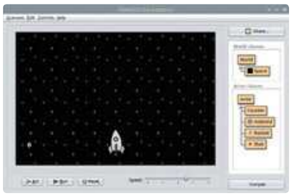
Node-RED

LANGUAGE: JAVASCRIPT/NODE-RED

Node-RED enables you to use a flowchart to program data flows for Internet of Things applications on Raspberry Pi. It uses JavaScript, the language of the web. Run the Node-RED console in Raspbian and then visit **http://localhost:1880** in your browser to program it. For help, see **nodered.org**.

Greenfoot

LANGUAGE: JAVA

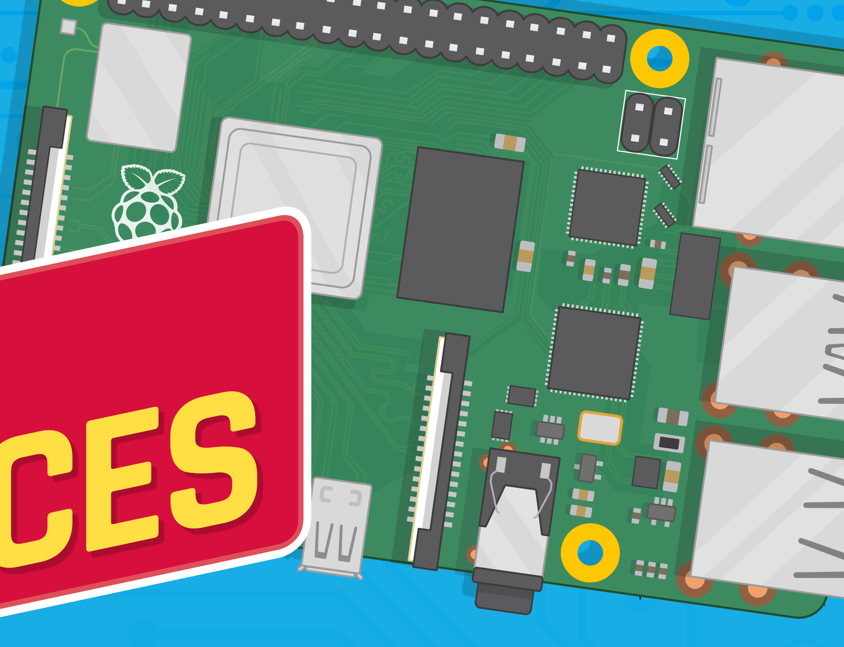


Java is one of the world's most popular programming languages. Greenfoot makes it easier to learn, by providing a friendly environment for building simple games. The editor incorporates a game world, and you add Java code to image objects to control their interactions. Find out more at **greenfoot.org**.

Geany

LANGUAGE: LOTS!

If you're looking for a lightweight but powerful IDE, try Geany. It supports many languages, including HTML, C, Java, PHP, JavaScript, and Perl. Its code auto-completion and syntax highlighting can help you code faster, and reduce errors. Our favourite feature? Code folding, used to show or hide logical chunks of code in a long program.



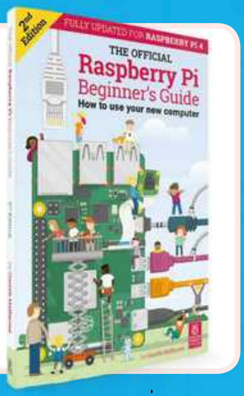
FURTHER RESOURCES

There's a wealth of resources available to support you as you learn more about Raspberry Pi and Raspbian

The Official Raspberry Pi Beginner's Guide

Available to buy in print and as a free PDF, this book shows you how to set up Raspberry Pi, and gets you started with programming it in Scratch and Python. It also covers the Sense HAT and Raspberry Pi Camera Module, with code examples you can build and tailor.

magpi.cc/BGbook


2nd Edition
FULLY UPDATED FOR RASPBERRY PI 4
THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer

Official Raspberry Pi Documentation

The documentation provides concise user guides for Linux, Minecraft, Sonic Pi, Scratch, and Python. It's also the place to go for advice on configuring Raspberry Pi, hardware specifications, and remote access tips. Useful for both beginners and power users.

rpf.io/docs

GPIO Zero Documentation

See how easy it is to start programming your own electronics projects for Raspberry Pi. The documentation for GPIO Zero shows you how to connect up sensors, LEDs, motors, and more. With a bag of cheap components, you can start building your first circuits.

rpf.io/gpiozero

Conquer the Command Line

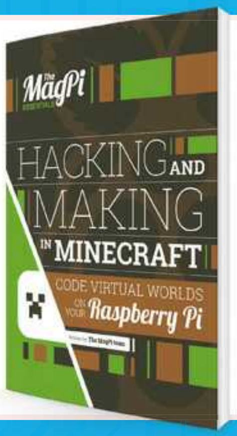
We publish a series of short books called *The MagPi Essentials*, and this one explains the Terminal, including using it for connecting disks, compiling software, and backing up. You can download all the books in the series, and past issues of *The MagPi*, at **magpi.cc/issues**.

magpi.cc/CLIbook

Hacking and Making in Minecraft

Minecraft on Raspberry Pi is a great way into the world of coding. This book, another in *The MagPi Essentials* series, shows you how to shape the world with Python code, use a connected LED as a treasure sensor, and use Sonic Pi and Node-RED together with Minecraft.

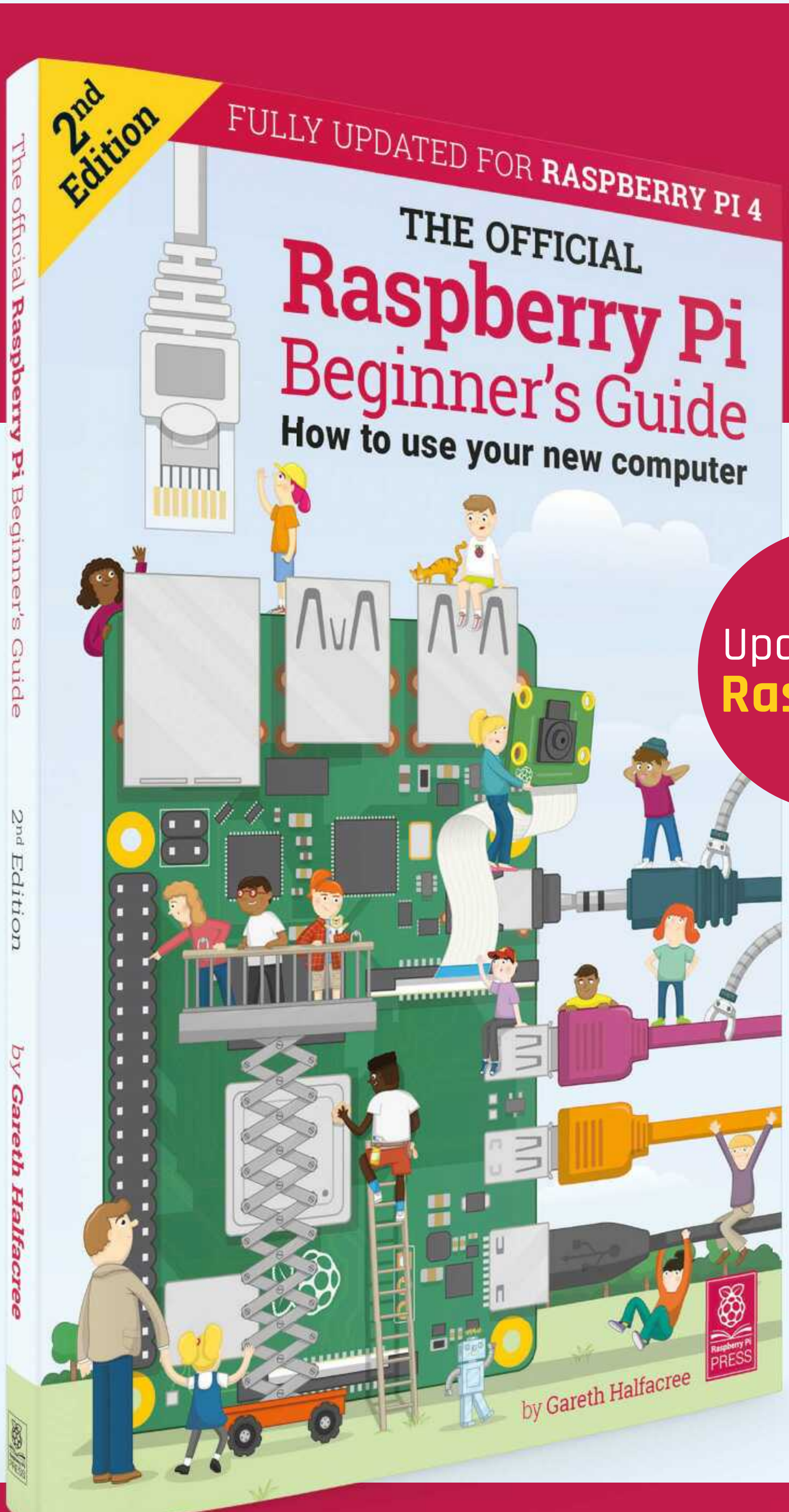
magpi.cc/MCbook


The MagPi
HACKING AND MAKING IN MINECRAFT
CODE VIRTUAL WORLDS ON YOUR Raspberry Pi

Python Games

Go behind the scenes of the Python games that come pre-installed in Raspbian with Al Sweigart's free online book. It explains the program code for seven of the games in detail, and lets you study another four listings yourself.

magpi.cc/RTfkvD



Fully Updated for **Raspberry Pi 4**

Build a low-cost robot

You want to build a robot without breaking the bank. Let's see what parts we need, where they fit, and how to keep the cost down

MAKER

Danny Staple

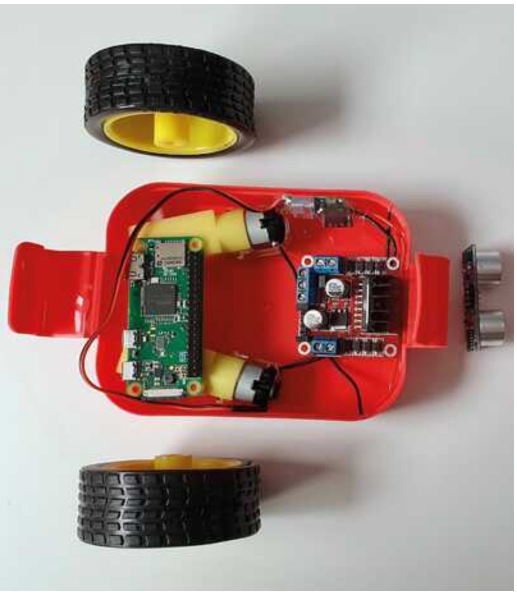
Danny makes robots with his kids as Orionrobots on YouTube, and is the author of *Learn Robotics Programming*.

orionrobots.co.uk

To make a robot, be it a wheeled rover, flying drone, factory robot, or autonomous spacecraft, you will need common classes of components. We'll discover what they are for, focusing on those needed for a wheeled robot.

We'll look at what options there are for the components, and how we might be able to save money. We'll go through the trade-offs needed for these options, the tools you might need, and their relative difficulty.

Any robot starts with a computer to run code, using sensors to collect data about the world. There are output systems to drive motors and actuators to affect the world. It needs power systems to get the right voltage and current to the right parts. The robot will need mechanical parts for the motors to drive, along with connecting the sensors and a body holding it all together.



These parts are ready to be built into a lunchbox chassis with plastic gear motors and plastic wheels. We will go into more detail on turning a lunchbox into a robot

01 An overview of robot parts

To make a wheeled robot, you are going to need some common part types to make it work:

- A chassis or body to hold everything together. You will need brackets for sensors eventually, too.
- Wheels and motors to drive them. This includes balance wheels or castors.
- A main controller to run your code: Raspberry Pi.
- A motor controller or driver to connect your Raspberry Pi safely to outputs.
- Batteries and power regulation for your electronics.
- Sensors to get data from the real world, like distance sensors and a camera.

02 Going low-cost

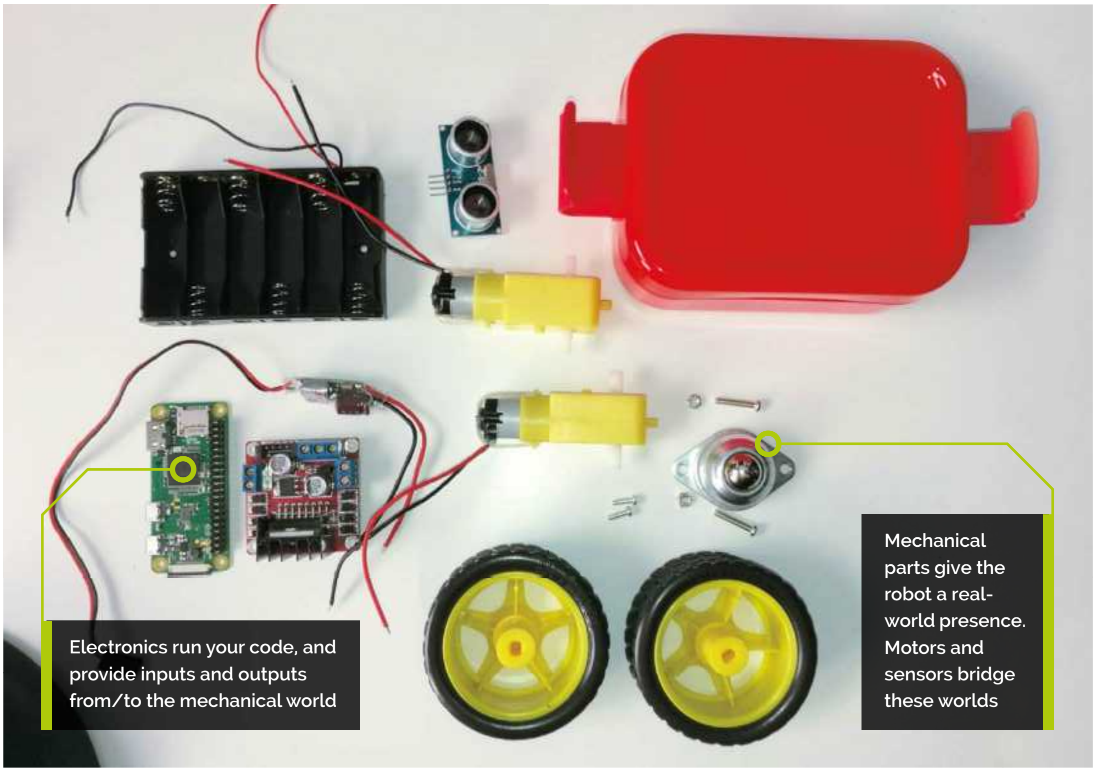
To go low-cost, you going to have to be a little creative. This will mean substituting parts, or finding parts that may not be the obvious choice. You will be able to save by shopping around, and waiting for parts that will take longer to ship will usually reduce cost.

Having parts pre-soldered or ready-made usually adds quite a lot to their cost, so be prepared to solder things together for the electronics, and to bodge or repurpose things for the body.

Robots can be made of anything, although we're not advocating it; we've even seen a robot made with vegetables for a body and wheels.

03 Raspberry Pi

The robot needs a Raspberry Pi to run your code. How would you save here? Well, the first cheapest Raspberry Pi is the one you already have!


Electronics run your code, and provide inputs and outputs from/to the mechanical world

Mechanical parts give the robot a real-world presence. Motors and sensors bridge these worlds

Top Tip

Get creative and repurpose

Be on the lookout for unused brackets and plastic shapes that might be handy to use in robot builds.

Make sure it's one with wireless LAN and a 40-pin header if you can (Raspberry Pi 3, 4, and Zero W models are all good choices).

Our favourite low-cost, and low-space, option is Raspberry Pi Zero WH (**magpi.cc/zerowh**). This is the smallest model, with wireless LAN and a GPIO header pre-soldered to the board. Since a robot isn't often connected to the screen, the lack of DSI port shouldn't be a problem. If you have a Raspberry Pi Zero W, then soldering pins to the GPIO header is a fun project (**magpi.cc/soldering**).

04 The chassis or body

The chassis holds the robot together. It's a fundamental decision on how your robot is made. Options for a wheeled chassis are:

- **Easy**: A laser-cut chassis – these are cheap to buy and easy to work with. They have space to extend the robot. They can be flimsy and do break, but are the simplest option and do not require much in the way of tools and time. If you go for a kit, a two-motor variant is advisable. Not as much fun as the lunchbox option.
- **Easy**: Adapt a lunchbox. This does require a little measuring and drilling of holes, but is still quite an easy option in terms of construction. You will need to choose smaller parts to fit in the lunchbox.
- **Intermediate**: Cut a chassis from wood or sheet material – this requires access to woodworking tools and CAD/drawing skills.
- **Intermediate**: The toy hack – one of our personal favourite options is taking a cheap motorised toy and swapping its electronics for a Raspberry Pi and motor board.
- **Hard**: The least cost for a chassis is not to have one and go for a free-form robot, strapping motors directly to control boards and batteries. This is a lot of fun, but takes experience and practice. Cable ties may be all you need to buy for this.

> "A good first robot is the lunchbox robot. It's a good compromise of saving cost and complexity"

A good first robot is option two, the lunchbox robot. It's a good compromise of saving cost and complexity, it's fun, and has a little bit of character. So that's what we're making here.

05 Not reinventing the wheel

Drive wheels will be attached to your motors. For a low-cost robot, two driven wheels is a good number.

Top Tip

Shop around more

Prices on robot parts vary by up to ten times. Swap words like board for module, and robot for Arduino, in your searches.

Unless trying to make experimental robots, plastic wheels with rubber tyres make the best drive wheels and are cheap and readily available.

- Both wheels need to be the same diameter.
- They need good grip.
- Axles should be aligned in the centre, and not slip.

Making your own is possible, but inadvisable. Instead, we recommend buying plastic wheels with tyres (search for this phrase with 'robot'), available for less than £5, and often with the motors for not much more.

06 Castor wheel or ball

A castor wheel, ball, or skid is mostly on the robot to balance it, without causing too much friction. You can get away with a simple bottle lid facing down here or, for a better cheap hack, half a ping-pong-ball. You can buy robot castor wheel or roller ball assemblies for under £2.

The important thing is that this component does not introduce friction, and can easily be attached firmly to the robot.

07 Types of drive motors

What kinds of motors are there?

- DC motors simply spin. They are cheap, but are easily stopped with any load.
- Stepper motors move in 'steps', a fraction of a full turn. They are not cheap, but can be salvaged from old printers and scanners.
- Servo motors can be moved to a particular position but not make a full revolution. They can be controlled directly from a Raspberry Pi. These can be modified for continuous rotation, but this can be complicated or expensive.
- DC gear motors combine a gearbox with a DC motor to drive heavier loads. They are cheap and easy to find. We recommend this option.

These are some motor examples. From left to right: a DC motor, a plastic gear motor, a metal gear motor, and a servo motor

08 Choosing the gear motors

Buy motors with a gear ratio above 40:1. Although they'll never match perfectly, get them in pairs, as motors that look similar may not have the same speed.

They come in plastic and metal geared flavours. Metal tends to last longer but cost more. Plastic gear motors are cheaper, but larger and not as sturdy. In most of our robots, plastic motors are fine.

90-degree motors fit better in a limited space. Ensure the axles match your wheels. Adapting axles adds complexity and cost.

Gear motors can be salvaged from an electronic toy like an RC excavator or tank.

09 The motor controller

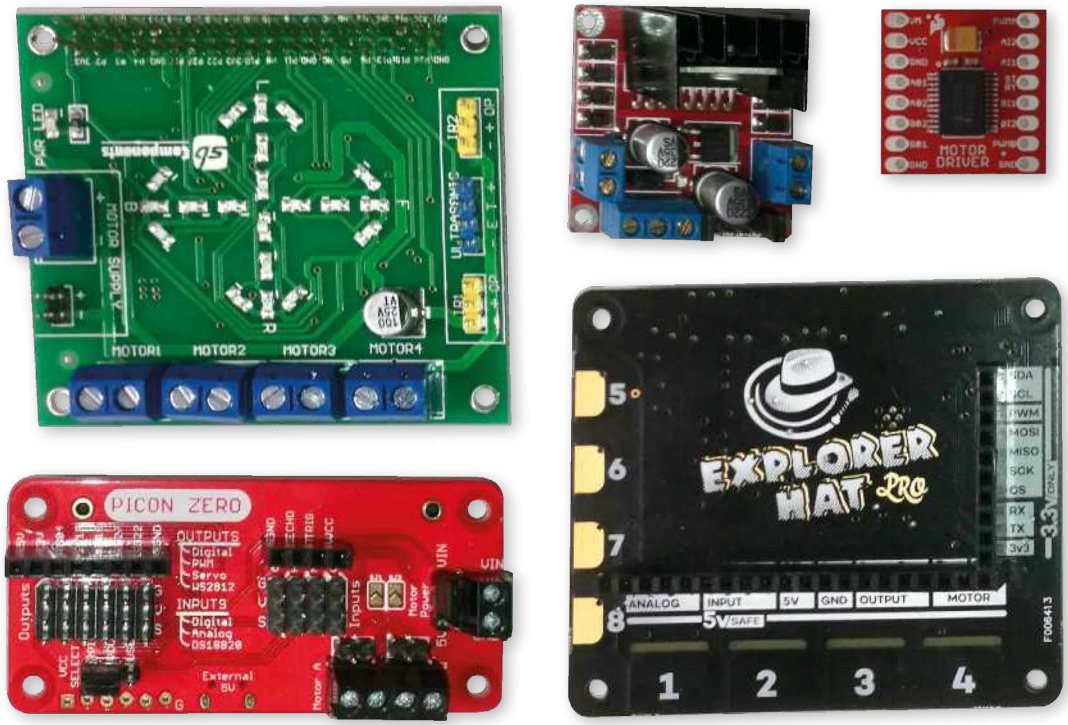
This connects your Raspberry Pi to the motors. Ensure it can control DC motors.

- Buy a Raspberry Pi HAT designed for motors. This is more expensive than other options, but may have additional functionality like logic shifting for sensors, servo motor control, or power regulation. It'll cost £15–£25.
- A DC motor control breakout module. These should have two channels. The cheaper options tend to be based around the L298N or DRV8833 chips, good enough for our purposes. Should be under £8. A simple and reliable option.
- Someone with more electronics knowledge could construct their own H-bridges, but this probably won't cost less than the DC motor modules.

10 Power up

Robots need power for the motors and electronics. A Raspberry Pi needs a smooth 5V to run, with upto 3A capacity. Motors introduce noise that could interfere with your Raspberry Pi.

- Separate power, using a USB power bank for Raspberry Pi and other batteries for motors. Power banks can be pricey, but we've seen smaller ones given away at shows.
- A single set of batteries with a regulator like



A selection of motor controllers. The DRV883 and L298 are small and cheap, with the larger types having more features

a UBEC (universal battery eliminator circuit). These can be bought to provide 5 V and more than 2 A. A UBEC can be found for around £6.

- LiPo/Li-ion are expensive and tricky, only recommended for experienced builders.

11 Making sense

For more interesting code, a robot needs sensors to detect things. Those that don't require extra conversion will save space and a little money. Be prepared to solder on headers.

An HC-SR04P is a cheap way to measure distance. Line-tracking sensors let the robot follow lines.

A camera can be used to follow lines and detect certain objects optically.

An optical flow sensor looks a bit like a camera and lets you measure x-y movement, but cannot tell when you have turned.

Motion sensors can tell you acceleration, rotation, and change in heading. 9DOF types are now around £12.

12 Fasteners and wiring

For fasteners, the 'standoff kits' found online are great; get a selection of M2, M2.5, and M3 including the bolts, nuts, and spacers.

Cable ties are cheap and handy. They can tidy up cabling but can also strap robot parts together.

> “For more interesting code, a robot needs sensors to detect things”

Hot glue is handy for stopping things rattling around; however, don't use it for load-bearing connections. Double-sided sticky tape, foam, and sticky tack can also be used this way.

Jumper wires are used to get signals and power between your components. We recommend getting male-to-female, male-to-male, and female-to-female jumper cable selections (for breadboarding). These will connect Raspberry Pi to most sensors and modules.

13 Tools

A hot glue gun has already been mentioned. Some kind of hand drill and a way to clamp parts will be needed for any custom building. DO NOT hold a part you are drilling in another hand!

Also useful is small screwdriver set, and miniature spanners for fastening parts. We recommend needlenose pliers, side cutters, a multimeter, and a soldering iron.

The best way to keep costs down is to borrow tools – if you have a local hackspace, makerspace, or can talk to a college about using their facilities, you won't need to buy all these tools.

Use SSH (Secure Shell) with Raspberry Pi

Remotely control your Raspberry Pi from a PC, Linux, or Mac computer and transfer files using SSH. By **Lucy Hattersley**

MAKER

Lucy Hattersley

Lucy is the editor of *The MagPi*. She likes retro gaming, building robots, coding, and making magazines and websites.

magpi.cc

SSH (also known as 'Secure Shell') is an encrypted networking technology that enables you to manage computers from the command line over a network.

