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

The MagPi

Issue 76 | December 2018 | magpi.cc

The official Raspberry Pi magazine

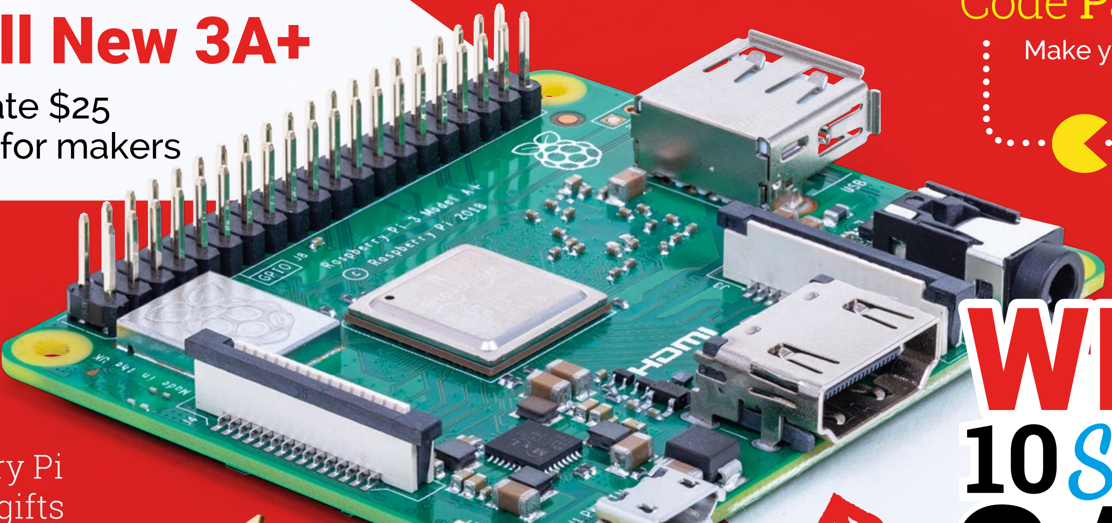
NEW FOR 2019

THE RASPBERRY PI SUPERGUIDE

- Get going guide
- Starter projects
- Tech jargon busted

The All New 3A+

The ultimate $25 computer for makers



Code Pac-Man

Make your own retro ghost-chaser

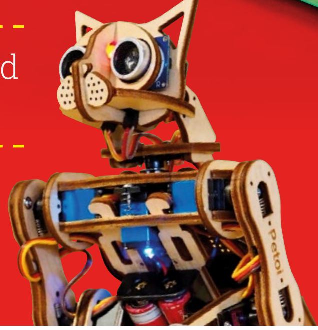
WIN! 10 Signed 3A+

PLUS

10 Raspberry Pi Christmas gifts

Script command line tasks

Meet Nybble: the powerful robotic kitten

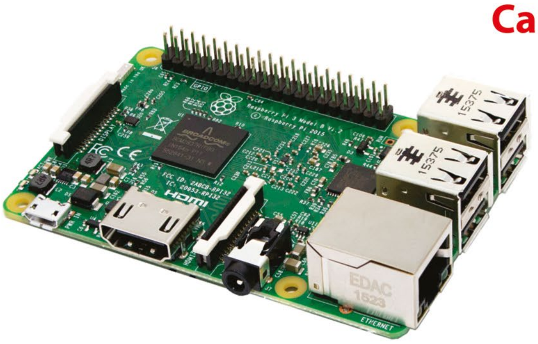


Make Smart Lights

Add to the festive fun with Pi-controlled house lighting

Scratch Resources | Raspberry Pi Boombox | GFX HAT

CanaKit™



CanaKit Raspberry Pi 3 Ultimate Starter Kit

Model B | 1 GB RAM | 1.2 GHz | Quad-Core CPU

> Learn to Code
> Explore Computing
> Get started with Electronics

KIT INCLUDES RASPBERRY PI 3 AND ...

PREMIUM CASE & HEAT SINKS

2.5A POWER ADAPTER

32 GB CLASS 10 MICROSD CARD

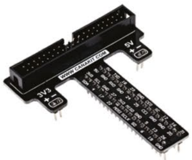
PRE-LOADED WITH OPERATING SYSTEM

USB MICROSD CARD READER

PREMIUM HDMI CABLE

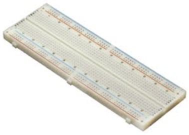
QUICK-START GUIDE

GPIO TO BREADBOARD INTERFACE BOARD



RIBBON CABLE

FULL-SIZE BREADBOARD



JUMPERS

MALE TO MALE & MALE TO FEMALE

LEDS

RESISTORS & PUSH-BUTTONS

Available for worldwide shipping at:

WWW.CANAKIT.COM

Raspberry Pi Zero W

Now available at CanaKit!

Wi Fi™ Bluetooth®

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO

WELCOME
to The MagPi 76

It's the most wonderful time of the year. The time when lots of folks see a wild Raspberry Pi appear under the tree for the first time.

And this month we're all being treated to a brand new Raspberry Pi, the super-sweet new 3A+ (page 66). 'Tis the season to be jolly and break out the new projects (especially ones that need a fast, but energy-efficient small computer).

If you're new to Raspberry Pi, then let me be the first to welcome you to the party. Grab some cake and help yourself to a drink!

Our Raspberry Pi Superguide (page 28) gives newcomers the all-star treatment. In it you'll find advice on how to set up a Raspberry Pi and get started with basic electronic components. You'll also learn how to set up simple projects like a retro gaming console and home theatre PC. It's proper computing with pins and programming.

Enjoy the party!

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

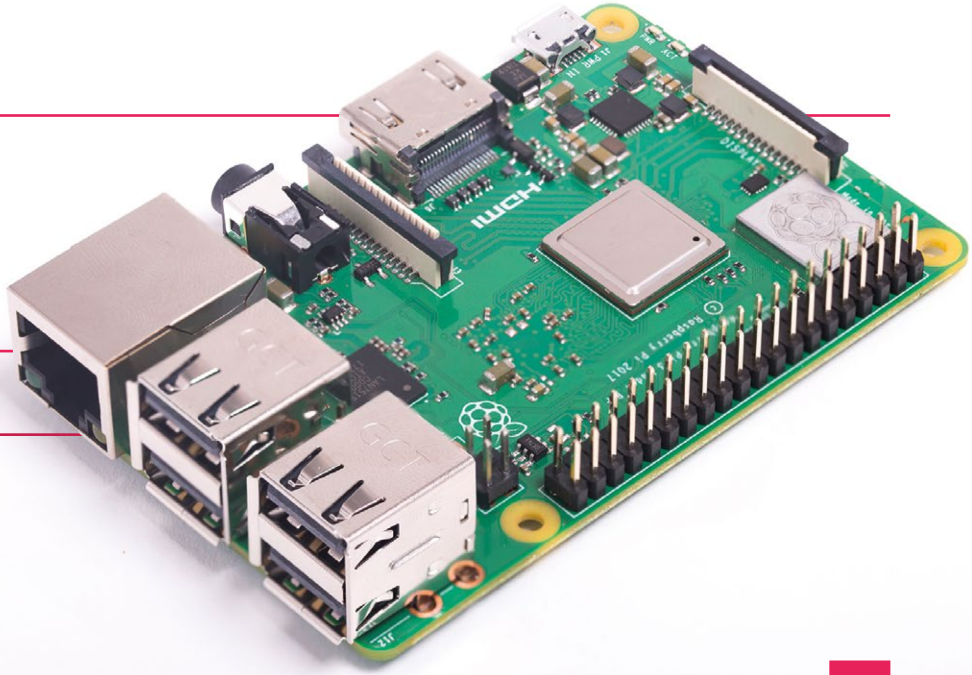
Editor of *The MagPi*. Lucy codes, crafts, and creates wonky robots. She speaks French (badly) and mangles the piano. One day she'll get that pet dog.

magpi.cc

Contents

> Issue 76 > December 2018

Cover Feature



28

Regulars

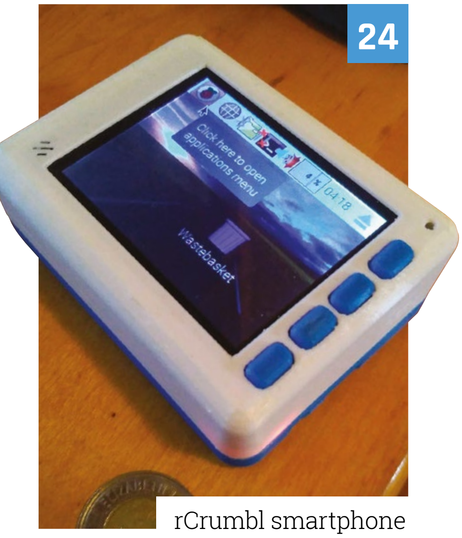
Project Showcases

12

As We Are

22

Tide Clock Weather Thing



24

rCrumbl smartphone

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.

Make your smart lights smarter

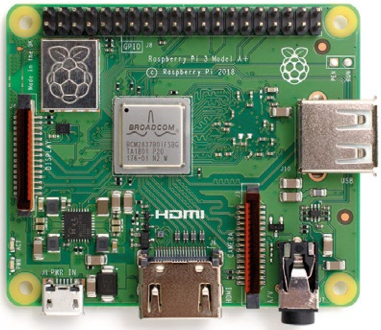
Pac-Man is back...

Code a fun game with your matrix of switchable lights

Christmas fun with these kits!

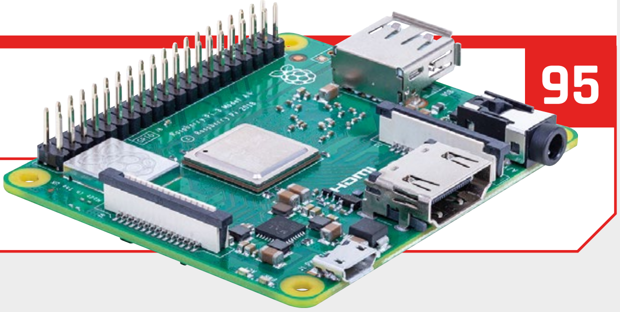
Tutorials

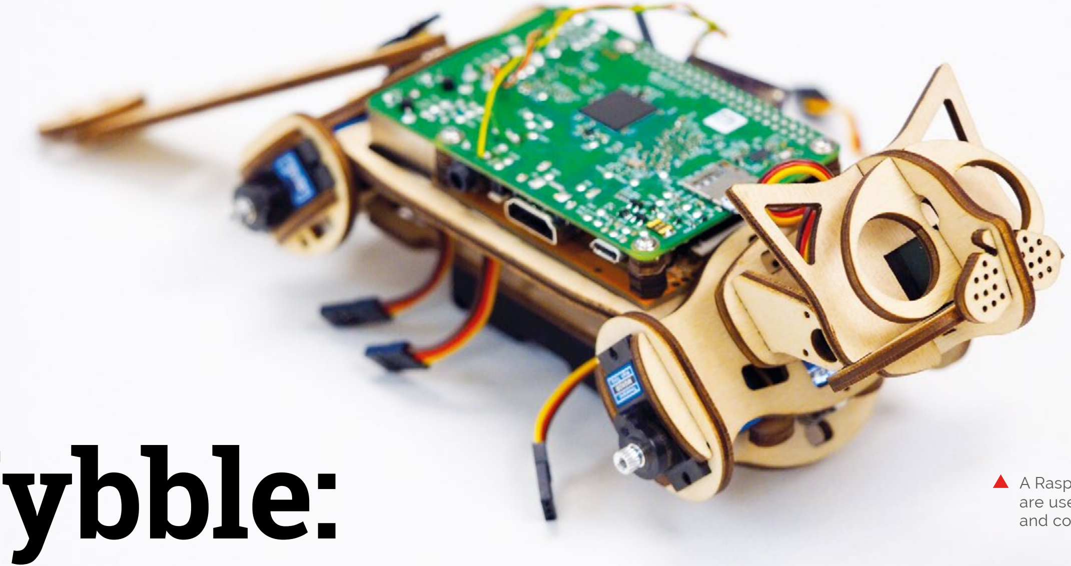
The Big Feature



Reviews

Community





▲ A Raspberry Pi and NyBoard are used to make decisions and control the cat's legs

Nybble: the open-source kitten

Raspberry Pi robotic cat leaps onto Indiegogo. By **Lucy Hattersley**

Back in *The MagPi* issue 68 (**magpi.cc/68**) we reported on OpenCat, a Raspberry Pi-based project by Chinese roboticist Rongzhong Li.

Fulfilling his dream to make the project self-supported, OpenCat has been renamed Nybble and Li has taken it to Indiegogo (**magpi.cc/FtXeeb**).

"Nybble is the lightest and fastest robotic cat that really walks," says Li. It's "not only a robotic cat that walks," he explains, "but a new type of pet."

The robotic cat has had a complete design makeover, and a wooden frame of interlocking parts replaces the 3D-printed parts of the original. "The wisdom of traditional Chinese woodwork was borrowed to make the major frame screw-free," writes Li.

▲ The wooden frame is designed to be assembled without using screws

An Arduino-compatible controller called NyBoard V0 is used to handle the motion of Nybble's legs. NyBoard sits on top of a Raspberry Pi, which controls higher-level functionality, such as "perception and decision-making," says Li.

The OpenCat software is being developed with community support, and you can take a look at it on GitHub: **magpi.cc/rThMwm**.

Serious cat

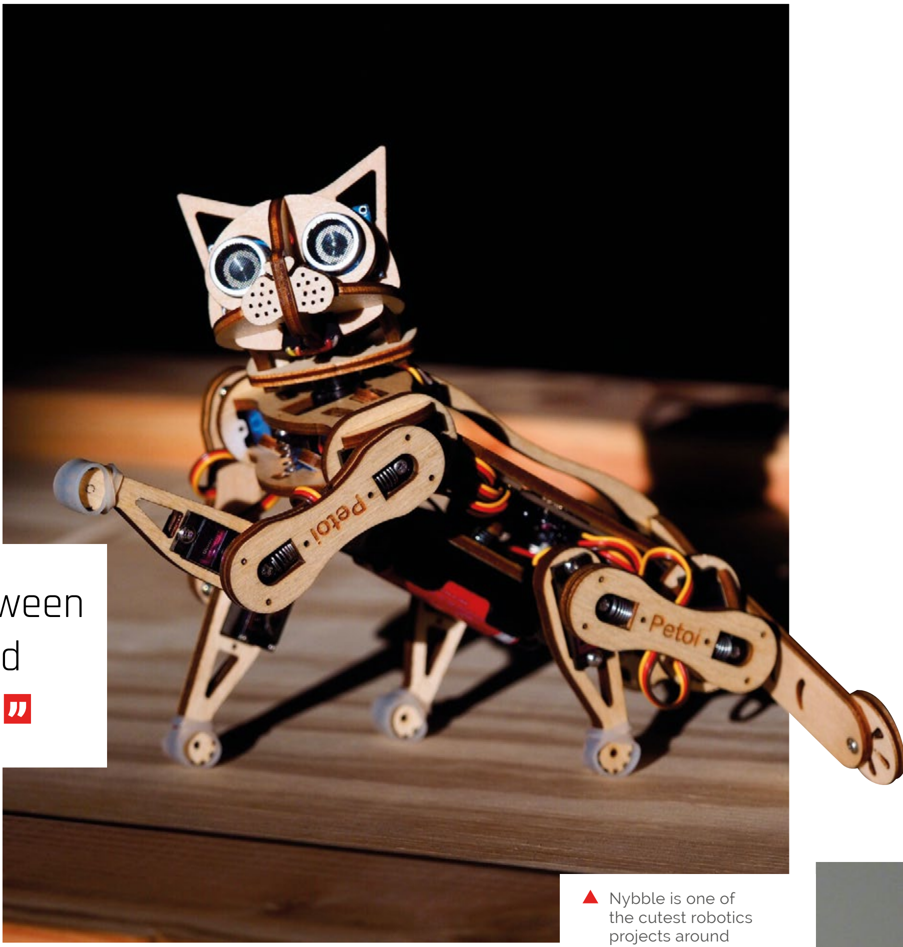
Don't be fooled by the project's cute appearance: Nybble aims to grab the attention of a serious robotics audience. Getting Nybble to walk may not be as straightforward as many users imagine, and with eleven servos it should be capable of complex movement. Nybble is adorable, but it aims to be a "precise instrument". Users will need to calibrate the limbs, for example.

> "We can break the barrier between those million-dollar quadruped robots and individual makers"

Li draws comparison between Nybble and high-profile robotics projects like those from Boston Dynamics (who build terrifying quadrupeds such as Spot and SpotMini).

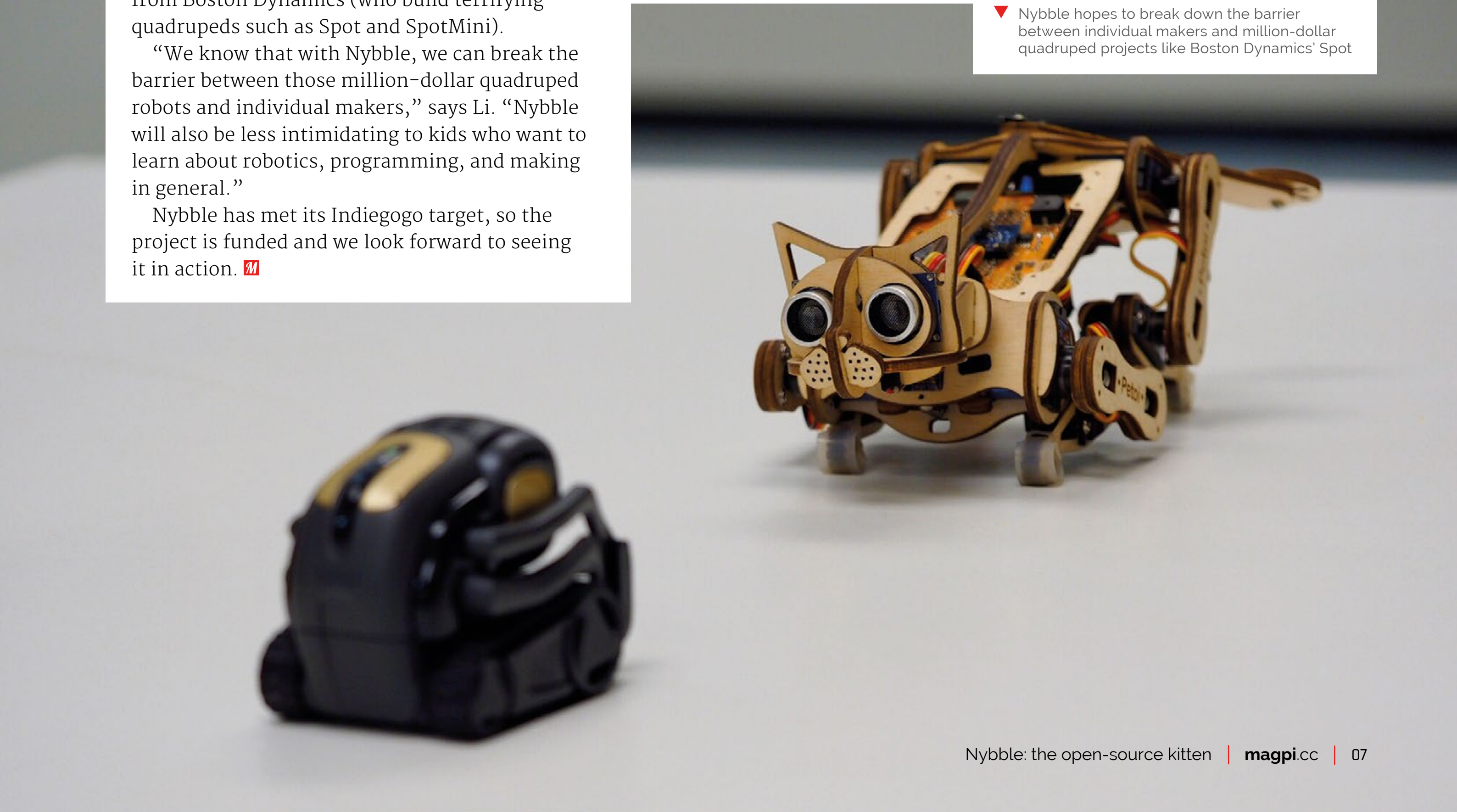
"We know that with Nybble, we can break the barrier between those million-dollar quadruped robots and individual makers," says Li. "Nybble will also be less intimidating to kids who want to learn about robotics, programming, and making in general."

Nybble has met its Indiegogo target, so the project is funded and we look forward to seeing it in action.



▲ Nybble is one of the cutest robotics projects around

▼ Nybble hopes to break down the barrier between individual makers and million-dollar quadruped projects like Boston Dynamics' Spot



Teachers will get resources, training, research, and certification as part of the programme

Raspberry Pi in £78m computing education boost

Dedicated funding for Raspberry Pi to transform UK computing education. By **Rosie Hattersley**

Computing education in England is about to get a much-needed jolt of funding with the help of the Raspberry Pi Foundation.

The welcome cash injection comes a year after The Royal Society reported that there was a 'once-in-a-generation opportunity' to transform the way computing is taught in schools and colleges. Commenting on the report, Raspberry Pi Foundation CEO Philip Colligan noted that "there's a long way to go before we can say that young people are consistently getting the computing education they need and deserve in UK schools."

> "A once-in-a-generation opportunity to transform the way computing is taught in schools and colleges"

The Raspberry Pi Foundation is one of the organisations that, jointly, have secured £78 million in UK government funding to make this vision a reality.

The Foundation is part of a consortium that also includes STEM Learning and the BCS (British Computer Society). Google has also pledged £1 million to support free online computing and computer science courses accessible to anyone.

While existing computing and ICT (information and communications technology) teachers are being directly targeted, the scheme will also upskill existing teachers in other disciplines to teach GCSE Computer Science.

Philip explains that the money will be used "to make sure every child in every school in England has access to a world-leading computing education."

▲ A Royal Society report last November drew attention to the scale of the challenge in transforming the way we teach computer science in the UK

The consortium will found a new National Centre for Computing – with a network of computing hubs where existing primary and secondary school computing teachers in England will be able to take part in fully funded CPD (continuing professional development) courses. Teachers will also have access to free resources enabling them to teach computing to students from Key Stage 1 right up to A-Level.

As part of an all-hands-on-deck approach to overhauling computing teaching in England, the Raspberry Pi Foundation and its consortium have more than 60 organisations signed up to offer practical assistance and expertise. Businesses, universities, and non-profit organisations are pooling their expertise and resources to provide the support that educators and schools require.

You can learn more, and get involved, at **teachcomputing.org**.