SSH is handy if you want to quickly connect to a Raspberry Pi from a terminal window on another computer. It's also ideal for lightweight distro installations that don't have graphical interfaces, and projects that don't have a screen (such as robots). It's especially useful when creating Internet of Things (IoT) projects, as these may be embedded inside other hardware.

We've already looked at VNC (Virtual Network Computing) – back in issue 71, **magpi.cc/71** – and Secure Shell offers a similar service. But while VNC shares the entire desktop, SSH works from the command line.

SSH also enables you to share files to, and from, a Raspberry Pi using SCP (secure copy) on the command line, or SFTP (SSH File Transfer Protocol).

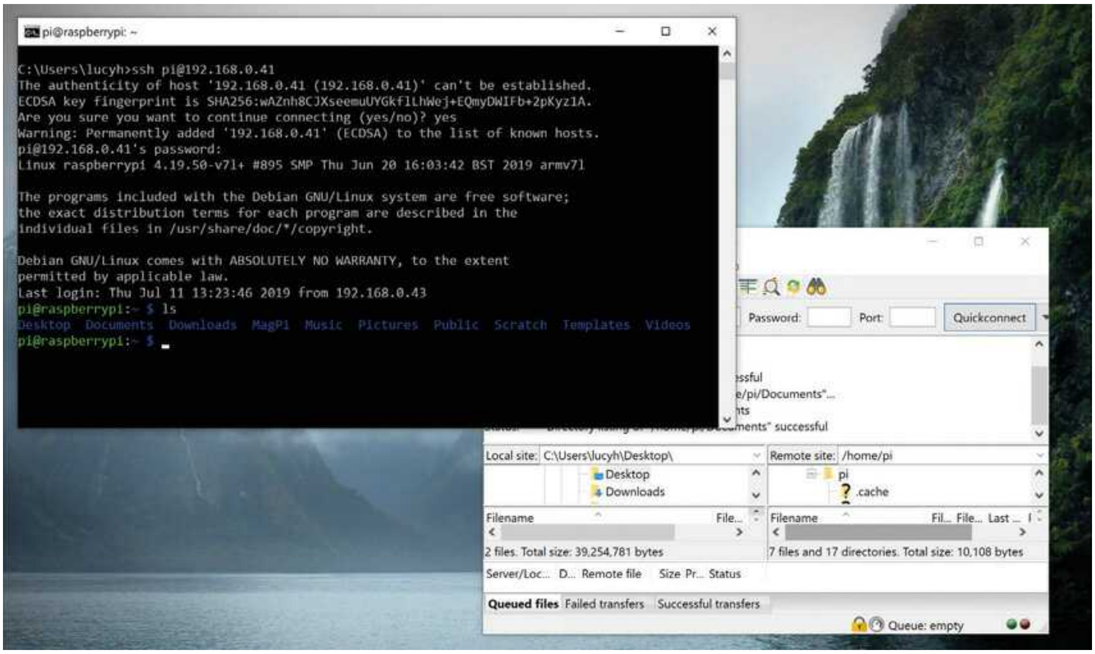
You don't need to install any software to start using Secure Shell. Linux, macOS, and Windows 10 have the SSH command-line application installed by default.

You'll need to use the password for your Raspberry Pi to log in. For security reasons, we strongly recommend changing the default password (**magpi.cc/tZdxha**).

SSH uses an encrypted network, so it doesn't send your password as plain text. More advanced users can control the encryption keys, using ssh-keygen. For now, we'll look at setting up and using Secure Shell.

You'll Need

- Raspberry Pi
- Network router
- Raspbian OS
- Second computer (Windows, macOS, or Linux)



pi@raspberrypi: ~

```
C:\Users\lucyh>ssh pi@192.168.0.41
The authenticity of host '192.168.0.41 (192.168.0.41)' can't be established.
ECDSA key fingerprint is SHA256:wAZnh8CJXseemuUYGkflLhWej+EQmyDWIFb+2pKyz1A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.41' (ECDSA) to the list of known hosts.
pi@192.168.0.41's password:
Linux raspberrypi 4.19.50-v7l+ #895 SMP Thu Jun 20 16:03:42 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 11 13:23:46 2019 from 192.168.0.43
pi@raspberrypi:~ $ ls
Desktop  Documents  Downloads  MagPi  Music  Pictures  Public  Scratch  Templates  Videos
pi@raspberrypi:~ $ _
```

```
essful
e/pi/Documents"...
nts
ments" successful
```

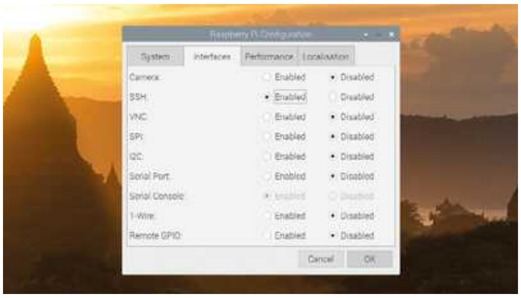
Local site: C:\Users\lucyh\Desktop\

Remote site: /home/pi

2 files. Total size: 39,254,781 bytes

7 files and 17 directories. Total size: 10,108 bytes

Queue: empty



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

▲ SSH is turned off by default in Raspbian and has to be enabled. The easiest way is to use Raspberry Pi Configuration Tool

01 Activate SSH in Raspbian

For security reasons, Secure Shell is not turned on by default in Raspbian. On your Raspberry Pi, choose Menu > Preferences > Raspberry Pi Configuration. Click on Interfaces and set SSH to Enabled. Click OK. You don't need to restart your Raspberry Pi, and SSH will be enabled whenever you use that installation of Raspbian from that point on (be sure to have updated your password from the default, which is 'raspberry').

02 Activate SSH Client in Windows

Linux and macOS both support SSH out-of-the-box; skip ahead to Step 3 if you are using one of those operating systems.

Windows 10 supports SSH, but you need to activate it. Click on Search and look for 'Manage Optional Features'. Click it in Search to open the Settings window.

Click 'Add a feature' and wait for the list of Optional Features to load. Scroll down the list to Open SSH Client (Beta). Click Install.

03 Get your IP address

Connect your Raspberry Pi to a local network. Use wireless LAN, or connect Raspberry Pi directly to a router with an Ethernet cable. Open a Terminal window and enter the following command:

```
hostname
```

You should see:

```
raspberrypi
```

In some circumstances you can use this host name, which is handy if you don't have the IP address, but it's more reliable to use the IP (internet protocol) address. Enter this to get your IP address:

```
hostname -I
```

This will return four numbers separated by dots. For instance, ours is:

```
192.168.0.41
```

Write this number down. You'll need it shortly.

04 Connect via SSH

Open Command Prompt on a Windows PC, or a Terminal window in Linux or macOS. Enter this command:

```
ssh pi@<IP>
```

Replace `<IP>` with the IP address of the Raspberry Pi. In our case, we enter `ssh pi@192.168.0.41`.

The first time you do this, you'll get a message saying that the authenticity can't be established, followed by a long cryptographic hash of letters and numbers. It will say, 'Are you sure you want to continue connecting?'

Enter **yes** and press **RETURN**. You'll be asked to enter the password for your Raspberry Pi.

05 On Raspberry Pi

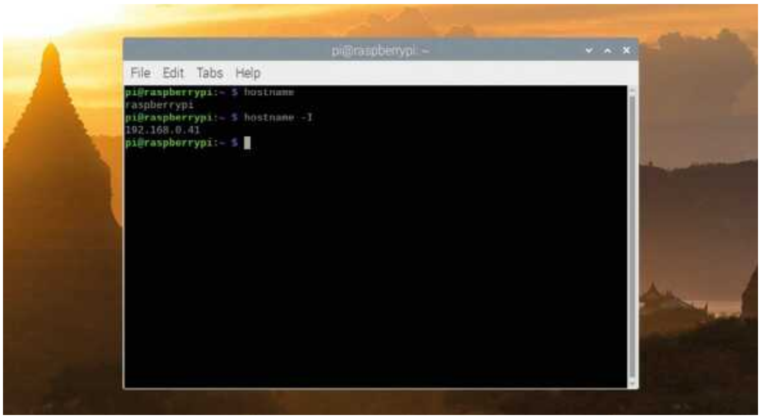
You will now see your usual command line replaced with **pi@raspberrypi: ~$**. You are now logged in and working on the command line from your Raspberry Pi. Enter `ls` and you'll see **Desktop**, **Downloads**, **Documents**, and the other unique Raspberry Pi folders and files. You can create, edit, move, and work with files as if you were using Terminal on your Raspberry Pi.

Top Tip

SSH from boot

You can activate SSH when setting up Raspbian by placing a file named **ssh**, without any extension, onto the boot partition of the microSD card from another computer. When Raspberry Pi boots, it looks for the **ssh** file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text, or nothing at all.

▼ Use the `hostname` command in Terminal to confirm the host name 'raspberrypi', and add the `-I` flag to get your assigned IP (internet protocol) address



```
pi@raspberrypi:~ $ hostname
raspberrypi
pi@raspberrypi:~ $ hostname -I
192.168.0.41
pi@raspberrypi:~ $
```

Top Tip

Turn SSH on and off

You can activate and start SSH from the command line on your Raspberry Pi:

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

And turn it off using:

```
sudo systemctl stop ssh
sudo systemctl disable ssh
```

06 Visual interfaces

There are limitations to SSH. A plain SSH connection can't open programs that use a graphical interface, so you should learn to use command-line alternatives (such as nano or vim instead of Leafpad for text editing). See How to Edit Text files on Raspberry Pi (**magpi.cc/BNFKeP**).

On Linux computers, you can forward your X session over SSH, to allow the use of graphical applications, by using the `-Y` flag:

```
ssh -Y pi@<IP>
```

Now you are on the command line as before, but you have the ability to open up graphical windows. For example, typing:

```
scratch &
```

...will open up Scratch in a graphical window.

Note that X11 is no longer present on Macs with macOS (**magpi.cc/AqEvnA)**, so you have to download and install it (**xquartz.org**) separately to access graphical windows over SSH.

07 Copy files

It's not easy to share files using Secure Shell from the command line, but it's possible and a handy skill to learn. If you're still accessing Raspberry Pi, enter `exit` at the command line to return to your local shell:

```
exit
```

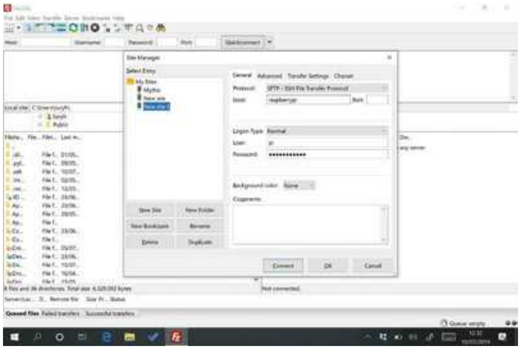
With SSH turned on, you can copy files from your source computer to the Raspberry Pi using SCP. We've downloaded *The MagPi* issue #83 (**magpi.cc/83**) PDF file to our **Downloads** folder. We're going to copy it to the MagPi directory on our Raspberry Pi.

Use the `scp` command followed by the path to the source file, and then the destination location. The destination is `pi@<IP>` followed by a colon ':' and the path to the destination location.

```
scp <source> pi@<IP>:<destination>
```

Replace the `<source>` and `<destination>` parts with the paths to the file and desired destination folder, and add your Raspberry Pi's IP address.

For instance, copying the PDF file on our machine looks like this:



▲ Use SFTP (Secure File Transfer Protocol) in a program such as FileZilla to transfer files to and from your Raspberry Pi via a graphical interface

```
scp Downloads\MagPi83.pdf pi@192.168.0.41:/
home/pi/MagPi
```

This copies *The MagPi* issue #83 PDF from the **Downloads** folder on our Windows PC to the **MagPi** directory on Raspberry Pi.

Notice that the directory path uses backslash ('\') for Windows source and slash ('/') for the Linux destination (if you're using a Mac or Linux machine, it'll be a slash in both source and destination.)

08 Copying files

If you copy a lot of files back and forth from a computer to Raspberry Pi, then it's a good idea to set up SFTP using a dedicated FTP program, such as FileZilla (**filezilla-project.org**).

Install FileZilla on your computer (it is available for Linux, Windows, and macOS).

Open FileZilla and choose File > Site Manager.

Now click New Site and you can give the site a custom name, such as 'Raspberry Pi'.

Change the Protocol to 'SFTP – SSH File Transfer Protocol'.

Enter 'raspberrypi' in the Host field, 'pi' in the User field, and your Raspberry Pi password in the Password field.

Click Connect and you will connect remotely to the Raspberry Pi. You will see all your Raspberry Pi files in the right-hand 'Remote' side of the display. On the left, in the 'Local Site' section will be all the files on your computer. Double-click directory folders in either side to navigate into them, and use the '..' folders at the top of each list to navigate back up to the parent folder.

Double-click a file in Local Site to send it to the Remote section, and vice versa. Follow the FileZilla tutorial if you need more guidance (**magpi.cc/WovBgn**).

RETRO GAMING SALE

Vilros Raspberry Pi 3 Model B Retro Pi Arcade Gaming Kit with Classic USB Gamepad

~~$79.99~~ **$75.99**

Vilros Raspberry Pi 3 Model B+ Retro Arcade Gaming Kit With Retro Gaming Controller Set

~~$114.99~~ **$109.99**

Vilros Raspberry Pi 3 Model B Plus Retro Gaming Kit with 2 Gamepads & Fan-Cooled Retro Gaming Case

~~$89.99~~ **$85.49**

Vilros Retro Gaming USB Classic Controller Set of 5

~~$39.99~~ **$37.99**

USE THE CODE **RETRO** FOR 5% OFF

Check us out at ***www.vilros.com*** /vilrosusa

Crack Coupon Code for ADDITIONAL 20% Savings! M ☊ Ƹ Ұ Ծ

"*Restrictions Apply" "**Offer Expires May 31, 2019"

Control servos with CircuitPython and Raspberry Pi

Want to use your powerful new Raspberry Pi 4 to control a robot? Adafruit's Blinka library makes it very easy

MAKER

Melissa LeBlanc-Williams

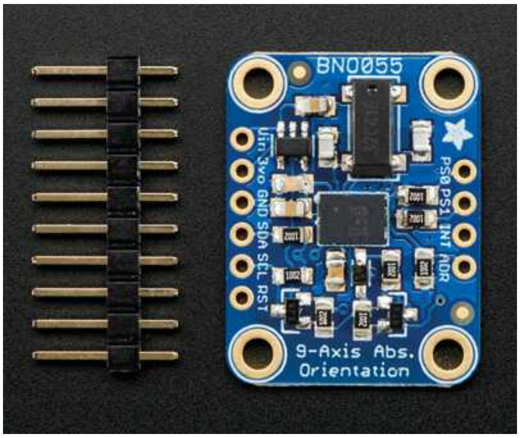
Melissa works on the compatibility layer for Python that allows Raspberry Pi to use many of the CircuitPython libraries. She is currently working full-time on Adafruit projects.

@makermelissa

Raspberry Pi 4 is the perfect computer for controlling robots. In celebration of Adafruit's upcoming CircuitPython Day (8 August 2019), we're going to take a look at how easy it is to use CircuitPython on Raspberry Pi.

CircuitPython is designed to control low-cost microcontroller boards. For running robotics, Adafruit has many different breakout sensors and boards. We're going to be looking at two boards that make a great combination in robotics: the BNO055 9-Degrees-of-Freedom (DOF) Absolute Orientation Breakout and the 16-Channel 12-bit PWM/Servo Driver.

The BNO055 is unique in that the sensor includes a built-in microprocessor that takes all of the data from the other sensors and calculates everything you want to know, for improved accuracy.



Rather than spending weeks or months fiddling with algorithms of varying accuracy and complexity, you can have meaningful sensor data in minutes

The 16-channel servo driver allows you to drive up to 16 different servos, which is excellent for robotics. Both of these boards can be driven through I^{2}C, so they only take up two pins on Raspberry Pi. For this project, you will need both male-to-male and male-to-female jumper wires.

01 Solder the headers

If you haven't already soldered the headers that came with the breakouts to the boards, let's do it. One of the easiest ways is to cut the headers to length and insert them into some breadboard with the long pins facing down. Then place the breakout on the pins, apply heat, and solder.

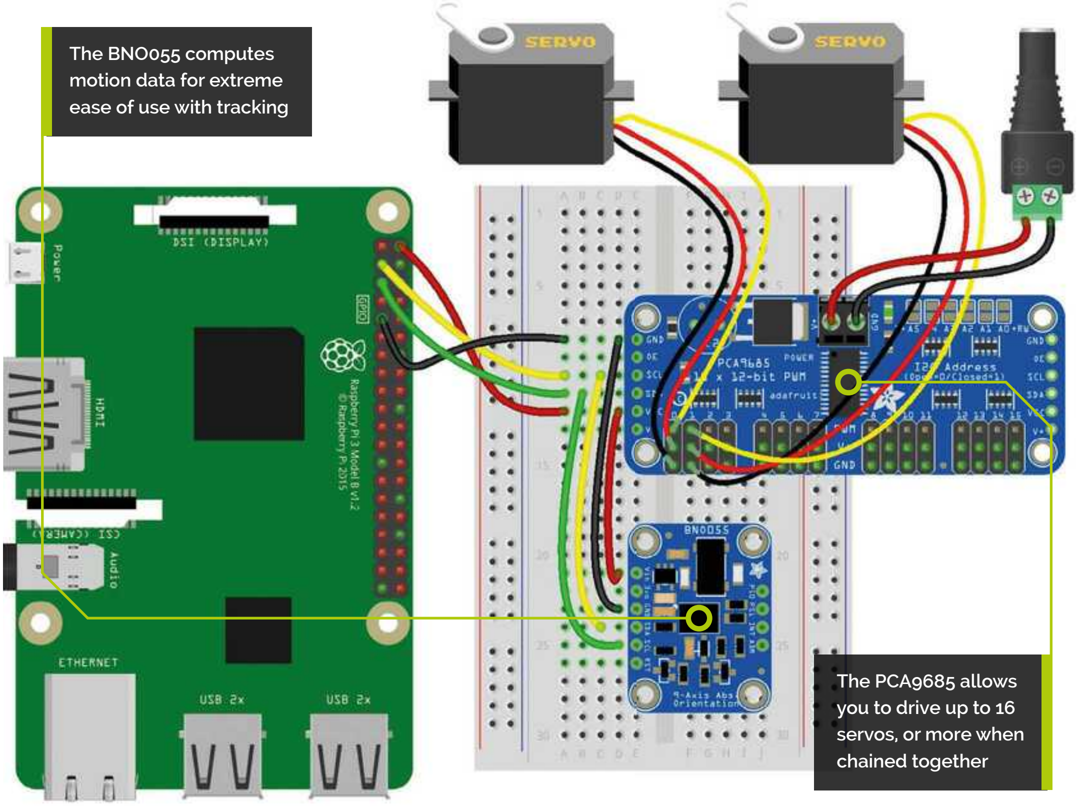
Once you have all of the headers connected, take the servo driver, flip it over, and solder the terminal block to the top side. Optionally, you can also add a capacitor to the servo driver board, especially if you have a lot of servos.

02 Connect the servo driver

The first item that we'll hook up is the 16-channel servo driver. Insert the breakout into the breadboard and wire it up to your Raspberry Pi according to the diagram (**Figure 1**). Take the female DC power adapter, loosen the screws, insert a male-to-male jumper wire in each hole, and tighten down the screws. Connect the other ends of the jumper wire into the terminal block on the servo driver according to the diagram. Be sure to connect the 5 V wire to VCC and not V+.

You'll Need

- 16-Channel Servo Driver **magpi.cc/fepZYG**
- BNO055 9-DOF Breakout **magpi.cc/UBiKnr**
- 2 × Servos **magpi.cc/syOCun**
- 5 V power supply **magpi.cc/kzPRHr**
- Female DC power adapter **magpi.cc/fjPTMp**
- Breadboard
- Jumper wires



The BNO055 computes motion data for extreme ease of use with tracking

The PCA9685 allows you to drive up to 16 servos, or more when chained together

Figure 1

Top Tip

Easy alignment

A little trick that makes soldering connectors easier is to solder the first pin, check alignment, and reheat if necessary.

03 Connect the orientation sensor

The next item we'll hook up is the BNO055 Absolute Orientation Sensor. Insert the breakout board into the breadboard. Connect the four wires according to the **Figure 1** diagram. Since this is using I²C like the previous board, all four wires should be connected in parallel. Once that is connected, you can plug the two servos into the servo driver in positions 0 and 1, minding the orientation of the plug. Lastly, don't forget to apply external power to the servo driver through the barrel connector we wired up in the last step.

04 Prepare your Raspberry Pi

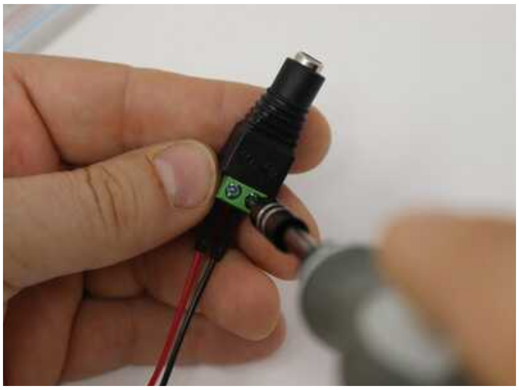
Start by downloading and flashing the latest version of Raspbian Buster onto a microSD card. You should be able to use any of the available OS versions for this project, since we will be doing most of the steps from the command line.

Place the microSD card into your Raspberry Pi, power up, and follow the usual setup prompts. When requested, update and reboot if necessary. If you are running Buster Lite, be sure to run the following command:

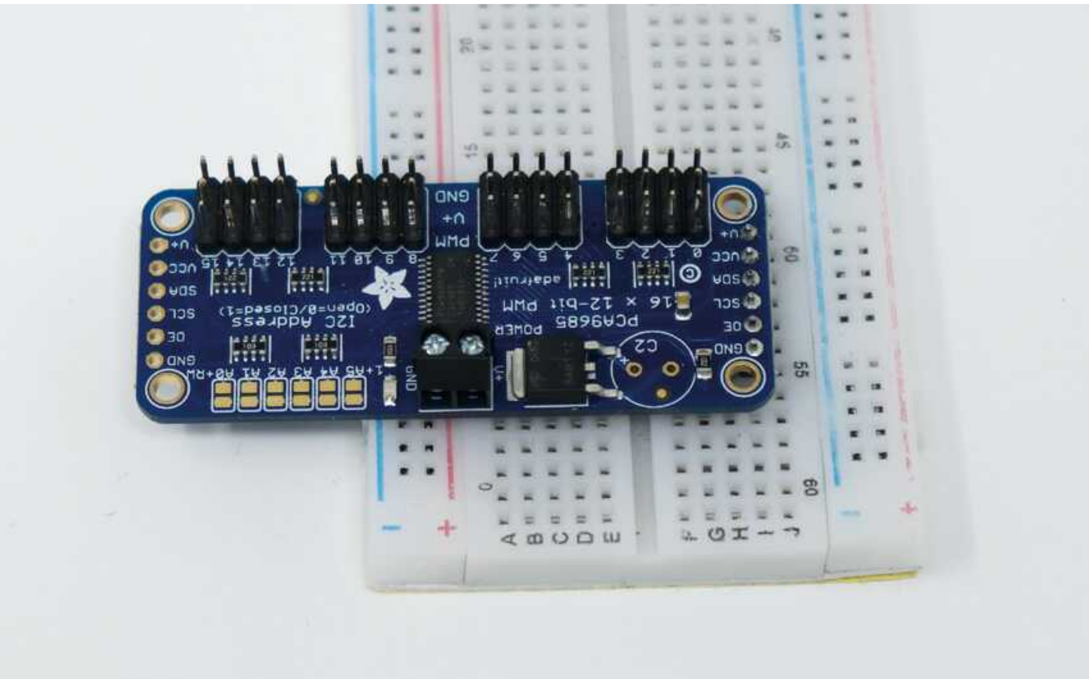
```
sudo apt update && sudo apt upgrade
```

If you are running the desktop version, you will automatically be prompted to update. Updating is very important.

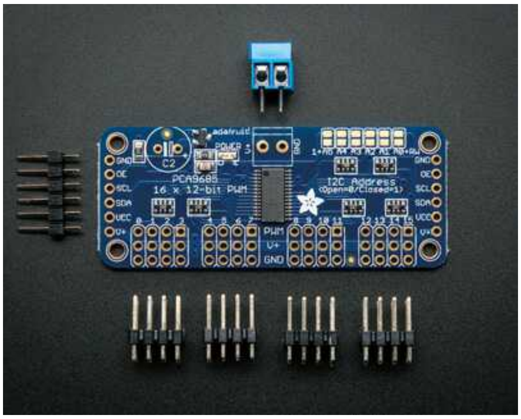
> “Both of these boards can be driven through I²C, so they only take up two pins on Raspberry Pi”



Use a screwdriver to connect the male-to-male jumpers to the female DC adapter



Six address-select pins so you can wire up to 62 of these on a single I^{2}C bus – a total of 992 outputs



The PCA9685 servo driver is able to drive up to 16 servos and only uses a few pins on Raspberry Pi

Top Tip

Reuse an old board

Did you upgrade your old Raspberry Pi for a new Raspberry Pi 4? Put it to good use: Blinka runs on older Raspberry Pi boards too.