Raspbian updated

New version of Raspberry Pi OS features VLC and Thonny version 3. By **Lucy Hattersley**

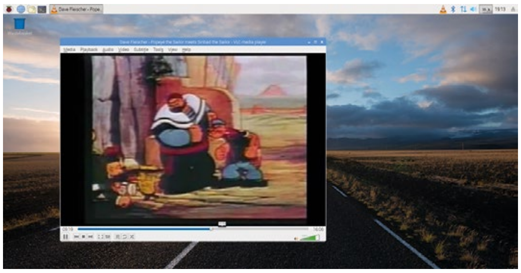
Raspberry Pi has updated Raspbian alongside its launch of the new Pi 3A+ computer. The latest update (2018-11-13) introduces major new features, and a host of under-the-hood tweaks.

A new hardware-accelerated version of VLC media player enables media playback on the Raspberry Pi. It's been a "glaring omission" according to Simon Long, the Software Engineer at Raspberry Pi responsible for Raspbian.

"Windows has Windows Media Player, macOS has QuickTime Player and iTunes, but we've had a big hole where something similar ought to be for Raspbian," writes Simon on the Raspberry Pi blog (**magpi.cc/bavZxx**).

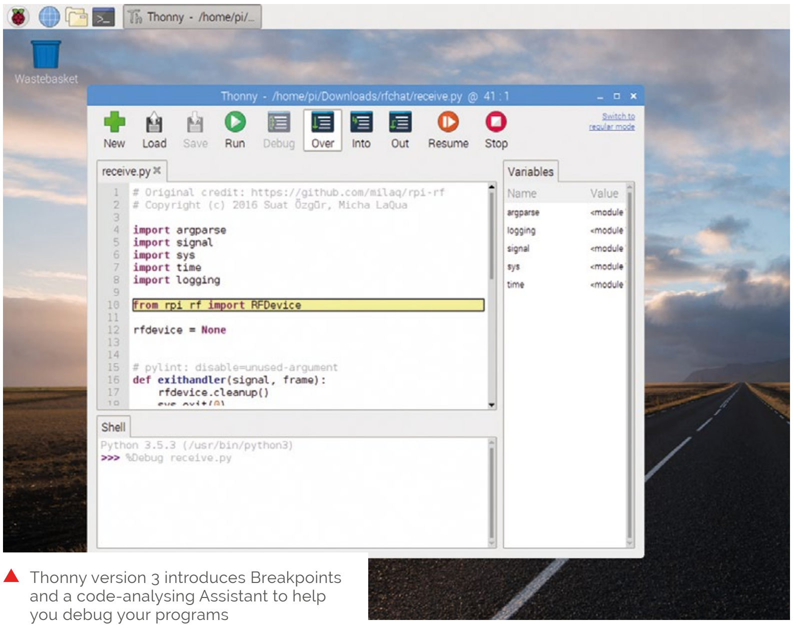
> " Major new features, and a host of under-the-hood tweaks "

VLC is a powerful piece of software that can play (or stream) virtually any video or audio file. "It uses software codecs for many video formats," reports Simon, and "VideoCore's video engine to accelerate playback of H.264, MPEG-2, and VC-1 video." The software is free, but users will need to buy additional licences to play MPEG and VC-1 video.



▲ A hardware-accelerated version of VLC media player enables powerful media playback on Raspberry Pi

Thonny, the standard Python IDE (integrated development environment) included with Raspbian, has been updated to version 3 (**thonny.org**). The new edition introduces welcome features such as Breakpoints and a code-analysing Assistant.



```
# Original credit: https://github.com/milaq/rpi-rf
# Copyright (c) 2016 Suat Özgür, Micha LaQua

import argparse
import signal
import sys
import time
import logging

from rpi rf import RFDevice

rfdevice = None


# pylint: disable=unused-argument
def exithandler(signal, frame):
    rfdevice.cleanup()
    sys.exit(0)
```

```
Name       Value
argparse   <module
logging    <module
signal     <module
sys        <module
time       <module
```

```
Python 3.5.3 (/usr/bin/python3)
>>> %Debug receive.py
```

▲ Thonny version 3 introduces Breakpoints and a code-analysing Assistant to help you debug your programs

Recommended software

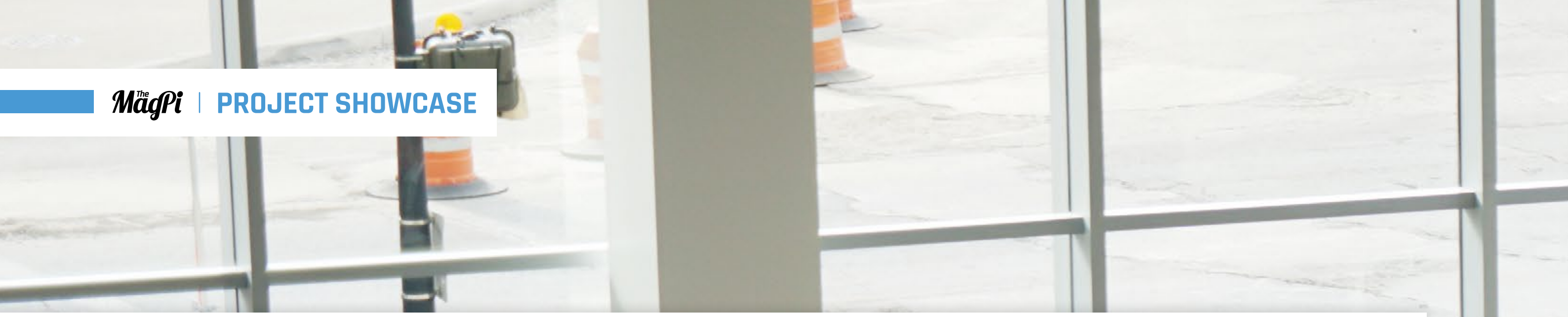
Raspberry Pi has chosen to create three OS images this time: Raspbian Stretch with Desktop and Recommended Software (also known as 'Raspbian Full'), Raspbian Stretch with Desktop, and Raspbian Stretch Lite (which does not feature the desktop interface).

Raspbian Full includes familiar extra programs such as LibreOffice, Scratch, Sonic Pi, Thonny, and Mathematica. NOOBS will feature Raspbian Full as the default installation. For information on using NOOBS to set up a Raspberry Pi, take a look at our QuickStart guide (**magpi.cc/quickstart**).

Both images are available for download from the Raspberry Pi website (**rpf.io/downloads**).