05 Laying the foundation

Next, we're going to install some of the dependencies that we will need. If you are running the desktop versions, start by opening up a Terminal window (**CTRL+ALT+T**). Raspbian Buster should come with Python 3 and pip3, which are required; if not, you can run the following command to install them:

```
 sudo apt-get install -y python3 git
python3-pip
```

Next, install or upgrade setuptools by typing the following command:

```
sudo pip3 install --upgrade setuptools
```

The setuptools package in Python will help with installing Blinka when we get to it in a few steps.

06 Enabling the interfaces

Now we need to enable the I^{2}C interface, which both of the boards use. We're going to use the raspi-config tool for this. If not already at a command prompt, open up a Terminal window. Type the following command:

```
sudo raspi-config
```

Once raspi-config opens up, open up Interfacing Options, select 'I2C', and choose Yes to enable the I^{2}C interface.

Next, go back into Interfacing Options again and do the same thing, except this time selecting 'SPI' to enable the SPI interface so that you'll have it available.

07 Enable clock-stretching

Certain I^{2}C sensors such as the BNO055 need to have clock-stretching enabled. This slows down the I^{2}C clock so the sensor can keep up with the speedy Raspberry Pi and not drop any data. We'll want to slow it down from 100kHz to 10kHz. We do that by adding a line to the configuration file. To enable clock-stretching, you'll want to open up **/boot/config.txt** in your favourite editor (e.g. nano). At the end of the file, add the following line:

```
dtparam=i2c_arm_baudrate=10000
```

After that, save the file and reboot your Raspberry Pi. Clock-stretching is now enabled.

Blinka is the name of the CircuitPython compatibility layer that runs on Raspberry Pi, and is named after the CircuitPython mascot

08 Install Blinka

Next, we want to install any remaining Python libraries that Blinka requires. For the first one, we'll use RPI.GPIO, which should already be installed. You can verify this by going to a command prompt and typing:

```
pip3 install RPI.GPIO
```

Finally, we'll actually install Blinka. As well as Blinka itself, this will install a few dependencies such as PureIO and PlatformDetect. Type the following command to install Blinka and any other dependencies:

```
pip3 install adafruit-blinka
```

09 Install driver libraries

Next, we need to install the required libraries to use Blinka with the BNO055 and servo driver. Make sure you have a Terminal window open and type the following commands:

```
pip3 install adafruit-circuitpython-pca9685
pip3 install adafruit-circuitpython-servokit
pip3 install adafruit-circuitpython-bno055
```

This will install the drivers for both boards, as well as the ServoKit helper library that makes using servos very easy.

The PCA9685 servo driver allows Raspberry Pi to drive some fairly large and powerful servos

> Enabling clock-stretching slows down the I²C clock so the sensor can keep up with the speedy Raspberry Pi and not drop any data

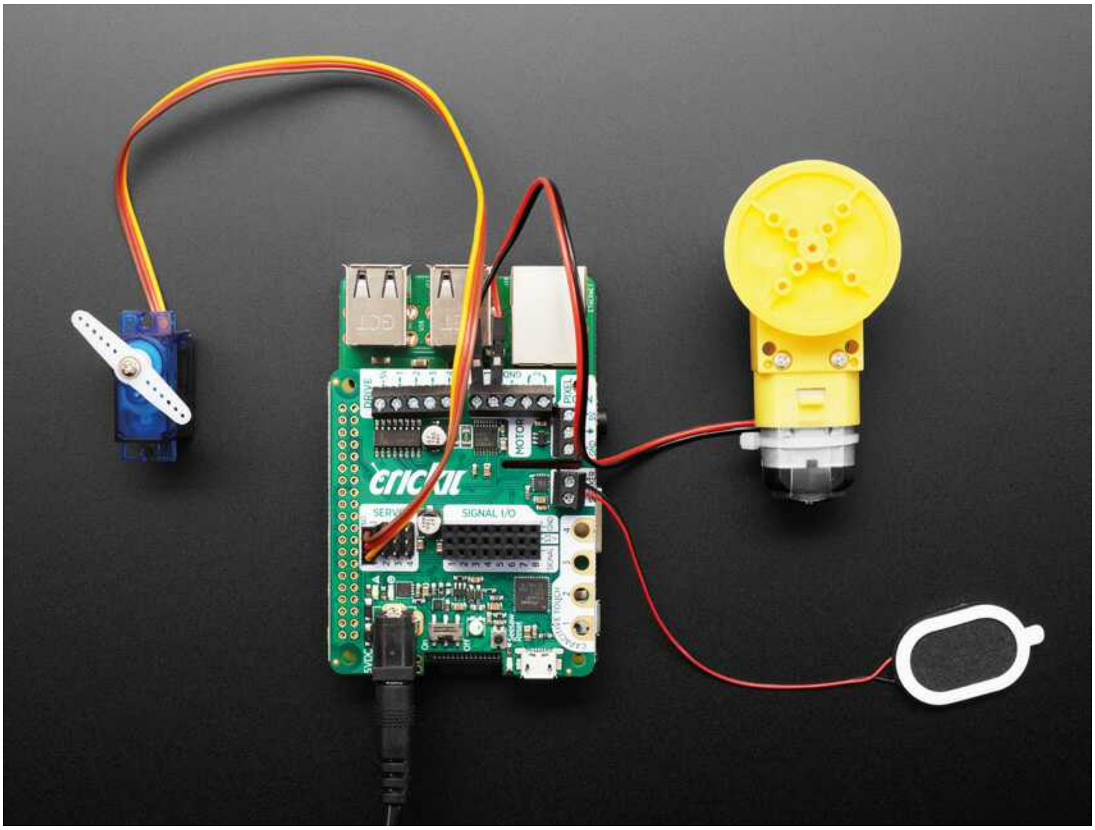
tiltdemo.py

Language: **Python 3**

DOWNLOAD THE FULL CODE:

magpi.cc/GNXPon

```python
"""
In this Demo we will drive two servos based on the Tilt along the Y and Z axis
of the BNO055 9-Degrees of Freedom IMU Sensor. This could easily be extended
to drive servos on all three axis as well as use a host of other information
including lateral acceleration.
"""

import board
import busio
from adafruit_servokit import ServoKit
import adafruit_bno055

# Set channels to the number of servo channels on your kit.
kit = ServoKit(channels=16)

# Setup the BNO055 to read data
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_bno055.BNO055(i2c)

while True:
    # Get the Euler angles from the BNO055
    (x, y, z) = sensor.euler

    # Euler angles are between -180 and 180
    # We want to translate them to the Servo angles between 0-180
    try:
        kit.servo[0].angle = (y + 180) / 2
        kit.servo[1].angle = (z + 180) / 2
    except ValueError:
        # Pass on any values that are out of range
        pass
```



The CRICKIT HAT makes robotics easy on Raspberry Pi

> As you tilt the breadboard in various directions, the servos should respond depending on the axis

10 Running the code

Now we want to save the demo code in your home directory (**/home/pi**) on Raspberry Pi. You can either open up your favourite editor on Raspberry Pi and enter it directly, or copy the file with an SCP or FTP utility if you have those services enabled. Once you have the code in your home directory, you can run it from the command line by typing the following:

```
python3 demo.py
```

Once you have the code running, pick up your breadboard and start moving it around. As you tilt it in various directions, the servos should respond depending on the axis that you tilt around. You could always add a third servo and respond to the third axis and slightly modify the code.

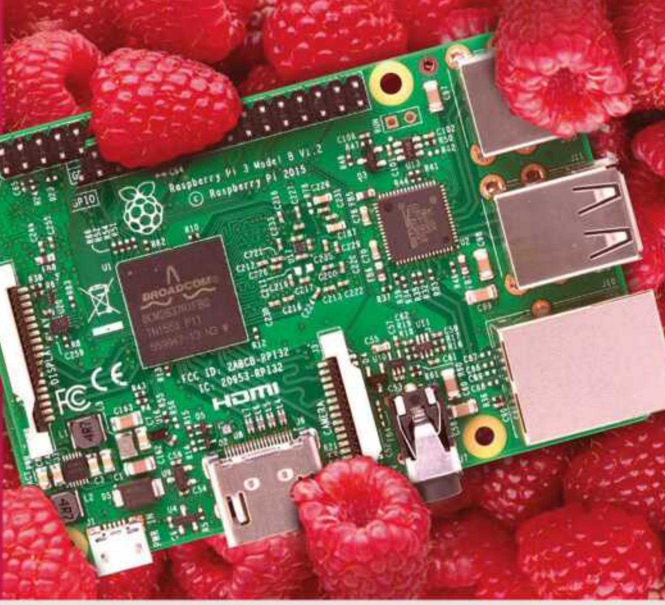
11 Other options

Adafruit offers plenty of other options for using Raspberry Pi for robotics, including the CRICKIT HAT. This plugs right into the GPIO header on Raspberry Pi and allows up to four servos, but is also able to control stepper motors and DC motors, drive NeoPixels, play sound through a speaker, and has capacitive touch. It has eight digital/analogue inputs that can also be configured as digital outputs. All the features on this board are wrapped up in one nice CircuitPython library.

12 More learning resources

Adafruit offers many more learning guide resources to show you what the Raspberry Pi is capable of and so much more. At the time of writing, there are close to 2000 guides available. You can go to **learn.adafruit.com** and browse through all of the guides available.

CircuitPython is being added to more boards all of the time, with guides being written for them. You can always go to **circuitpython.org/blinka** to see which other boards are supported.



Build Demolition Man's verbal morality ticketing machine

Use voice recognition and a thermal printer to issue printed fines if your friends say a swear word

MAKER

PJ Evans

PJ is a writer, trainer, developer, and wrangler of Milton Keynes Raspberry Jam. He wishes you joy-joy and mellow greetings.

@mrpjevans

You'll Need

- USB microphone **magpi.cc/NwjF0a**
- Speaker pHAT **magpi.cc/TkYcvO**
- Thermal printer **magpi.cc/xbfAtk**
- 3D-printed case **magpi.cc/fHeMry**

In the 1993 action movie *Demolition Man*, Sylvester Stallone stars as a 1990s cop transported to the near-future. Technology plays a central role in the film, often bemusing the lead character. In a memorable scene, he is repeatedly punished by a ticketing machine for using bad language (a 'violation of the verbal morality statute').

In the future, an always-listening government machine detects every banned word and issues a fine in the form of a receipt from a wall-mounted printer. This tutorial shows you how to build your own version using the Google Voice API and a thermal printer. Not only can we replicate detecting banned words, but it also doubles as a handy voice-to-paper stenographer (if you want a more serious use).

▲ Luckily, the printer's connector matches up with the GPIO pins and can be directly attached

01 Prepare the hardware

We built a full 'boxed' project, but you can keep it simple if you wish. Raspberry Pi needs a method for listening, speaking, and printing. The easiest solution is to use USB for all three. After prototyping using Raspberry Pi 4 and various USB devices, we settled on a Raspberry Pi Zero W with a small USB mic and Pimoroni Speaker pHAT to save space. A Pico HAT Hacker allowed the connection of both the printer and Speaker pHAT, as they don't share GPIO pins. This bit of space-saving means the full assembly can be installed inside the 3D-printed case, along with the printer.

02 Connect the printer

To issue our receipts we're using a thermal printer, the kind found in supermarket tills. This particular model is surprisingly versatile, handling text and graphics. It takes standard 2.25-inch (57 mm) receipt paper, available in rolls of 15 metres. When printing, it does draw a lot of current, so we advise using a separate power supply. Do not attempt to power it from your Raspberry Pi itself. You may need to fit a barrel connector and source a 1.5 A, 5 V supply. The printer uses a UART/TTL serial connection which neatly fits on to the GPIO. Although the printer's connection is listed as being 5 V, it is in fact 3.3 V, so can be directly connected to the ground, TX, and RX pins (physical pins 6, 8, 10) on the GPIO.

This thermal printer does not require ink and uses standard receipt rolls

Tucked inside are a Raspberry Pi Zero W, speaker, and microphone

03 Install and configure Raspbian

Get yourself a copy of Raspbian Buster Lite and burn it to a microSD card using a tool like Etcher. You can use the full version of Buster if you wish. Perform the usual steps of getting a WiFi connection and updating to the latest version using `sudo apt update && sudo apt -y upgrade`. From a command prompt, run `sudo raspi-config` and go to 'Interfacing options', then 'Enable serial'. When asked if you would like the login shell to be accessible, respond 'No'. On the next question, 'Would you like the serial port hardware to be enabled?' reply 'Yes'. Now reboot your Raspberry Pi.

04 Test the printer

Let's make sure the printer is up and running. Double-check you've connected the header to the GPIO correctly and power up the printer. The LED on the printer should flash every few seconds. Load in the paper and make sure it's feeding correctly. We can talk to the printer directly, but the Python 'thermalprinter' library (**magpi.cc/JbfiPV**) makes coding for it so much easier. To install the library:

```
sudo apt install python3-pip
pip3 install thermalprinter
```

Create a file called **printer.py** and enter in the code in the relevant listing. Run the code using:

printer.py

Language: **Python 3**

DOWNLOAD THE FULL CODE:

magpi.cc/WYrUcu

```python
from thermalprinter import ThermalPrinter

with ThermalPrinter(port='/dev/serial0') as printer:
    printer.out("Hello, world!")
    printer.feed(2)
```

speech_to_text.py

Language: **Python 3**

```python
import io
import os

# Imports the Google Cloud client library
from google.cloud import speech
from google.cloud.speech import enums
from google.cloud.speech import types

# Instantiates a client
client = speech.SpeechClient()

# The name of the audio file to transcribe
file_name = os.path.join(
    os.path.dirname(__file__),
    'test.wav')

# Loads the audio into memory
with io.open(file_name, 'rb') as audio_file:
    content = audio_file.read()
    audio = types.RecognitionAudio(content=content)

config = types.RecognitionConfig(
    encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,
    sample_rate_hertz=44100,
    language_code='en-GB')

# Detects speech in the audio file
response = client.recognize(config, audio)

for result in response.results:
    print('Transcript: {}'.format(
result.alternatives[0].transcript))
```

Any mix of USB speakers and microphones will do, although the better the mic, the more accurate the voice recognition

Top Tip

Better mic for better results

The quality of the microphone is key, especially if you want this to work at a distance.

```
python3 printer.py
```

If you got a nice welcoming message, your printer is all set to go.

05 Test the microphone

Once your microphone is connected to Raspberry Pi, check the settings by running:

```
alsamixer
```

This utility configures your various sound devices. Press **F4** to enter 'capture' mode (microphones), then press **F6** and select your device from the list. Make sure the microphone is not muted (**M** key) and the levels are high, but not in the red zone. Back at the command line, run this command:

```
arecord -l
```

This shows a list of available recording devices, one of which will be your microphone. Make a note of the card number and subdevice number.

To make a test recording, enter:

```
arecord --device=hw:1,0 --format S16_LE
--rate 44100 -c1 test.wav
```

If your card and subdevice numbers were not '0,1', you'll need to change the device parameter in the above command.

Say a few words, then use **CTRL+C** to stop recording. Check the playback with:

```
aplay test.wav
```

06 Choose your STT provider

STT means 'speech to text' and refers to the code that can take an audio recording and return recognised speech as plain text. Many solutions are available and can be used in this project. For the greatest accuracy, we're going to use Google Voice API. Rather than doing the complex processing locally, a compressed version of the sound file is uploaded to Google Cloud and the text returned. However, this does mean Google gets a copy of everything 'heard' by the project. If this isn't for you, take a look at Jasper (**magpi.cc/OsiutD**), an alternative that supports local processing.

07 Create your Google project

To use the Google Cloud API, you'll need a Google account. Log in to the API Console at **console.developers.google.com**. We need to create a project here. Next to 'Google APIs', click the drop-down, then 'New Project'. Give it a name. You'll be prompted to enable APIs for the project. Click the link, then search for 'speech'. Click on 'Cloud Speech-to-Text API', then 'Enable'. At this point you may be prompted for billing information.

Don't worry, you can have up to 60 minutes of audio transcribed for free each month.

08 Get your credentials

Once the Speech API is enabled, the screen will refresh and you'll be prompted to create credentials. This is the info our code needs to be granted access to the speech-to-text API. Click on 'Create Credentials' and on the next screen select 'Cloud Speech-to-text API'. You're asked if you're planning to use the Compute Engine; select 'no'. Now create a 'service account'. Give it a different name from the one used earlier, change the role to 'Project Owner', leave the type of file as 'JSON', and click 'Continue'. A file will be downloaded to your computer; transfer this to Raspberry Pi.

09 Test Google recognition

When you're happy with the recording levels, record a short piece of speech and save it as **test.wav**. We'll send this to Google and check our access to the API is working. Install the Google Speech-To-Text Python library:

```
sudo apt install python3-pyaudio
pip3 install google-cloud-speech
```

Now set an environment variable that the libraries will use to locate your credentials JSON:

```
export GOOGLE_APPLICATION_CREDENTIALS="/
home/pi/[FILE_NAME].json"
```

(Don't forget to replace `[FILE_NAME]` with the actual name of the JSON file)

Using a text editor, create a file called **speech_to_text.py** and enter the code from the relevant listing. Then run it:

```
python3 speech_to_text.py
```

If everything is working correctly, you'll get a text transcript back within a few seconds.

10 Live transcription

Amazingly, Google's speech-to-text service can also support streaming recognition, so rather than capture-then-process, the audio can be sent as a stream and a HTTP stream of the recognised text comes back. When there is a pause in the speech, the results are finalised, so then we can send the results to the printer. If all the code you've entered so far is running correctly, all you need to do is download the **stenographer.py** script from **magpi.cc/WYrUcu** and start it using:

```
python3 stenographer.py
```

You are limited on how long you can record for, but this could be coupled with a 'push to talk' button so you can make notes using only your voice!

> In the event anyone mentions a banned word, a buzzer will sound and a fine will be printed

11 Banned word game

Back to *Demolition Man*. We need to make an alarm sound, so install a speaker (a passive one that connects to the 3.5 mm jack is ideal; we used a Pimoroni Speaker pHAT). Download the **banned.py** code from **magpi.cc/WYrUcu** and edit it in your favourite text editor. At the top is a list of words. You can change this to anything you like (but don't offend anyone!). In our list, the system is listening out for a few mild naughty words. In the event anyone mentions one, a buzzer will sound and a fine will be printed. Make up your list and start the game by running:

```
python3 banned.py
```

Now try one of your banned words.

12 Package it up

Whatever you decide to use this project for, why not finish it up with a 3D-printed case so you package up the printer and Raspberry Pi with the recording and playback devices and create a portable unit? Ideal for pranking friends or taking notes on the move!

See if you can invent any other games using voice recognition, or investigate the graphics capability of the printer. Add a Raspberry Pi Camera Module for retro black and white photos. Combine it with facial recognition to print out an ID badge just using someone's face. Over to you.

Top Tip

Print more

Thermal printing is great fun. Explore the Python libraries to generate barcodes and images.

Part 02

Learn Lua with PICO-8

You've played other people's PICO-8 games, now it's time to make your own! Create a retro space-shooter whilst learning the Lua language

MAKER

Dan Lambton-Howard

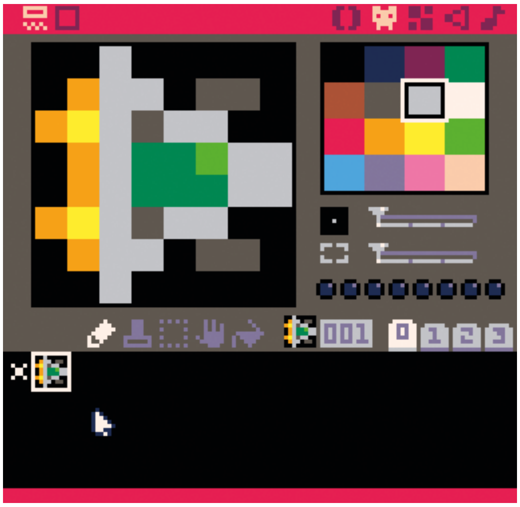
Dan is an independent game designer based in Newcastle upon Tyne, where he is lucky enough to make games for his PhD.

@danhowardgames

Coding in PICO-8 is done in a lightweight and easy-to-learn language called Lua. It's quick, powerful, and is by far the most popular scripting language in game development today, having been used in everything from Dark Souls to World of Warcraft. So even if you're just a little bit interested in game dev, it's a good skill to have. This tutorial will walk you through using Raspberry Pi and PICO-8 to make a simple retro space-shooter, a great foundation for things to come.

01 Launch sequence initiated

First things first, launch PICO-8 and, from the console, hit **ESC**. You should now be staring at the code editor. It isn't the most beautiful text editor, but you'll sure grow to love it! We want to start with a blank slate, so if you already have a cart loaded you might need to reboot in the console. Before we start with the code, two things to note: PICO-8 doesn't use upper case letters, everything is lower case (so hands off that Caps-Lock). Secondly, similar to Python, there is no need for semicolons to end lines.



Our cute spaceship. Sorry, we mean, hyper advanced warfighter… ahem

02 The holy trinity of PICO-8

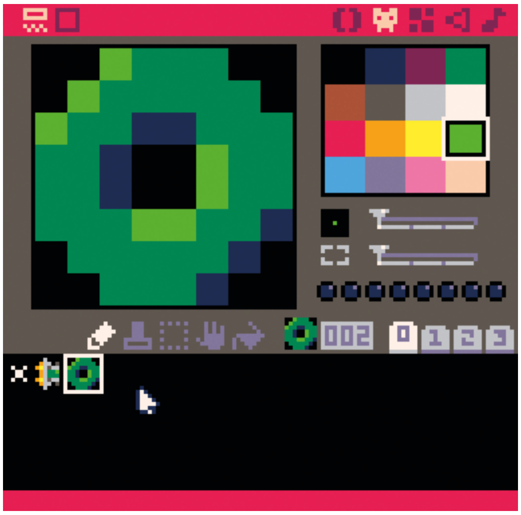
PICO-8 has three special functions that structure any PICO-8 program. The first, `_init()`, is run once at program startup, whilst `_update()` and `_draw()` are called 30 times a second, meaning games are 30 fps by default. Define these three functions in your code, as in **Figure 1**. You can also give your game a title by using `--` to comment. We've chosen something suitably B-movie for our retro space-shooter. Hit **ESC** to return to the terminal and type `save yourgamename` to save your cart (you should do this often), then **ESC** again to hop back to the code editor.

03 Ready Player One

No space-shooter is complete without a solitary pilot flying a super-advanced experimental warfighter. Switch to the sprite editor (using the tabs at the top right) and draw our ship. Don't worry too much about graphics as we'll be covering that in a later tutorial. Doodle a spaceship facing right in sprite slot 001. Write the following code into your `_init()` function to declare the player as a table: `player = {x=20,y=64,sprite=1}`. Tables are very useful in Lua; this one contains a reference to your player's x and y coordinates, as well as what sprite to draw.

You'll Need

- PICO-8 **magpi.cc/pico8**
- Keyboard and mouse



Those blobs came from the moon! Don't worry too much about graphics at this stage

04 Moving the player

Now, within the `_draw()` function, add `cls()`. Then, on a new line, add `spr(player.sprite,player.x,player.y)`. This will tell PICO-8 to clear the screen each frame, then draw the player at the x and y coordinates stored in the `player` table. You can test this by hitting **CTRL+R**. You should see your little ship on the screen. Now let's get them moving! The following code placed in the `_update()` function should move the player when the direction keys are pressed.

```
  if btn(0) then player.x-=2 end
  if btn(1) then player.x+=2 end
  if btn(2) then player.y-=2 end
  if btn(3) then player.y+=2 end
```

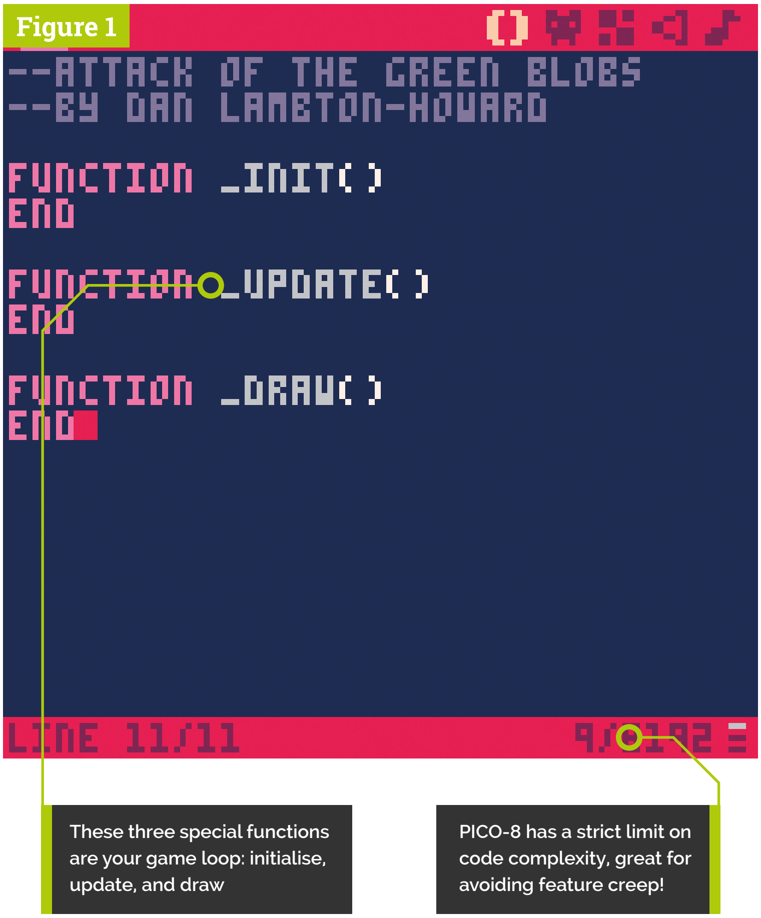
05 The enemy reveal themselves

But what are we fighting against? Those evil green blobs from outer space, that's who! Draw a suitably alien-looking creature in sprite slot 002. We want our enemies to be attacking in waves. You can see the full code in the source, but briefly we are declaring a new empty table in `_init()` named `enemies`. Then we write a new function `create_enemies()` which creates a new enemy (similar to how we created the player) and then adds it to the `enemies` table. Lastly, a new function `create_wave()` spawns a number of enemies.

06 They're coming for us!

To actually draw the enemies, we need to write a `for` loop in `_draw()`. This loops over all the enemies in our `enemies` table, once per frame, and draws them on the screen. Aliens are no threat if they just sit there, so we need them to come towards the player. A simple way of doing this is to write another loop in `_update()`that alters each enemy's x value per frame. Now let's actually spawn some. Add `create_wave(rnd(6)+5)` into `_init()`. This will call our enemy wave function that we wrote earlier, and create five to ten aliens on startup.

Figure 1



```
--ATTACK OF THE GREEN BLOBS
--BY DAN LAMBTON-HOWARD

FUNCTION _INIT()
END

FUNCTION _UPDATE()
END

FUNCTION _DRAW()
END
```

LINE 11/11 9/8192

These three special functions are your game loop: initialise, update, and draw

PICO-8 has a strict limit on code complexity, great for avoiding feature creep!

> Aliens are no threat if they just sit there, so we need them to come towards the player

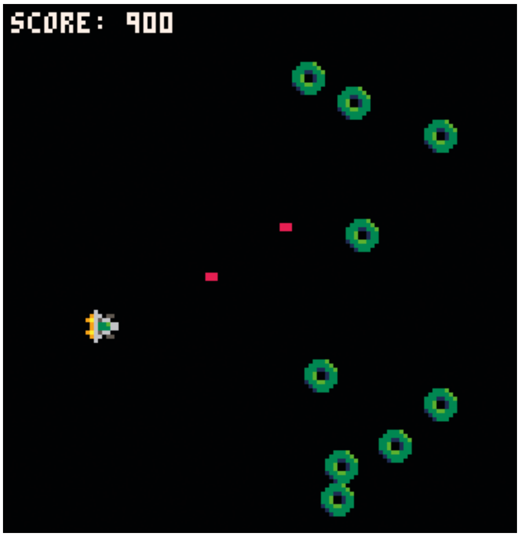
07 Our pilot strikes back

Run your game and you should be immediately swarmed by aliens. We need some way of fighting back! Let's code some lasers. We do this in a very similar way to enemies, by declaring an empty `lasers` table, making a new function to create a laser, and writing a `for` loop to update each laser's position, and one to draw

Top Tip

Don't trip over a table

Lua's tables are a flexible and powerful data type. But beware: indexing starts at 1 and not 0 like in most other languages!



It's amazing how quickly you can get a game up and running on PICO-8

Top Tip

Organise your code

A great way to keep your code under control is to use PICO-8's built-in tabs, found in the top-left corner of the editor.

each laser (as a red rectangle). The difference is we add `if btnp(4) then create_laser(player.x+5,player.y+3) end` after our player movement. This creates a new laser in front of the player when they press **X** (or B button on a controller).

08 High-speed collision

You've probably noticed that our lasers are entirely ineffective against the alien scum. That's because we haven't coded any collision detection. There are many ways to do this – entire books have been written about the topic – but let's keep things simple. We'll declare a new `enemy_collision()` function that checks if a point is inside an 8×8 pixel square around an enemy. If so, it returns `true`. Next, within our enemies update loop (step 6) we'll also loop through the `lasers` table to check collisions; if so, we delete both the laser and the enemy, destroying them both. Kerpow!

> Upon a collision, we delete both the laser and the enemy, destroying them both

09 Game over

The tides of battle have turned, but it's hardly a fair fight. Let's reuse the same collision function to check if an enemy has struck the player. Again, within the enemy update loop, we check for collision with a point in the player's ship. If we find a collision, we'll declare a new variable `gameover = true` (cue dramatic music). We will then wrap the player move and draw code in a conditional `if not gameover then [code] end`, so that the player can't keep playing, and a `print` statement in `_draw()` to really hammer the point home.

10 They just keep coming

So now we have our pilot, lasers, aliens, and some collisions. But let's increase the tempo and have aliens arriving in ever-increasing waves. To do this we will create a timer that increments each frame, and spawn a new wave every three seconds. Declare `wavetimer = 0` and `waveintensity = 5` in `_init()` and then, in `_update()`, increment the timer by one. Let's also include a conditional that spawns a new wave, and increases the intensity, when the timer hits 90 (30 frames per second × 3).

11 Space debris

Now we need to do a bit of tidying up. For example, those lasers you've been firing? They don't actually stop off screen, you know. They continue forever, and will eventually start slowing down PICO-8 as it tries to process thousands of off-screen lasers. The same for aliens. To fix this, within the laser and enemy update loops, check if each is out of screen bounds (0–127 for both x and y) and delete any strays. Additionally, to prevent the player from going off screen, add `player.x = mid(0,player.x,120)`, and the same for y, in `_update()`.

12 Scoring big

Survival is one thing, but high scores are better. To cap this tutorial off, create a new variable `score = 0` in `_init()` and add a new line when an alien is destroyed that adds to `score`. Choose whatever amount you want, but 100 sounds good, doesn't it? Adding `print('score: '..score,2,2,7)` to the end of `_draw()` should show the score on screen. That's all for now, but we'll be looking at graphics and sound in the next few issues, as well as giving our little space-shooter some more oomph!

Lua is used extensively in World of Warcraft as a scripting language

SpaceShooter.p8

DOWNLOAD THE FULL CODE:

magpi.cc/eDDppk

Language: **Lua**

```lua
--attack of the green blobs
--by dan lambton-howard
function _init() -- called once at start
    player = {x=20, y=64, sprite=1} --player table
    enemies = {}
    lasers = {}
    create_wave(rnd(6)+5) --start game with a wave
    wavetimer = 0
    waveintensity = 5
    score = 0
end

function _update() -- called 30 times per second
    wavetimer+=1
    if not gameover then --only move the player if not gameover
        if btn(0) then player.x-=2 end
        if btn(1) then player.x+=2 end
        if btn(2) then player.y-=2 end
        if btn(3) then player.y+=2 end
        if btnp(4) then create_laser(player.x+5,player.y+3) end
    end
    --stop player going off screen edges
    player.x=mid(0,player.x,120)
    player.y=mid(0,player.y,120)

    for enemy in all(enemies) do --enemy update loop
        enemy.x-=enemy.speed --move enemy left
        for laser in all(lasers) do --check collision w.laser
            if enemy_collision(laser.x,laser.y,enemy) then
                del(enemies,enemy)
                del(lasers,laser)
                score+=100
            end
        end
        --check collision w/ player
        if enemy_collision(player.x+4,player.y+4,enemy) then
            gameover = true
        end
        --delete enemy if off screen
        if enemy.x<-8 then
            del(enemies,enemy)
        end
    end

    for laser in all(lasers) do --laser update loop
        laser.x+=3 --move laser to the right
        if laser.x>130 then --delete laser if off screen
            del(lasers,laser)
        end
    end

    if wavetimer==90 then --every 3 seconds spawn wave
        create_wave(rnd(6)+waveintensity)
        wavetimer=0 -- reset timer
        waveintensity+=1
    end
end

function _draw() --called 30 times per second
    cls() --clear screen
    if not gameover then
        spr(player.sprite,player.x,player.y) --draw player
    end

    for enemy in all(enemies) do --draw enemies
        spr(enemy.sprite,enemy.x,enemy.y)
    end

    for laser in all(lasers) do --draw lasers
        rect(laser.x,laser.y,laser.x+2,laser.y+1,8)
    end

    if gameover then --print game over to screen
        print('game over',50,64,7)
    end
    print('score: '..score,2,2,7) --show score on screen
end
--creates an enemy at x,y with random speed 1-2
function create_enemy(x,y)
    enemy={x=x,y=y,speed=rnd(1)+1,sprite=2}
    add(enemies,enemy)
end
--spawns a wave of enemies off screen
function create_wave(size)
    for i=1,size do create_enemy(256,rnd(128)) end
end

function create_laser(x,y)
    laser = {x=x,y=y}
    add(lasers,laser)
end
--returns true if x,y are within a 8x8 rectangle around enemy
function enemy_collision(x,y,enemy)
    if x>=enemy.x and x<=enemy.x+8 and y>=enemy.y and y<=enemy.y+8 then
        return true
    end
    return false
end
```

Part 02

The Squeeze: LED Racer

Warning!
High voltage

The dynamo torches in this project can produce high voltage, so be careful.

Have you got what it takes to win the Formula One LED race? Run a race on your own LED track layout using your Squeeze controller

MAKER

Mike Cook

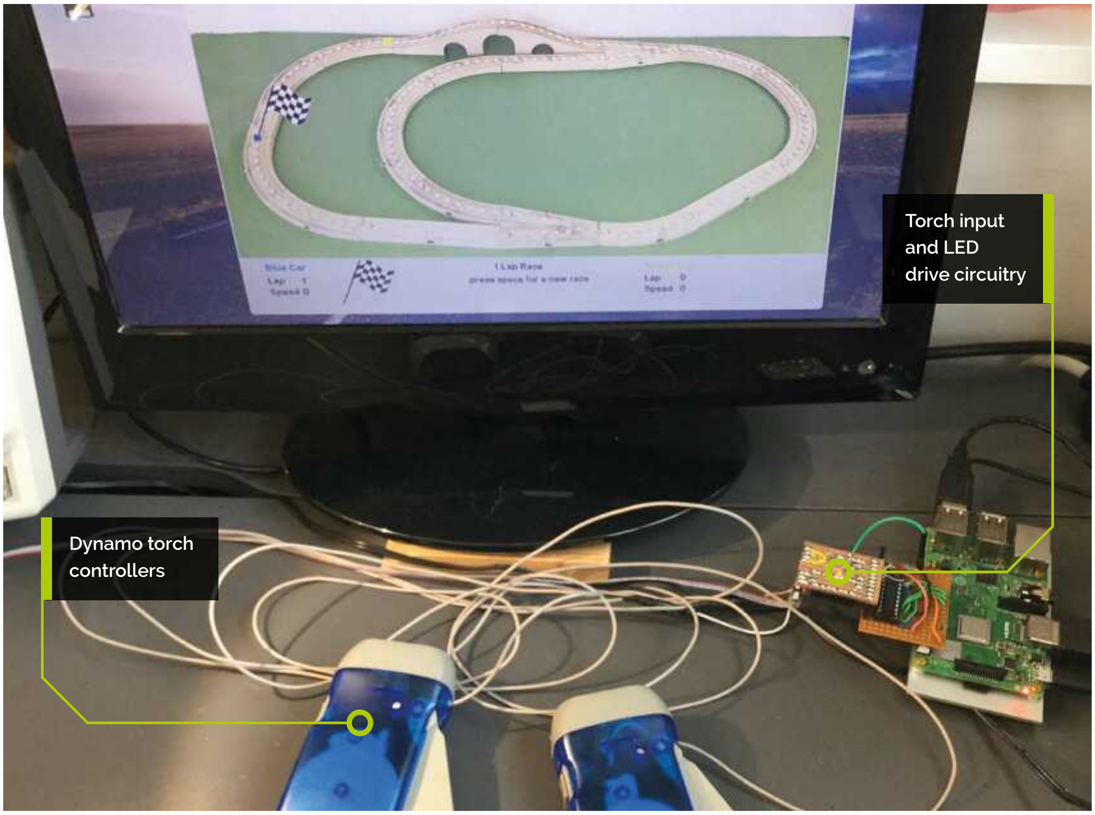
Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/TPaUfT

Last month, we saw how to convert a dynamo torch into a controller. The more you squeezed it, the bigger signal you got. Now, we are going to put this to good use, by making an LED racer game. Back in *The MagPi* #41, we used a simple IKEA train to help teach reading, 'The Story Train'. So we've used the wooden track from this as the basis of our game. You can still get these, although the bridge has changed a bit and more extra track sets are available. Other wooden train tracks could be used; just search online.

01 The layout

The track layout we chose to use has a bridge and two sets of points and is shown in **Figure 1**. This shows each individual piece of track, the sort of track section it is, along with the LED numbers we use. One lap of the track starts at piece S1 and goes round to the Points 1 piece and loops back to the Points 2 piece. The second time the Points 1 piece is reached, the other branch is taken and the track continues round to the start again. So we get a total track length of 231 steps from only 184 LEDs.



Torch input and LED drive circuitry

Dynamo torch controllers

You'll Need

- IKEA Lillabo train set with track extension pack **magpi.cc/FpUZix**
- A/D converter from *The MagPi* #68 **magpi.cc/68**
- Squeeze controller from *The MagPi* #83 **magpi.cc/83**
- 184 × WS2812 LEDs on PCB **magpi.cc/vsYpFc**

Figure 2

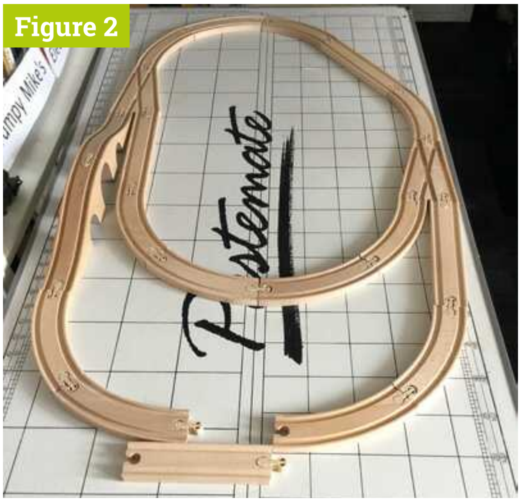


Figure 2 The need for a small piece of track

02 Making the layout

The desired track could not be made with the pieces we had, because we needed an extra short piece. This piece is in the newer sets, but not the one we had, so we had to make one – see **Figure 2**. We sawed a longer piece to the required length, then placed it over the hole end of another straight piece and marked the hole with a pencil. Then, we drilled a 10 mm hole where it was marked, and removed the end material with a saw. A combination of round and straight files made it fit the plastic stud from another piece – see **Figure 3**.

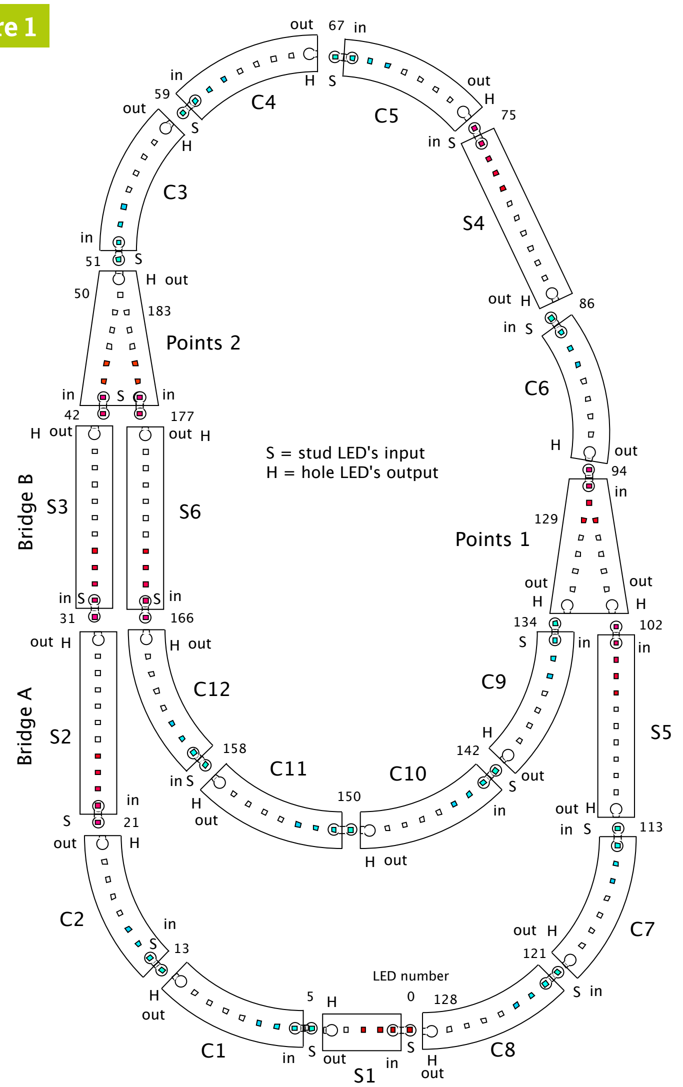
03 Components

We used individual WS2812B LEDs mounted on a PCB, and spaced them out such that the gap between the two plastic studs – used for joining two sections of track – was maintained throughout the rest of the track in LEDs. As we wanted to be able to take this project apart for storage, and transportation, we made each piece of track a standalone LED piece. This had a four-pin connector on the outside of each track piece roughly in the middle. This meant that we could make a wiring loom to connect it all up and store that separately.

04 Templates

In order to make a nice curve, we made the orientation of the individual LEDs change throughout the curve. To do this we drew a full-scale picture of a curve piece rotating each LED 11.25° so the 90° curve was completed with eight LEDs. We also drew the points and straight sections before printing these out on paper. Next,

Figure 1



12 Clockwise Curves – connector long arc – 96 LEDs
5 Straight long pieces – including bridge section – 55 LEDs
1 Straight short piece – custom shortened – 5 LEDs
2 Points – 15 + 13 LEDs
Total number LEDs = 184 LEDs

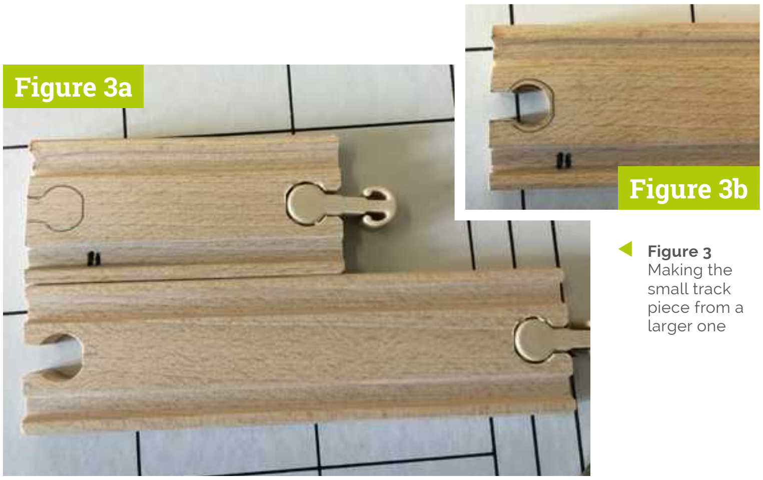
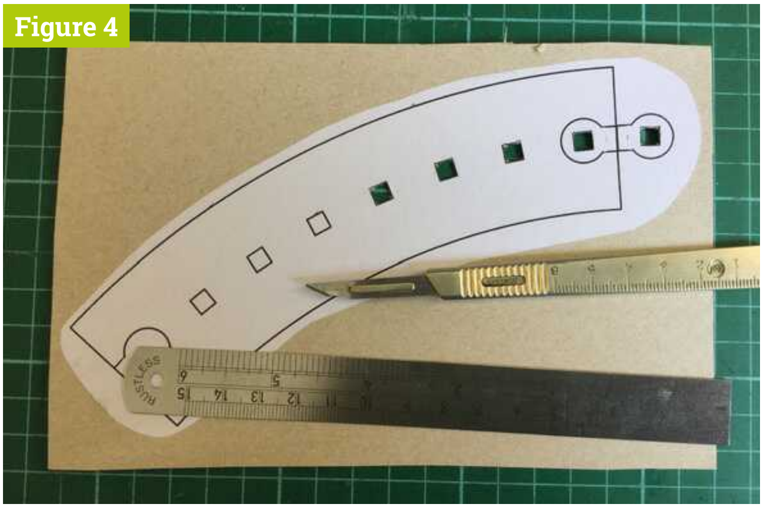


Figure 3 Making the small track piece from a larger one

Figure 4



▲ **Figure 4** Cutting out the LED holes in the template

▶ **Figure 5** Adding an extra layer of cardboard to the template

Top Tip

Wiring the LEDs

It helps if the wire making the power rails are curved to begin with: stretch it with your hands to approximate the final shape first.

▼ **Figure 7** Track section connector – use glue if it is not a tight fit

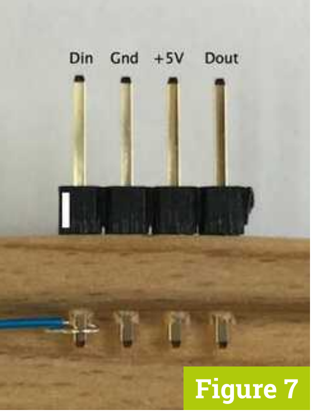
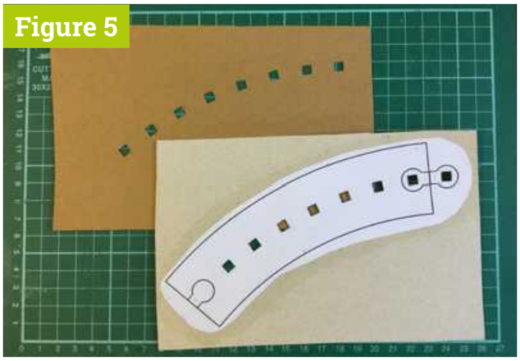


Figure 7

Figure 5



we glued each section to a piece of cardboard from a cereal box and cut out the LEDs' squares with a scalpel. **Figure 4** shows the LED squares being cut out of the template.

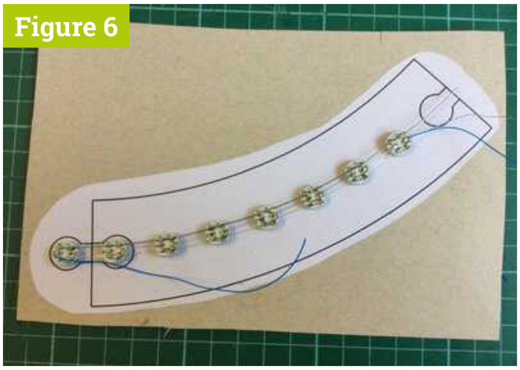
05 Finishing off the template

Then we used another piece of cardboard to draw through the square holes and cut then out. We made sure the holes were aligned by pushing an LED at each end and then the two pieces of cardboard were glued together – **Figure 5**. When the glue was set, we gave each template three coats of clear varnish to toughen them up and, when dry, placed them under heavy books for a few days to remove any curl. We also made the straight track and the points template. The template PDF files can be found on our GitHub page.

06 Constructing the LED strips

For strength, we used 22 SWG wire to make the power and ground connections. Start at one end and solder the wire to one power pad of the LED. Then, use pliers to bend the wire just after this joint so that it lines up with the second LED's pad, and solder that. Repeat this and when a run is finished, go back and solder the other power pad. Wire the data-in to the data-out of the central data chain with 28 SWG wire (we used stripped-back Kynar wire). **Figure 6** shows the wiring of a finished curve.

Figure 6



▲ **Figure 6** Wiring up a track section; note the extension wires on the data-in and data-out of the strip

07 The track connector

First off, we need to drill four holes, about 1 mm down from the top edge of the track in the approximate centre. We pushed the header pins into the wood so they made tiny indentations, then marked these with a sharp pencil, and drilled 1 mm holes. Then we could push the long header into the wood and have just enough sticking out to connect some Kynar wire to the strip assembly. **Figure 7** shows how we wire the connector for all the tracks, with a white mark (from typing correction fluid) on the left side to mark the data-in signal.

08 Fixing the strip to the track

We found the quickest way to do this was with hot-melt glue, although if you have the time it might be simpler to use clear silicon adhesive. We put a generous helping of glue on the two plastic pieces at the end of the track, and quickly pressed the first two LEDs into it. When the glue had set, we lifted up a middle LED, applied a little glue under it, and pushed that down, then we repeated this for the final LED in the strip. You can see a full track section in **Figure 8** (overleaf).