To update an existing Raspbian image, open a Terminal window and enter:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install vlc
```



As We Are

Ever wanted your face to be 14-foot high? Now you can with this sculpture covered in LEDs and powered by many, many Pi boards. **Rob Zwetsloot** tries to get his head around it

MAKER

Matthew Mohr & Mac Pierce

Matthew is an artist who came up with the idea for As We Are, and Mac Pierce is the creative technologist that helped make it a reality.

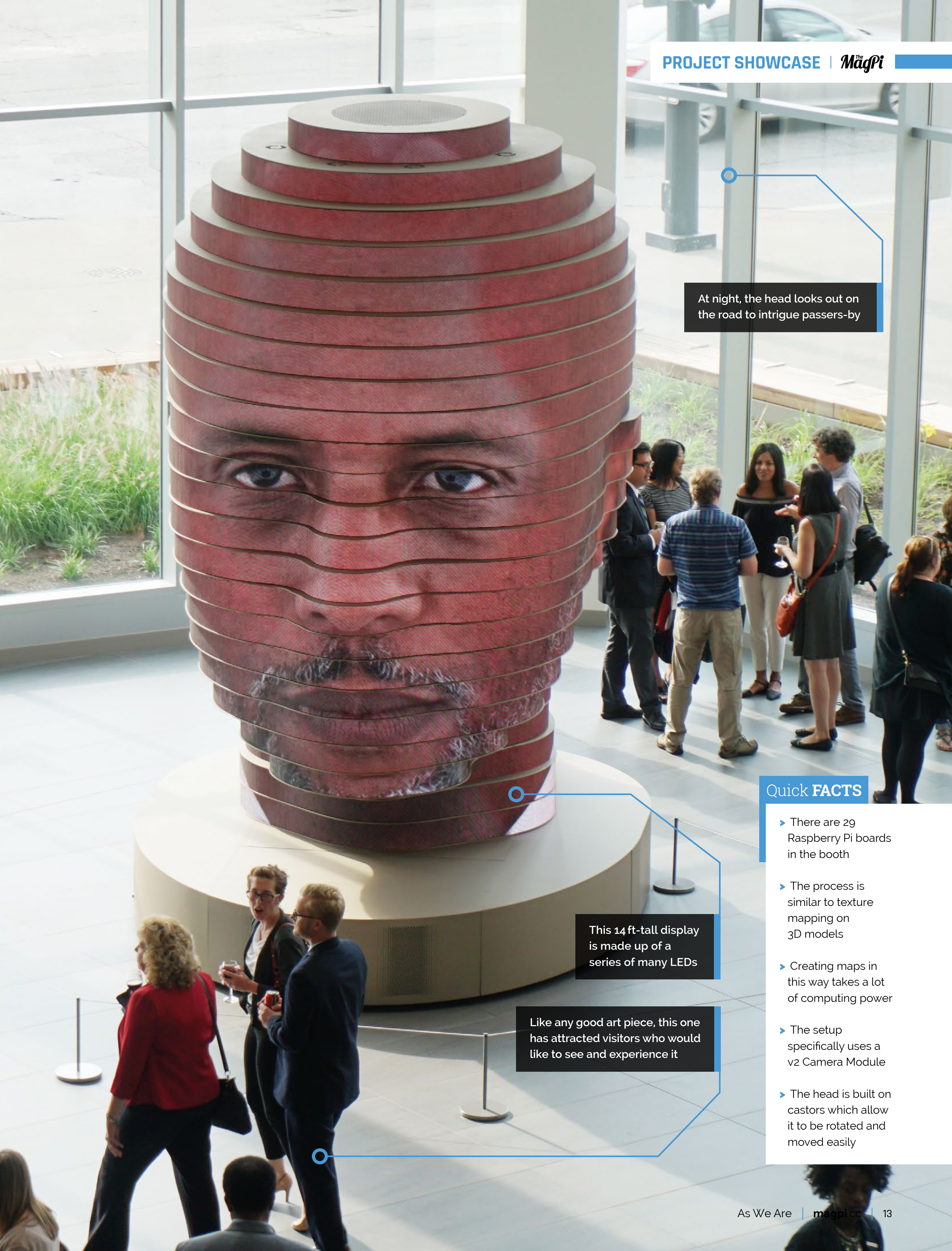
magpi.cc/UjTkbW

Imagine walking through the doors of a convention centre and seeing someone's face staring down at you. Not just in statue form, or even on a big TV either – no, this is a 14 ft (4.3 m) high sculpture of a head, covered in LEDs that are hooked up to a load of Raspberry Pi boards to display a face. That must be somewhat disconcerting.

"The idea was concepted and artistically directed by Matthew Mohr," Mac Pierce, the production manager of the piece, tells us. "It was commissioned by the Greater Columbus Convention Center (**columbusconventions.com**) as the centrepiece of their new collection of local art."

It's certainly striking. One feature that is easy to miss is the Pi-powered photo booth that is inside the head, allowing people to get their face scanned and the images 3D-mapped to the screens. It's not just showing off famous people, or even Matthew the creator, but instead folks around you, or yourself. Mac opted to use Raspberry Pi boards in the booth after researching the standard solutions.

"Looking at network-attachable cameras, none of the versions we found would work for our application," says Mac. "Either they were scientific or industrial inspection cameras that required complex lensing and large mounting requirements – not to mention being prohibitively expensive – or they had impenetrable APIs and closed software that would limit their usefulness. The Pi 3 and Camera Module combo ended up fitting nicely for our application, as the camera could be mounted separate from the main body of the Pi, which allowed the overall size of the booth to have a lot less wasted space. The Pis could also be easily attached to the network, and were open so that custom programs could be written on it to handle the photo processing and network handling."



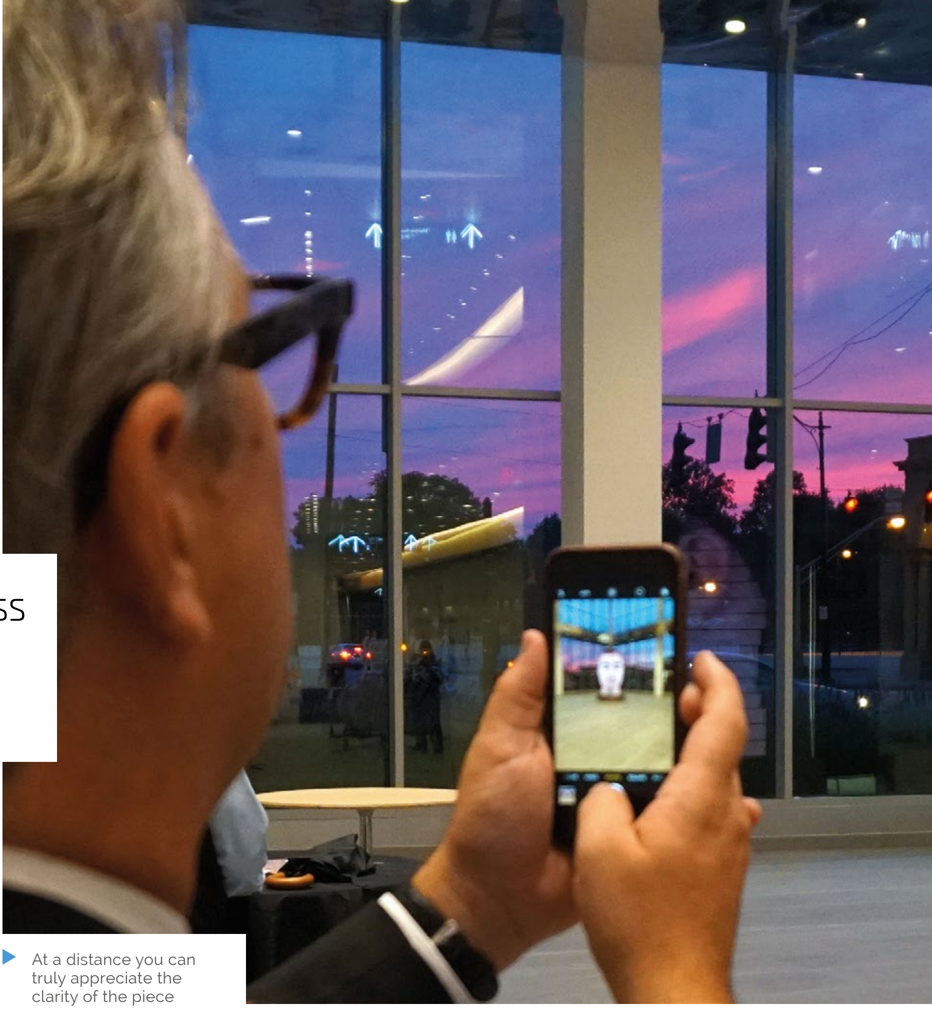
At night, the head looks out on the road to intrigue passers-by

This 14 ft-tall display is made up of a series of many LEDs

Like any good art piece, this one has attracted visitors who would like to see and experience it

Quick FACTS

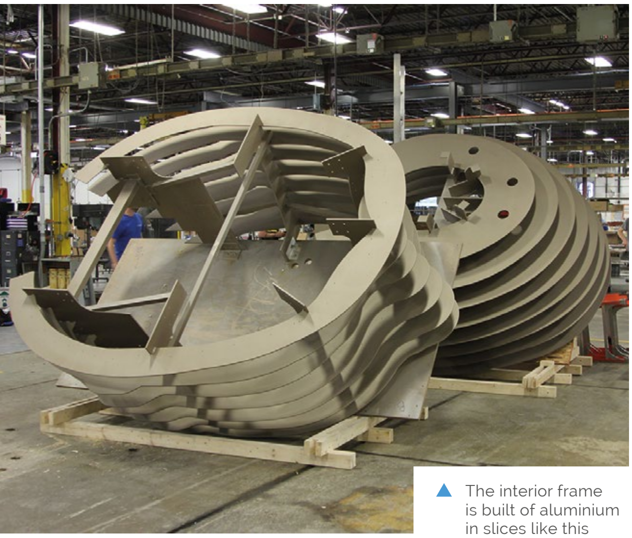
- There are 29 Raspberry Pi boards in the booth
- The process is similar to texture mapping on 3D models
- Creating maps in this way takes a lot of computing power
- The setup specifically uses a v2 Camera Module
- The head is built on castors which allow it to be rotated and moved easily



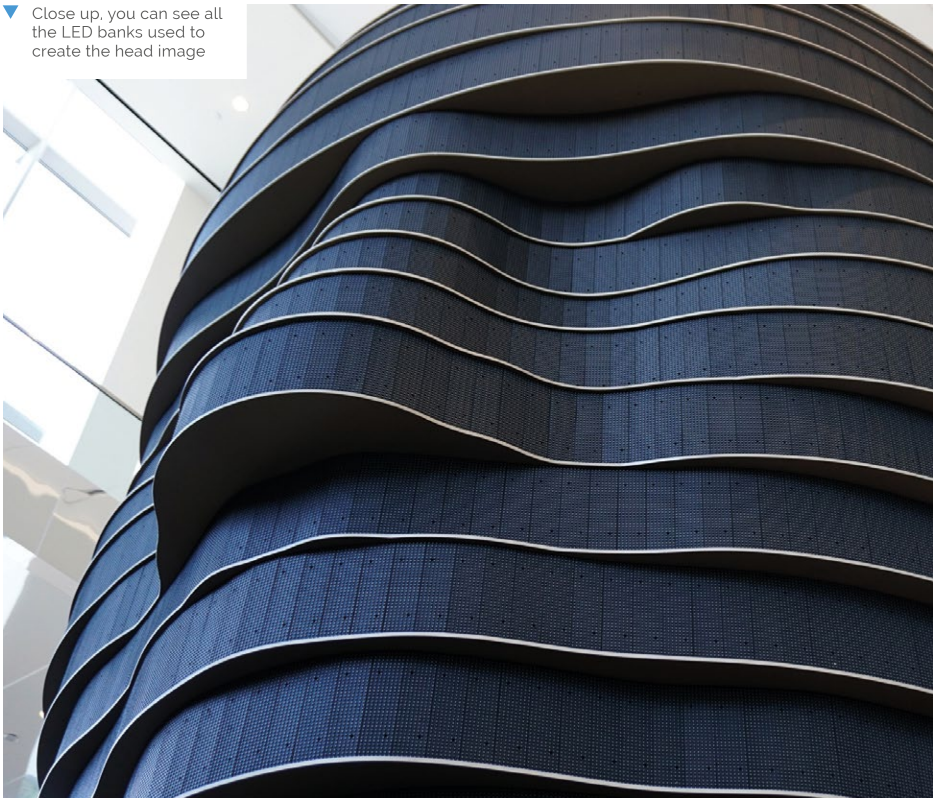
Mac explains, “The overall sculpture itself is built around an aluminium frame that gives the piece structure and [enables] mounting of the LEDs. The LEDs themselves were custom-manufactured by Sansi North America (**snadisplays.com**) to be able to form the curved layers of the head with only minimal faceting. The autonomous system of the scanning is done through a series of large processing servers, a half-rack of which exists in the top of the head, and another full rack that operates remotely.”

> “One feature that is easy to miss is the Pi-powered photo booth that is inside the head”

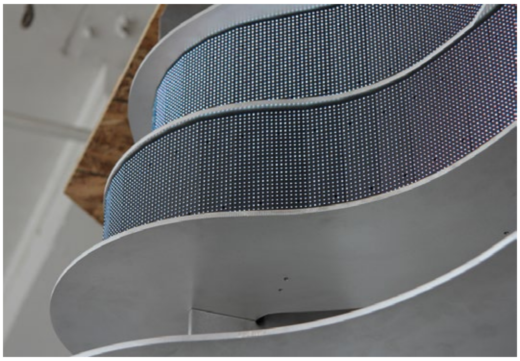
At a distance you can truly appreciate the clarity of the piece



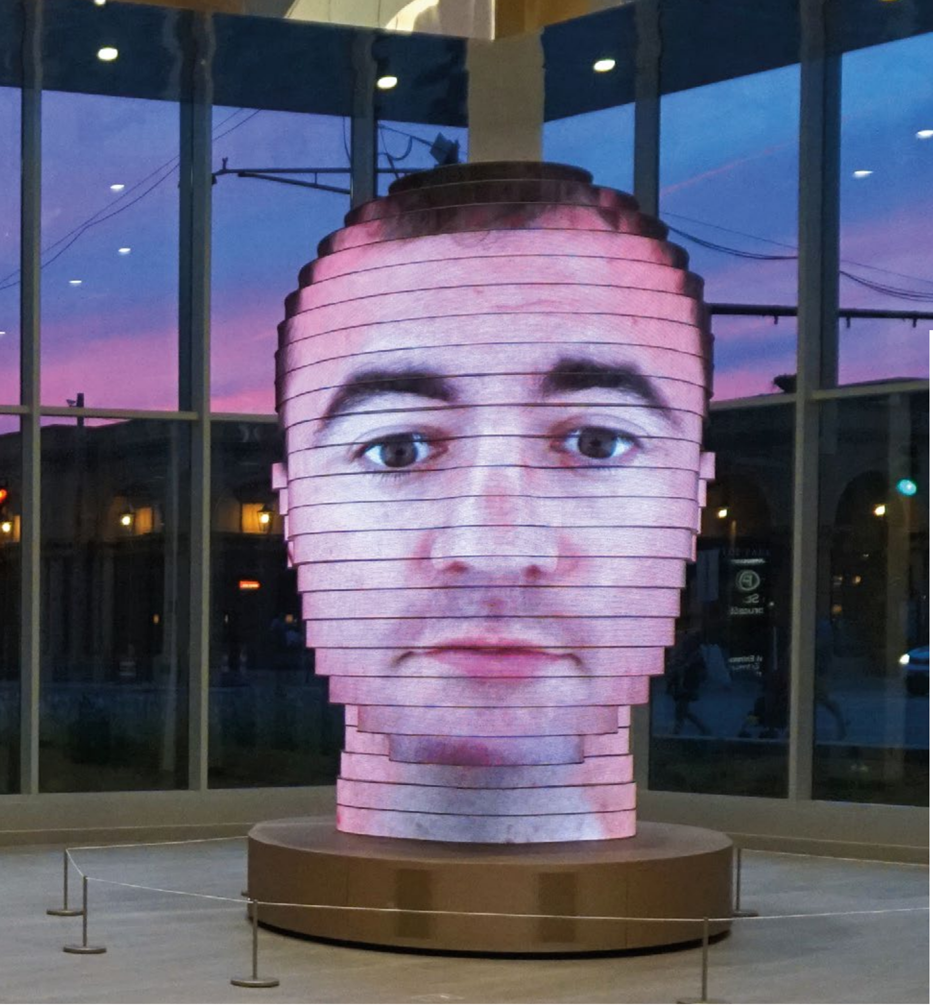
The interior frame is built of aluminium in slices like this



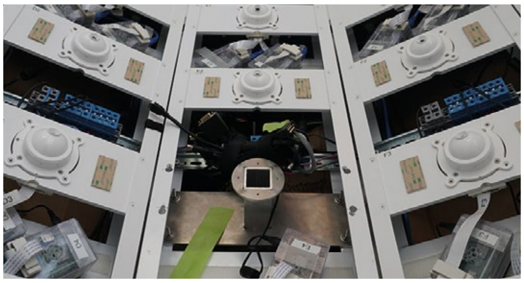
Close up, you can see all the LED banks used to create the head image



These are individual LEDs, not just screens



It takes a lot of Raspberry Pi boards to capture enough images for the project



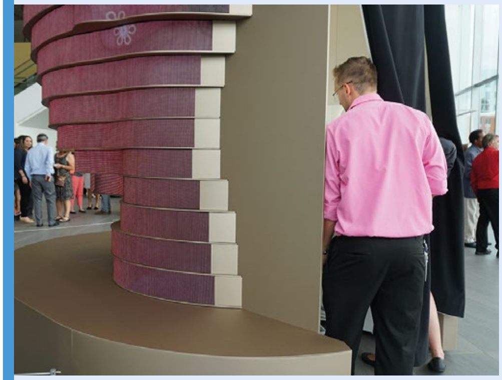
The photo booth part operates via a touchscreen which guides the users on where to look and how to place their head for an optimal result. The Raspberry Pi boards then take the photos required by the servers.

See your reaction

The project has received a fair bit of attention, with one person calling it the ‘Ultimate Selfie Machine’, and it continues to operate in the Columbus Convention Center, where it remains open to the public. It even turns around to look out at the street at night.

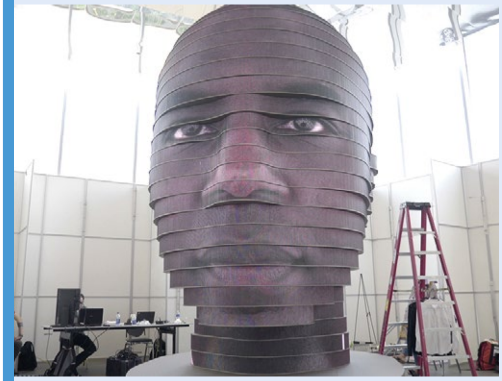
As for future projects, projection mapping has taken Mac’s interest: “I’m right now looking at a few different versions of projection-mapping software that runs on a Pi, which would make it perfect for permanently installed projection pieces. Also, there are a wealth of applications for the Pi in digital signage, of which I’m hoping to do some projects with soon.”

Your face on the big screen



01 While you may be distracted by the head sculpture itself, make sure to walk around behind it to find a curtain. Walk through it to find where the magic happens.

02 Follow the on-screen instructions to take the perfect photo (or 30). These pictures are then sent to be processed by a server above you, and a server elsewhere.



03 Head back outside and you’ll soon see your face looking back at you. Now’s a perfect time to take a selfie of yourself with yourself.

Working Model of the Trinity Buoy Wharf Lighthouse

Building ten working models of a lighthouse for your wedding day is a hugely impressive feat – **Nicola King** shines a light on an ambitious project

MAKER

Dave

Dave has a lot of Pi-related experience, including attaching a Pi to his Roomba so he could record time-lapse videos of it hoovering the flat.

magpi.cc/BZVvgm

When Dave knew that he would be getting married in a unique London lighthouse, he decided to make some very original wedding table centrepieces: working models of the very lighthouse in which the nuptials were taking place. Not only that, but he resolved to make ten of them – quite an undertaking. As Dave explains, "With my love of 3D printing, Raspberry Pis, and needlessly complicated projects, recreating the lighthouse as a working model seemed like the way to go."

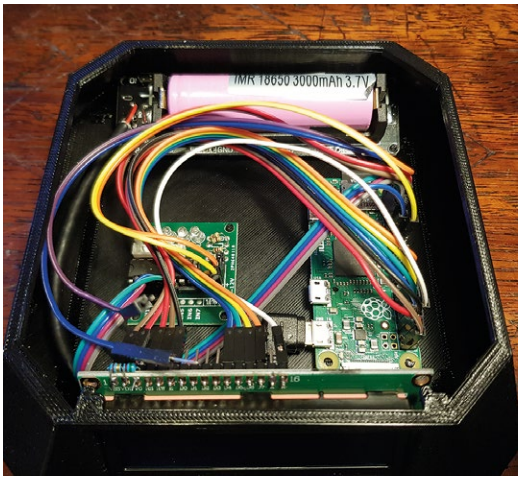
Buoy oh buoy!

Over a three- to four-month period, before his August 2018 wedding, Dave designed, 3D-printed, and completed the models with, unsurprisingly, "the last month being pretty non-stop."

A time-consuming project, modelling the lighthouse necessitated a lot of work, and Dave used many photographs of the real-life lighthouse to guide his progress. In addition, as he explains, "The software I used on the Pi was pretty simple, although I tweaked it a few times. The two most challenging parts were fitting in all the printing time, and the mechanism for the rotating light. I went through a couple of major versions of that, a lot of minor versions, and still never got it 100% reliable – turns out rotating and maintaining a circuit is quite hard!"

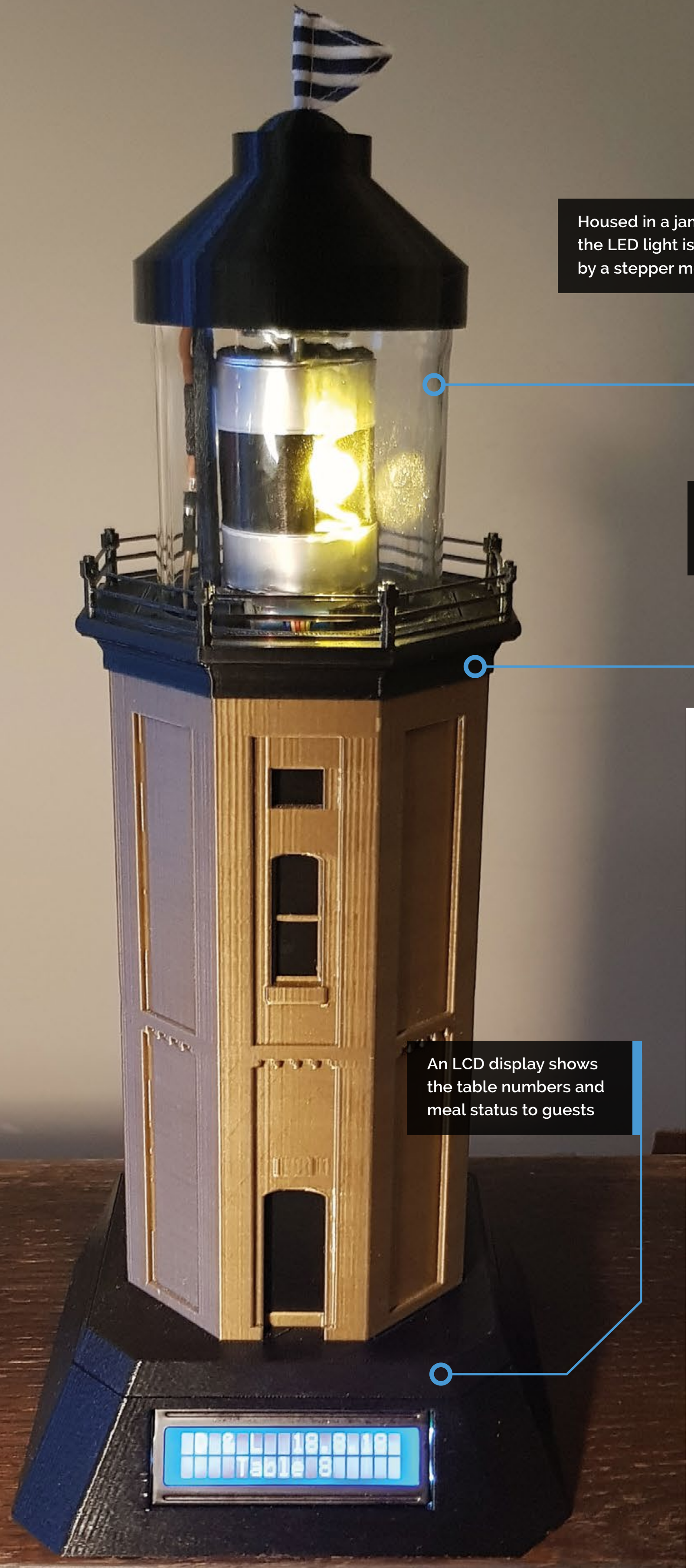
The 3D printing time was indeed considerable, as each model required 44 hours of printing in order to produce the six key pieces: outer, inner, railing, cap, base, and base lid.

In terms of function, the models include an 18650 battery (in a shield), a Raspberry Pi, and an LCD display in the base. At the top is a warm white LED light on



Concealed in the base are a Pi Zero, battery, motor board, and LCD; the wiring proved challenging

A production line of lighthouses, ready for the big day



Housed in a jam-jar, the LED light is rotated by a stepper motor

The body and five other parts took 44 hours to 3D-print for each model

An LCD display shows the table numbers and meal status to guests

Quick FACTS

- Trinity Buoy Wharf is the site of London's only lighthouse
- The lighthouse model was designed in Autodesk 123D
- Dave used his Prusa MK2.5 to 3D-print components
- A jam-jar is used for the glass light casing
- Dave gave most of the models to wedding guests – one is now in Australia

top of a stepper motor. "Wires run up the middle and when you turn on the battery, it powers on the light and the Pi," reveals Dave. "The Pi then starts turning the motor (and light) and displaying various messages on the display."

Not plain sailing

Understandably, the project was far from straightforward. Dave tells us that he needed to tweak the wiring, and there were a number of adjustments to the lights. "I ended up with a factory line of Raspberry Pis updating themselves and then updating their host names on the network when they were done – so I could drop onto each one the startup script, the display script, and the motor script. The display script used a Python scheduler and pulled the bespoke messages from a JSON file, so I could use the same code and then drop the right messages file on each one."

Dave's impressive model-making accomplishment was such an ingenious idea, and something that was, ultimately, very well received by his wedding guests, as he tells us: "I didn't have much trouble giving them away, which was great … one is [now] in the wardroom of a navy minesweeper."

Dave clearly has a great talent for model-making, but tells us that his next model project may be something slightly less time-consuming, "I'll no doubt make some other things, but I don't know what yet. I'm looking forward to making only one of something."

MAKER

Sarab Sethi

Sarab is a PhD student at Imperial College London working across the Applied Mathematics, Design Engineering, and Tropical Forest Ecology departments.

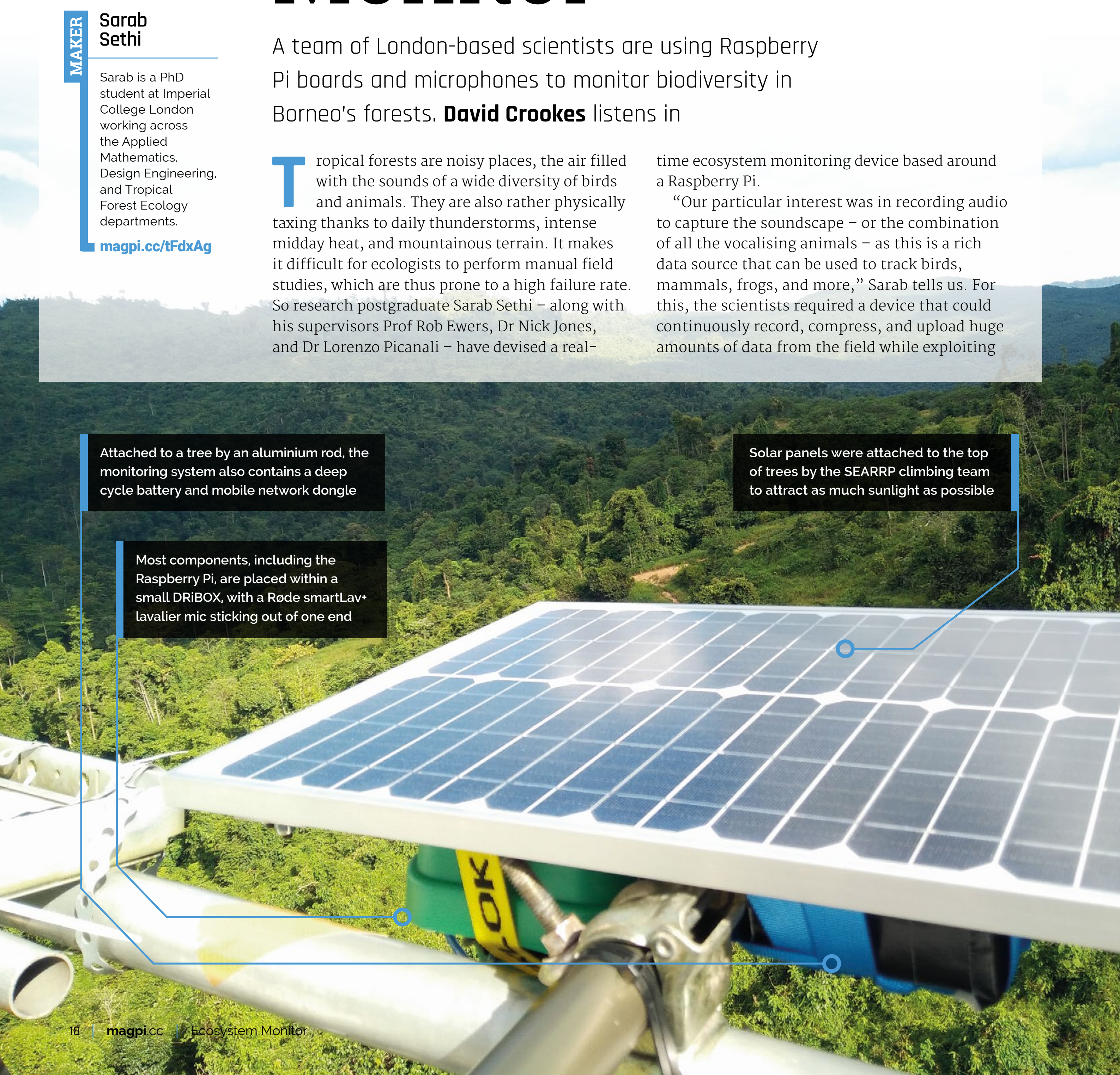
magpi.cc/tFdxAg

Ecosystem Monitor

A team of London-based scientists are using Raspberry Pi boards and microphones to monitor biodiversity in Borneo's forests. **David Crookes** listens in

Tropical forests are noisy places, the air filled with the sounds of a wide diversity of birds and animals. They are also rather physically taxing thanks to daily thunderstorms, intense midday heat, and mountainous terrain. It makes it difficult for ecologists to perform manual field studies, which are thus prone to a high failure rate. So research postgraduate Sarab Sethi – along with his supervisors Prof Rob Ewers, Dr Nick Jones, and Dr Lorenzo Picanali – have devised a real-time ecosystem monitoring device based around a Raspberry Pi.

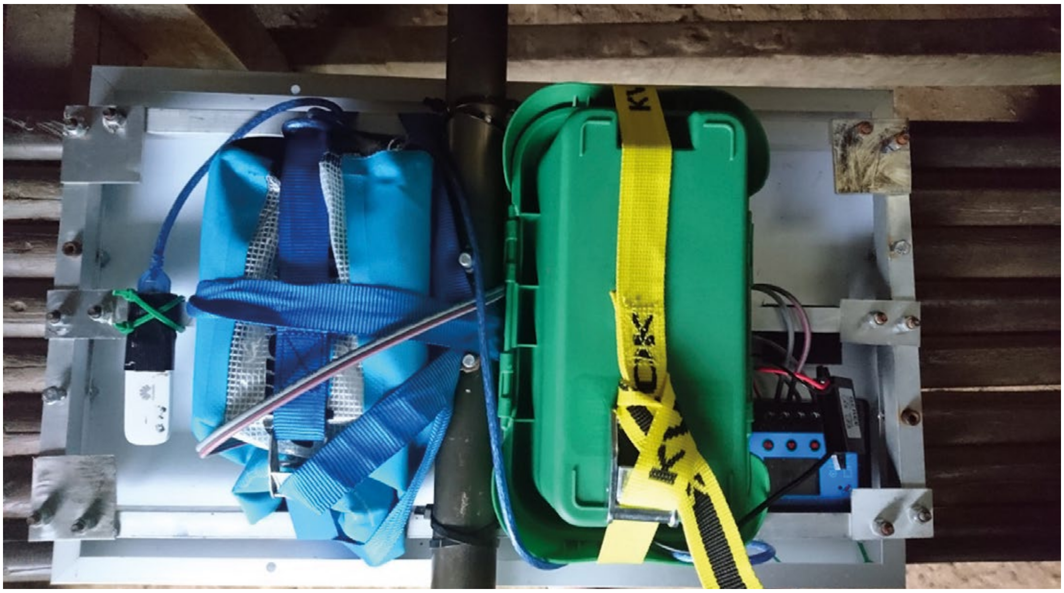
"Our particular interest was in recording audio to capture the soundscape – or the combination of all the vocalising animals – as this is a rich data source that can be used to track birds, mammals, frogs, and more," Sarab tells us. For this, the scientists required a device that could continuously record, compress, and upload huge amounts of data from the field while exploiting



Attached to a tree by an aluminium rod, the monitoring system also contains a deep cycle battery and mobile network dongle

Solar panels were attached to the top of trees by the SEARRP climbing team to attract as much sunlight as possible

Most components, including the Raspberry Pi, are placed within a small DRiBOX, with a Røde smartLav+ lavalier mic sticking out of one end



This is the view from underneath the solar panel, with everything firmly strapped or taped into place

a patchy mobile signal to remotely transfer data to a server. “The Raspberry Pi was ideal as a low-cost, relatively low-power device with a usable amount of computing power and large support for a wide range of sensors,” says Sarab.

Jungle sounds

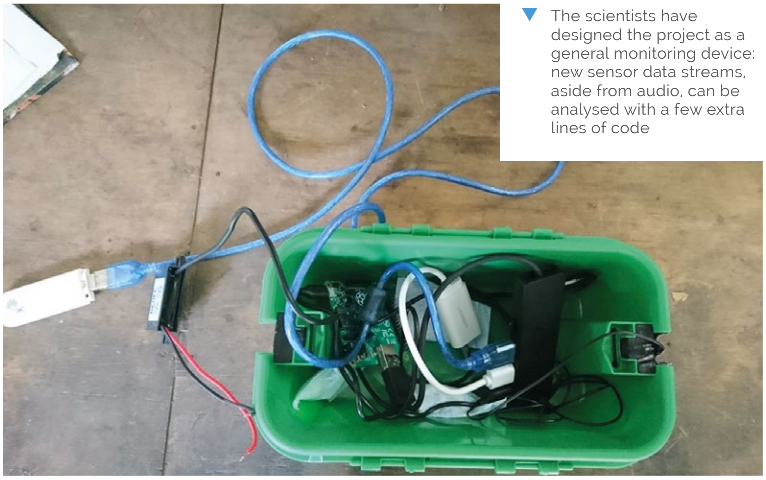
With their field site in Sabah, Borneo, in mind, they set about creating a system that could monitor the effects of oil palm plantations and logging on the region’s biodiversity by listening out for the sounds of animals. It involved using a Røde smartLav+ microphone to provide high-quality audio recordings, along with an external USB audio card, solar energy, and a 3G dongle to connect to the internet. The ultimate aim is to use artificial intelligence to pick up on the audio and make sense of the data.

For now, however, the Python-programmed software runs two threads concurrently: one continuously records data from a sensor and stores it in uncompressed files, and the second compresses this data and robustly uploads it using FTP to a remote server. “It is important that the device is networked to minimise the amount of times a scientist or research assistant has to go to visit the device to manually collect the data – freeing up time to be better spent on other more efficient and less exhausting tasks,” explains Sarab. “Large animals also love to play with (or more likely destroy) any equipment left in tropical forests. Continuously uploading data serves as an instant backup system.”

Testing times

The system is being tested at the Stability for Altered Forest Ecosystems (SAFE) project in Borneo where it’s been up-and-running since February 2018. It currently uses a network of twelve acoustic monitoring devices which are placed across a gradient of logging intensity, from old-growth untouched forest to oil palm plantations, allowing animals to be tracked in specific environments.

“Our major headache has been the solar power system, as the quality and range of batteries available in this region of Malaysia is generally pretty poor and importing batteries across borders is difficult,” Sarab says. “However, the monitoring device itself has fared surprisingly well, especially considering the near 100 percent humidity and regular movement due to shaking of trees, animal interference, etc. To date, we have collected over 15 000 hours of audio using these devices, and more is coming in each day.”



The scientists have designed the project as a general monitoring device: new sensor data streams, aside from audio, can be analysed with a few extra lines of code

Quick FACTS

- The overall system costs £230 to make
- 64GB can store a month of animal sounds
- The Pi boards are powered by 20 to 30 W solar panels
- They’re exposed to temperatures of 2 to 31.5°C
- They withstand 614 mm of precipitation per month

Project BoxBoom

This modern take on a retro boombox is sure to (boom) shake the room, as **David Crookes** discovers

MAKER

Dane Hermse & Nicole Horward

Dane Hermse and Nicole Horward spend much of their spare time making fun items, from a machine that automates goodbyes to an IoT hamster wheel.

magpi.cc/LwDONF

There were two main methods of listening to music back in the 1980s: privately through a pair of headphones and a Walkman, or at full blast through a portable music player so that other people could hear, whether they wanted to or not. With Project BoxBoom, Pi enthusiasts and makers Dane Hermse and Nicole Horward hope to evoke memories of the latter. "When we found a boombox in a second-hand store, it instantly made me happy," Dane tells us. "It reminded me of *The Fresh Prince of Bel-Air* and a colourful decade."

The pair had wanted to revamp old equipment for some time and they soon got to work on their new purchase. "The broad strokes of the build were already in my head – remove the broken parts, add a screen, hook up the audio, run some sort of music app, and add a power source," Dane says. Hoping to make use of the old knobs and buttons and have them work with a Raspberry Pi while retaining the old look and feel of the original player, he envisioned that a touchscreen would replace the cassette deck.

Sounds good

After opening the old boombox, the pair first checked which components still worked and they looked for parts that could be reused. "We found everything was broken except for the speakers, so I went looking for the parts we needed," says Dane. "This meant measuring what screen would fit, how much power the amplifier would need, and how big the power bank could be. I was also keen to ensure the inside mechanics would keep working and the outside stayed nice."

> When we found a boombox in a second-hand store, it instantly made me happy

Aside from a Raspberry Pi 3 Model B, Dane ordered an Adafruit HDMI 5-inch 800×480 display backpack with a resistive touchscreen, along with an Adafruit TPA2012 2.1W stereo audio amplifier, and a 2.4A power bank. He cleaned up the hole where the cassette player used to be at the front of the boombox, and 3D-printed two frames to hold the screen.

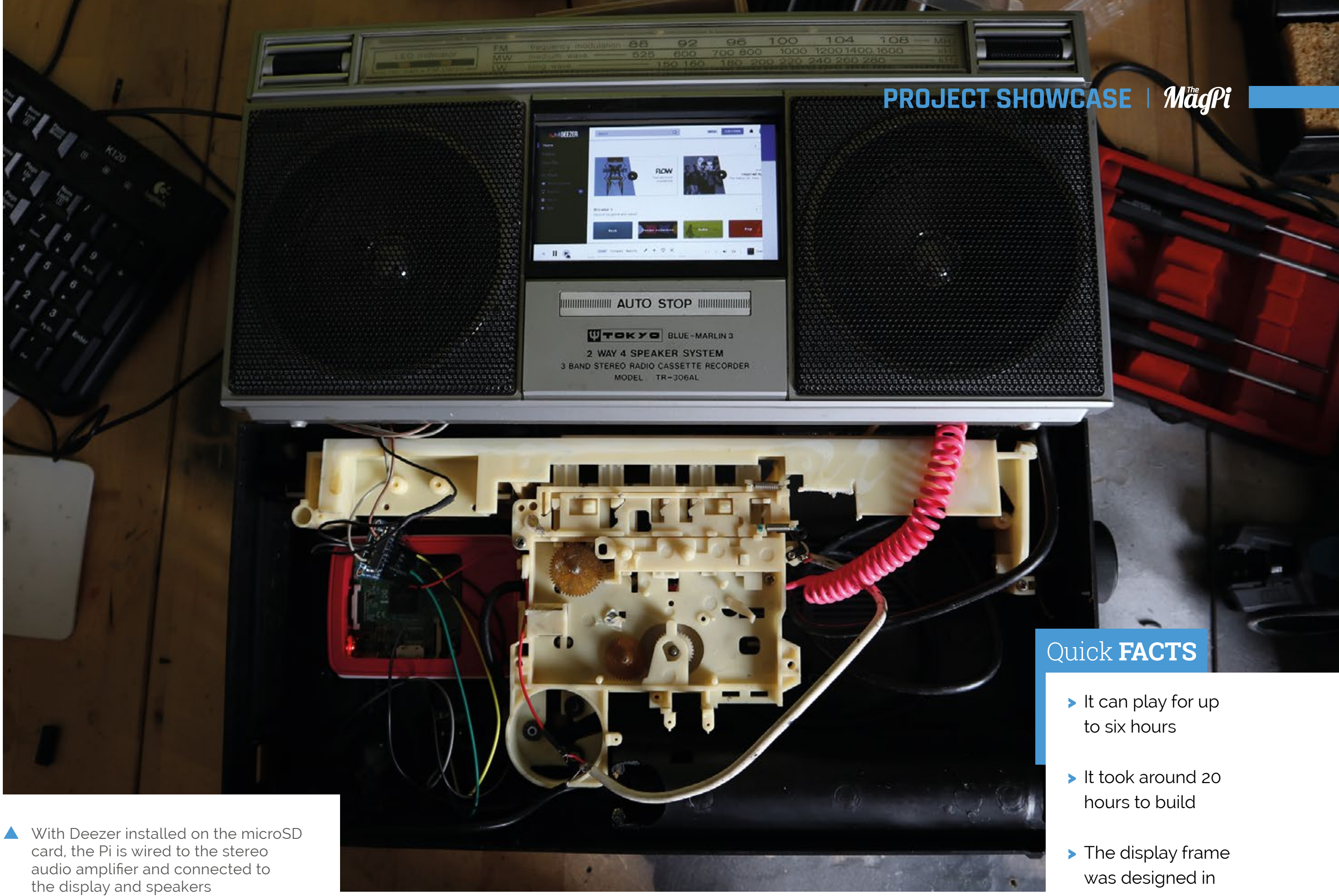
The screen is ready to be installed. Dane says the GPIO pins could be connected to control LEDs, too

"Next, I soldered the amplifier chip, connected the speakers to the amplifier, added the Raspberry Pi, and hooked up the wires," Dane explains. Here the Pi came into its own. "Its in-built internet connectivity, four USB ports, audio jack, and HDMI port were super-useful," he continues. "But then for every project we do, the Pi is a default: you can do anything with them."

Streaming ahead

Getting a web music player to co-operate was trickier. "Spotify didn't work on Chromium, which was something to do with Widevine digital rights management, but Deezer worked on the first try," Dane says, explaining that SoundCloud also worked. "Adding an on-screen keyboard extension to the browser meant the boombox was ready to be used."

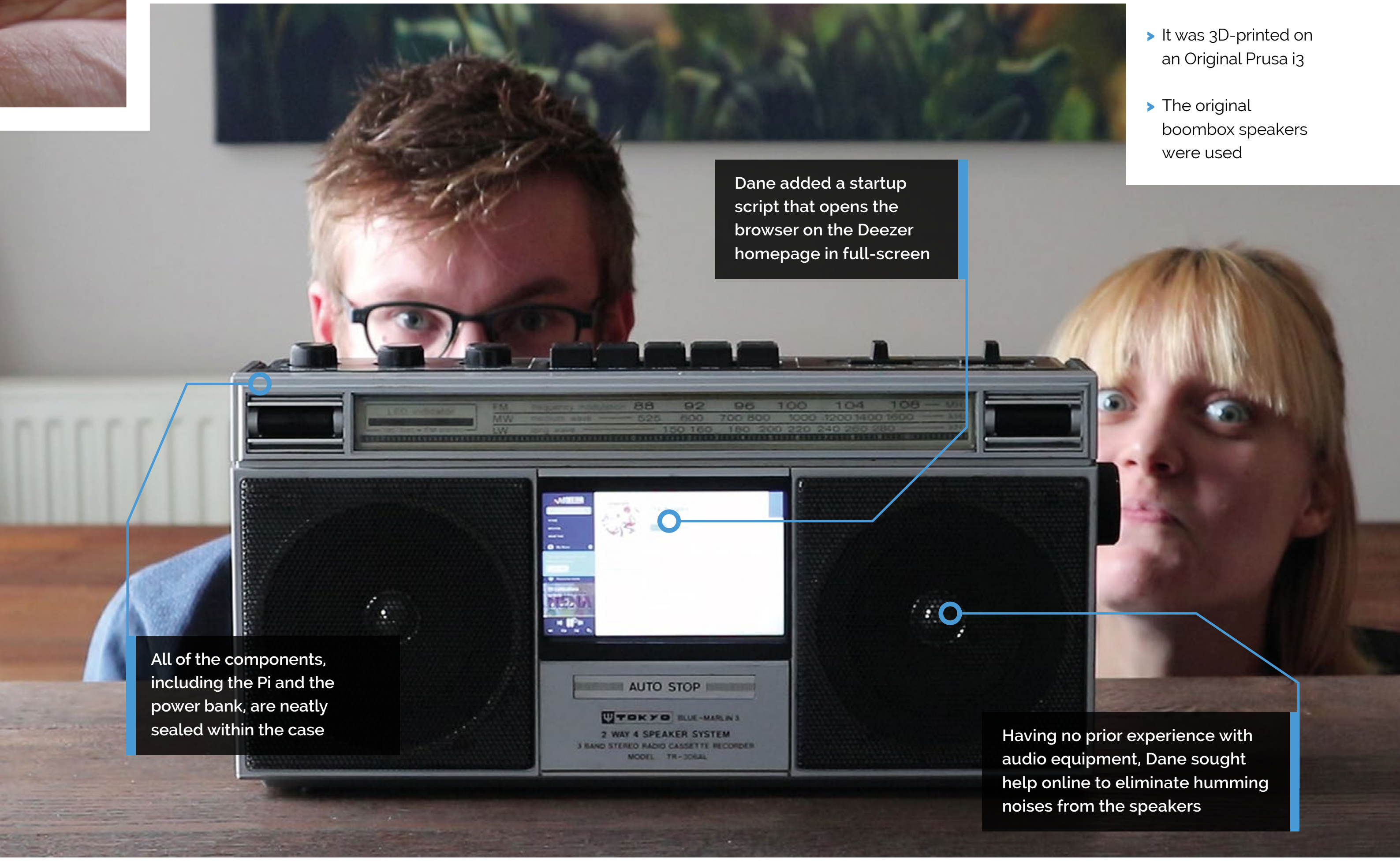
For the finishing touches, Dane converted the old antenna into a touchscreen stylus, painted the screen frames black, cleaned the entire build, and put the old screws back. "The most difficult part was to make it look nice," he muses. "But it's had very positive responses." Music to his (and everyone else's) ears, we'd suspect.



▲ With Deezer installed on the microSD card, the Pi is wired to the stereo audio amplifier and connected to the display and speakers

Quick FACTS

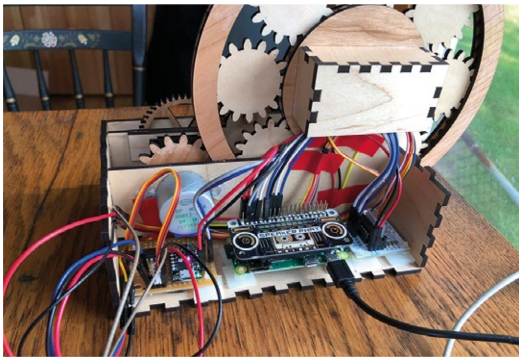
- It can play for up to six hours
- It took around 20 hours to build
- The display frame was designed in Tinkercad
- It was 3D-printed on an Original Prusa i3
- The original boombox speakers were used

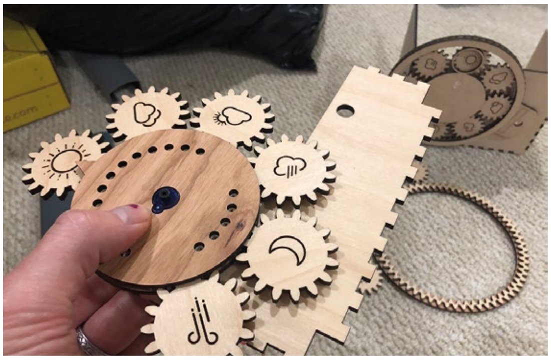


Dane added a startup script that opens the browser on the Deezer homepage in full-screen

All of the components, including the Pi and the power bank, are neatly sealed within the case

Having no prior experience with audio equipment, Dane sought help online to eliminate humming noises from the speakers





Assembling the laser-cut hub and planetary gears, which are rotated by a stepper motor

A Pi Zero runs the code, a Speaker pHAT provides chime sounds, while a Trinket M0 controls the NeoPixels

Tide Clock Weather Thing

This beautifully crafted device provides a detailed weather forecast and low tide times. **Phil King** is on cloud nine

MAKER

Fin Hopkins

A software engineer on the City of Boston's Digital Team, Fin is into board games, web development, and social justice.

magpi.cc/ZOSXTQ

Seeking to make a gift for the in-laws looking after the kids in Maine over the summer, Fin Hopkins decided to build a 'Tide Clock Weather Thing' to help predict the weather and tides for days out at the beach or kayaking.

"I remember [the reaction from the recipients] was something like 'wow, that's beautiful! What is it?' I had to point out what all the lights and dials were, since there aren't any markings on the case," recalls Fin.

> "I had to point out what all the lights and dials were, since there aren't any markings on the case"

How it works

The large wheel on the device shows the current weather conditions on top; as they change, it rotates planetary gears to bring a new icon to the top. In the middle of the wheel, a finger points to the current temperature, with the forecasted daily range lit by coloured LEDs. Five more LEDs below light up blue for impending rain, filling up to show when it's 60, 45, 30, 15, or 5 minutes away.

24 LEDs at the bottom of the device represent each hour of the day, lighting up in different colours for forecast weather conditions – including blue for rain, yellow for sunny, dim white for cloudy, and green for windy. Just above this strip, a moving bar with two pointers shows when the two daily low tides will occur.

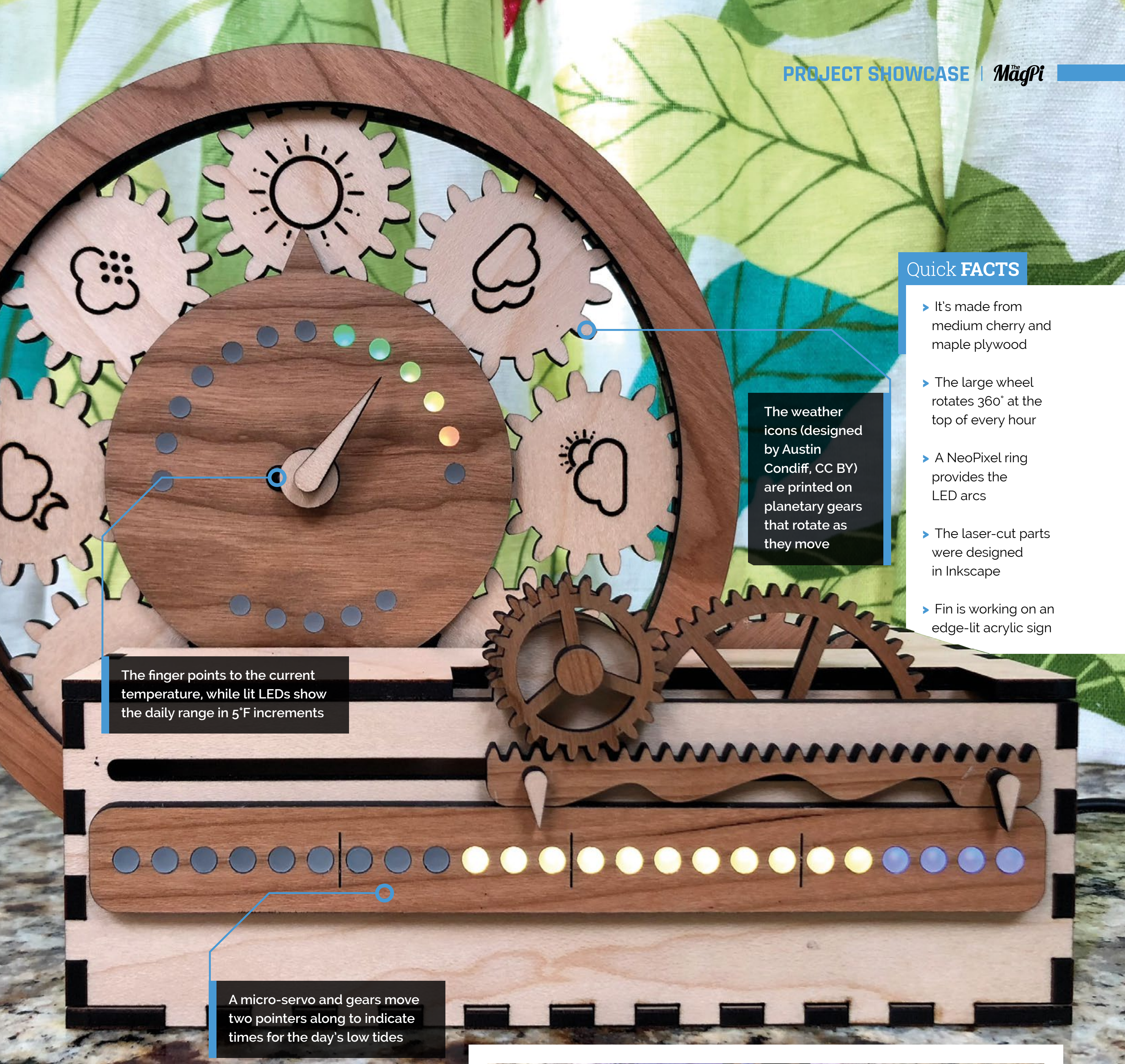
A chime is also sounded for low and high tides using a Speaker pHAT connected to a Pi Zero, which runs the Python software and controls the NeoPixel LEDs via a Trinket M0 microcontroller. All the weather data is sourced from the Dark Sky API, while tide data comes from the NOAA's Tides and Currents site.

Time to make

From the first gear prototypes to a working version of the device took Fin about three weeks, working nights and weekends. "We were going to visit my in-laws for the 4 July holiday, so it was a sprint to a tight deadline at the end. I got all the hardware and wiring done, and then ended up finishing the coding while I was up there."

The design adapted and fell into place as Fin prototyped, starting with the laser-cut planetary gears – for the weather symbols – which rotate around a central hub. "Once I saw how neat the large version [of the wheel] looked, it was 'OK, what else can I put with this?'"

A lot of the changes made concerned reducing the scope. "I wanted to show high and low tides over two days, with each indicator separately



The weather icons (designed by Austin Condiff, CC BY) are printed on planetary gears that rotate as they move