09 LED driver

As there are so few LEDs on at any one time, and they are not shining at full blast, it is safe to power them off the 5 V pins of the GPIO connector.

led_racer.py

DOWNLOAD THE FULL CODE:

magpi.cc/dhaAam

Language: **Python**

```python
#!/usr/bin/env python3
#LED Racer - by Mike Cook June 2019
# Needs to run in supervisor mode
# Can take over 5 seconds to initialise

import os, pygame, sys, csv, time, spidev
# comment out next line if no LEDs attached
import board, neopixel

pygame.init()
pygame.mixer.quit()
pygame.mixer.init(frequency=22050, size=-16, channels=2, buffer=512)
os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
pygame.display.set_caption("LED Racer")
pygame.event.set_allowed(None)
pygame.event.set_allowed([pygame.KEYDOWN, pygame.QUIT])
screen = pygame.display.set_mode([920,498],0,32)
textHeight=22 ; font = pygame.font.Font(None, textHeight)
done = False ; trackLEDs = False ; screenUpdate = False
lastPos = [0,0]
position = [0.0,0.0] # progress of player
lap = [0,0] ; raceLaps = 1 # number of laps in a race
nAv = 10 # number of samples to average
avPoint = [0.0,0.0] ; average = [0.0,0.0]  ; speed = [0.0,0.0]
p1 = [0] * nAv  ; p2 = [0] * nAv ; runningAv = [p1,p2]
backCol = (180,180,180) ; black = (0,0,0)
scale = [5000, 4100] # adjust if two controllers are not the same

def main():
    global position,lap,screenUpdate, done, winner, speed
    print("LED Racer - use number keys to set the number of laps")
    init()
    initLEDs() # comment out if no LEDs attached
    while 1:
        winner = -1 ; speed[0] = 0 ; speed[1] = 0
        position = [0.0,0.0] ; lap[0] = 0 ; lap[1] = 0
        drawScreen(0,0) ; time.sleep(2.0)
        whistle.play()
        while not done:
            time.sleep(0.005)
            checkForEvent()
            readVoltage()
            updatePosition()
            if screenUpdate :
                drawScreen(path[int(position[0])], path[int(position[1])])
                updateLEDs(path[int(position[0])], path[int(position[1])])
                screenUpdate = False
        drawWords("press space for a new race",350,430,black,backCol)
        pygame.display.update()
        end.play()
        while done: # wait for space bar or quit
            checkForEvent()

def updatePosition():
    global position, screenUpdate, lap, lastPos, done, winner, speed
    for i in range(0,2):
        position[i] += speed[i] + speedChange[int(position[i])]
        if lastPos[i] != int(position[i]):
            screenUpdate = True
            lastPos[i] = int(position[i])
        if position[i] >=len(path):
            position[i] = 0
            lap[i] += 1
            if lap[i] >= raceLaps:
                winner = i
                done = True
                speed[0] = 0 ; speed[1] = 0

def updateLEDs(pos0,pos1):
    global pixels
    if(trackLEDs):
        pixels.fill((0, 0, 0))
        if pos0 == pos1:
            pixels[pos1] = (255,0,255)
        else:
            pixels[pos0] = (0,100,200)
            pixels[pos1] = (200,200,0)
        pixels.show()

def drawScreen(pos0,pos1):
    screen.blit(track, (0,0) )
    pygame.draw.rect(screen,backCol,(0,399,920,99),0)
    drawWords("Blue Car",35,407,(0,100,200),backCol)
    drawWords("Yellow Car",630,407,(140,140,0), backCol)
    drawWords("Lap",35,430,black,backCol)
    drawWords("Lap",630,430,black,backCol)
    drawWords("Speed",35,450,black,backCol)
    drawWords("Speed",630,450,black,backCol)
    drawWords(str(raceLaps)+ "Lap Race",390,407,black,backCol)
    drawWords(str(lap[0]), 87,430,black,backCol)
    drawWords(str(lap[1]), 688,430,black,backCol)
    drawWords(str(int(speed[0]*100)), 87,450,black,backCol)
    drawWords(str(int(speed[1]*100)), 688,450,black,backCol)
    if pos0 == pos1:
        pygame.draw.rect(screen,(200,0,200),( led_pointX[pos0],led_pointY[pos0],10,10),0)
    else:
```

Use Lego figures, farm animals, flags or buildings to enhance the look of the layout

Therefore, all we need to drive them is a normal 74HTC14 buffer to step up the 3.3 V data signal from Raspberry Pi to the 5 V required by the LEDs. We have used this circuit many times before when we have used addressable LEDs. We managed to squeeze this onto last month's input signal board, and the layout of this is shown in **Figure 9**.

10 The base board

We took a 1070 mm by 600 mm piece of 18 mm thick P5 floor boarding and painted it green – two coats of primer, two of undercoat, and two of top coat – then we laid out the track on it. At each side of every other piece, we drilled a 3 mm hole through the track and board and inserted a 20 mm M3 screw to fix the track in place – we didn't fasten these down, just the screw pushed through both holes alone held it in place. We did this one piece at a time, keeping the rest of the track in place at all times.

Top Tip

LED fit

If the LEDs don't fit firmly in the template and pop out, secure them in place with Blu Tack.

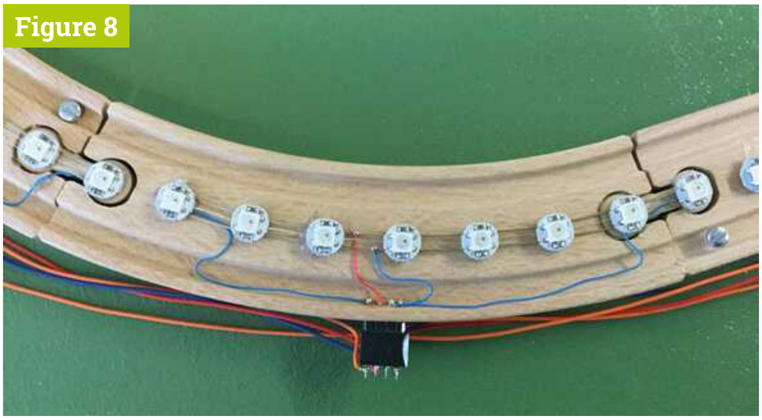


Figure 8

Figure 8 A completed track section

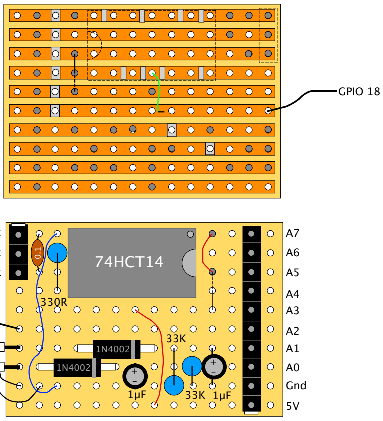


Figure 9 Layout of the LED driver and signal conditioner board

11 The wiring loom

The connectors on each track piece now need to be wired. The 5 V and ground go to each section of track; the data flow was shown in **Figure 1**. LEDs 0 to 128 form the outer ring, and LEDs 129 to 183 form the extension loop. The software controls the order these are lit, so that it looks like a continuous track. Most of the track pieces have their data-out connected to the data-in of the next piece – except for the loop, where you connect the data-out of track C8 and connect it to the data-in of the Points 1 section.

12 The software

The software to drive this is shown in the **led_racer.py** listing. The `path` list translates the position round the track into what LED is lit. Sloping sections of the track, like the bridge, speed up or slow down the cars, so you have to squeeze more rapidly to crawl up the hill. With the number keys, you can set how many laps a race should last. The red and blue cars are represented by their colours, but when they are on the same spot it is a purple colour. The screen display mirrors the LED track, so you could play it without the physical track if you wanted.

In conclusion

We have another version of this program on the GitHub page that imposes speed limits around some sections of the curved track. You spin off and get a time penalty for going too fast. Also see this page for notes on generating your own layout.

led_racer.py (continued)

Language: **Python**

```python
        pygame.draw.rect(screen,(0,100,200),(led_pointX[pos0],led_pointY[pos0],10,10),0)
        pygame.draw.rect(screen,(200,200,0),(led_pointX[pos1],led_pointY[pos1],10,10),0)
    if winner !=-1:
        screen.blit(flag, (64,122) )
        if winner == 0:
            screen.blit(flag, (145,402) )
        else:
            screen.blit(flag, (761,402) )
    pygame.display.update()

def drawWords(words,x,y,col,backCol) :
    textSurface = font.render(words, True, col, backCol)
    textRect = textSurface.get_rect()
    textRect.left = x # right for align right
    textRect.top = y
    screen.blit(textSurface, textRect)
    return textRect

def readVoltage():
    global average, avPoint, speed, runningAv
    for i in range(0,2):
        adc = spi.xfer2([1,(8+i)<<4,0]) # request channel
        reading = (adc[1] & 3)<<8 | adc[2] # join two bytes together
        runningAv[i][int(avPoint[i])] = reading
        avPoint[i]+=1
        if avPoint[i] >= nAv:
            avPoint[i] = 0
        average[i] =  0
        for j in range(0,nAv): # calculate new running average
            average[i] += runningAv[i][j]
        average[i] = average[i] / nAv
        speed[i] = average[i] / scale[i]

def init():
    global track, flag, led_pointX, led_pointY, path, spi, speedChange, whistle, end
    whistle = pygame.mixer.Sound("sounds/whistle.ogg")
    end = pygame.mixer.Sound("sounds/end.ogg")
    track = pygame.image.load("images/Layout.jpg")
    flag = pygame.image.load("images/flag.png").convert_alpha()
    path = [] # the sequence of LED numbers to make a lap
    for i in range(0,97):
        path.append(i)
    for i in range(129,184):
        path.append(i)
    for i in range(50,129):
        path.append(i)
    #print('number of steps in a lap',len(path))
    led_pointX = []
    led_pointY = []
    x_Off = -5
    y_Off = -5
    with open('Points.csv',newline='') as File:
        reader = csv.reader(File)
        for row in reader:
            led_pointX.append(int(row[0])+x_Off)
            led_pointY.append(int(row[1])+y_Off)
        speedChange = [0.0] * len(path) # speed increment start off with zero
    # for bridge
    for i in range(22,32):
        speedChange[i] = -0.12 # retarding speed on up
    for i in range(33,43):
        speedChange[i] = 0.1 # increasing speed on down
    spi = spidev.SpiDev()
    spi.open(0,0)
    spi.max_speed_hz=1000000

def initLEDs():
    global pixels, trackLEDs
    trackLEDs = True
    pixel_pin = board.D18
    num_pixels = 184
    ORDER = neopixel.GRB
    pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=0.15, auto_write=False, pixel_order=ORDER)
    updateLEDs(0,0)

def terminate(): # close down the program
    pygame.mixer.quit()
    pygame.quit() # close pygame
    if trackLEDs :
        pixels.fill((0, 0, 0)) # turn off LEDs
        pixels.show()
    os._exit(1)

def checkForEvent(): # see if we need to quit
    global reading, screenUpdate, average, done, raceLaps
    event = pygame.event.poll()
    if event.type == pygame.QUIT :
        terminate()
    if event.type == pygame.KEYDOWN :
        if event.key == pygame.K_ESCAPE :
            terminate()
        if event.key == pygame.K_SPACE :
            done = False
        if event.key == pygame.K_RETURN :
            done = True # reset a race
            screenUpdate = True
        if event.key > pygame.K_0 and event.key <= pygame.K_9 :
            raceLaps = event.key - pygame.K_0
            screenUpdate = True

if __name__ == '__main__':
    main()
```

Part 04

GUI user input with C and GTK

Enable users to enter text, and to select options using check and radio buttons

MAKER

Simon Long

Simon Long is a software engineer working for Raspberry Pi, responsible for the Raspberry Pi Desktop on both Raspbian and Debian.

rpf.io

We've seen how we can display text on a window using a label – in last issue's tutorial (magpi.cc/83) – but what about if we want to read in some text that the user has typed? GTK provides the GtkTextEntry widget for this.

Text entry

Modify the example code from last issue to add a new cell to the table with a GtkTextEntry widget, like so:

```
#include <gtk/gtk.h>

GtkWidget *txt;

void end_program (GtkWidget *wid, gpointer ptr)
{
  gtk_main_quit ();
}

void copy_text (GtkWidget *wid, gpointer ptr)
{
  const char *input =
gtk_entry_get_text (GTK_ENTRY (txt));
  gtk_label_set_text (GTK_LABEL (ptr), input);
}

void main (int argc, char *argv[])
{
  gtk_init (&argc, &argv);

  GtkWidget *win =
gtk_window_new (GTK_WINDOW_TOPLEVEL);
  GtkWidget *btn =
gtk_button_new_with_label ("Close window");
  g_signal_connect (btn, "clicked",
G_CALLBACK (end_program), NULL);
  g_signal_connect (win, "delete_event",
G_CALLBACK (end_program), NULL);
  GtkWidget *lbl = gtk_label_new ("My label");

  GtkWidget *btn2 =
gtk_button_new_with_label ("Copy button");
  g_signal_connect (btn2, "clicked",
G_CALLBACK (copy_text), lbl);

  txt = gtk_entry_new ();

  GtkWidget *tab = gtk_table_new (2, 2, TRUE);
  gtk_table_attach_defaults (GTK_TABLE (tab),
lbl, 0, 1, 0, 1);
  gtk_table_attach_defaults (GTK_TABLE (tab),
btn2, 1, 2, 0, 1);
  gtk_table_attach_defaults (GTK_TABLE (tab),
btn, 0, 1, 1, 2);
  gtk_table_attach_defaults (GTK_TABLE (tab),
txt, 1, 2, 1, 2);
  gtk_container_add (GTK_CONTAINER (win), tab);
  gtk_widget_show_all (win);
  gtk_main ();
}
```

An Introduction to C & GUI Programming

For further tutorials on how to start coding in C and creating GUIs with GTK, take a look at our new book, *An Introduction to C & GUI Programming*. Its 156 pages are packed with all the information you need to get started – no previous experience of C or GTK is required!
magpi.cc/GUIbook

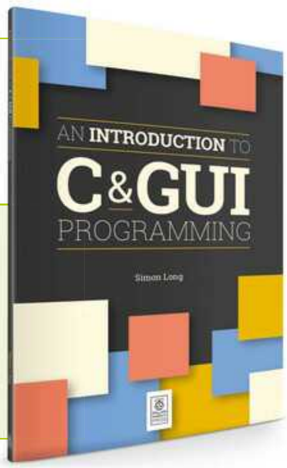
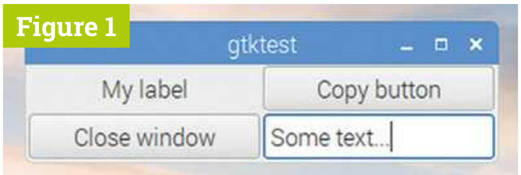


Figure 1



▲ **Figure 1** A GtkEntry added to the bottom-right of the table

In this example, we declare a pointer to the GtkTextEntry widget, `txt`, as a global variable. This is because we need to access it from a button handler, which also needs to access the GtkLabel widget. We previously used the general-purpose pointer in the `g_signal_connect` function to pass widget information to a handler, but in this case, the handler needs access to two separate widgets, and we've only got one pointer. (There are ways of using a single pointer for this, but using a global variable is a lot easier to understand.)

We've added a new function, `copy_text` – this uses the function `gtk_entry_get_text` to get a pointer to the text buffer used by the GtkEntry widget; this stores whatever the user has typed into the entry box. Note that the pointer to this buffer of chars is declared with the modifier `const` – this indicates a variable that cannot be changed by the programmer. We then pass this buffer to `gtk_label_set_text`, which copies the contents of the GtkEntry to the GtkLabel.

Finally, we've created the GtkEntry itself with `gtk_entry_new`, and added it to the table. (Note we've also moved the attachment points of btn so that it only takes up one cell of the table rather than two, so as to give enough space for the GtkEntry.)

Build the code and run it; try typing something into the entry box and pressing the 'Copy button' to see what happens (**Figure 1**).

Spin buttons

There are various other ways for the user to enter data. Quite often they need to select one of a number of options, rather than being given a box and asked to type into it. GTK provides a number of widgets for this purpose; one, which is closely related to the GtkEntry, is the GtkSpinButton.

A spin button is used to enter a numeric value in a certain range. You specify the minimum and maximum values, and by how much the value should change on each press of the button. This gives a text entry control – which the user can still type into – with an easy way of changing the value. If the user does type in a value which is outside the limits you set, it is automatically rounded to the closest value within the limits when you try to read it.

We only need to change one line to convert the GtkEntry to a GtkSpinButton in our example above. Replace:

```
    txt = gtk_entry_new ();
```

...with:

```
    GtkObject *adj = gtk_adjustment_new (0, -10,
 10, 1, 0, 0);
    txt = gtk_spin_button_new (GTK_ADJUSTMENT
 (adj), 0, 0);
```

This changes `txt` from a GtkEntry to a GtkSpinButton. We can see that one of the arguments to the `gtk_spin_button_new` function is something called a GtkAdjustment – what's that?

A GtkAdjustment is used to set the range of values that a spin button (and a few other widgets with similar behaviour) can take. So the `gtk_adjustment_new` function takes arguments which specify, in order, the default value (0 in this case), the minimum value (-10), the maximum value (10), and the step size (1). (The last two arguments specify 'page' increments, which are used in some controls where there are two different sets of buttons to change the values – a spin button doesn't use these, so they can just be set to zero.)

The GtkAdjustment is passed as an argument to the `gtk_spin_button_new` function. The other two arguments are the climb rate – how fast the value changes when the button is held down – and the number of decimal places to be shown on the value.

Build and run the new version of the code; the text entry box now has two small buttons at the right-hand side (**Figure 2**); these can be used to change the value. You can still type into the box; if you type a value that isn't a number, or is outside the range specified in the GtkAdjustment, it will be corrected to a value within the range when one of these buttons – or 'Copy button' – is pressed.

▼ **Figure 2** A GtkSpinButton instead of the GtkEntry

Figure 2

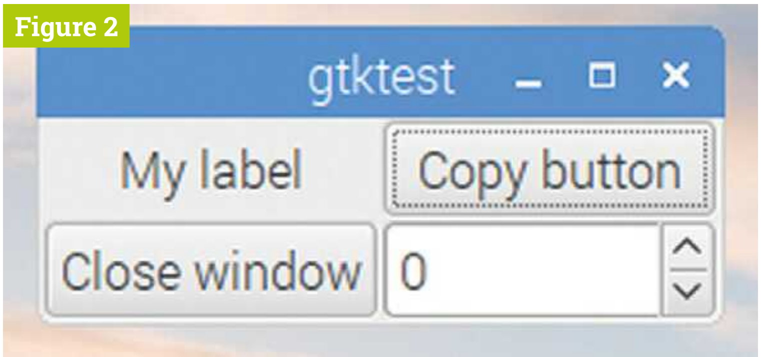
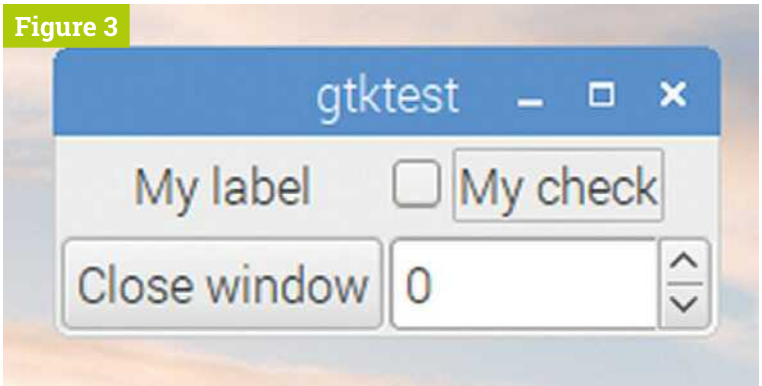


Figure 3



▲ **Figure 3** A GtkCheckButton at the top-right of the window

Toggle buttons

One useful way to get input from the user is by the use of buttons to select an option. GTK offers several widgets for this purpose; two of the more useful are check buttons – boxes that can be ticked or unticked – and radio buttons – selectors in which only one member of a group can be active at a time. These are both children of the GtkToggleButton widget, which is itself a child of the GtkButton widget we have already used several times; a toggle button is a button which can be in either a selected or an unselected condition.

Check buttons

Check buttons are the simpler of the two, so we'll look at those first. A GtkCheckButton is a box which can either be empty or have a tick mark in it; clicking on it toggles it between those two states. Creating one is as easy as:

```
    GtkWidget *chk =
gtk_check_button_new_with_label ("My check");
```

This creates a check button and a label to its left with the supplied text in it. (There is also a function `gtk_check_button_new`, which creates just the check box without the label – this makes things tidier if you have a label for it somewhere else in your window.) Try replacing `btn2` in the previous example with a check button like `chk` – it should look like **Figure 3**.

By default, a check button which has just been created is in the unchecked state. To check the button, you call:

```
    gtk_toggle_button_set_active
(GTK_TOGGLE_BUTTON (chk), TRUE);
```

Setting the button's active state to `TRUE` checks the box; setting it to `FALSE` unchecks it.

There's a corresponding function to read the active state of a toggle button. To read whether a check button is currently checked into a variable, do:

```
    int state = gtk_toggle_button_get_active
(GTK_TOGGLE_BUTTON (chk));
```

The value of `state` will be 1 if the button is checked, or 0 otherwise.

You can also connect a handler to the `toggled` signal of a check button with `g_signal_connect` – this is equivalent to connecting to the `clicked` signal of a regular GtkButton, as we have seen in previous examples:

```
    g_signal_connect (chk, "toggled",
G_CALLBACK (check_toggle), NULL);
```

The handler for a `toggled` call looks like this:

```
  void check_toggle (GtkWidget *wid,
gpointer ptr)
  {
    printf ("The state of the button is %d\n",
gtk_toggle_button_get_active (GTK_TOGGLE_BUTTON
(wid)));
  }
```

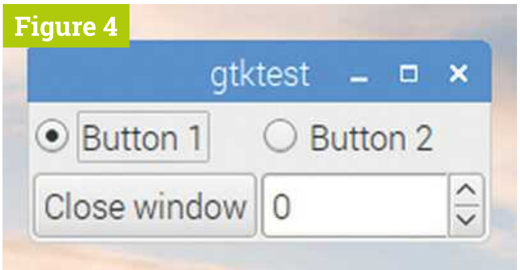
Note that the first argument of any handler callback is always a pointer to the widget that generated the signal, so we check the state of the check button by calling `gtk_toggle_button_get_active` on the pointer received as the first argument to its handler.

Radio buttons

Radio buttons can be viewed as a group of check buttons, in which one and only one of each group is selected at a time – checking any button in the group automatically unchecks all other buttons in the group. These are shown as circular indicators: The code controlling radio buttons is almost identical to that for check buttons; their active state can be set or checked with the same commands, and they generate a `toggled` signal when clicked.

The significant difference between radio button and check button code is that radio buttons need to be assigned to a group of other radio buttons; this is required so that the link between the set of buttons

Figure 4



▲ **Figure 4** Two GtkRadioButtons at the top of the window is established so that GTK knows which buttons need to be unchecked when another is checked.

Radio buttons are created using the function `gtk_radio_button_new_with_label` – this takes one additional parameter over `gtk_check_button_new_with_label`, which is the group to which the button belongs. When the first button in a group is created, this parameter is set to `NULL`:

```
    GtkWidget *rad1 =
gtk_radio_button_new_with_label (NULL, "Button 1");
```

When second and subsequent buttons are created, the group of the first button can be read with the function `gtk_radio_button_get_group` and passed into `gtk_radio_button_new_with_label` calls:

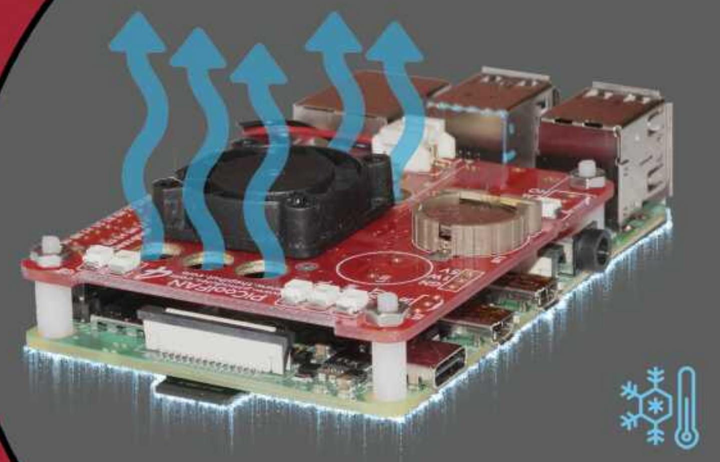
```
    GSList *group = gtk_radio_button_get_group
(GTK_RADIO_BUTTON (rad1));

    GtkWidget *rad2 =
gtk_radio_button_new_with_label (group,
"Button 2");
```

> “The significant difference between radio button and check button code is that radio buttons need to be assigned to a group of other radio buttons”

Every button in the group needs to be associated with another button in the same group in this manner in order for the link to work correctly. (Any radio button which is not linked with a group will not function correctly; it will always be selected and clicking on it has no effect.)