The finger points to the current temperature, while lit LEDs show the daily range in 5°F increments

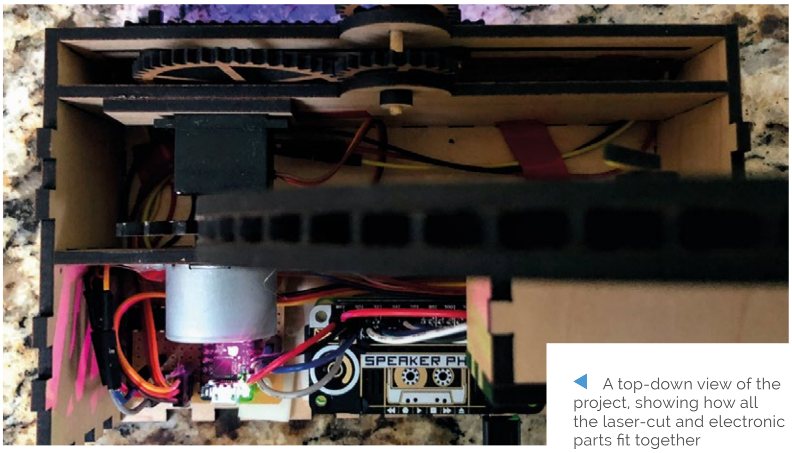
A micro-servo and gears move two pointers along to indicate times for the day's low tides

Quick FACTS

- It's made from medium cherry and maple plywood
- The large wheel rotates 360° at the top of every hour
- A NeoPixel ring provides the LED arcs
- The laser-cut parts were designed in Inkscape
- Fin is working on an edge-lit acrylic sign

controlled. Once I got into 'OK, now how would I build that?' it became one day, then not separately controlled, then just low tides," explains Fin. "Even after I simplified my goal for the tide indicator, it took a number of prototypes to get right."

Making their debut Raspberry Pi – and electronics – project was an educational process for Fin. "I learned how to solder, how to crimp wire, and I finally got my head around the difference between voltage and current. I learned about stepper motors, GPIO pins, I²C communication, PWM, and a ton of other topics."



A top-down view of the project, showing how all the laser-cut and electronic parts fit together

rCrumbl

Sometimes things don't always go to plan, but that's the way the cookie crumbles. **David Crookes** talks to Dylan Radcliffe about his attempt to create a 3G phone

MAKER

Dylan Radcliffe

As well as getting his hands dirty as a keen maker, Dylan loves the great outdoors and he's passionate about the environment, hiking, food, and gardening.

magpi.cc/iTVuOC

Back in 1982, everyone's favourite extra-terrestrial, E.T., needed to phone home, so an attempt was made, via a coffee can filled with electronics connected to a tinfoil-lined umbrella, and a Speak & Spell. For many of today's modern makers, that would be an alien concept (not least because it just wouldn't work). But it's one Dylan Radcliffe may well have paid some thought to, given his struggles to create a 3G smartphone based around a Raspberry Pi.

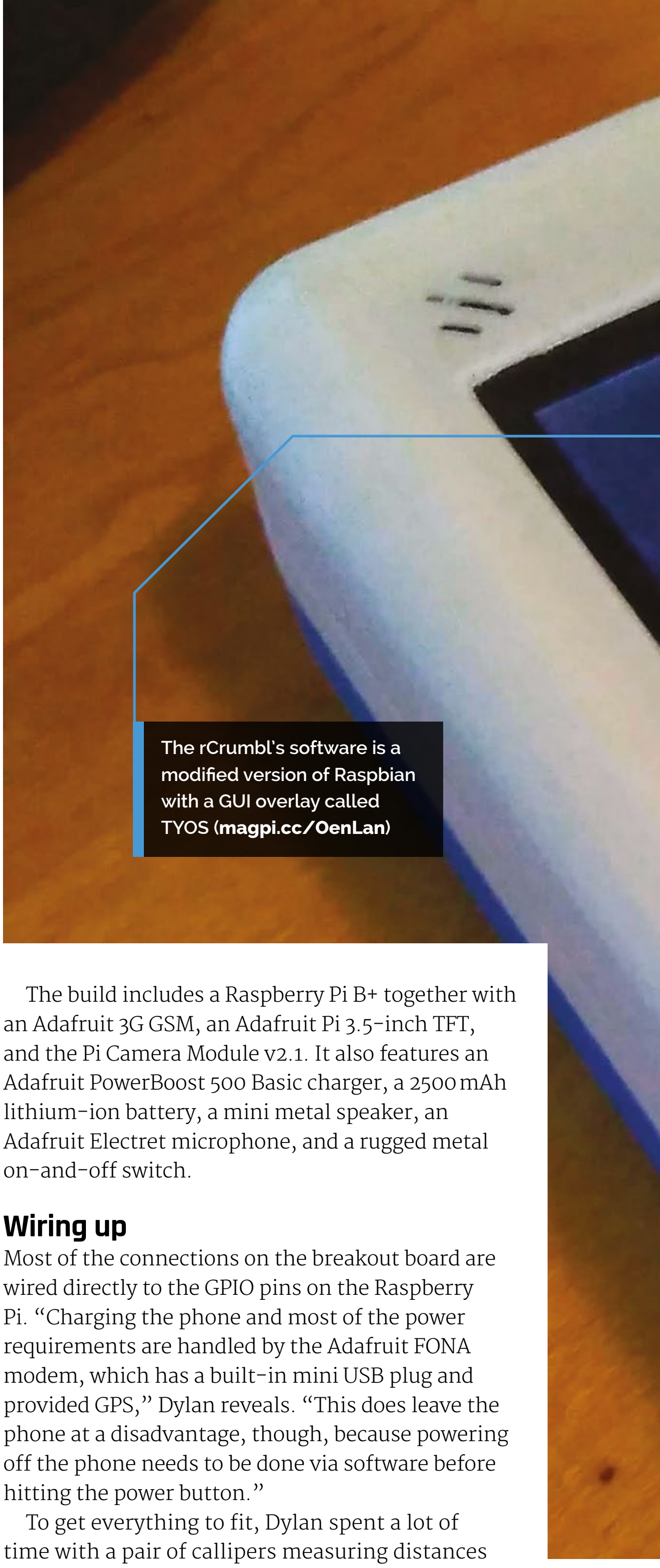
Usually our project showcases highlight the finished product, but in this case Dylan needs some help. When he began planning his project, he based it on the TyTelli DIY smartphone, a device made almost four years ago by Tyler Spadgenske that could snap photos, send texts, and make and receive calls. It used a specially written OS in Python and it was housed in a 3D-printed case. But it only worked on a 2G cellular network and Dylan found, to his horror, that his home country of Canada had finally phased it out.

2G or not 2G

"I'd purchased a PCB which was only designed to work with a 2G network, but I was lucky because the supplier had also released a 3G chip," he says. "Unfortunately, all of the software produced by others for their phones won't work with this new chip, so this is my challenge." So far, he's been able to call his landline using the command line, but he is vowing to do better and he hopes readers of *The MagPi* will be able to help.

> "His primary aim of creating a Pi-based phone that could fit into his pocket has been a success"

As it stands, Dylan has a functional pocket computer running Linux, but it remains a great example of determination. His primary aim of creating a Pi-based phone that could fit into his pocket has been a success, and that has been down to a great deal of time spent determining exactly how everything would need to be wired together in order to create a functional piece of electronics.



The rCrumbl's software is a modified version of Raspbian with a GUI overlay called TYOS **(magpi.cc/OenLan)**

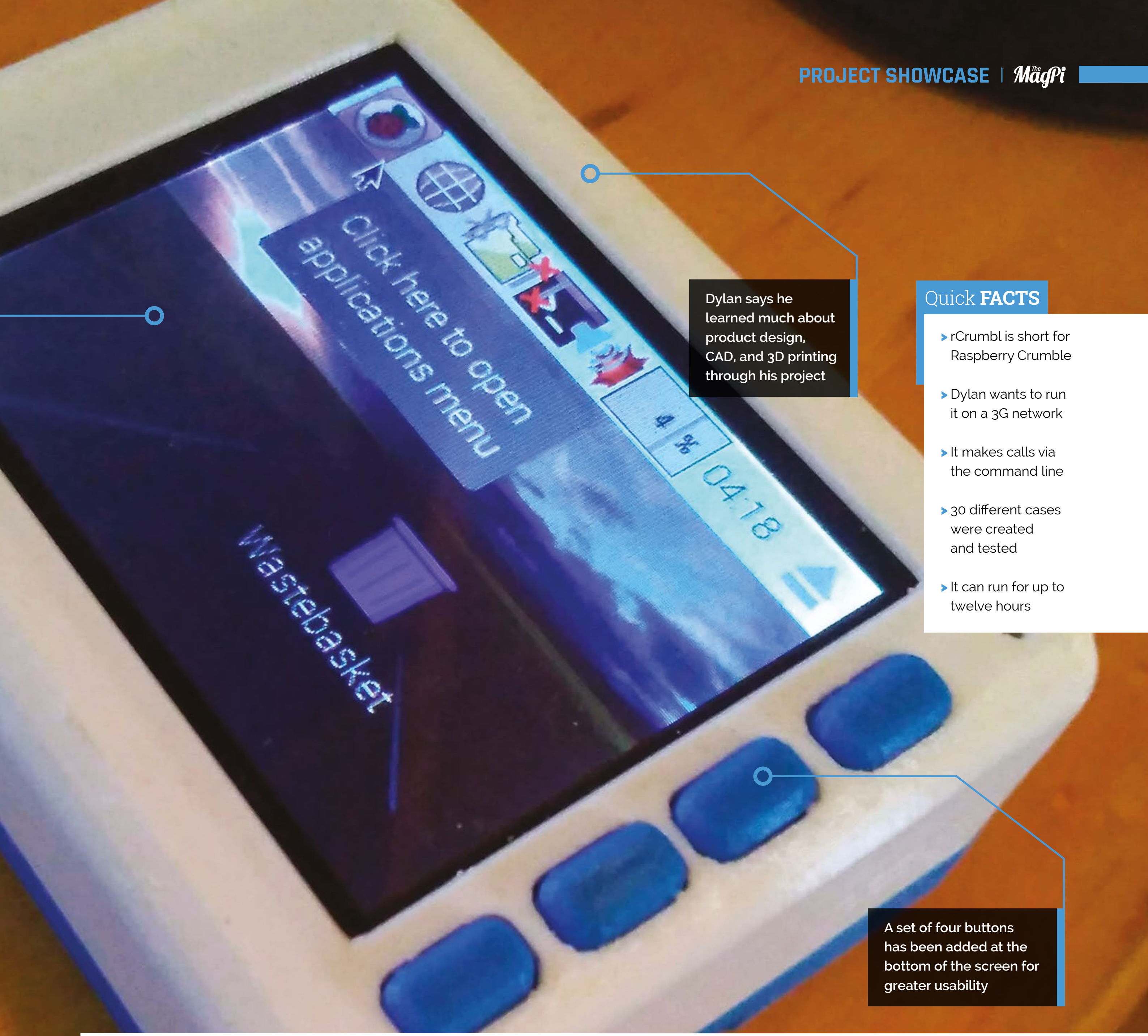
The build includes a Raspberry Pi B+ together with an Adafruit 3G GSM, an Adafruit Pi 3.5-inch TFT, and the Pi Camera Module v2.1. It also features an Adafruit PowerBoost 500 Basic charger, a 2500 mAh lithium-ion battery, a mini metal speaker, an Adafruit Electret microphone, and a rugged metal on-and-off switch.

Wiring up

Most of the connections on the breakout board are wired directly to the GPIO pins on the Raspberry Pi. "Charging the phone and most of the power requirements are handled by the Adafruit FONA modem, which has a built-in mini USB plug and provided GPS," Dylan reveals. "This does leave the phone at a disadvantage, though, because powering off the phone needs to be done via software before hitting the power button."

To get everything to fit, Dylan spent a lot of time with a pair of callipers measuring distances between components and wires. "Once I had everything working, I created a small box on my 3D printer and tried to cram everything in, in a way that made sense," he says, knowing that he would end up soldering in the tightest environment he'd ever experienced.

He then spent a couple hours thinking about how to optimise the design and writing notes before firing up his CAD software again. In the process, a small bracket was designed and used to attach some of the smaller PCBs to hold them in place. "All I have to do is get 3G working, but I'm sure I'll get there in the end," discloses Dylan.

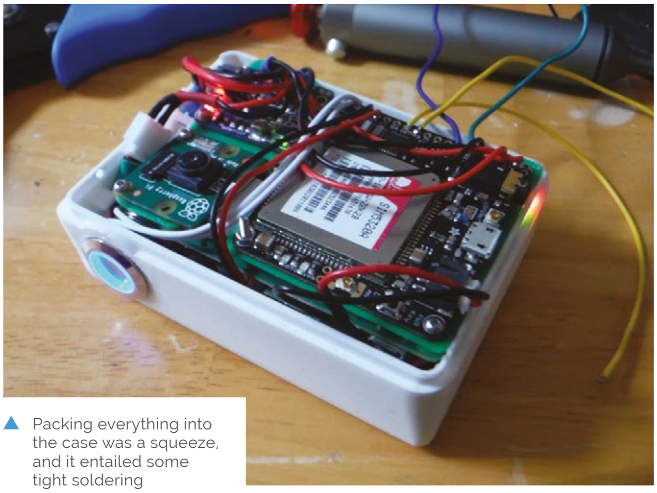


Dylan says he learned much about product design, CAD, and 3D printing through his project

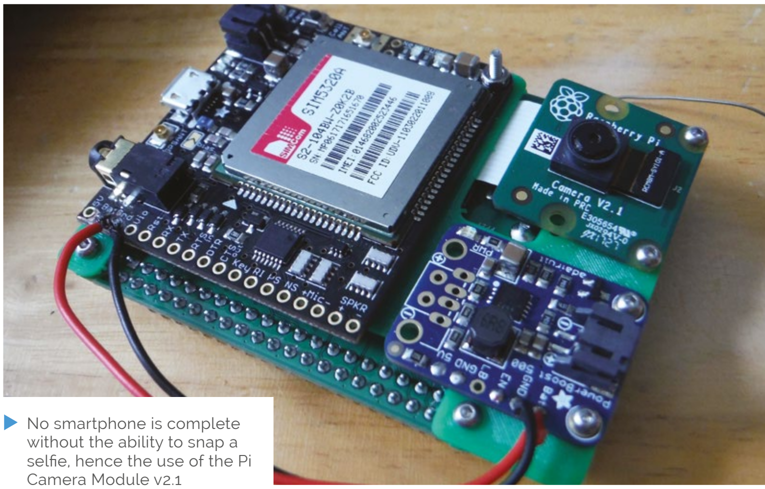
A set of four buttons has been added at the bottom of the screen for greater usability

Quick FACTS

- rCrumbl is short for Raspberry Crumble
- Dylan wants to run it on a 3G network
- It makes calls via the command line
- 30 different cases were created and tested
- It can run for up to twelve hours



▲ Packing everything into the case was a squeeze, and it entailed some tight soldering



▶ No smartphone is complete without the ability to snap a selfie, hence the use of the Pi Camera Module v2.1

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Issue 76 | December 2018 | magpi.cc
NEW FOR 2019
THE RASPBERRY PI SUPERGUIDE
• Get going guide • Starter projects • Tech jargon busted
Code Pac-Man
Make your own retro ghost-chaser
The All New 3A+
The ultimate $25 computer for makers
WIN! 10 Signed 3A+
PLUS
10 Raspberry Pi Christmas gifts
Script command line tasks
Meet Nybble: the powerful robotic kitten
Make Smart Lights
Add to the festive fun with Pi-controlled house lighting
Scratch Resources | Raspberry Pi Boombox | GFX HAT

Subscriber Benefits

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

Rolling Monthly Subscription

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

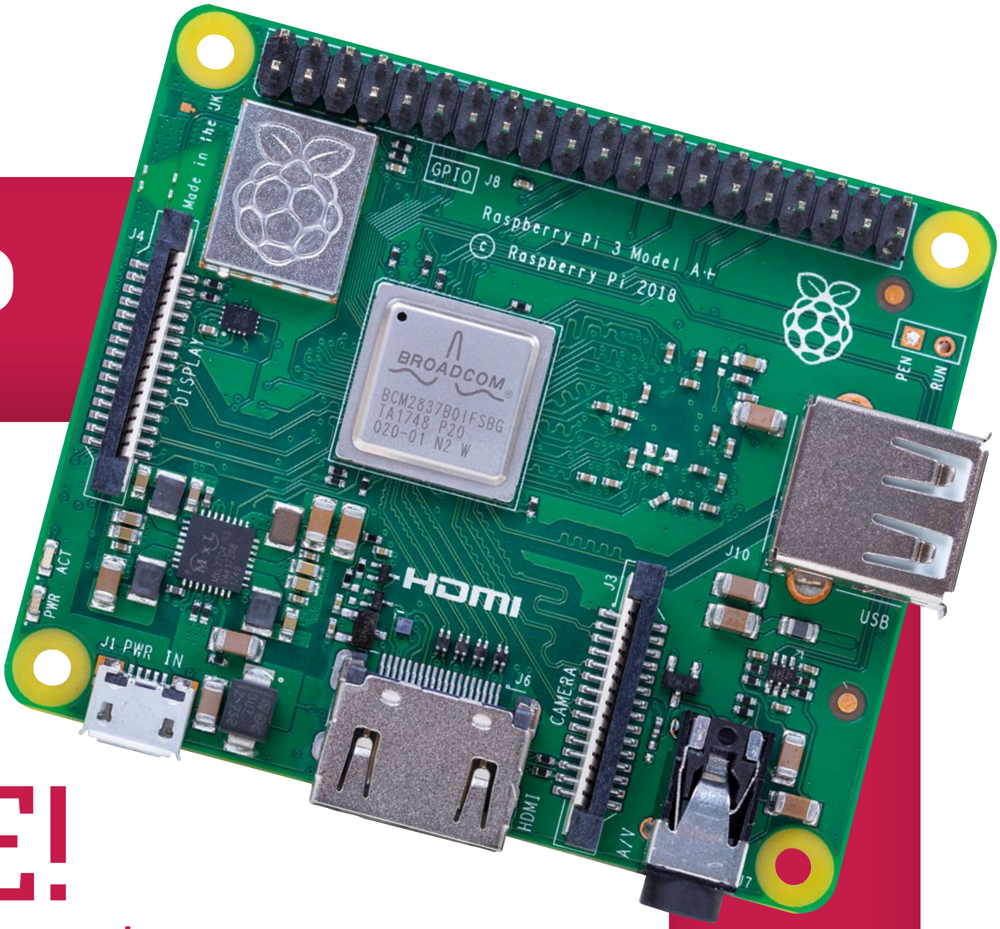
Subscribe for 12 Months

£55 (UK)	£90 (USA)
£80 (EU)	£95 (Rest of World)

Free Raspberry Pi 3A+ with 12 Month upfront subscription only
(no Raspberry Pi 3A+ with Rolling Monthly Subscription)

Subscribe online: magpi.cc/subscribe

LIMITED OFFER!



FREE!

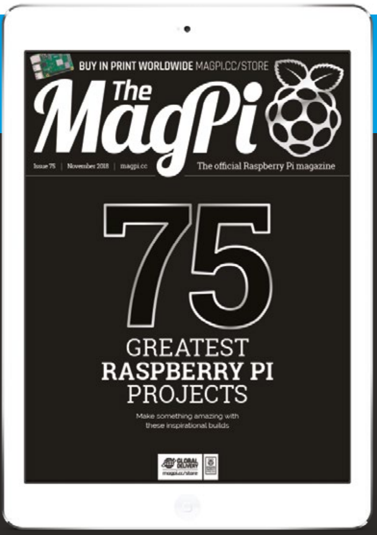
Raspberry Pi 3A+

WITH YOUR 12 MONTH PRINT SUBSCRIPTION

This is a limited offer. It replaces our usual offer of a free Pi Zero W.
Offer subject to change or withdrawal at any time.

WORTH **£25**

Buy now: magpi.cc/subscribe


BUY IN PRINT WORLDWIDE MAGPI.CC/STORE
The MagPi
The official Raspberry Pi magazine
75
GREATEST RASPBERRY PI PROJECTS
Make something amazing with these inspirational builds

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

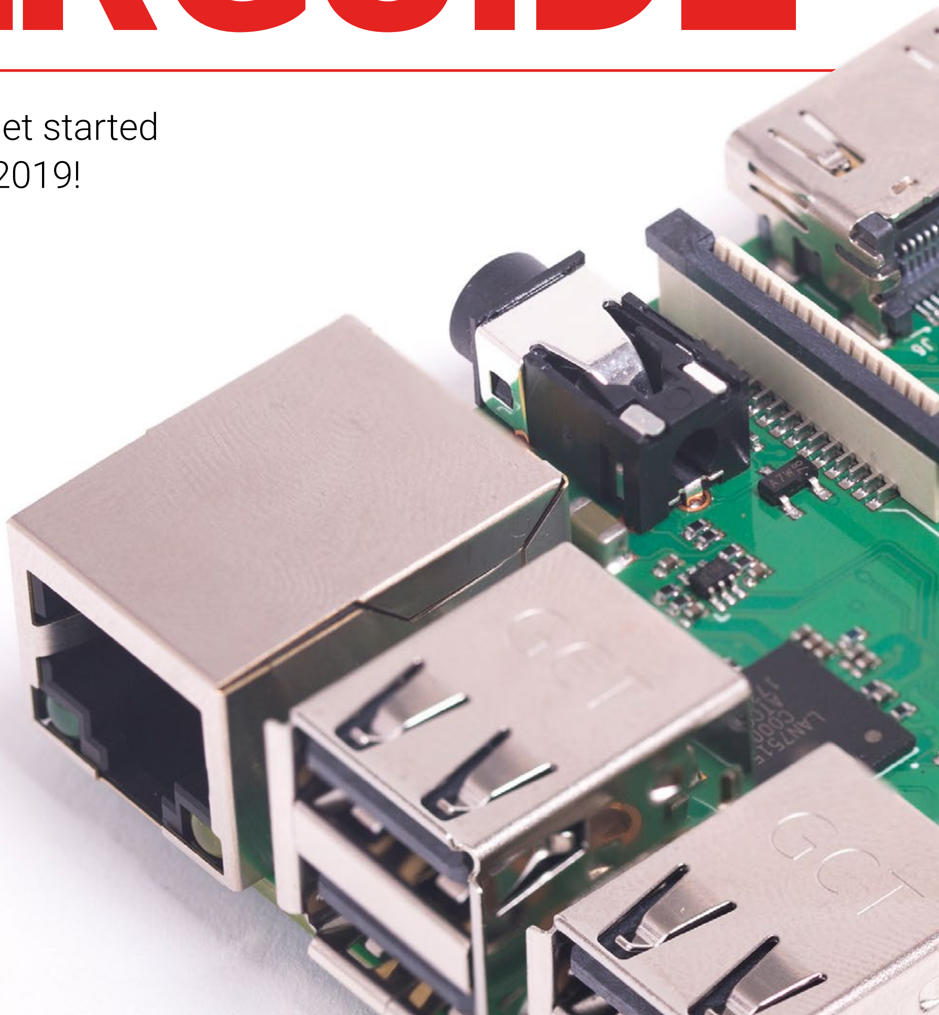
THE RASPBERRY PI SUPERGUIDE

Everything you need to know to get started with Raspberry Pi – updated for 2019!

Welcome to our ultimate guide for the Raspberry Pi! There's always new Raspberry Pi software and hardware coming out and while we like to think we keep you updated on it all, it does mean that our complete guides can sometimes feel a little outdated.

That's why we've put together this: a completely new guide to getting started. If you are (or someone you know is) new to the Raspberry Pi, this guide has you covered.

So, whether you have an original Model B or a brand-new 3A+, get ready to enter the world of Raspberry Pi. It's a lot of fun!

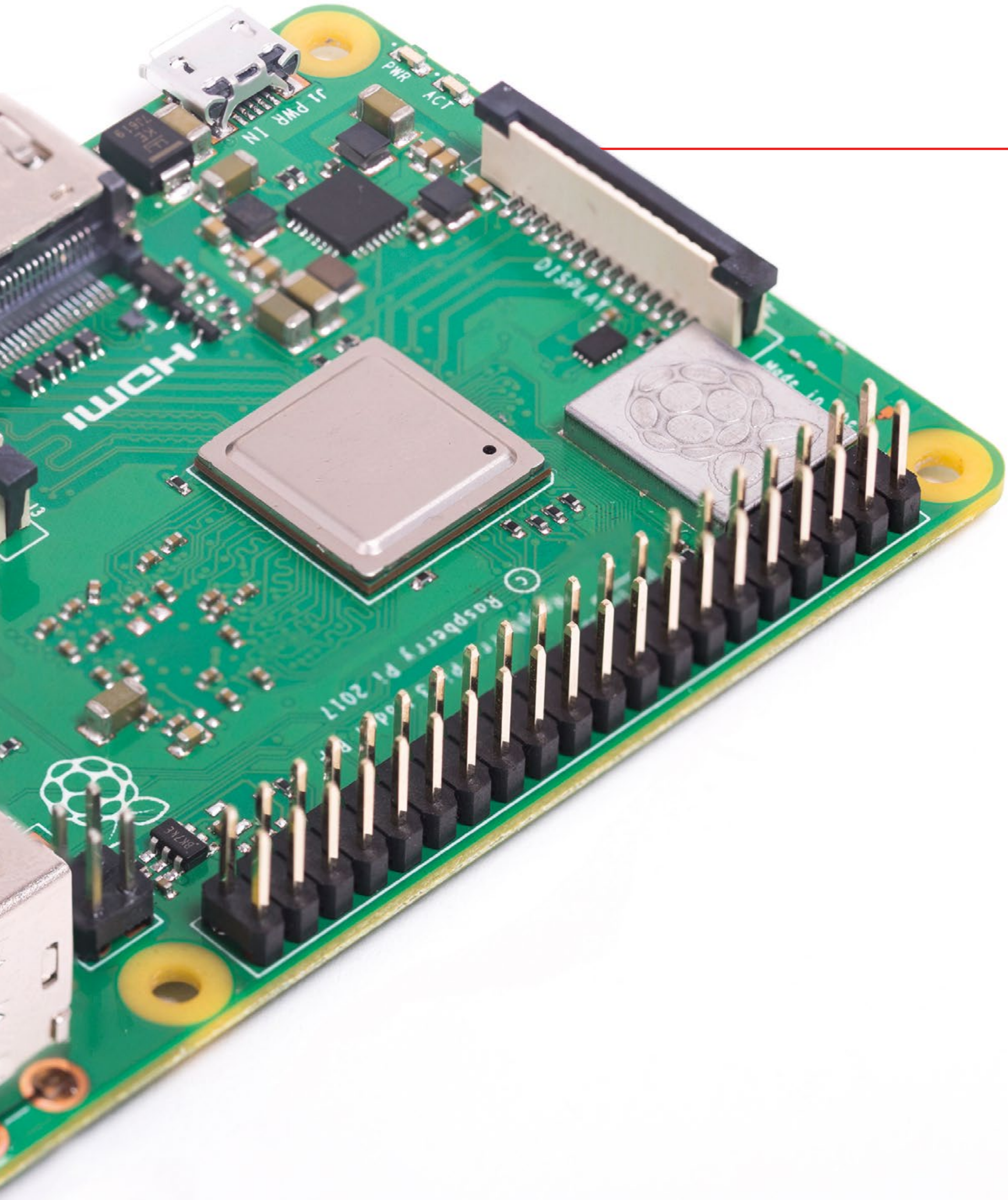


QUICKSTART GUIDE

Get your Raspberry Pi working

Got your Raspberry Pi but still not sure how to even get it turned on? It's very simple – you just need to download an operating system from the Raspberry Pi website (**magpi.cc/noobs**), put it on a microSD card, and plug it in.

Let us guide you through this process step-by-step in our QuickStart guide, which you can download here: **magpi.cc/quickstart**



OFFICIAL RESOURCES

Want some more ideas of how to use your Raspberry Pi? Head online…

RASPBERRY PI PROJECTS SITE

rpf.io/projects

Hundreds of projects from the Raspberry Pi Foundation, Code Club, and CoderDojo can be found here, covering a wide array of programming and electronics projects. A lot of them are suitable for younger makers as well!

DOCUMENTATION

rpf.io/docs

If you need some in-depth info about your Raspberry Pi and Raspbian, the official documentation is very thorough and even contains examples to help you understand what everything means.

FORUMS

rpf.io/forums

The Raspberry Pi community can be found on the official forums, and they have plenty of expertise with using the Pi. You'll usually be able to find a solution to any issues you have here – and if not, you can always ask about it.

WELCOME TO THE RASPBERRY PI DESKTOP

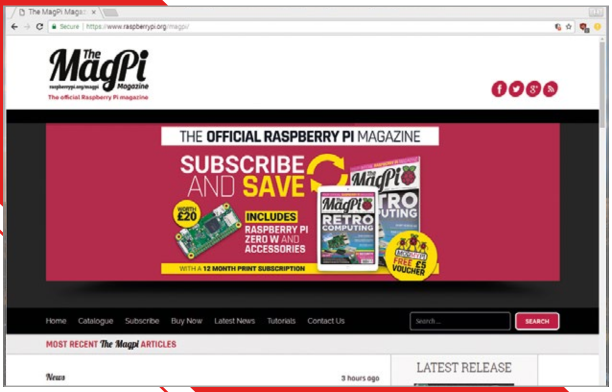
New to the world of Raspberry Pi? **PJ Evans** shows you around

Once you've got Raspbian with Desktop and Recommended Software installed and running, you may be wondering what to do next. Let's look at some of the key software installed, how to add more, and some of the differences between Raspbian and its peers, Windows and macOS.

CHROMIUM

Web browser

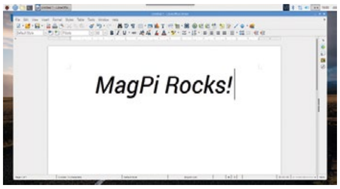
An open-source version of Google Chrome, Chromium is a fast and fully-featured web browser that can handle the demands of modern websites.


The MagPi Magazine
https://www.raspberrypi.org/magpi/
The MagPi Magazine
The official Raspberry Pi magazine
THE OFFICIAL RASPBERRY PI MAGAZINE
SUBSCRIBE AND SAVE
WORTH £20
INCLUDES RASPBERRY PI ZERO W AND ACCESSORIES
WITH A 12 MONTH PRINT SUBSCRIPTION
Home
Catalogue
Subscribe
Buy Now
Latest News
Tutorials
Contact Us
Search...
SEARCH
MOST RECENT The MagPi ARTICLES
News
3 hours ago
LATEST RELEASE

LIBREOFFICE

Office suite

Word processing, spreadsheets, presentations, databases, and drawing. A complete suite of office tools free with your Raspberry Pi. Compatible with other packages such as Microsoft Office.

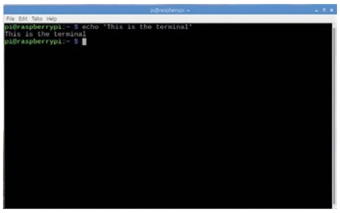


MagPi Rocks!

TERMINAL

Command-line access

The gateway to thousands of powerful text-based applications and tools, you may find yourself using the Terminal a lot to run scripts or install software.

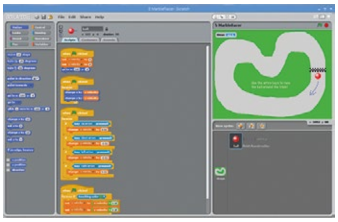


```
pi@raspberrypi:~ $ echo 'This is the terminal'
This is the terminal
pi@raspberrypi:~ $
```

SCRATCH

Fun programming environments

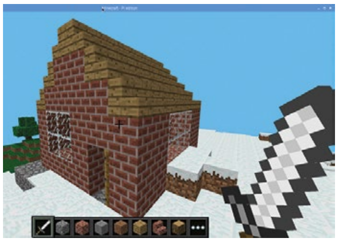
Scratch is the perfect introduction to coding. You can build games or even control lights using the Pi's GPIO pins. Its graphical block system makes coding friendly.



MINECRAFT PI

Open-world game

Not only can you run Minecraft on a Pi, you can learn by interacting with its virtual world through code. The only limit is your imagination.



Credit: Rob Zwetsloot

SONIC PI

Music generation

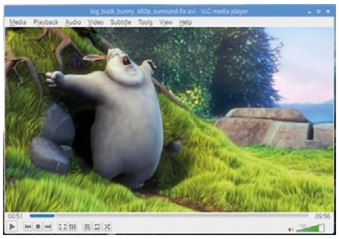
Known as 'live coding', creating music through code has been thrust into the mainstream by this phenomenal application. Your Raspberry Pi is now a serious musical instrument!

```
# Ocean

# Coded by Sam Aaron

with_fx :reverb, mix: 0.5 do
  live_loop :oceans do
    s = synth [:bnoise, :cnoise, :gnoise].choose, amp: rrand(0.
    control s, pan: rrand(-1, 1), cutoff: rrand(60, 110)
    sleep rrand(2, 4)
  end
end
```

VLC

Media player

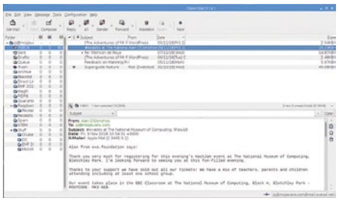
The Swiss Army knife of media playback, supporting a multitude of video and audio formats. If it can be played, VLC will probably be able to play it.


big_buck_bunny_480p_surround-fix.avi - VLC media player
Media Playback Audio Video Subtitle Tools View Help

CLAWS MAIL

Email client

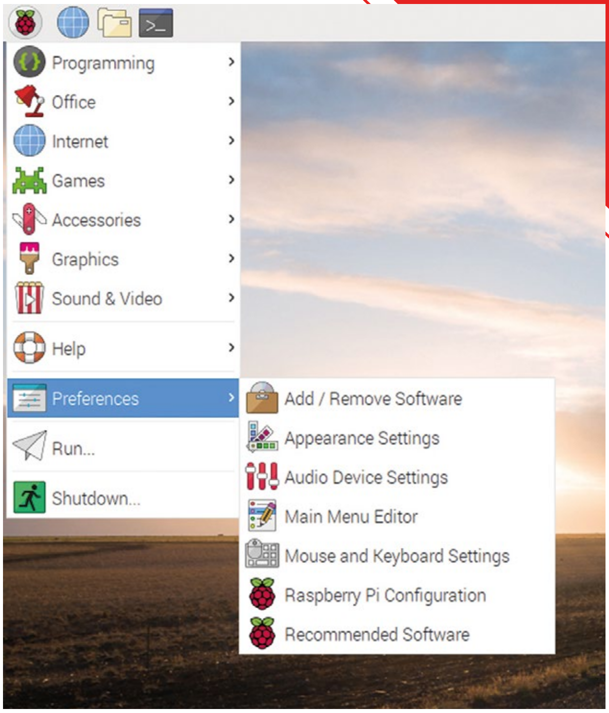
If you need more than webmail, Claws provides all the common features you'll need. It's easy to configure and provides an elegant interface.



INSTALLING SOFTWARE

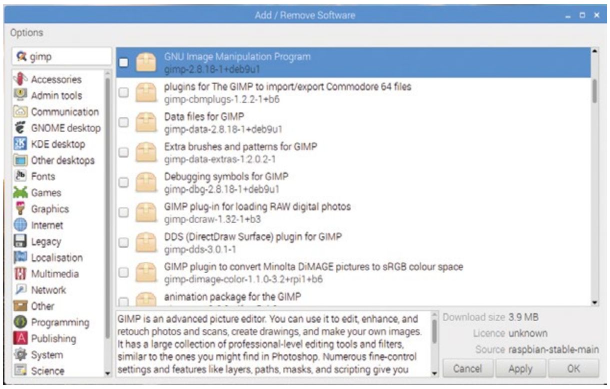
01 Open your package manager

The most straightforward way to install software on Raspbian with Desktop is to use the 'Add/Remove Software' application. Start by clicking the Raspberry Pi logo to open the menu, then select 'Preferences' followed by 'Add/Remove Software'.



02 Get searching

We're going to install GIMP, so it's as simple as entering 'gimp' in the search box and pressing **RETURN**. After a few seconds, a list of matches appears. Click the one just labelled 'GNU Image Manipulation Program'. Any other required packages, known as dependencies, will automatically be installed.



gimp

□ GNU Image Manipulation Program
gimp-2.8.18-1+deb9u1

□ plugins for The GIMP to import/export Commodore 64 files
gimp-cbmplugs-1.2.2-1+b6

□ Data files for GIMP
gimp-data-2.8.18-1+deb9u1

□ Extra brushes and patterns for GIMP
gimp-data-extras-1:2.0.2-1

□ Debugging symbols for GIMP
gimp-dbg-2.8.18-1+deb9u1

□ GIMP plug-in for loading RAW digital photos
gimp-dcraw-1.32-1+b3

□ DDS (DirectDraw Surface) plugin for GIMP
gimp-dds-3.0.1-1

□ GIMP plugin to convert Minolta DiMAGE pictures to sRGB colour space
gimp-dimage-color-1.1.0-3.2+rpi1+b6

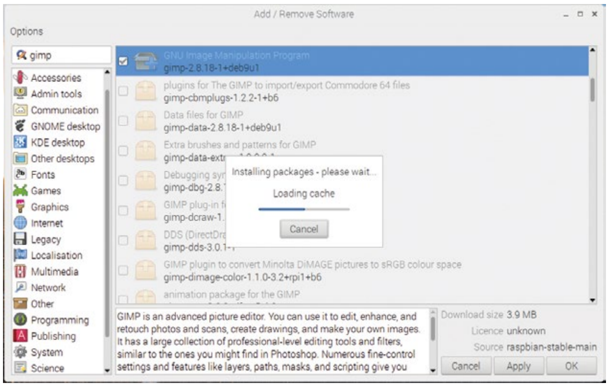
□ animation package for the GIMP

GIMP is an advanced picture editor. You can use it to edit, enhance, and retouch photos and scans, create drawings, and make your own images. It has a large collection of professional-level editing tools and filters, similar to the ones you might find in Photoshop. Numerous fine-control settings and features like layers, paths, masks, and scripting give you

Download size 3.9 MB
Licence unknown
Source raspbian-stable-main

03 Install

Now the good bit. Click 'Apply'. You will need to enter your password. Your application will then be downloaded and installed for you. Installation time will depend on your bandwidth and the size of the package in question. GIMP typically takes a couple of minutes to install. Once complete, you can find GIMP in the application menu under 'Graphics'.



gimp

☑ GNU Image Manipulation Program
gimp-2.8.18-1+deb9u1

□ plugins for The GIMP to import/export Commodore 64 files
gimp-cbmplugs-1.2.2-1+b6

□ Data files for GIMP
gimp-data-2.8.18-1+deb9u1

□ Extra brushes and patterns for GIMP
gimp-data-extr

□ Debugging sy
gimp-dbg-2.8.

□ GIMP plug-in f
gimp-dcraw-1.

□ DDS (DirectDr
gimp-dds-3.0.1-

□ GIMP plugin to convert Minolta DiMAGE pictures to sRGB colour space
gimp-dimage-color-1.1.0-3.2+rpi1+b6

□ animation package for the GIMP

Installing packages - please wait...

Loading cache

Cancel

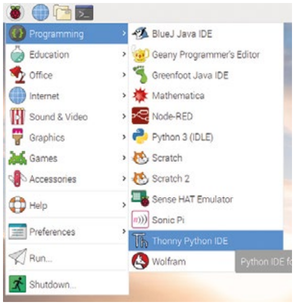
GIMP is an advanced picture editor. You can use it to edit, enhance, and retouch photos and scans, create drawings, and make your own images. It has a large collection of professional-level editing tools and filters, similar to the ones you might find in Photoshop. Numerous fine-control settings and features like layers, paths, masks, and scripting give you

Download size 3.9 MB
Licence unknown
Source raspbian-stable-main

CODE HELLO WORLD

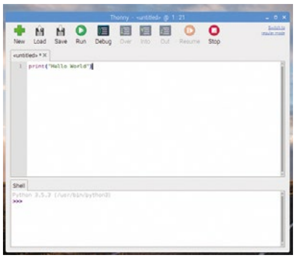
01 Start Thonny

The Raspberry Pi is designed to make it easy to learn to program a computer. There are lots of programming languages, but the best one to start with is Python. You can use Python on a Raspberry Pi using a program called Thonny. From the application menu, select Programming > Thonny Python IDE.



02 Hello World

It's traditional for any new program in a language to display Hello World on the screen. Click on the main screen in Thonny and enter:



```
print("Hello World")
```

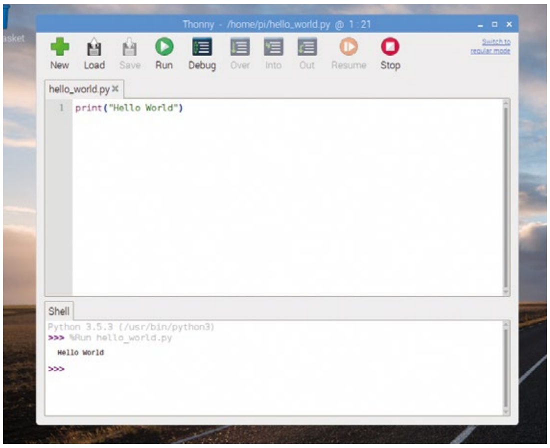
```
Python 3.5.3 (/usr/bin/python3)
>>>
```

```
print("Hello World")
```

Click Save and call your program **hello_world.py**.

03 Run your code

Click on the green Run button. Congratulations! You've just run your first computer program. Python is a very easy language to learn. Visit **magpi.cc/learnpython** to read our starter guide.

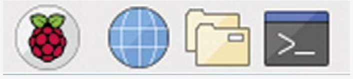


```
print("Hello World")
```

```
Python 3.5.3 (/usr/bin/python3)
>>> %Run hello_world.py
  Hello World
>>>
```

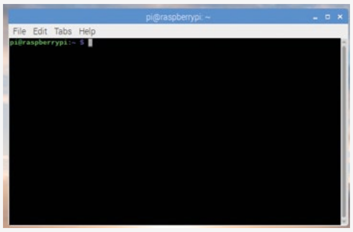
SWITCH TO RASPBIAN

Raspbian is the name for the Raspberry Pi's official operating system and it's a Linux distribution. 'Raspbian with Desktop' comes with a graphical user interface. Here are some key differences between the Desktop and other systems.



APPLICATION MENU

The application launch menu operates in a similar way to Windows, but is located at the top. Clicking the Raspberry Pi logo (top left) displays the menus where you can find pre-installed software and access to various utilities. Favourite apps can be pinned on the left and some utilities can be controlled on the right.



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

TERMINAL CONFUSION

Chances are, you'll use the Terminal a lot to run scripts and issue commands to the operating system. This command-line interface is common to all operating systems, but often goes by different names. 'Command Prompt' (Windows), 'Console', and 'Terminal' all refer to the same thing. Learn more at **magpi.cc/terminal**.

DIRECTORY LAYOUT

All users on Raspbian have a home directory in **/home**. Just like Windows and macOS, you'll see directories such as 'Music' and 'Pictures' already created for you.

FILE MANAGER

Like other operating systems, you can graphically browse the file system. Your default user is 'pi' and your home directory is **/home/pi**. Don't worry about looking around but if you're asked for a password before changing a file, the system might be trying to protect it, so take care.

REMOTE CONTROL

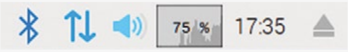
Want to get remote control of your Pi? Windows has 'Remote Desktop', macOS has 'Back to my Mac', and Raspbian has VNC. A server is installed by default but not active. Enable it in Preferences > Raspberry Pi Configuration > Interfaces.

SSH

If you just need remote command-line access to your Pi, SSH can be enabled to provide a 'Secure Shell'. This can be accessed from other Pi boards, Linux distributions or macOS using Terminal, or the PuTTY app for Windows.

WASTEBASKET

While macOS has 'Trash' and Windows has a 'Recycle Bin', the Raspberry Pi has the 'Wastebasket'. It works in exactly the same way.



WHAT'S WITH THE PERCENTAGES?

If you're scratching your head over the bar graph placed in the top-right, it's a live reading of the CPU's activity. If things appear slow and the percentage is high, it may be the machine is 'having a think' or too many tasks are running.

ACCESSORIES

The Accessories menu provides applications for many common tasks. Both Image Viewer (in the Graphics menu) and PDF Viewer have the functionality of macOS's Preview. There's also a Task Manager which, like Windows (or Activity Monitor in macOS), allows you to track what's going on with your machine.

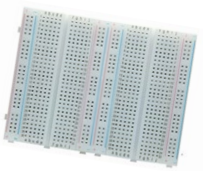
RASPBERRY PI AND THE PHYSICAL WORLD

Take your coding further and break out of the screen into the real world

The one piece of hardware that separates the Raspberry Pi from the average desktop PC or laptop is its GPIO (general-purpose input/output) header. This collection of pins on the edge of the board provides a simple, safe way to connect circuits to the Pi so it can interact with its surroundings. From blinking a light to controlling a home security system, it's the GPIO that makes it all possible.

ELECTRONICS COMPONENTS AND KIT

BREADBOARD



▶ **magpi.cc/tswjyk**

An easy way to connect components together and to the Pi itself. Cheap, easy to use, and essential for prototyping your next masterwork.

JUMPER WIRES

▶ **magpi.cc/ecmNPS**

Jumper wires are used to make connections on the breadboard and to the Raspberry Pi's GPIO pins. Look out for multipacks containing different lengths and types.

LEDS

▶ **magpi.cc/gqdVMP**

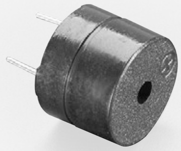
Add some colour to your project with light-emitting diodes. Inexpensive and a fun component for your first projects. Available in multiple colours.

TACTILE BUTTONS

▶ **magpi.cc/aCtkRt**

The GPIO handles input as well as output, so a great way to learn is with a tactile button. These fit straight on to breadboards for easy building.

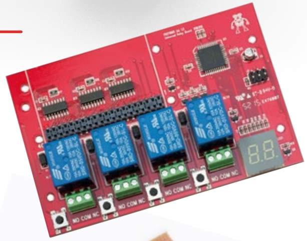
BUZZERS



▶ **magpi.cc/ijcKzz**

Make some noise! Simple buzzers come in a wide variety of types and sizes, and are straightforward to add to projects.

RELAYS



▶ **magpi.cc/nRidMi**

Want to switch something on but not be connected directly to its circuit? Relays are a simple solution where you don't 'cross the streams'.

STRIPBOARD

▶ **magpi.cc/mirhkq**

A cheap alternative to custom PCBs for small circuits. Strips (or pads) of copper on a board on to which you can solder components.

SOLDERING IRON

▶ **magpi.cc/gnxPQs**

If you want to transfer a breadboarded circuit to a PCB or stripboard, you'll need a soldering iron, a stand, and some lead-free solder.

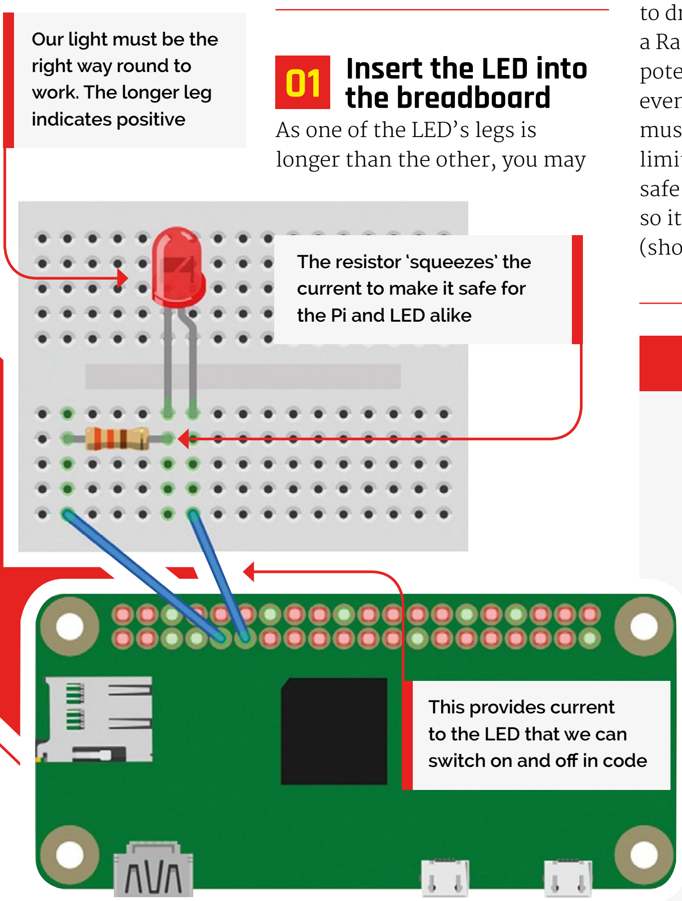
LIGHTING AN LED

When you first learn to code, chances are you'll write a program to display 'Hello, World!' on the screen.

When learning electronics, you'll make something light up. No exceptions here! Let's see how, with just a few lines of code, the GPIO can light up your life.

01 Insert the LED into the breadboard

As one of the LED's legs is longer than the other, you may need to bend it (carefully!) so both legs will fit securely in the breadboard. LEDs will only work one way round in a circuit, so the longer leg indicates the positive side.


Our light must be the right way round to work. The longer leg indicates positive

The resistor 'squeezes' the current to make it safe for the Pi and LED alike

This provides current to the LED that we can switch on and off in code

02 Add a resistor

An LED will attempt to draw more current than a Raspberry Pi can handle, potentially damaging the LED or even the Pi. To prevent this, we must add a resistor which will limit the flow of electricity to safe levels. Place a 330 Ω resistor so it connects with the negative (shorter) leg of the LED.

03 Wire to the Pi

Orientate the Pi so the Ethernet and USB ports are at the bottom (six o'clock). Now pin 1 of the GPIO is the top pin on the left. GPIO17 is the sixth pin down on the left. Connect a jumper cable to this pin and then to the positive leg of the LED (via a hole in the same breadboard column).

04 Grounding

Finally, for the current to flow, it needs a ground. Connect another jumper cable to one of the GND pins of the GPIO. There's one directly above GPIO17. This connects to the other side of the resistor (again, via a breadboard hole in the same column).

05 Code

We'll use Python with the GPIO Zero library to control the LED. Under 'Programming' in the main menu, open 'Thonny Python IDE' and enter the code below exactly as shown, including the indentations. Then click 'Run' to see the LED blink.

Example Code

```
# Extend Python's capabilities
from gpiozero import LED
from time import sleep