Try replacing the check button and the label in the previous example with the pair of grouped radio buttons shown above – you should get something that looks like **Figure 4**.

If you’ve linked them correctly, then when you click ‘Button 2’, it will be selected and ‘Button 1’ will be unselected, and vice versa.



SUBSCRIBE AND
SAVE UP TO
35%
on the cover price
MAKE BUILD HACK CREATE
25 PAGES
HackSpace
TECHNOLOGY IN YOUR HANDS
Raspberry Pi 4
MAKER'S GUIDE
DIY POLLUTION MONITORING
KITCHEN GLASS-MAKING
Add this powerful computer to your next project
MARBLES PYTHON TETRIS CLOCK

Scorching Summer PROJECTS

Take your Raspberry Pi outside for some fun making outdoors

Look, we understand. The WiFi outside isn't very good, screen glare can be terrible, and not every tree has a power socket you can use. That doesn't mean that the outdoors can't be fun… or full of tech.

So let's go outside and enjoy the sunshine, while also having a bit of fun with making some IoT projects.

Summer Essentials

- Suntan lotion
- A bottle of water
- Sunglasses
- Raspberry Pi
- Backpack of parts

Hydroponic Gardening

Automate growing some veggies

magpi.cc/62

While this is something you can do all year round, what better time to get it all set up than while it's warm and sunny? It's also very cheap, costing less than £40 to build the automatic system to keep all your plants watered.

This system that Kent Elchuk built is fairly simple: it tests soil moisture levels and it can also control pumps that deliver nutrients and water to the plant, based on the level of moisture. It even includes a handy camera so you can visually inspect your garden remotely.

Kent's setup is very impressive, with dozens of plants growing in towers in his garden. However, you can always start simple with a trough or even just a single potted plant and grow from there.

Zero GPS Logger

Track your every movement

magpi.cc/40

Our colleague Matt Richardson, from Raspberry Pi Foundation North America, created this little project when Raspberry Pi Zero first came out, using a simple GPS locator and some code. You can use it to track your walks, runs, hikes, bike rides, commutes, drives, and more, for analysis later.

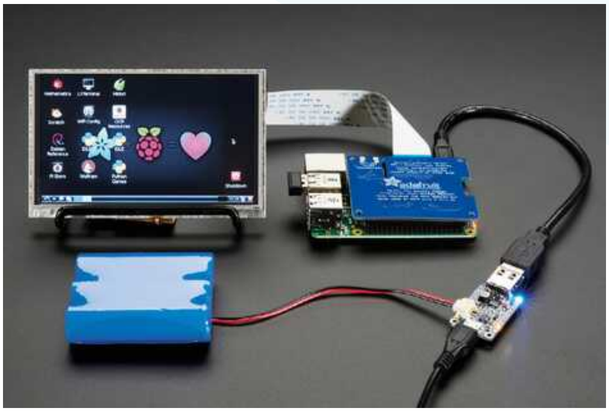
You could easily combine this with one of the boat projects, or even a HAB, to get a better idea of where you're going and where you've been – although it's probably not quite accurate enough for full automation in some cases.

The Zero GPS in its case, attached to a backpack

A Raspberry Pi Zero is connected to a GPS receiver

Powering your Raspberry Pi

Portable power for outdoor projects



How do you keep your Raspberry Pi on while outside? Well, if you're just powering a Raspberry Pi and a few small components hooked up to the GPIO pins or USB ports, a portable battery pack / phone charger will do the trick. You can turn off the graphical desktop in Raspberry Pi Configuration to reduce power consumption as well, so that the battery lasts longer. If you're feeling particularly adventurous, you can look at an Adafruit PowerBoost and get a lithium battery to power your creation.

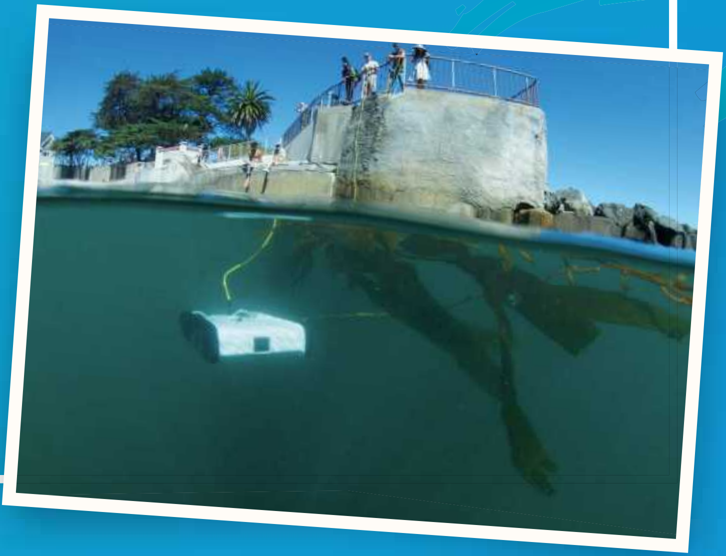
Trident ROV

Explore the world beneath the waves | sofarocean.com

We quite love the Trident. It's an underwater drone that is pretty rugged and is also powered by Raspberry Pi – we've seen pictures of it exploring sunken wrecks in big lakes, and that is pretty cool.

It is mighty expensive, though, at $1695 (£1355). However, it's built upon the OpenROV platform, which is an open-source hardware and software system for building your own underwater drones and such.

There are various other underwater drone projects you can look up as well, including one that uses PVC piping to stay watertight. It makes for a fun way to explore under the water without needing to get a scuba suit or an entire submarine.



Robotic Sailbot

Captain Raspberry Pi sails away

ubcsailbot.org

Sailing is cool, but what if the boat could pilot itself sometimes? Everyone needs a break, after all, and the University of British Columbia has been working on creating an automated sailing boat that also has a bit of solar power in it as well.

Now, this is quite a large boat, but you can scale it down to something smaller. A team at Aberystwyth University has made much smaller ones (**abersailbot.co.uk**), and we've also seen speedboat-like RC toys outfitted with a Raspberry Pi, souped up to make them a little more exciting.

> You can scale it down to something smaller

High Altitude Ballooning

Send your Raspberry Pi to the sky

daveakerman.com

We talked a bit about HAB (high-altitude ballooning) last issue, which is essentially where you strap your Raspberry Pi to a special balloon and send it over 30 km into the sky. Preferably with a parachute so that you can retrieve it afterwards as well!

Weather is a big factor in making sure it's safe to launch, so a clear day with low wind offers optimal conditions to then launch your balloon, before chasing it down once it inevitably goes pop. Building it is fun as well – you can do it all from a Raspberry Pi, including GPS trackers to chart its progress, a Raspberry Pi Camera Module to take photos and videos of the Earth from up high, along with any other sensors you fancy attaching to it.

Naturebytes Nature Trap

What's living in your garden? | naturebytes.org

Being outdoors means getting to see nature. However, nature isn't always accommodating to humans, so you might not see that special bird or fox that you know visits your garden. With a nature trap, you can improve your chances of seeing elusive creatures, or just document how many animals visit your garden.

The Naturebytes kit is great, as it comes with the essentials: a motion sensor, camera, weatherproof case, and a real-time clock for extra precision.

You could of course make something like this yourself – you don't even need a waterproof case if you have it pointing out of a window! You can even attach an infrared Raspberry Pi NoIR Camera Module to capture night-time beasties.

Outdoor Displays

Easy-to-read monochrome screens

Traditional screens aren't always the easiest to see in daylight, even on modern smartphones. If you're planning to create something that might need a display of some kind outdoors, there are two great alternatives: simple black and white LED displays such as the GFX HAT (**magpi.cc/nZseKc**), or an e-ink display like Inky wHAT (**magpi.cc/LZWpAq**).

> “Traditional screens aren't always the easiest to see in daylight”

Raspberry Pi Camera

Point and shoot your summer findings

Phone cameras are great, but have you ever built your own camera by hand? There are tons of projects around that let you build a camera with a Raspberry Pi – from stuff like PolaPi-Zero (**magpi.cc/h0WD0R**) which includes a thermal printer for instant photos, to awesome infrared photography (**magpi.cc/LUjIKX**) that creates surreal and picturesque images.

Remember, Raspberry Pi Camera Modules also have video and slow-motion modes, and you can even program one to take a series of photos and turn them into an animated GIF!

> “ Have you ever built your own camera by hand? ”

Raspimower Dalek

This robot exterminates grass | magpi.cc/41

Robot vacuum cleaners are cool, but how about a robot lawnmower? Maker Ron Ostafichuk decided to automate trimming his grass, and created a robot out of spare parts that would do it – controlled by a Raspberry Pi!

The next logical step was to turn it into a Dalek for Halloween. You don’t have to go that far, but if you’re looking for a way to do your garden chores while enjoying some outdoor fun, an automatic mower might be a great idea.

Pibow Coupé 4 and Fan SHIM

PIBOW COUPÉ 4 ▶ Pimoroni ▶ **magpi.cc/enLWLt** ▶ £9/$9 | **FAN SHIM** ▶ Pimoroni ▶ **magpi.cc/qZYBWd** ▶ £10/$10

Raspberry Pi 4 layout means new cases, including a new version of the famous Pibow. **Rob Zwetsloot** checks it out, along with a cooling Fan SHIM

SPECS

COLOURS: Rainbow, Red, Ninja

WEIGHT: 50 g

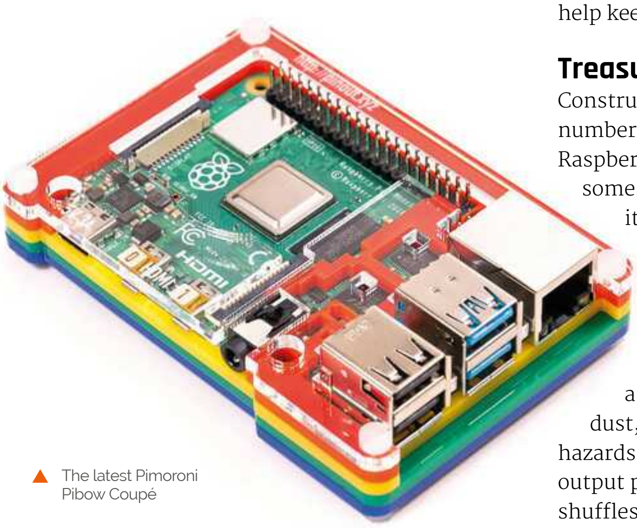
DIMENSIONS: 99 × 66 × 15 mm

One glance at Raspberry Pi 4 and you'll notice that it looks quite a bit different from your standard Raspberry Pi Model B. Since the original Raspberry Pi B+ came out, all Raspberry Pi boards have had a standard port layout, but with new tech comes new features, which is why Raspberry Pi 4 has some extra ports and a shuffling around of the USB and Ethernet.

You can read why the board layout has changed in issue 83 (**magpi.cc/83**); however, it does mean you'll either need to hack apart an old Raspberry Pi case or get a new one. Which is where Pimoroni comes in, as it so often does, with the new Pibow Coupé for Raspberry Pi 4.

Currently the Coupé is the only case style that Pimoroni is making for Raspberry Pi 4, lacking the top layers of a full Pibow. However, it still partially covers and protects a Raspberry Pi 4 while giving full access to the GPIO and ports. It also comes with a bonus feature: the ability to add a special fan to help keep your Raspberry Pi 4 nice and cool.

▲ The Fan SHIM is very small but pretty powerful

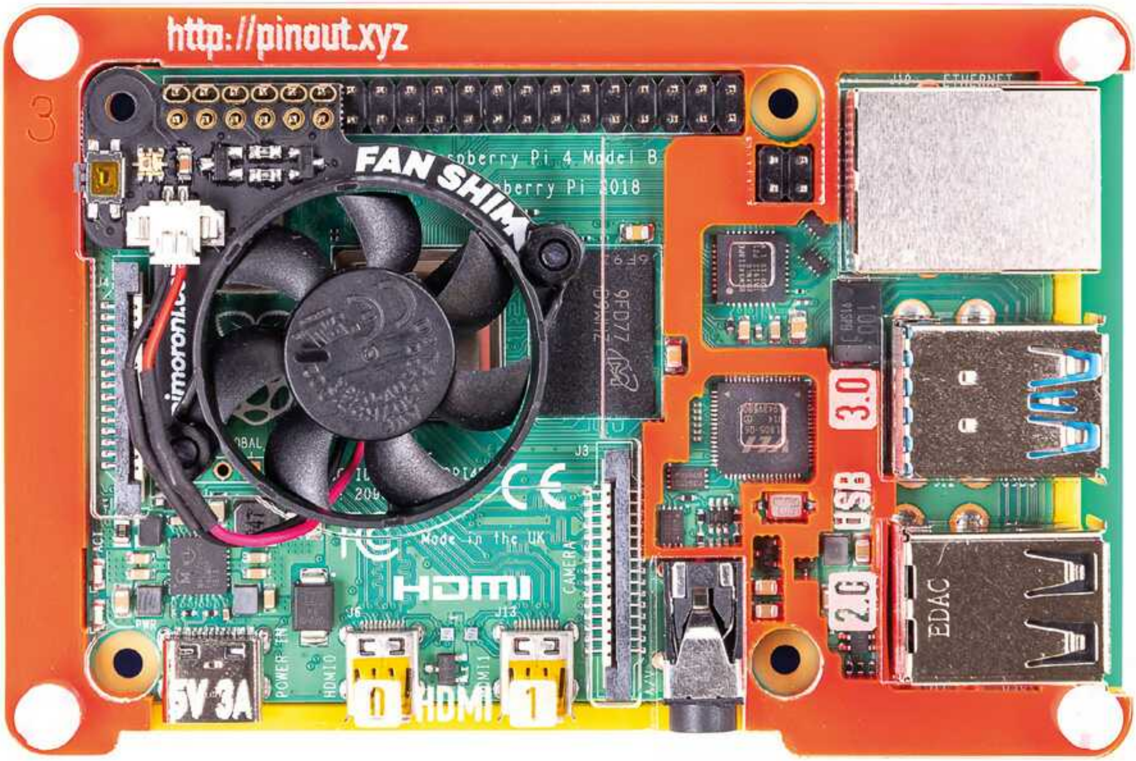


▲ The latest Pimoroni Pibow Coupé

Treasure box

Construction of the case is simple. There are five numbered slices of plastic that slot on and around Raspberry Pi 4, which are then tightened with some plastic nuts and bolts. We were able to put it together in a couple of minutes; if you're having trouble, however, there's a handy online build video from Pimoroni.

If you've ever had a Pibow case, you know the score – it's sturdy, light, and looks pretty nice. Importantly, it keeps a Raspberry Pi protected pretty well from dust, grubby fingers, and other environmental hazards. The top plate labels what each input/output port is, which is especially handy with the shuffles and additions for this hardware release.



▲ The case design allows for easily adding the Fan SHIM

> Specific care has been taken to keep the important chips open to the air

Specific care has been taken to keep the important chips open to the air, allowing for better ventilation of the board. While you don't need any special heat sinks or fans for Raspberry Pi 4, it doesn't hurt to add them, and the Pibow Coupé is specifically designed to fit the new Fan SHIM.

Biggest fan

Again, assembly is simple. Screw the fan onto the PCB, plug it in, then you can slip it over the GPIO pins while your Raspberry Pi is off. Quick and easy. However, you will need to install some software to control it, which is where it gets really fun.

Very basically, the software lets you turn the fan on and off. However, with different scripts, you can have it activate at specific temperatures and even change what the LED and button do. It's very hackable and, according to tests, very effective as well. It also doesn't sound like a hoover, which is always a bonus with a case fan.

Raspberry Pi computers have been around for over seven years now, and Pimoroni making a nice case for one is hardly a shocker – however, it's nice to see that the first round of Raspberry Pi 4 cases are great.

▲ The Fan SHIM is very small but pretty powerful

Verdict

A great first case for Raspberry Pi 4, keeping it very accessible for tinkering and hacking. The optional Fan SHIM is ideal for cooling it during periods of high CPU load.

9/10

Keybow MINI

▶ Pimoroni ▶ **magpi.cc/zywRLL** ▶ £20 / $21

A mini mechanical keypad with light-up keys. By **Phil King**

SPECS

KEY SWITCHES: 3 × Kailh Speed (Gold – clicky, or Silver – linear)

LIGHTING: 3 × APA102 RGB LEDs

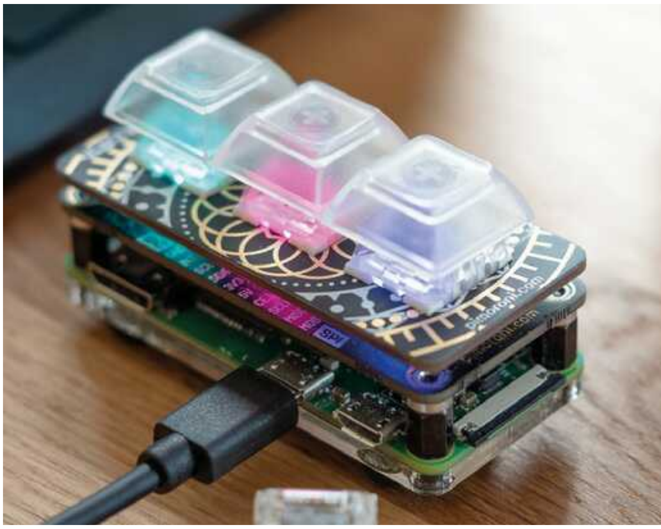
BREAKOUT HEADER: I²C, SPI

DIMENSIONS: 68.5 × 31.75 × 39 mm

Honey, I shrunk the keypad. The Keybow MINI is a three-key version of the original 12-key Keybow reviewed in issue 79 (**magpi.cc/79**). This time you need to supply your own Raspberry Pi Zero (with a GPIO header), but otherwise the principle is the same: connect the keypad to another computer – with the supplied micro USB to USB-A cable – and you can customise the key mapping to use it for a wide range of purposes, such as a games controller, hotkey pad for applications, or to insert text/code snippets with a single key press.

Solderless assembly is very similar to the original Keybow and only takes 10–15 minutes. After attaching your Raspberry Pi Zero to the PCB and acrylic base, you simply push-fit three switches into a key plate (making sure they're all orientated the same way) and add the translucent key caps before slotting into the PCB.

As before, you'll need to supply your own microSD card for the tiny (26.6MB) Keybow OS. Upon connecting the Keybow to a computer via the USB OTG cable, the OS is loaded into a RAM disk and it boots up in just 10–15 seconds.



▲ The keys can be lit up in static or animated patterns

Unlike with the full-size Keybow, we had no problem inserting the cable into Raspberry Pi Zero's USB OTG port as the latter is now situated on the edge of the device rather than tucked away inside.

▲ You have the choice of clicky or linear (quiet) keys

Light it up

Once connected, the keys light up in an animated rainbow pattern (determined by a PNG file). To change this, and customise the key mapping, you need to slot the microSD card back into your computer and edit a Lua file to select a sample key layout or your own custom one, as explained in the online tutorial: **magpi.cc/WdAWZc**. As noted there, you do need to add a few lines to the default Keybow layout code, and alter some function names, to make it work on the MINI. Alternatively, there's now a Keybow Python library that can be used within Raspbian to light and map the keys.

Either way, the real power of the Keybow MINI lies in the ability to create advanced macros to trigger a whole series of key combinations and presses with just a single key press.

Verdict

Considerably less expensive than the full-size Keybow, with the same functionality, just fewer keys. It looks cool when lit up, features a nice key action, and has multiple possible uses.

8/10

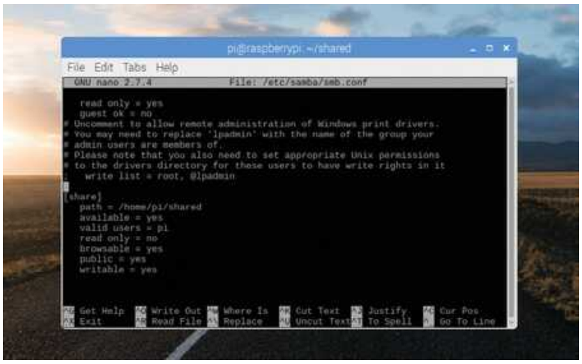
Wireframe
LIFTING THE LID ON VIDEO GAMES
£3
HARD MODE
Serbia's growing games industry
ART OF FIGHTING
The devs making accessible brawlers
BASIC INSTINCT
The creator of Assassin's Creed evolves the survival genre
SUPE
FO
Prison Architect
Introversion's dramatic fall and triumphant rise
Groan
Point-and-click
Exploring the handmade world of LUNA

10 Best:

Projects to upgrade with Raspberry Pi 4

Raspberry Pi 4 is very powerful, and extremely fast. Here are some projects you could upgrade with one…