# Set 'red' to represent the LED
red = LED(17)

# Loop forever
while True:

    # Send current to the LED for one second
    red.on()
    sleep(1)

    # Stop the current for one second
    red.off()
    sleep(1)
```

USING A BUTTON

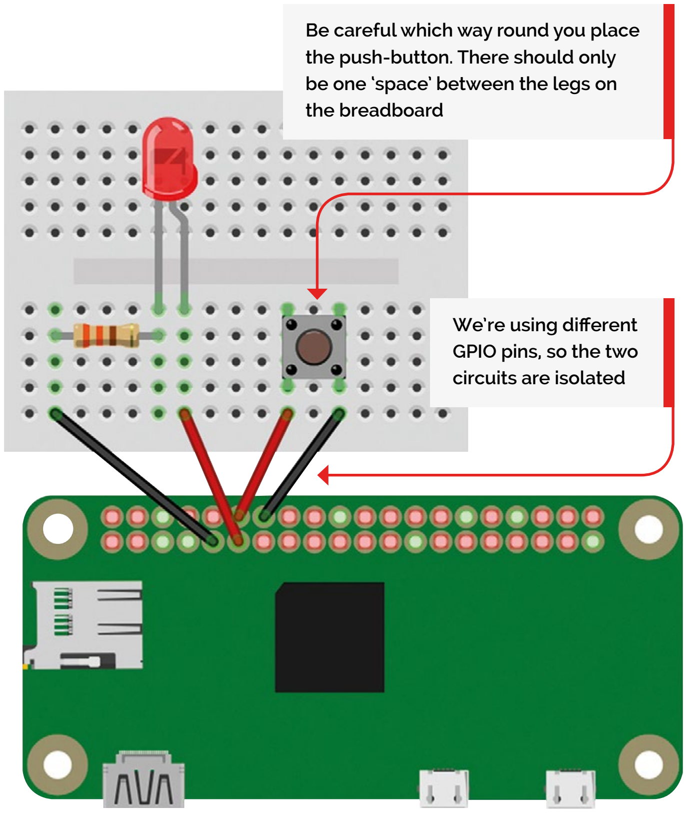
Let's add some input to our new circuit. We'll add a push-button that can control the LED.

01 Orientate the button

Start by placing your push-button in the breadboard, being careful not to interfere with the LED circuit. The button has four pins which protrude on two sides. The pins opposite each other are always connected, so the protruding sides needs to 'bridge' the lines of the breadboard.

02 Connect to the GPIO

We need to connect the button to the GPIO so we can monitor its state. We'll use GPIO 18 for this, which is the pin directly to the right of GPIO 17, the one we used in the LED circuit. Now connect the other end of the jumper cable to either of the button's legs on the breadboard.



Be careful which way round you place the push-button. There should only be one 'space' between the legs on the breadboard

We're using different GPIO pins, so the two circuits are isolated

03 Ground the button

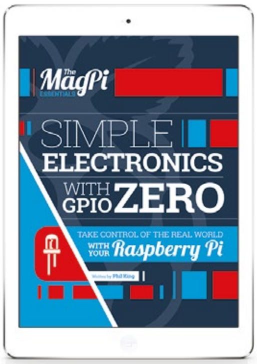
Just like the LED, we need to ground the button so current will flow. There is a spare ground point directly below GPIO 18. Connect this to the other leg on the short side of the button. Before switching anything on, check and check again that the GPIO is wired up correctly.

04 More code

Just like before, enter the code below into Thonny and click 'Run'. Now, when the button is pressed, the LED will light up. What else can you make it do?

Example Code

```
# Extend Python's capabilities
from gpiozero import LED, Button
from signal import pause

# Assign these variables to our LED and button
led = LED(17)
button = Button(18)

# Assign events that are triggered when the button is used
button.when_pressed = led.on
button.when_released = led.off

# This command prevents Python from exiting
pause()
```



The MagPi ESSENTIALS
SIMPLE ELECTRONICS WITH GPIO ZERO
TAKE CONTROL OF THE REAL WORLD WITH YOUR Raspberry Pi

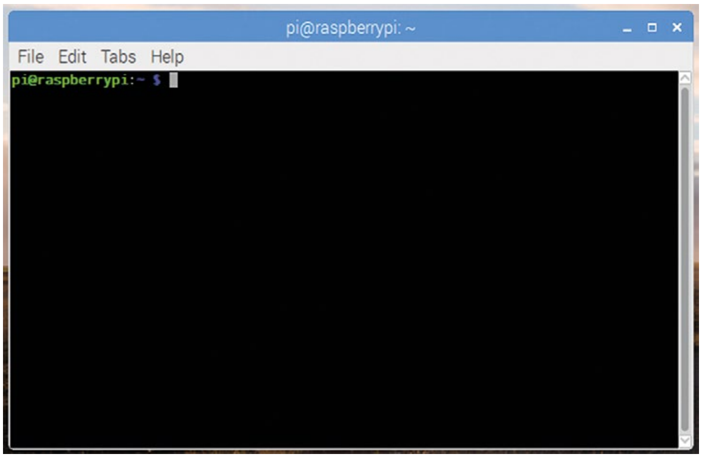
MAGPI ESSENTIALS
SIMPLE ELECTRONICS WITH GPIO ZERO

Get up-and-running with electronics quickly with our GPIO Zero Essentials Guide.
magpi.cc/gpio-zero

COMMAND-LINE SUPER BASICS

The command line is a powerful and efficient way to interact with your Raspberry Pi

The Terminal application is your gateway to the Raspbian command line. Yes, having a black screen with a flashing cursor and little else may seem a little intimidating, but with practice the command line becomes a go-to environment for getting things done. Click the Terminal icon in the Task Bar to start..



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

COMMANDS

sudo

'Super/substitute user do'. Some commands require extra privileges to work. Prefixing the command with 'sudo' allows this. If you forget, type `sudo !!` to repeat the previous command.

```
sudo raspi-config
```

apt

APT is Raspbian's package manager. It can be used to install a huge range of applications and also keeps existing installations up to date.

```
sudo apt install gimp
```

cd directory path

Raspbian stores files in directories. 'cd' is short for 'change directory', so `cd /boot` will take you to the Pi's **boot** directory. `cd` on its own always takes you to your home directory.

```
cd /home/pi/MagPi
```

ls directory

View the contents of a directory (**lis**t). If you omit the directory, the current one will be listed. Add `-l` before the directory for a more detailed output.

```
ls -l /boot
```

cp, mv, and rm

'Copy, move, and remove'. These are used to make copies of files or directories, move files around, rename or delete them. `cp` takes two arguments: source and destination.

```
cp /home/pi/MagPi/MagPi75.pdf /home/pi/Desktop/MagPi75.pdf
```

mkdir and rmdir

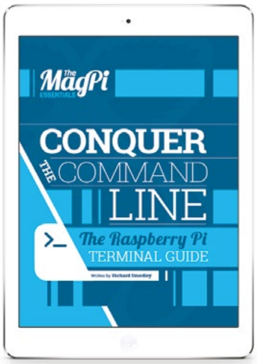
Make and remove directories. You can have as many as you like, and have directories within directories.

Tip: To create multiple levels in one command, use `-p`.

```
mkdir -p /home/pi/my/new/directory/tree
```

exit

When you're done in the Terminal, simple type `exit` to cleanly close everything down.



MAGPI ESSENTIALS
CONQUER THE COMMAND LINE

Master your computer using command-line code in Terminal with our free guide.
magpi.cc/essentials-bash

JARGON BUSTER

Worried about your boot? Pip got you pipped? Here are some of the more common terms you'll hear in the Raspberry Pi universe

APT

Advanced Package Tool. This is the Raspbian package manager for installing and upgrading applications.

ARM

Designer of the Raspberry Pi's processor architecture.

Broadcom

Manufacturer of the Raspberry Pi's main processor.

Burn

The process of writing an operating system image to an SD card.

Distribution

The many versions of Linux, including Raspbian, are known as Linux distributions.

DSI

Digital Serial Interface. The DSI connector on the Raspberry Pi can be used to connect displays.

Ethernet

A wired networking system.

GPIO

General-purpose input/output. An array of pins for connecting physical circuits to the Pi.

GPIO Zero

A Python library of tools for communicating via the GPIO.

GPU

Graphics processing unit. On the Pi, this assists the main processor to provide smooth, fast graphics.

HDMI

All Raspberry Pi models provide an HDMI output to connect to modern televisions and monitors.

I^{2}C

Inter-Integrated Circuit. Like SPI, allows multiple devices to be connected to the Pi via the GPIO.

Kernel

A small piece of code that is the heart of the operating system. Everything runs 'on top' of the kernel.

Linux

A family of open-source operating systems, including Raspbian.

OS

Operating system – the code that handles the interaction between the processor, its devices, and the outside world.

Packages

Distributions of applications that are handled by APT.

pip

A package manager for Python to ease the installation of Python libraries and other applications.

Prompt

As in 'prompt for input'. The $ symbol before your cursor in the Terminal.

Raspbian

The official operating system of the Raspberry Pi.

SoC

'System on a Chip'. The main processor on the Raspberry Pi handles many functions of a computer in a single package.

SPI

Serial Peripheral Interface. A way for devices to communicate. The GPIO features an SPI interface.

SSH

'Secure Shell'. A way of accessing a command-line interface on a remote computer.

sudo

'Super/substitute user do'. Run a command with the highest-possible privileges.

Terminal

A program that gives access to the Raspbian command-line interface.

UART

'Universal asynchronous receiver-transmitter'. A very simple way for computers to communicate.

EASY HOME PROJECTS

Build something small and useful with your Raspberry Pi

MAGPI ISSUE 66

Discover a range of easy-to-build media player projects in *The MagPi* #66: **magpi.cc/66**

After setting up your Raspberry Pi and learning about all its features, the first thing you'll want to do is to build something useful. Nothing helps you learn like a project, and the Raspberry Pi can become all kinds of different things. We've picked three simple and fun projects to help you get started.

HOME THEATRE **PC**

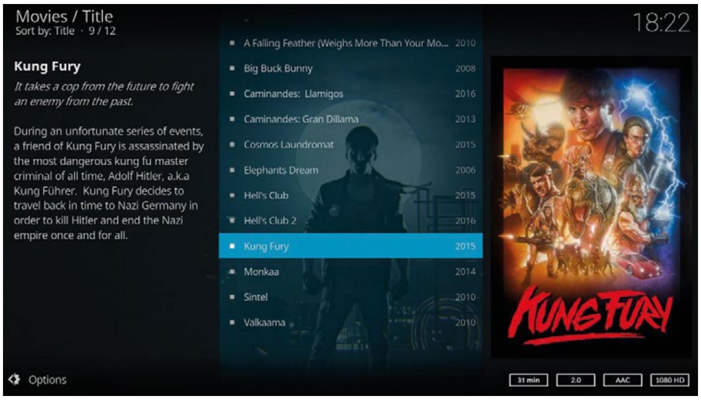
YOU'LL NEED

- Raspberry Pi (any model)
- LibreELEC (**libreelec.tv**)
- HDMI cable
- Power supply (**magpi.cc/power**)
- Remote control (**magpi.cc/cxudSB**)

01 Install LibreELEC

Kodi is our favourite media centre software. With Kodi you can play your own movie and music files. Kodi is available on the Raspberry Pi in a range of different operating systems. The best in our eyes is LibreELEC, available in NOOBS (New Out Of Box Software). Follow our Raspberry Pi QuickStart guide (**magpi.cc/quickstart**), but choose LibreELEC instead of Raspbian in Step 5.

£10 PROJECT



Movies / Title
Sort by: Title · 9 / 12

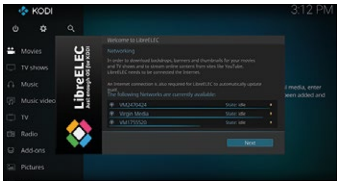
18:22

Kung Fury

It takes a cop from the future to fight an enemy from the past.

During an unfortunate series of events, a friend of Kung Fury is assassinated by the most dangerous kung fu master criminal of all time, Adolf Hitler, a.k.a Kung Führer. Kung Fury decides to travel back in time to Nazi Germany in order to kill Hitler and end the Nazi empire once and for all.

..	
A Falling Feather (Weighs More Than Your Mo...	2010
Big Buck Bunny	2008
Caminandes: Llamigos	2016
Caminandes: Gran Dillama	2013
Cosmos Laundromat	2015
Elephants Dream	2006
Hell's Club	2015
Hell's Club 2	2016
Kung Fury	2015
Monkaa	2014
Sintel	2010
Valkaama	2010



Welcome to LibreELEC

Networking

In order to download backdrops, banners and thumbnails for your movies and TV shows and to stream online content from sites like YouTube, LibreELEC needs to be connected the Internet.

An Internet connection is also required for LibreELEC to automatically update itself.

The following Networks are currently available:

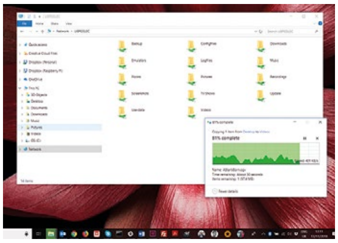
VM2470424	State: idle
Virgin Media	State: idle
VM1755520	State: idle

02 Set up Kodi

LibreELEC will boot up into the Kodi interface. For the setup process, it is useful to plug a keyboard into your Raspberry Pi (although you can use Kodi just fine with a remote control). Go through the Welcome process and connect to your local wireless LAN network (or use an Ethernet cable). Samba remote access should be activated by default; ensure it's set to on. Use SSH and take note of the IP address if you're using a Mac.

03 Copy movies across

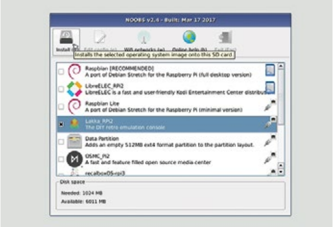
With Samba enabled, you can copy video files from a Windows PC to your Raspberry Pi pretty easily. Open a File Explorer window in Windows, and click on Network in the Navigation pane; double-click LIBREELEC. Drag and drop movie files to the Videos folder (and likewise for Music, TV Shows, and Pictures). Use Kodi to select the video and start playing it.



RETRO CONSOLE

01 RetroPie

Turning a Raspberry Pi into a retro games console is another easy (and fun) project. It's incredibly easy and good value too. You can set it up with a keyboard, but attach a game controller for a low-cost retro games console. Use NOOBS to set up a Raspberry Pi and make sure you have an internet connection (click 'Wifi networks' to join your network during installation). This enables you to access a wider range of operating systems, including Lakka_RPi2.



NOOBS v2.4 - Built: Mar 17 2017

Installs the selected operating system image onto this SD card

- □ Raspbian [RECOMMENDED] – A port of Debian Stretch for the Raspberry Pi (full desktop version)
- □ LibreELEC_RPi2 – LibreELEC is a fast and user-friendly Kodi Entertainment Center distribu…
- □ Raspbian Lite – A port of Debian Stretch for the Raspberry Pi (minimal version)
- ☒ Lakka_RPi2 – The DIY retro emulation console
- □ Data Partition – Adds an empty 512MB ext4 format partition to the partition layout.
- □ OSMC_Pi2 – A fast and feature filled open source media center
- recalboxOS-rpi3

Disk space
Needed: 1024 MB
Available: 6011 MB

02 Connect to network

Connect Lakka to your network. Using your gamepad (or a keyboard), navigate to Settings > WiFi and connect to your wireless LAN. Now navigate to Settings > Services and set SAMBA Enable to On.

03 Transfer ROM

Games are supplied separately as ROM files, and you must find your own. There are many homebrew games (modern games developed for classic consoles) and one of our favourites is Blade Buster, developed for NES hardware. You can get Blade Buster from the developer's website (**magpi.cc/bladebuster** – it's in Japanese, but click Download). The ROM is a zip file (do not unzip it). Copy the zip file into Network > Lakka > ROMs.

Get a wireless USB controller like this one from The PiHut **magpi.cc/gamepad**

04 Playing games

Now go Load Content > Start Directory and you should see the zip file you transferred at the bottom (ours is **BB_20120301.zip**). Select the file and choose Load Archive and NES / Famicom (FCEUmm). The game should load. Enjoy shooting the baddies in this modern classic vertical shoot-'em-up.

MAGPI ISSUE 67

Discover a wealth of retro goodness in *The MagPi* #67.
magpi.cc/67

Part 06

Code your own Pac-Man game

Pac-Man captured the hearts and pocket money of many young people in the eighties. Since then, it has made its way onto just about every computer system and console

MAKER

Mark Vanstone

Educational software author from the nineties, author of the ArcVenture series, disappeared into the corporate software wasteland. Rescued by the Raspberry Pi!

magpi.cc/YiZnxl

@mindexplorers

You'll Need

- Raspbian Jessie or newer
- An image manipulation program such as GIMP, or images available from **magpi.cc/nBSXKz**
- The latest version of Pygame Zero (1.2)
- USB joystick or gamepad (optional)

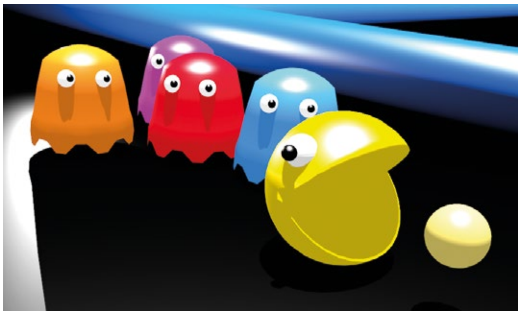
The concept of *Pac-Man* is quite simple. Pac-Man eats dots in a maze to score points. Avoid the ghosts unless you have just eaten a power-up, in which case ghosts are tasty. In this series we have gradually introduced new elements of Pygame Zero and also concepts around writing games. This is the first instalment in a two-part tutorial which will show you some more tricks to writing arcade games with Pygame Zero. We will also use some more advanced programming concepts to make our games even better. In this first part, we will put together the basics of the *Pac-Man* game and introduce the concept of adding extra Python modules to our program.

01 Let's get stuck in

As with the more recent episodes of this series, let's jump straight in, assuming that we have our basic Pygame Zero setup done. Let's set our window size to `WIDTH = 600` and `HEIGHT = 660`. This will give us room for a roughly square maze and a header area for some game information. We can get our gameplay area set up straight away by blitting two graphics – 'header' and 'colourmap' – to `0,0` and `0,80` respectively in the `draw()` function. You can make these graphics yourself or you can use ours, which can be found at **magpi.cc/nBSXKz**.

02 It's amazing

The original game had a very specific layout to the maze, but many different ones have appeared in later versions. The one we will be using is very similar to the original, but you can make your own design if you want. If you make your own, you'll also have to make two more maps (we'll come to those in a bit) which help with the running of the game. The main things about the map is that it has a central area where the ghosts start from and it doesn't have any other closed-in areas that the ghosts are likely to get trapped in (they can be a bit stupid sometimes).



▲ Because Pac-Man didn't involve shooting things, the Japanese designers thought it would appeal more to girls

03 Hmmm, pizza

Our next challenge is to get a player actor moving around the maze. For some unknown reason, the game's creator, Toru Iwatani, decided to make the main character a pizza that ate dots. Well, the eighties were a bit strange and that seemed perfectly reasonable at the time. We'll need two frames for our character: one with the mouth open and one with it closed. We can create our player actor near the top of the code using `player = Actor("pacman_o")`. This will create the actor with the mouth-open graphic. We will then set the actor's location in an `init()` function, as in previous programs.

04 Modulify to simplify

We can get our player onto the play area by setting `player.x = 290` and `player.y = 570` in the `init()` function and then call `player.draw()` in the `draw()` function, but to move the player character we'll need to get some input from the player. Previously we have used keyboard and mouse input, but this time we are going to have the option of joystick or gamepad input. Pygame Zero doesn't currently directly support gamepads, but we are going to borrow a bit of the Pygame module to get this working. We are also going to make a separate Python module for our input.

05 It's a joystick.init

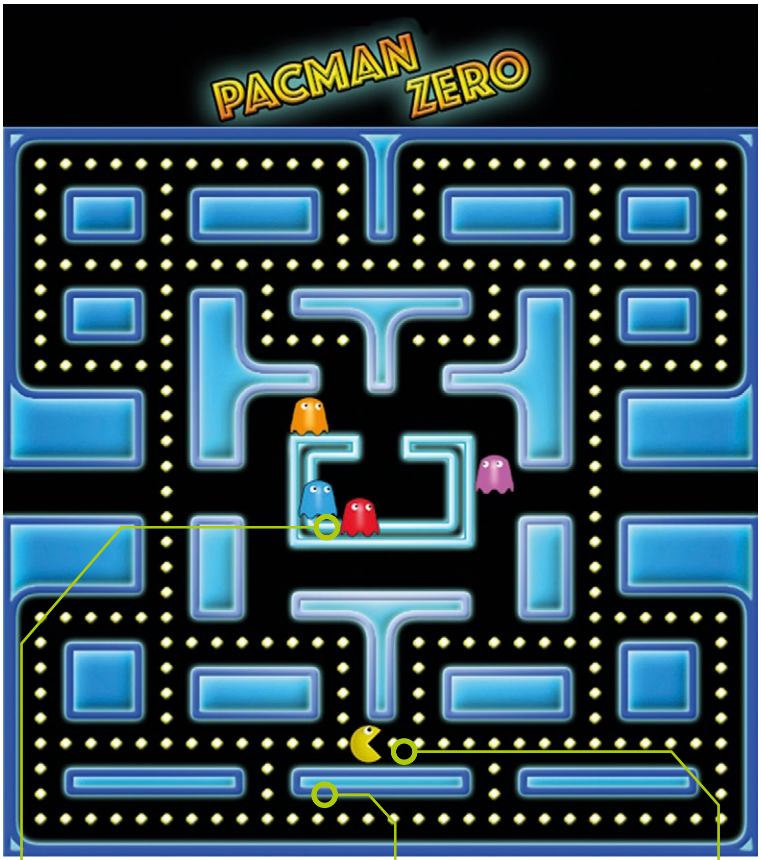
Setting up a new module is easy. All we need to do is make a new file, in this case **gameinput.py**, and in our main program at the top, write `import gameinput`. In this new file we can import the Pygame functions we need with `from pygame import joystick, key` and `from pygame.locals import *`. We can then initialise the Pygame joystick object (this also includes gamepads) by typing `joystick.init()`. We can find out how many joysticks or gamepads are connected by using `joystick_count = joystick.get_count()`. If we find any joysticks connected, we need to initialise them individually – see **figure1.py**.

06 Checking the input

We can now write a function in our gameinput module to check input from the player. If we define the function with `def checkInput(p):` we can get the x axis of a joystick using `joyin.get_axis(0)` and the y axis by using `joyin.get_axis(1)`. The numbers that are returned from these calls will be between -1 and +1, with 0 being the central position. We can check to see if the values are over 0.8 or under -0.8, as, depending on the device, we may not actually see -1 or 1 being returned. You may like to test this with your gamepad or joystick to see what range of values are returned.

07 Up, down, left, or right

The variable `p` that we are passing into our `checkInput()` function will be the player actor. We



The maze is made of corridors and maze walls

Ghosts move around the maze, looking for Pac-Man

The player is represented by the Pac-Man character that moves around the maze, eating dots

figure1.py

```
# gameinput Module

from pygame import joystick, key
from pygame.locals import *

joystick.init()
joystick_count = joystick.get_count()

if(joystick_count > 0):
    joyin = joystick.Joystick(0)
    joyin.init()
    # For the purposes of this tutorial
    # we are only going to use the first
    # joystick that is connected.
```

Top Tip

Modules

Using separate modules means not only is your code easier to follow, but it's easier for a team to work on.

figure2.py

```python
def checkInput(p):
    global joyin, joystick_count
    xaxis = yaxis = 0
    if joystick_count > 0:
        xaxis = joyin.get_axis(0)
        yaxis = joyin.get_axis(1)
    if key.get_pressed()[K_LEFT] or xaxis < -0.8:
        p.angle = 180
        p.movex = -20
    if key.get_pressed()[K_RIGHT] or xaxis > 0.8:
        p.angle = 0
        p.movex = 20
    if key.get_pressed()[K_UP] or yaxis < -0.8:
        p.angle = 90
        p.movey = -20
    if key.get_pressed()[K_DOWN] or yaxis > 0.8:
        p.angle = 270
        p.movey = 20
```

figure3.py