Whenever a new Raspberry Pi model comes out, we end up dreaming about what we can do with it. Raspberry Pi 4 is no different, especially with all the extra power it has! Here are ten excellent Raspberry Pi projects that would be made so much better with a Raspberry Pi 4. M



```
  GNU nano 2.7.4              File: /etc/samba/smb.conf

   read only = yes
   guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

[share]
   path = /home/pi/shared
   available = yes
   valid users = pi
   read only = no
   browsable = yes
   public = yes
   writable = yes

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

▲ File Server

Network storage

A lot of folks already have a Raspberry Pi file server in their homes. Their size and low power footprint make them perfect for the task. The only issue with them has been networking and storage bottlenecks, due to USB 2.0 and 100MB Ethernet. Now, with super-speed USB 3.0 ports and proper Gigabit Ethernet, Raspberry Pi 4 will give any file server a big speed boost.

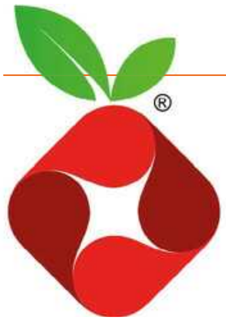
magpi.cc/samba

▲ Retro Games Console

Emulation station

Early tests have shown far better emulation of fifth-generation video game consoles. At the time of writing, the RetroPie team are hard at work on a version of their excellent retro gaming OS for Raspberry Pi 4.

magpi.cc/newpicade



◀ Pi-hole

Ad-blocking for extra security

We hear about people using Pi-hole all the time, especially with the amount of malicious and suspicious ads that exist on the internet these days. This blocks these ads before they even get to your computer – and with improved networking speeds, Raspberry Pi 4 makes this even better.

pi-hole.net

▶ Robots

Powered-up automatons

There are many pros in using a Raspberry Pi 4 as your robot's brain: the faster processing speeds and improved network connection makes for better control, whether by a remote or code. However, it does eat a bit more power than previous Raspberry Pi models, so consider what your needs are.

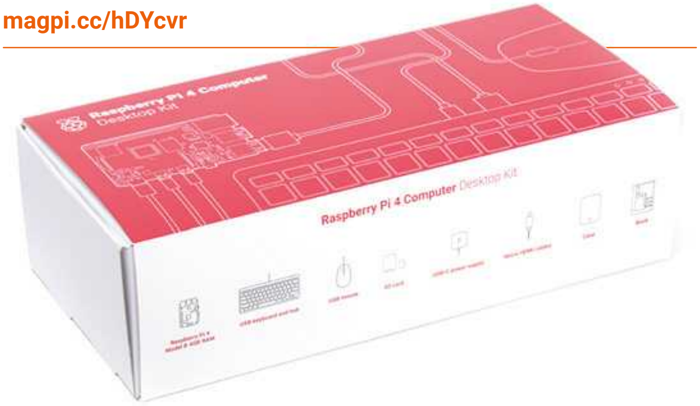
magpi.cc/QCZsyp

Desktop Replacement

Ditch Windows and Mac

Raspberry Pi 4's extra power means it can adequately act as a desktop replacement. Especially with the two micro-HDMI ports for dual displays. We'll be doing a proper test of this in a future issue, so look out for it!

magpi.cc/hDYcvr


Raspberry Pi 4 Computer Desktop Kit
Raspberry Pi 4 Computer Desktop Kit

Wireless Bridge

Share some internet

At an older hotel that only provides internet via an Ethernet cable? Fancy being a bit more secure on an open network? Using a Raspberry Pi 4 as a wireless bridge means you'll get a much faster internet and general network connection in this case.

magpi.cc/LhVfpE

Supercomputer Cluster

Parallel computing

Beowulf clusters made with Raspberry Pi are pretty cool for big maths problems and other number-crunching tasks. With about three to four times more raw processing power in Raspberry Pi 4 than Raspberry Pi 3B+, you'll be able to pull off some amazing calculations.

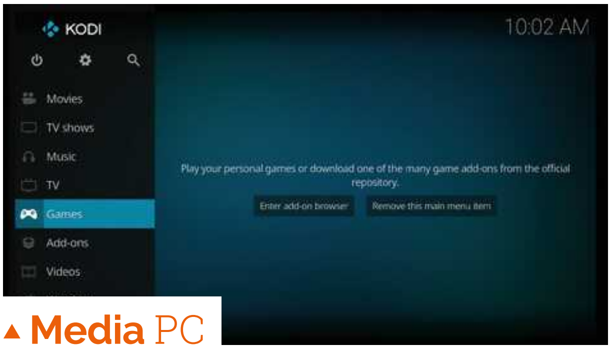
magpi.cc/VhUiPc

Machine Learning

4GB of artificial intelligence

More power and better network speeds mean you'll have a much better time using something like OpenCV for computer vision, or connecting with any number of online APIs to do some image or voice recognition.

magpi.cc/YKzcLp



Play your personal games or download one of the many game add-ons from the official repository.

Media PC

Raspberry Pi on telly

Putting Kodi on a Raspberry Pi is a time-honoured tradition, whether it's via LibreELEC, OSMC, or another dedicated media OS. It's already pretty great, so how can Raspberry Pi 4 make it better? 4K H.265 hardware decoding is how. This means 10-bit videos will play properly, and 4K video is possible. Even if you don't have anything like that to play, the better networking is sure to help.

libreelec.tv

Laptop Pi

Portable Raspberry Pi power

If Raspberry Pi 4 can be a desktop replacement, it can also be a laptop replacement. Custom Raspberry Pi laptop builds can be a bit tricky, but as Raspberry Pi 4 has largely the same design as previous Raspberry Pi models, you should be able to easily replace older ones.

magpi.cc/awpkvX

SPECS AND BENCHMARKS

Want to see just how good Raspberry Pi 4 is? Check out our full specs and benchmark breakdowns in issue 83 of *The MagPi* (**magpi.cc/83**)

Learn Sense HAT with Raspberry Pi

Essential resources for the space-borne add-on board for Raspberry Pi. By **Lucy Hattersley**

Astro Pi Resources

AUTHOR

Astro Pi

Price: Free

astro-pi.org/resources

The Sense HAT is an add-on board for Raspberry Pi. It adds a range of motion sensors to Raspberry Pi, including a gyroscope, accelerometer, and magnetometer, plus temperature, pressure, and humidity sensors. On top of the Sense HAT board sits an 8×8 LED matrix and five-button joystick.

Sense HAT is a lot of fun. It's a great way to get feedback on an environment with Raspberry Pi, and was made especially for the Astro Pi mission (where two Raspberry Pi units were sent into space).

Astro Pi is an ESA (European Space Agency) project run with the Raspberry Pi Foundation. It enables young people to conduct scientific investigations in space by writing computer programs that run on Raspberry Pi computers aboard the International Space Station.

Students up to 19 years old can take part. If you're older than that, you can still get involved by volunteering for a Code Club (**codeclub.org**) and mentoring students.

Even if you don't want to take part, Astro Pi has a solid set of resources designed to help students set up the Sense HAT and collect data from it.

Video Learning

Set up the Sense HAT by watching a video

ASTRO PI KIT – HOW TO ASSEMBLE THE SENSE HAT

This video from the ESA takes you through setting up the Sense HAT on a Raspberry Pi and using it to take part in the Astro Pi project.

magpi.cc/uriDhA

HOW TO BUILD A 3D-PRINTED ASTRO PI FLIGHT CASE

The Sense HAT units used in the Astro Pi program are housed inside special cases designed to endure the rigours of space flight. Discover how to 3D-print your own special Astro Pi case in this video.

magpi.cc/axK0OB

TIM PEAKE WITH ASTRO PI ON THE INTERNATIONAL SPACE STATION

Take a tour of the ISS and the Sense HAT kits being used on board the space station with British ESA astronaut Tim Peake.

magpi.cc/dacYpU

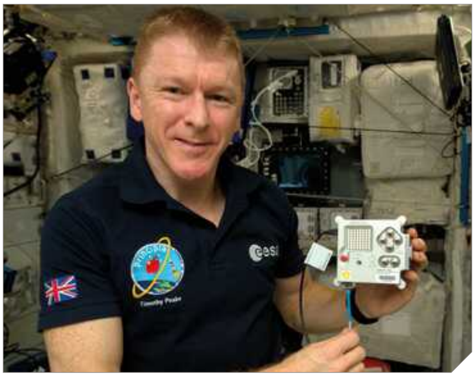
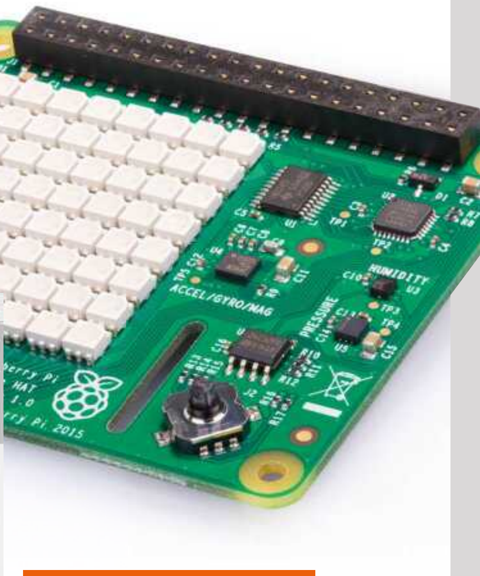


Image credit: ESA

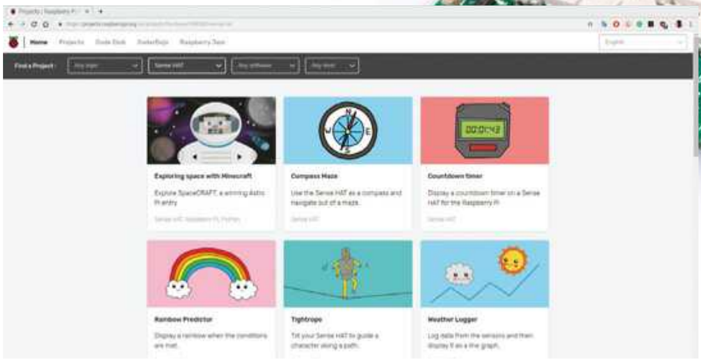


Getting Started with Sense HAT

AUTHOR
Raspberry Pi Foundation

Price:
Free

rpf.io/projects



Find a Project: Any topic | Sense HAT | Any software | Any level

Exploring space with Minecraft
Explore SpaceCRAFT, a winning Astro Pi entry
Sense HAT, Raspberry Pi, Python

Compass Maze
Use the Sense HAT as a compass and navigate out of a maze.
Sense HAT

Countdown timer
Display a countdown timer on a Sense HAT for the Raspberry Pi
Sense HAT

Rainbow Predictor
Display a rainbow when the conditions are met.

Tightrope
Tilt your Sense HAT to guide a character along a path.

Weather Logger
Log data from the sensors and then display it as a line graph.

If you're looking to get up-and-running with Sense HAT, the Raspberry Pi Foundation has you covered. The Getting Started with the Sense HAT tutorial (**magpi.cc/QBynLN**) shows you how to display text and images on the LED matrix display, set the orientation, and sense the environment and movement.

After that, head over to the Raspberry Pi Foundation's projects website, click 'Browse all projects', and change the 'Any hardware' drop-down to Sense HAT (**magpi.cc/FCCwsc**). Here you'll find a range of fun projects that explore all the different elements of the Sense HAT – from building a tight-rope game controlled by the accelerometer, to a weather logger and rainbow predictor. The projects are sparkly and designed to appeal to kids, but they're a great way to learn data capture and basic programming concepts.

Experiment with the Sense HAT

AUTHOR
Raspberry Pi Education Team

Price:
Free

magpi.cc/sense-hat-book

Essentials Guides are short books designed to make it easy to learn Raspberry Pi subjects. One of the most popular and enduring is *Experiment with the Sense HAT*.

You can still pick up a digital copy of this book for free, although print copies are now hard to find.

Inside the book, you'll find eight chapters that cover everything from setting up the Sense HAT to building projects such as a Magic 8 Ball and Interactive Pixel Pet.

While the Astro Pi mission has evolved beyond that described in the book, the tutorials have stood the test of time. They remain a great way to understand the inner workings of the Sense HAT and how to control it with code.

Web resources

Get the most from your Sense HAT with this online documentation

SENSE HAT API

Make sure you bookmark the Sense HAT API webpage. This reference outlines all the functions used to control the HAT (and used in a variety of different tutorials).
magpi.cc/cDbEPq

SENSE HAT GITHUB

Take a look at the Python module used to control the Sense HAT hardware on Raspberry Pi. Here you will find the underlying code being used to control the device.
magpi.cc/f0tNFp

SENSE HAT DOCUMENTATION

Raspberry Pi documentation for the Sense HAT explains how to set up and calibrate the device, as well as update the firmware. Check out the link to the schematics while you're there.
magpi.cc/fNqSCn

Anne Carlill

Veteran programmer and founder of York Raspberry Pi Jam

› Day job **Retired** | › Community role **Raspberry Jam runner** | › Website **magpi.cc/YVgvmU**

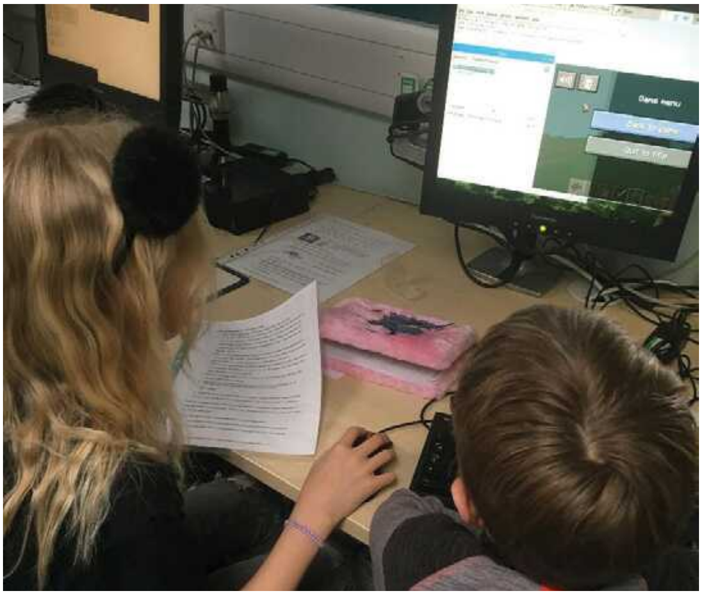
"York is a small city but I felt it had potential to have a regular Jam," Anne Carlill explains to us. For those outside of the UK, York is a historic city, even if it is a little small, and a symbol of the North of England.

Anne herself definitely has the know-how to run a Jam in York. She's been coding for 45 years, passing on her skills to her children with a BBC Micro in the eighties. She's also been a college teacher, and now, although she's retired, she has been running a Code Club for four years.

"There had been a couple of big Jams at the STEM Centre but no follow-up," Anne continues. "I've been going to Leeds Raspberry Jam for about four years and have learnt so much from their organiser Claire Garside – and it was Claire who suggested I should consider running a Raspberry Jam. I went to Cambridge for the training, took a while to find venues, and started running my own at the beginning of 2017. I think of YRJ as a little offshoot of Leeds, as our attendance is small, but we are growing. York Explore Libraries have been very supportive and both of our venues are libraries: one in West and one in East York."

What are the York Raspberry Jam attendees like?

They are all ages and abilities, but mostly parents with children: they're beginners right up to people who know a lot more than me. Luke (**@YorkPiJam**) and I began organising together in March 2018. It's a bit odd having a 16-year-old and a grandma, in her late 60s, planning together,



Learning Minecraft Pi seems to still be popular everywhere

Luke runs the Jam with Anne – you may have seen him in the magazine writing tutorials in the past

York Pi Jam was one of the many Raspberry Jams that celebrated Raspberry Pi's seventh birthday this year!

but it works. We plan by social media as we don't see each other between Jams. Luke is the brains, and I'm the facilitator who makes sure the kit works and there are biscuits, etc.

> What excites me most is to see people who haven't got much cash really taking to the Raspberry Pi!

How did you learn about Raspberry Pi?

I heard a report on the Radio 4 *Today* programme about six years ago and just had to have one. My Model B setup cost £60 and I have about £20 a month to save up or spend on electronics kit. What excites me most is to see people who haven't got much cash really taking to the Raspberry Pi! And I'm very keen to know how well Raspberry Pi is going down in places like India and Africa where the cost really matters. I love hearing from people in those places about what they're managing to do with their frugal kit of Raspberry Pi devices.

What have you made with a Raspberry Pi?

I'm a beginner with physical computing, but I really enjoy having a go. I've made the CamJam EduKit #3 robot buggy and several light-up wearables. I think my best project so far has been the My Naturewatch Camera Trap, though, because I was given a Raspberry Pi Camera Module a few years back and didn't know what to do with it. Then I heard about this Raspberry Pi Zero project, I just had to have a go – and it works! So far, my pictures have been rubbish, but I'm determined to persist.

Any Raspberry Pi projects you'd love to make?

I'm really keen on the environment and looking at climate change, so I'd love to get a full environmental monitor, such as the Enviro+ with particulate matter sensor, because I'd love to do some citizen data science and be part of a large-scale data project.

York Pi Jam

Want to check out York Pi Jam? Keep an eye on the Eventbrite page for it to find out when new Jams are planned: **magpi.cc/YvgvmU**. They will also be in our Events list on page 90 when they're coming up.

This Month in Raspberry Pi

Raspberry Pi 4 in the wild!

Last month, Raspberry Pi 4 was released – here's some of the cool stuff we've seen with it so far!

Can you believe it's only been a month since Raspberry Pi 4 came out? Hopefully everyone that wanted it has managed to grab one by now. However there are still plenty of people that have already been making amazing things with their new Raspberry Pi.

01. People were very happy to get their Raspberry Pi 4 computers

02. Gavin Stephens wasted no time putting his Raspberry Pi 4 boards in a cluster

03. Raspberry Jams are already getting Raspberry Pi 4 computers in to play with!

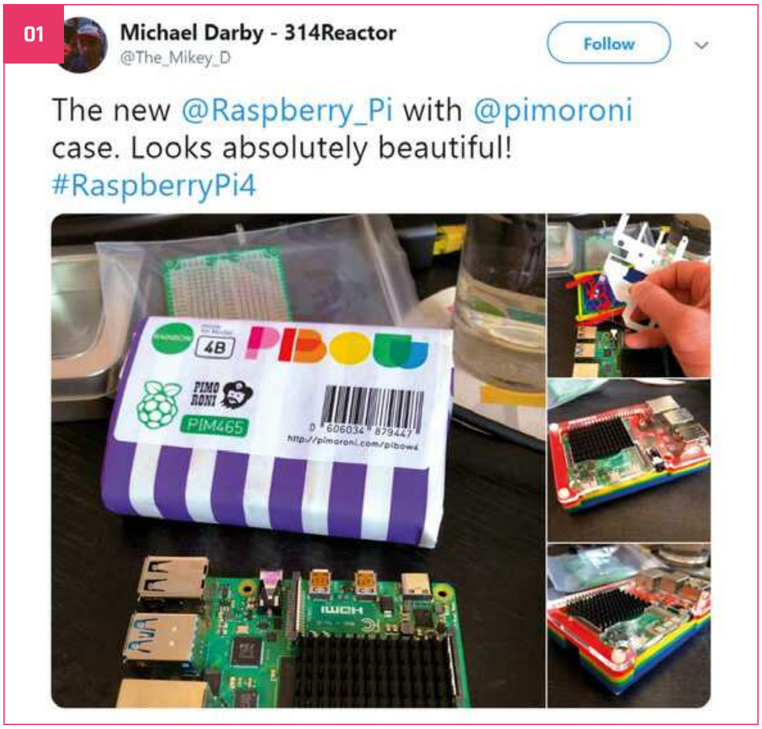
04. That's one way to play your own music in your car!

05. Some folks flexed their 3D printing skills to do stuff like this VESA mount

06. The desktop replacement Raspberry Pi 4 devices are already out there!

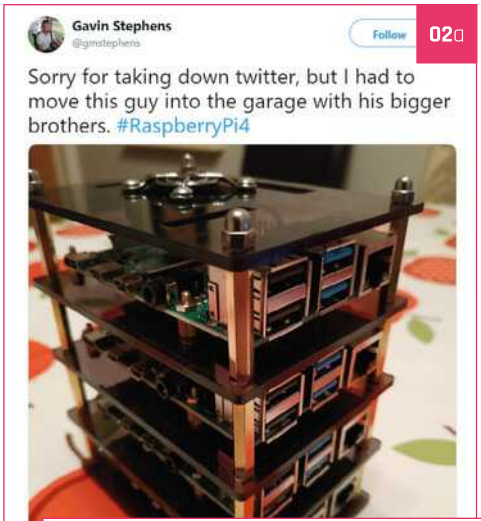
07. Reddit user xCP23x took some amazing X-ray photos of Raspberry Pi 4! Check them out here: **magpi.cc/pDHhTC**

01



Michael Darby - 314Reactor
@The_Mikey_D

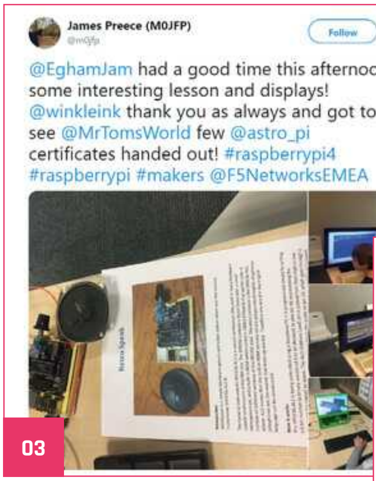
The new @Raspberry_Pi with @pimoroni case. Looks absolutely beautiful! #RaspberryPi4



02a

Gavin Stephens
@gmstephens

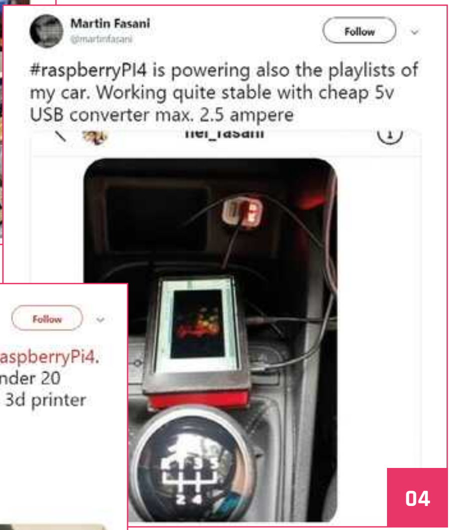
Sorry for taking down twitter, but I had to move this guy into the garage with his bigger brothers. #RaspberryPi4



03

James Preece (M0JFP)
@m0jfp

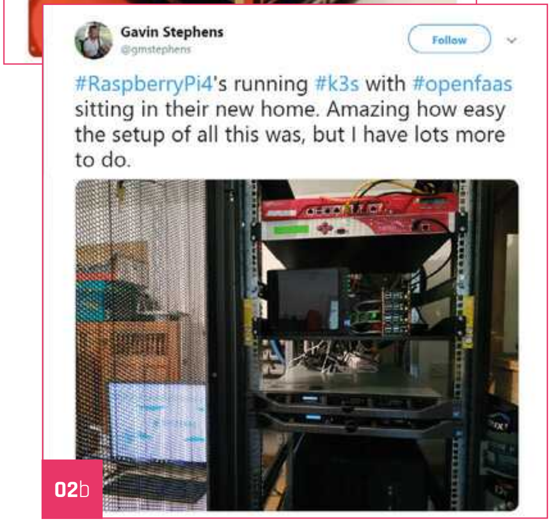
@EghamJam had a good time this afternoon some interesting lesson and displays! @winkleink thank you as always and got to see @MrTomsWorld few @astro_pi certificates handed out! #raspberrypi4 #raspberrypi #makers @F5NetworksEMEA



04

Martin Fasani
@martinfasani

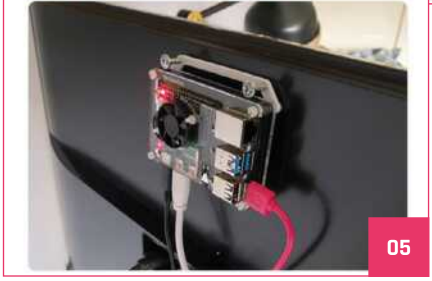
#raspberryPI4 is powering also the playlists of my car. Working quite stable with cheap 5v USB converter max. 2.5 ampere



02b

Gavin Stephens
@gmstephens

#RaspberryPi4's running #k3s with #openfaas sitting in their new home. Amazing how easy the setup of all this was, but I have lots more to do.



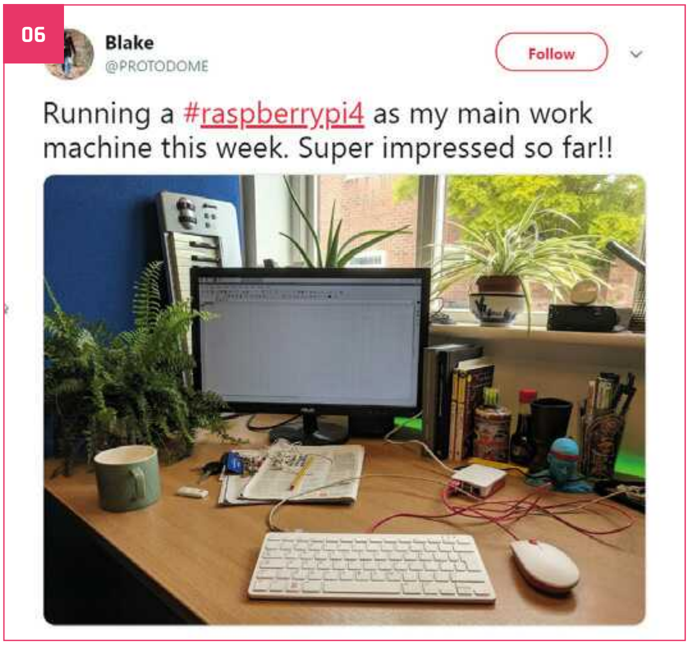
05

Nick O'Leary
@knolleary

Created a VESA mount for my #RaspberryPi4. From cad in blender to done in under 20 minutes. The benefits of having a 3d printer on hand.

@pimoroni

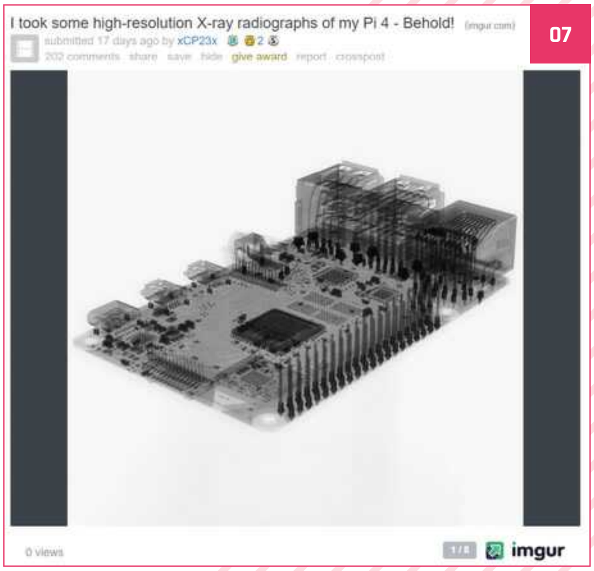
instagram.com/p/BznvXssJLr9



06

Blake
@PROTODOME

Running a #raspberrypi4 as my main work machine this week. Super impressed so far!!



07

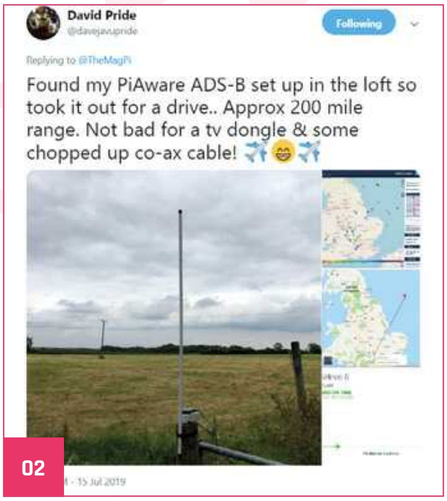
I took some high-resolution X-ray radiographs of my Pi 4 - Behold! (imgur.com)
submitted 17 days ago by xCP23x 2
202 comments share save hide give award report crosspost

0 views

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 is a small fraction of them. Follow along at **#MagPiMonday**.



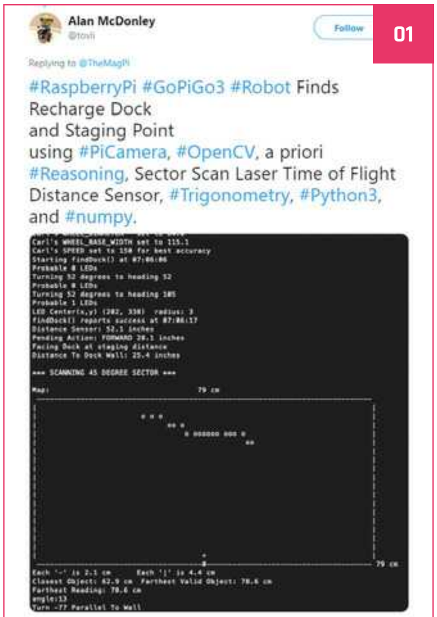
02

David Pride
@davejavupride
Following

Replying to @TheMagPi

Found my PiAware ADS-B set up in the loft so took it out for a drive.. Approx 200 mile range. Not bad for a tv dongle & some chopped up co-ax cable! ✈😁✈

1 - 15 Jul 2019



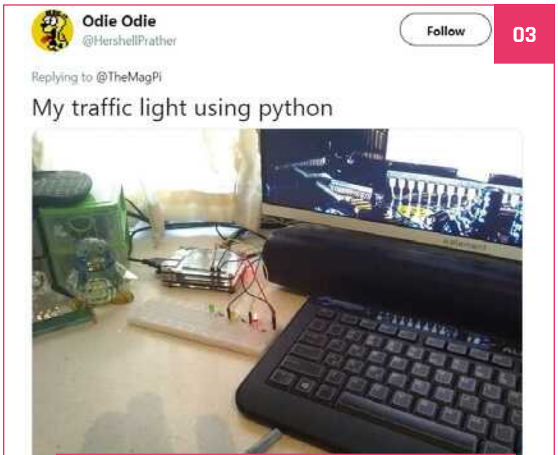
01

Alan McDonley
@tovli
Follow

Replying to @TheMagPi

#RaspberryPi #GoPiGo3 #Robot Finds Recharge Dock
and Staging Point
using #PiCamera, #OpenCV, a priori #Reasoning, Sector Scan Laser Time of Flight Distance Sensor, #Trigonometry, #Python3, and #numpy.

```
Carl's WHEEL_BASE_WIDTH set to 115.1
Carl's SPEED set to 150 for best accuracy
Starting findDock() at 07:06:06
Probable 0 LEDs
Turning 52 degrees to heading 52
Probable 0 LEDs
Turning 52 degrees to heading 105
Probable 1 LEDs
LED Center(x,y) (202, 330)  radius: 3
findDock() reports success at 07:06:17
Distance Sensor: 52.1 inches
Pending Action: FORWARD 28.1 inches
Facing Dock at staging distance
Distance To Dock Wall: 25.4 inches

*** SCANNING 45 DEGREE SECTOR ***

Map:                              79 cm
 ----------------------------------------------------------------
|                                                                |
|                  * * *                                         |
|                       ** *                                      |
|                          * ****** *** *                         |
|                                        **                       |
|                                                                |
|                                                                |
|                                 +                               |
 ---------------------------------#------------------------------ 79 cm
Each '-' is 2.1 cm      Each '|' is 4.4 cm
Closest Object: 62.9 cm  Farthest Valid Object: 78.6 cm
Farthest Reading: 78.6 cm
angle:13
Turn -77 Parallel To Wall
```

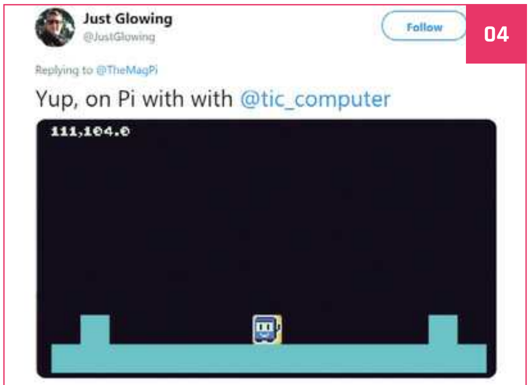


03

Odie Odie
@HershellPrather
Follow

Replying to @TheMagPi

My traffic light using python

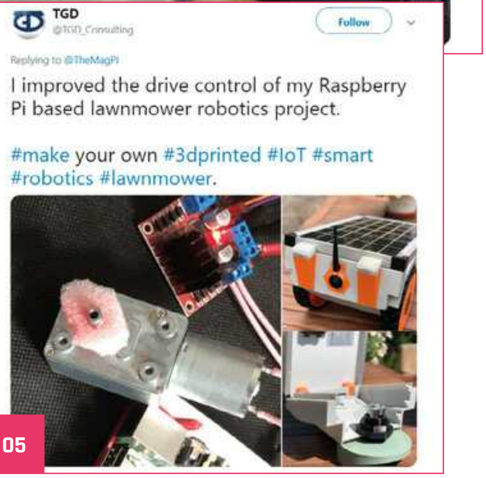


04

Just Glowing
@JustGlowing
Follow

Replying to @TheMagPi

Yup, on Pi with with @tic_computer



05

TGD
@TGD_Consulting
Follow

Replying to @TheMagPi

I improved the drive control of my Raspberry Pi based lawnmower robotics project.

#make your own #3dprinted #IoT #smart #robotics #lawnmower.

01. This is an extremely impressive bit of coding that gets a robot to charge itself like a Roomba

02. A 200-mile radius is pretty incredible for this kind of device

03. Even doing something as simple as traffic light LEDs is a first step for someone

04. Programming games is always fun on Raspberry Pi

05. A solar-powered lawn-mowing robot? Look out for more on this in a future issue…

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month

LYRA

This very slim and lovely-looking handheld console uses a Raspberry Pi Compute Module, which helps it stay quite slim. It sports a D-pad, four face buttons, and a couple of shoulder buttons, which is enough to play pretty much every game made before 1995. You can even get a cheaper DIY kit so you can have fun building it yourself.

▶ **kck.st/2Jih2r4**

CROWDFUNDING A PROJECT?

If you've launched a Raspberry Pi-related project, let us know!

magpi@raspberrypi.org

Portable Retro Game Console

Another portable games console! This one uses a Raspberry Pi 3A+ to keep it slim, and also includes a few more input buttons and a more colourful design. It has a much larger screen, at 7.9 inches, which is a lot bigger than even the Wii U gamepad.

▶ **kck.st/2IPBDCL**

Best of the rest!

Here are some other great things we saw this month

SD CARD FIX

'Broken SD card slot? Not anymore!' While it's a clever fix, every engineer we showed it to made many uncomfortable noises. Now you have to look at it as well.

▶ **magpi.cc/ZLXFDT**

RASPBERRY PI NETBOOK

Are netbooks still a thing, or did tablets effectively replace them? Either way, we do love a good laptop conversion project, and this one using an old portable DVD player and a Raspberry Pi Zero is pretty spectacular.

▶ **magpi.cc/SFqcgS**

Raspberry Jam Event Calendar

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

01. Hythe Library Jam Summer Coding Session

- Tuesday 30 July
- Hythe Library, Hythe, UK
- **magpi.cc/jetzUj**

A morning of amazing coding! Design cool games and take part in a Summer Reading Challenge coding mission.

02. Exeter Raspberry Jam

- Saturday 3 August
- Exeter Library, Exeter, UK
- **magpi.cc/MExQAp**

A monthly meeting for everyone interested in all things microcomputers, microcontrollers, robotics, and making.

03. Seattle Raspberry Jam

- Wednesday 21 August
- Bellevue Library, Bellevue, WA, USA
- **magpi.cc/NvsBzz**

Come and participate in the monthly project, share your knowledge, or show a project you've created.

04. Scratch Conference Europe

- Friday 23 August to Sunday 25 August
- Churchill College, Cambridge, UK
- **magpi.cc/PAsskt**

Hosted by the Raspberry Pi Foundation, this event will have loads of exciting activities to do and talks to attend.

05. Preston Raspberry Jam

- Saturday 31 August
- Fulwood Library, Preston, UK
- **magpi.cc/utJqiR**

A Saturday morning digital making event, with children, adults, and families, from beginners to experts.

06. Stafford Raspberry Jam

- Tuesday 10 September
- Stafford Library, Stafford, UK
- **magpi.cc/hRUXcz**

Welcoming anyone who wants to show off their projects, or see other people's builds!

07. Roanoke Raspberry Jam

- Saturday 21 September
- Roanoke South County Library, Roanoke, VA, USA
- **roanokeraspberryjam.info**

Join in the digital making community at the Roanoke Raspberry Jam hosted by the South County Library.

08. Castro Valley Jam

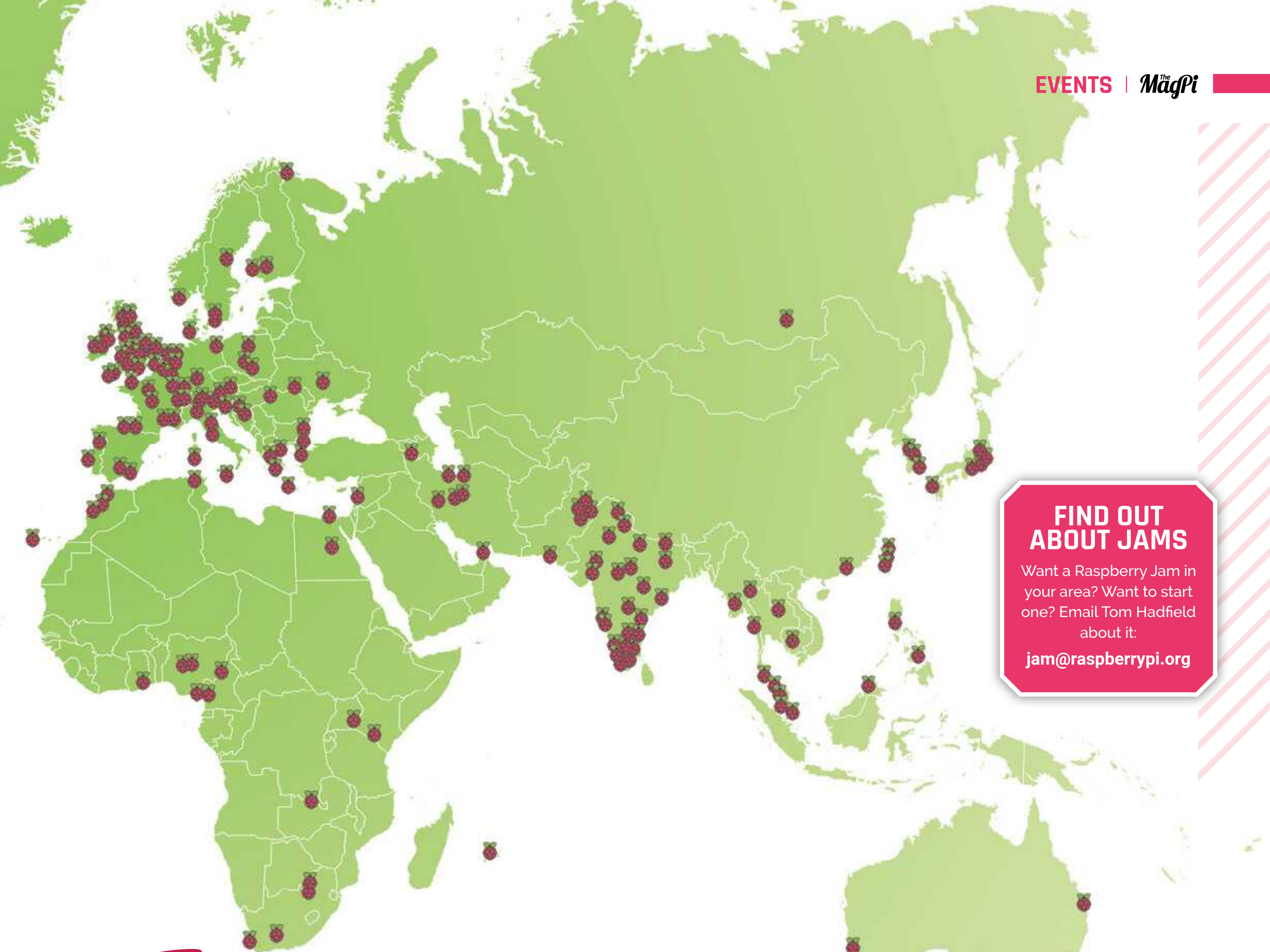
- Saturday 5 October
- Castro Valley Library, Castro Valley, CA, USA
- **magpi.cc/eXpRiF**

If you're interested in coding and want to learn about Raspberry Pi in Castro Valley, check out this Jam.

FULL CALENDAR

Get a full list of upcoming events for August and beyond here:

rpf.io/jam

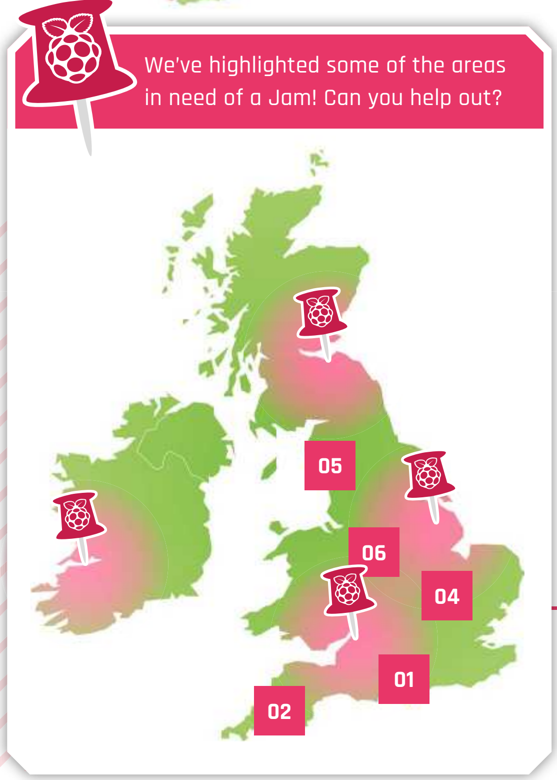


FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one? Email Tom Hadfield about it:

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?

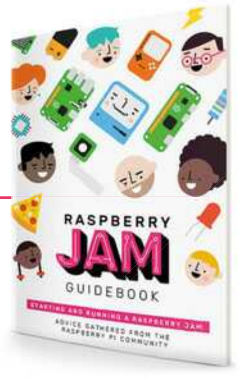


Raspberry Jam advice:
Activities

"When a competition like Astro Pi, or a programme like Pioneers comes out, we'll do talks and invite people along to take part in workshops at the Jam. It's good because it gives an incentive for people to come along and take part, and gives them motivation to work. We celebrate the entrants, and those who've won, and invite them to do talks. We ask them to mentor at future Raspberry Jams, or run workshops."

Andy Melder – Southend Raspberry Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **magpi.cc/2q9DHfQ**


RASPBERRY
JAM
GUIDEBOOK
STARTING AND RUNNING A RASPBERRY JAM
ADVICE GATHERED FROM THE RASPBERRY PI COMMUNITY

Your Letters

Past issues

If I subscribe in print from the USA, will I get access to the previous issues online? Do I get access once I set up the 'My Account' section?

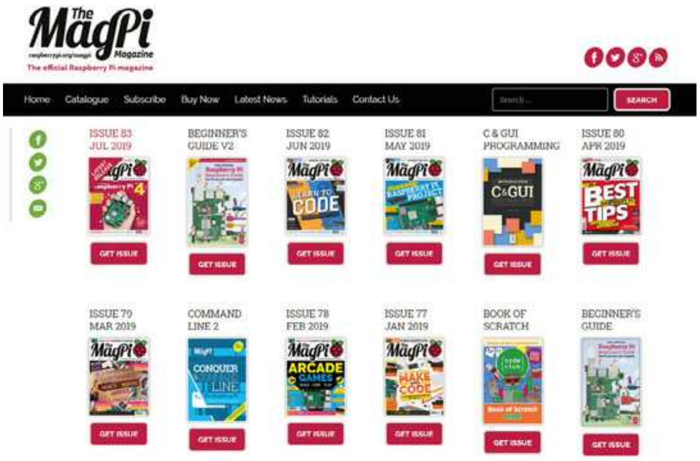
Daniel via email

Every issue is available as a free PDF from our website at **magpi.cc/issues**, so you'll be able to get them even if you don't have a subscription! Just find the magazine or book you want to read, and click on it to find out more.

The My Account section allows you to manage your subscription, as well as input some payment and address details in case you make further purchases on the Raspberry Pi Press Store. It's just a way to make it easier for our users.

Contact us!

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



▲ Make sure to check out our catalogue page for any old issues you wish to read as a PDF

On boot

I love *The MagPi* and always eagerly await my copy in the post every month. However, different article authors seem to have different opinions about the best way to run a script (say a Python program) when a Raspberry Pi boots. I will be needing to run a script on boot for the first time for my Times Square New York project, and I was wondering how best to do that, as some say you should add a line to crontab, some say to add a line to /etc/rc.local, some say to add a line to /etc/profile. I would like to know the best way to run a script on boot – *The MagPi*'s verdict.

Jonathan via email

The reason you see us do it a few different ways is mostly down to personal taste, and a bit to do with the kind of project people are making.

The two best methods in our opinion are using **/etc/rc.local** and **/etc/profile**. The former is considered best practice, as it makes the script part of the startup of Raspbian, so should not have as much effect on the way you use a Raspberry Pi otherwise. The official documentation on **rc.local** covers how to do this: **magpi.cc/SmPFZY**.

The **profile** method is quick and dirty, although it generally achieves the same results. While both are good if you're using a headless Raspberry Pi setup (i.e. no monitor or need for a desktop environment), the **rc.local** method lets you dial into a Raspberry Pi a little more cleanly.

Give both a try and see what works for you!

OS test

Hello, I am a fan of *The MagPi* and would like to suggest something to put in *The MagPi*! Since Raspberry Pi 4 has 2GB of RAM and 1.5GHz of processing power, this unlocks more opportunities for new and more resource-hungry operating systems. A good idea for 'The Big Feature' or a cover feature would be the top ten operating systems for Raspberry Pi 4! And maybe a tutorial on how to install some of them! If this does make its way into *The MagPi*, please include Android as one of these operating systems.

Joseph via email

We've done this sort of feature before, testing out all the available operating systems for Raspberry Pi. However, we're overdue an update and Raspberry Pi 4 is a good reason to do a new one at some point.

Unfortunately Raspberry Pi 4, while it is more powerful (and can have up to 4GB of RAM!), can't just have any operating system installed on it. They need to be specifically made for Raspberry Pi 4 as it has an ARM processor and not an x86/x64 processor like a standard PC. So while we'll try out as many operating systems as we can, we won't be able to check out full Windows 7 or 10, etc.

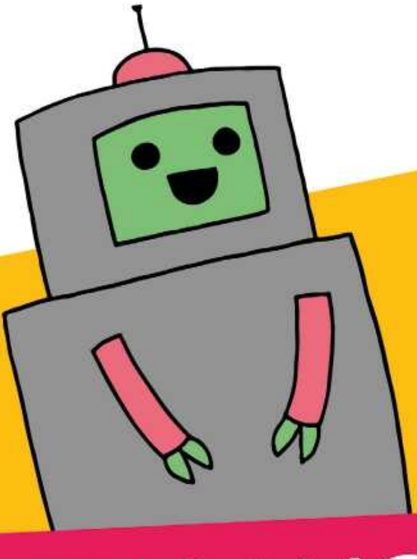
As for Android, that's up to Google! We would love to see it on Raspberry Pi 4, and we'd be more than happy to test it if it gets ported.

Apollo flight

I'm looking forward to seeing how Dave Akerman's Apollo anniversary flight goes in this issue! I've been wanting to get into high-altitude ballooning myself, and it's inspiring me to look into it!

Stella via Facebook

Dave's Apollo anniversary balloon launch was due to happen just as we were going to press, so we don't have any amazing photos of it quite yet – please look out for them in the next issue! Dave has even created a custom Apollo Guidance Computer (AGC) display for this special launch (**magpi.cc/ethRqK**).



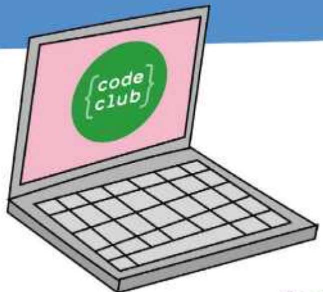
{code club}

CAN YOU HELP INSPIRE THE NEXT GENERATION OF CODERS?

Code Club is a network of volunteers and educators who run free coding clubs for young people aged 9-13.

We're always looking for people with coding skills to volunteer to run a club at their local school, library, or community centre.

You can team up with friends or colleagues, you will be supported by someone from the venue, and we provide all the materials you'll need to help children get excited about digital making.



To find out more, join us at

www.codeclubworld.org

Code Club is part of the Raspberry Pi Foundation. Registered Charity Number 1129409

WIN One of three RASPBERRY PI 4 Desktop Kits

The full Raspberry Pi Desktop Kit includes just about everything you need to use a Raspberry Pi 4! Along with a Raspberry Pi 4 itself, it comes with a power supply, two micro-HDMI to HDMI cables, 16GB microSD card, case, keyboard and mouse, and our 252-page *Raspberry Pi Beginner's Guide*.

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

Terms & Conditions

Competition opens on **24 July 2019** and closes on **29 August 2019**. 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.

CoderDojo

Can I start a CoderDojo club in my local area?

CoderDojo is a global network of free, volunteer-led, project-based programming clubs for children aged 7–17. Dojos are championed by individuals all around the world who are passionate about giving young people the opportunity to learn to code.

Starting a Dojo is a fun and incredibly rewarding experience

You don't need to possess technical skills to start a Dojo. The most important attribute is that you can bring people together for a shared goal.

We're ready to support you by providing:

- **Learning resources and guides**
- **A free event management system**
- **Certificate templates, digital badges, and giveaways**

> "I started a Dojo to give my kids a place to meet other children also interested in programming and making games. I get to see them making new friends, learning from one other, and they loved it. Realising how I had created such a wonderful place for children has ignited a spark in me."
>
> - Maroes, CoderDojo NL

Start your own club. Join us at CoderDojo.com

NEXT MONTH | The MagPi

RASPBERRY PI 4 DESKTOP PC

Using Raspberry Pi as your only computer. Discover the best setup, accessories, and apps

THE MAGPI #85 ON SALE 29 AUGUST

Plus!

Make a Raspberry Pi NAS router

Design video game sprites in PICO-8

Back to school: Raspberry Pi in education

Play Go with TensorFlow and Coral USB Accelerator

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.org

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.org

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors
Phil King and Nicola King

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

DESIGN
criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Sam Ribbits

Illustrator
Sam Alder

CONTRIBUTORS
Mike Cook, David Crookes, PJ Evans, Rosemary Hattersley, Daniel Lambton-Howard, Nicola King, Melissa LeBlanc-Williams, Simon Long, Sean McManus, Danny Staple

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Director of Communications
Liz Upton

CEO
Eben Upton

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

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

recycle When you have finished with this magazine, please recycle it.

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.

Raspberry Pi PRESS

ISSN: 2051-9982.

The joys of making jam

For **PJ Evans**, running a Raspberry Jam is more about the people than the computers

It's Friday night. The kitchen table is stacked with boxes. It's mostly paperwork, stationery, computers, and some rather tempting cookies. Tomorrow, my wife and I will be up early to head to The National Museum of Computing, where we will hold the next Milton Keynes Raspberry Jam. This will be the ninth meet I have supervised, although the history of the MK Jam goes right back to the start, Peter Onion founding the event shortly after the launch of the original Raspberry Pi.

> Younger kids squeal with delight as Zobbie the robot springs into life

As usual, I'm a bunch of nerves. Will people show up? Have we remembered everything? Will the attendees like it? Every Friday before the Jam, I'm convinced we are heading for an embarrassing failure.

Saturday comes. We arrive at a sleepy museum and start erecting tables and putting up posters. One-by-one the volunteers arrive and my fears quickly dispel. Smiling faces, full of enthusiasm, travelling far and giving up their time for free to make this event work. I'm just the caretaker here; without their dedication, there would be no event. Now we're a team with a common goal that reminds me of Lord Reith's mandate for the BBC: inform, educate, entertain.

Enthusiasm abounds

Attendees always start arriving before we're ready. I don't mind: it's fantastic that people can't wait to get started. Projects are set up, workshops get booked, and the number of curious children are matched only by the robots that are now scuttling by our feet. An hour in, our four rooms are full to bursting. Electronic noises, flashing LEDs, and wide, curious eyes. With everyone settled, it's about now I can pause and take it all in.

This is the good bit, the part where I remind myself why we do this. A sea of people engaging with technology. Some learning, others sharing knowledge. Younger kids squeal with delight as Zobbie the robot springs into life, and partners who have patiently accompanied their geeky other-half get drawn in and start asking questions. At the rear of the main room, a group of retired engineers are busy sharing the progress of their latest projects.

Great leveller

Afterwards, we head off for a debrief lunch and reflect how Raspberry Jams are communities. We get the opportunity to inspire young people and help them start their journeys into technology. There are also those retirees who get to indulge their passions with like-minded people who have become friends. For me, that's what Raspberry Pi is all about. The low cost and openness makes it a great leveller. There is rightly a lot of focus on Raspberry Pi in education, but it's also brought people together in later life who might otherwise be lonely. Raspberry Pi is for everyone.

Sunday morning. A quick email check, coffee, and breakfast. I create a new folder: MK Jam #10. I hope people turn up.

AUTHOR

PJ Evans

When not writing for *The MagPi* and others, PJ supervises the Milton Keynes Raspberry Jam and thinks you should come along to the next one..

mkraspberryjam.org

Your Logo
Here