```python
# inside update() function

    if player.movex or player.movey:
        inputLock()
        animate(player, pos=(player.x + player.movex, player.y + player.movey), duration=1/SPEED, tween='linear', on_finished=inputUnLock)

# outside update() function

def inputLock():
    global player
    player.inputActive = False

def inputUnLock():
    global player
    player.movex = player.movey = 0
    player.inputActive = True
```

can test each of the directions of the joystick at the same time as the keyboard and then set the player angle (so that it points in the correct direction for movement) and also how much it needs to move. We'll set these by saying (for example, if the left arrow is pressed or the joystick is moved to the left) `if key.get_pressed()[K_LEFT] or xaxis < -0.8:` and then `p.angle = 180` and `p.movex = -20`. See **figure2.py** for the full `checkInput()` function.

▲ You can plug a gamepad or joystick into one of the USB ports on your Raspberry Pi

08 Get a move on!

Now we have our input function set up, we can call it from the `update()` function. Because this function is in a different module, we need to prefix it with the module name. In the `update()` function we write `gameinput.checkInput(player)`. After this function has been called, if there has been any input, we should have some variables set in the player actor that we can use to move. We can say `if player.movex or player.movey:` and then use the `animate()` function to move by the amount specified in `player.movex` and `player.movey`.

09 Hold your horses

The way we have the code at the moment means that any time there is some input, we fire off a new animation. This will soon mean that layers of animation get called over the top of each other, but what we want is for the animation to run and then start looking for new input. To do this we need an input locking system. We can call an input lock function before the move and then wait for the animation to finish before unlocking to look for more input. Look at **figure3.py** to see how we can make this locking system.

10 You can't just move anywhere

Now, here comes the interesting bit. We want our player actor to move around the maze, but at the moment it will go though the walls and even off the screen. We need to restrict the movement only to the corridors of the maze. There are several different ways we could do this, but for this game we're going to have an image map marking the areas that the player actor can move within. The map will be a black and white one, showing just the corridors as black and the walls as white. We will then look at the map in the direction we want to move and see if it is black; if it is, we can move.

11 Testing the map

To be able to test the colour of a part of an image, we need to borrow a few functions from Pygame again. We'll also put our map functions in a separate module. So make a new Python file and call it **gamemaps.py** and in it we'll write `from pygame import image, Color`.

We must also load in our movement map, which we need to do in the Pygame way: `moveimage = image.load('images/pacmanmovemap.png')`. Then all we need to do is write a function to check that the direction of the player is valid. See **figure4.py** for this function.

12 Using the movemap

To use this new module, we need to `import gamemaps` at the top of our main code file and then, before we animate the player (but after we have checked for input), we can call `gamemaps.checkMovePoint(player)`,which will zero the `movex` and `movey` variables of the player if the move is not possible. So now we should find that the player actor can only move inside the corridors. We do have one special case that you may have noticed in **figure4.py**, and that is because there is one corridor where the player can move from one side of the screen to the other.

13 You spin me round

There is one more aspect to the movement of the player actor, and that is the animation. As Pac-Man moves, the mouth opens and shuts and points in the direction of the movement. The mouth opening and closing is easy enough: we have an image for open and one for closed and alternate between the two. For pointing in the correct direction, we can rotate the player actor. Unfortunately, this has a slight problem that Pac-Man will be upside-down when moving left. So we just need to have one version that is switched the other way round. See **figure5.py** for a function that sorts out all of this.

figure4.py

```
# gamemaps module
from pygame import image, Color
moveimage = image.load('images/pacmanmovemap.png')

def checkMovePoint(p):
    global moveimage
    if p.x+p.movex < 0: p.x = p.x+600
    if p.x+p.movex > 600: p.x = p.x-600
    if moveimage.get_at((int(p.x+p.movex), int(p.y+p.movey-80))) != Color('black'):
        p.movex = p.movey = 0
```

figure5.py

```
def getPlayerImage():
    global player
    # we need to import datetime at the top of our code
    dt = datetime.now()
    a = player.angle
    # this next line will give us a number between
    # 0 and 5 depending on the time and SPEED
    tc = dt.microsecond%(500000/SPEED)/(100000/SPEED)
    if tc > 2.5 and (player.movex != 0 or player.movey !=0):
        # this is for the closed mouth images
        if a != 180:
            player.image = "pacman_c"
        else:
            # reverse image if facing left
            player.image = "pacman_cr"
    else:
        # this is for the open mouth images
        if a != 180:
            player.image = "pacman_o"
        else:
            player.image = "pacman_or"
    # set the angle on the player actor
    player.angle = a
```

Top Tip

Pygame

Pygame Zero is based on Pygame, but if you want to use some of the Pygame functions, best to do it in a separate module to avoid confusion.

figure6.py

```
# This goes in the main code file.

def initDots():
    global pacDots
    pacDots = []
    a = x = 0
    while x < 30:
        y = 0
        while y < 29:
            if gamemaps.checkDotPoint(10+x*20, 10+y*20):
                pacDots.append(Actor("dot",(10+x*20, 90+y*20)))
                pacDots[a].status = 0
                a += 1
            y += 1
        x += 1

# This goes in the gamemaps module file.

dotimage = image.load('images/pacmandotmap.png')

def checkDotPoint(x,y):
    global dotimage
    if dotimage.get_at((int(x), int(y))) == Color('black'):
        return True
    return False

# This bit goes in the draw() function.

    pacDotsLeft = 0
    for a in range(len(pacDots)):
        if pacDots[a].status == 0:
            pacDots[a].draw()
            pacDotsLeft += 1
        if pacDots[a].collidepoint((player.x, player.y)):
            pacDots[a].status = 1
    # if there are no dots left, the player has won
    if pacDotsLeft == 0: player.status = 2
```

14 Spot on

So when we have put in a call to `getPlayerImage()` just before we draw the player actor, we should have Pac-Man moving around, chomping and pointing in the correct direction. Now we need something to chomp. We are going to create a set of dots at even spacings along most of the corridors. An easy way to do this is to use a similar technique that we're using for testing where the corridors are. If we make an image map of the places the dots need to go and loop over the whole map, only placing dots where it is black, we can get the desired effect.

15 Tasty, tasty dots

To get our dots doing their thing, we'll need to code a few things. We need to initialise actors for each dot, we need to draw each dot, and if the player eats the dot, we need to stop drawing it; **figure6.py** shows how we can do each of these jobs. We need `initDots()`, we need to add another function to **gamemaps.py** to work out where to position the dots, and we need to add some drawing code to the `draw()` function. In addition to the code in **figure6.py**, we need to add a call to `initDots()` in our `init()` function.

16 I ain't afraid of no ghosts

Now that we have our Pac-Man happily munching dots, we must introduce our villains to the mix. In the original game, the ghosts had names; in the English version they were known as Blinky, Pinky, Inky, and Clyde. They roam the maze looking for Pac-Man, starting from an enclosure in the centre of the map. We can initialise each ghost as an actor to appear at the centre of the maze and keep them in a list called `ghosts[]`. To start off with, we'll just make them move around randomly. The way we can do this is to set a random direction (`ghosts[g].dir`) for each and then keep them moving until they hit a wall.

17 Random motion

We can use the same system that we used to check player movement for the ghosts. Each time we move a ghost – `moveGhosts()` – we can get a list of which directions are available to it. If the current direction (`ghosts[g].dir`) is not available, then we randomly pick another direction until we find one that we can move in. We can also have a random occurrence of changing direction, just to make it a bit less predictable – and if the ghosts collide with each other, we could do the same. When we have moved the ghosts with the `animate()` function, we get it to count how many ghosts have finished moving. When they are all done, we can call the `moveGhosts()` function again.

Top Tip

Animations

When using the `animate()` function, it is best to use the callback function to see when it has finished, as different systems may work at different speeds.

18 Look like a ghost

The last thing to do with our ghosts is to actually draw them to the screen. We can create a function called `drawGhosts()` where we loop through the four ghosts and draw them to the screen. One of the details of the original game was that the eyes of the ghosts would follow the player; we can do this by setting the ghost image to reverse if the player is to the left of the ghost. We have numbered images so that ghost one is **ghost1.png** and ghost two is **ghost2.png**, etc. Have a look at the full **pacman1.py** program listing to see all the functions that make the ghosts work.

19 Game over

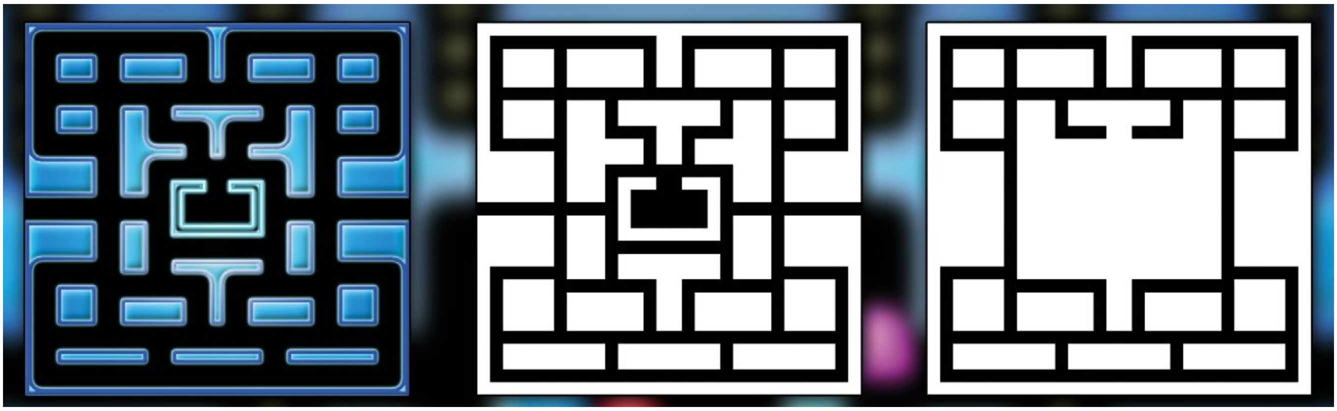
Of course, we need to deal with the end-of-the-game conditions and, as before, we can use a status variable. In this case we have previously set `player.status = 2` if the player wins. We can check to see if a ghost collides with the player and set `player.status = 1`. Then we just need to display some text in the `draw()` function based on this variable. And that's it for part one. In the next part we'll be giving the ghosts more brains, adding levels, lives, and power-ups – and adding some sweet, soothing music and sound effects.

gameinput.py

Language: **Python 3**

```python
# gameinput Module

from pygame import joystick, key
from pygame.locals import *

joystick.init()
joystick_count = joystick.get_count()

if(joystick_count > 0):
    joyin = joystick.Joystick(0)
    joyin.init()

def checkInput(p):
    global joyin, joystick_count
    xaxis = yaxis = 0
    if joystick_count > 0:
        xaxis = joyin.get_axis(0)
        yaxis = joyin.get_axis(1)
    if key.get_pressed()[K_LEFT] or xaxis < -0.8:
        p.angle = 180
        p.movex = -20
    if key.get_pressed()[K_RIGHT] or xaxis > 0.8:
        p.angle = 0
        p.movex = 20
    if key.get_pressed()[K_UP] or yaxis < -0.8:
        p.angle = 90
        p.movey = -20
    if key.get_pressed()[K_DOWN] or yaxis > 0.8:
        p.angle = 270
        p.movey = 20
```

Three maps are used: one which we see, one to check possible movements, and one to check where dots are to be placed



Colour Map

Movement Map

Dot Location Map

gamemaps.py

▸ Language: **Python 3**

```python
# gamemaps module

from pygame import image, Color
moveimage = image.load('images/pacmanmovemap.png')
dotimage = image.load('images/pacmandotmap.png')

def checkMovePoint(p):
    global moveimage
    if p.x+p.movex < 0: p.x = p.x+600
    if p.x+p.movex > 600: p.x = p.x-600
    if moveimage.get_at((int(p.x+p.movex), int(p.y+p.movey-80))) != Color('black'):
        p.movex = p.movey = 0

def checkDotPoint(x,y):
    global dotimage
    if dotimage.get_at((int(x), int(y))) == Color('black'):
        return True
    return False

def getPossibleDirection(g):
    global moveimage
    if g.x-20 < 0:
        g.x = g.x+600
    if g.x+20 > 600:
        g.x = g.x-600
    directions = [0,0,0,0]
    if g.x+20 < 600:
        if moveimage.get_at((int(g.x+20), int(g.y-80))) == Color('black'): directions[0] = 1
    if g.x < 600 and g.x >= 0:
        if moveimage.get_at((int(g.x), int(g.y-60))) == Color('black'): directions[1] = 1
    if g.x-20 >= 0:
        if moveimage.get_at((int(g.x-20), int(g.y-80))) == Color('black'): directions[2] = 1
    if g.x < 600 and g.x >= 0:
        if moveimage.get_at((int(g.x), int(g.y-100))) == Color('black'): directions[3] = 1
    return directions
```

pacman1.py

▸ Language: **Python 3**

```python
import pgzrun
import gameinput
import gamemaps
from random import randint
from datetime import datetime
WIDTH = 600
HEIGHT = 660

player = Actor("pacman_o") # Load in the player Actor image
SPEED = 3

def draw(): # Pygame Zero draw function
    global pacDots, player
    screen.blit('header', (0, 0))
    screen.blit('colourmap', (0, 80))
    pacDotsLeft = 0
    for a in range(len(pacDots)):
        if pacDots[a].status == 0:
            pacDots[a].draw()
            pacDotsLeft += 1
        if pacDots[a].collidepoint((player.x, player.y)):
            pacDots[a].status = 1
    if pacDotsLeft == 0: player.status = 2
    drawGhosts()
    getPlayerImage()
    player.draw()
    if player.status == 1: screen.draw.text("GAME OVER", center=(300, 434), owidth=0.5, ocolor=(255,255,255), color=(255,64,0) , fontsize=40)
    if player.status == 2: screen.draw.text("YOU WIN!", center=(300, 434), owidth=0.5, ocolor=(255,255,255), color=(255,64,0) , fontsize=40)

def update(): # Pygame Zero update function
    global player, moveGhostsFlag, ghosts
    if player.status == 0:
        if moveGhostsFlag == 4: moveGhosts()
        for g in range(len(ghosts)):
            if ghosts[g].collidepoint((player.x, player.y)):
                player.status = 1
                pass
        if player.inputActive:
            gameinput.checkInput(player)
            gamemaps.checkMovePoint(player)
            if player.movex or player.movey:
                inputLock()
                animate(player, pos=(player.x + player.movex, player.y + player.movey), duration=1/SPEED, tween='linear', on_finished=inputUnLock)

def init():
```

DOWNLOAD THE FULL CODE:

magpi.cc/jefHFU

```
    global player
    initDots()
    initGhosts()
    player.x = 290
    player.y = 570
    player.status = 0
    inputUnLock()

def getPlayerImage():
    global player
    dt = datetime.now()
    a = player.angle
    tc = dt.microsecond%(500000/SPEED)/(100000/SPEED)
    if tc > 2.5 and (player.movex != 0 or player.movey !=0):
        if a != 180:
            player.image = "pacman_c"
        else:
            player.image = "pacman_cr"
    else:
        if a != 180:
            player.image = "pacman_o"
        else:
            player.image = "pacman_or"
    player.angle = a

def drawGhosts():
    for g in range(len(ghosts)):
        if ghosts[g].x > player.x:
            ghosts[g].image = "ghost"+str(g+1)+"r"
        else:
            ghosts[g].image = "ghost"+str(g+1)
        ghosts[g].draw()

def moveGhosts():
    global moveGhostsFlag
    dmoves = [(1,0),(0,1),(-1,0),(0,-1)]
    moveGhostsFlag = 0
    for g in range(len(ghosts)):
        dirs = gamemaps.getPossibleDirection(ghosts[g])
        if ghostCollided(ghosts[g],g) and randint(0,3) == 0: ghosts[g].dir = 3
        if dirs[ghosts[g].dir] == 0 or randint(0,50) == 0:
            d = -1
            while d == -1:
                rd = randint(0,3)
                if dirs[rd] == 1:
                    d = rd
            ghosts[g].dir = d
        animate(ghosts[g], pos=(ghosts[g].x + dmoves[ghosts[g].dir][0]*20, ghosts[g].y + dmoves[ghosts[g].dir][1]*20), duration=1/SPEED, tween='linear', on_finished=flagMoveGhosts)

def flagMoveGhosts():
    global moveGhostsFlag
    moveGhostsFlag += 1

def ghostCollided(ga,gn):
    for g in range(len(ghosts)):
        if ghosts[g].colliderect(ga) and g != gn:
            return True
    return False

def initDots():
    global pacDots
    pacDots = []
    a = x = 0
    while x < 30:
        y = 0
        while y < 29:
            if gamemaps.checkDotPoint(10+x*20, 10+y*20):
                pacDots.append(Actor("dot",(10+x*20, 90+y*20)))
                pacDots[a].status = 0
                a += 1
            y += 1
        x += 1

def initGhosts():
    global ghosts, moveGhostsFlag
    moveGhostsFlag = 4
    ghosts = []
    g = 0
    while g < 4:
        ghosts.append(Actor("ghost"+str(g+1),(270+(g*20), 370)))
        ghosts[g].dir = randint(0, 3)
        g += 1

def inputLock():
    global player
    player.inputActive = False

def inputUnLock():
    global player
    player.movex = player.movey = 0
    player.inputActive = True

init()
pgzrun.go()
```

CHRISTMAS PROJECT

Hack smart lights

On and off? Dimming? We can do better than that. Take control of your lights with a Pi and link them to just about anything!

MAKER

PJ Evans

PJ Evans is a writer, software engineer, and Jam organiser. His Pi now controls more of his house than he does.

mrpjevans.com

@mrpjevans

Smart lighting has been around for a few years now and, in terms of reliability and functionality, the technology has come a long way. Best of all, the cost of implementation has plummeted. Not only are smart lights affordable, there's a great range of products to choose from and some are hackable! Using IKEA's Trådfri range, we're going to improve our smart lighting setup by adding some Pi magic, and then linking the lights to the outside world to create a bit of Christmas fun.

01 Zig-a-Zig(Bee)-Ah

The Trådfri range uses the popular ZigBee radio system to communicate. No central hub is needed for basic light switching – you just use the provided remotes. That's not the case if we want to hook things up to a Raspberry Pi or the internet. To allow smartphones to control lights, IKEA has released an inexpensive gateway that bridges the gap between ZigBee and your network. All it requires is USB power (charger included) and a wired connection to your network. Now, install the Trådfri app (iOS and Android) on your phone and follow the instructions.

02 Lighten up

As you work through the app's setup procedure, you will be asked if you want to configure your devices. You'll need to plug your light in now. All Trådfri lights have Edison screws – if you need an adapter for bayonet fittings, IKEA

You'll Need

- IKEA Trådfri Colour Smart Light **magpi.cc/zFhgHN**
- IKEA Trådfri Gateway **magpi.cc/AUqvsQ**
- IKEA KOPPLA Edison screw to bayonet converter (optional) **magpi.cc/kSMDAC**

Here's everything you need: a smart light, internet gateway, remote control, and, optionally, a socket adapter

sells these too for a princely £1. You'll also need the remote control provided to sync the light up with both it and the gateway. It's a reasonably painless process, and by the end you should be able to control the light with your phone as well as the remote. Don't proceed until you've confirmed this.

03 Be prepared

Once you've finished changing the light to all the colours of the rainbow, make sure you've got a Pi ready to go. You're only going to be working on the command line, so Raspbian Stretch Lite is perfect. If you've got Raspberry Pi Desktop for x86, any popular Linux distribution, or even macOS, the following instructions will work on these platforms too. This is a zero-solder project and we're not using the GPIO this time. All that's needed is a network connection. Just make sure you've done `sudo apt update && sudo apt upgrade` before you start.

04 We're all going on a gateway hunt

Before we can talk to the gateway, we need to find it, and that means we need its IP address. Probably the easiest place to discover it is by accessing your router's webpage, as most routers display a list of connected devices. Otherwise, Fing, a free app for Android and iOS, scans your network and displays the IP address of everything it finds. You're looking for a device with the name GW-XXXXXXXXXXXX, where the value of the Xs can be seen on the underside of the gateway, under 'MAC'. Fing will also display 'TRADFRI-Gateway'.

05 Learning to CoAP

The Trådfri gateway speaks CoAP, a protocol designed for the Internet of Things. It's lightweight, fast, small, and capable of communicating with millions of devices. Raspbian does not speak CoAP, so we need to teach it. From the command line, we'll download some libraries, compile, and install them. Enter the following, line-by-line from your home directory:

```
 sudo apt install git automake libtool python3-pip
 git clone --depth 1 --recursive -b dtls https://github.com/home-assistant/libcoap.git
 cd libcoap
 ./autogen.sh
 ./configure --disable-documentation --disable-shared --without-debug CFLAGS="-D COAP_DEBUG_FD=stderr"
 make
 sudo make install
 pip3 install pytradfri
```

Top Tip

Don't like meatballs?

There are many other hackable systems on sale, such as Philips Hue. If Trådfri doesn't meet your needs, shop around!

06 Open a communications channel

We've now installed libcoap and pytradfri, a Python 3 library for talking CoAP to the gateway. This library, written and maintained by 'ggravlingen' on GitHub, allows us to talk with the various devices easily and without knowing anything about the underlying protocols at work. It also comes with a testing script, so let's start with that.

```
python3 -i -m pytradfri <IP-Address>
```

Replace '<IP-Address>' with the address of the gateway, as discovered in Step 4. You'll be asked the security code on the base of the gateway, after which you'll see a list of available commands and the familiar Python prompt.

07 It's time to light the lights

Let's test the light. At the Python prompt (>>>), type...

```
lights
```

...followed by **ENTER**. All being well, a description of your light will be displayed as part of an array. Seeing this confirms that your Pi is talking to the gateway! Now, let's see if we can command our light. Make sure the light is on and enter the following at the Python prompt:

```
api(light.light_control.set_dimmer(0))
```

This instructs all light devices connected to the gateway to switch off. Illuminate them again with...

```
api(light.light_control.set_dimmer(254))
```

254 is the maximum value for the dimmer.

08 Coding light

Time for some code. When you entered your security code in the previous step, a file was created called **tradfri_standalone_psk.conf**. This contains your unique password for communicating with the gateway. Create all the following scripts in the same directory as this file.

In the script shown here, change the IP address to match your gateway and run using Python 3:

```
python3 brighten.py
```

The code locates the password in the saved file and then interrogates the gateway to discover available devices. Once we have our lights array, a simple loop takes our light from minimum to maximum brightness.

09 See the light

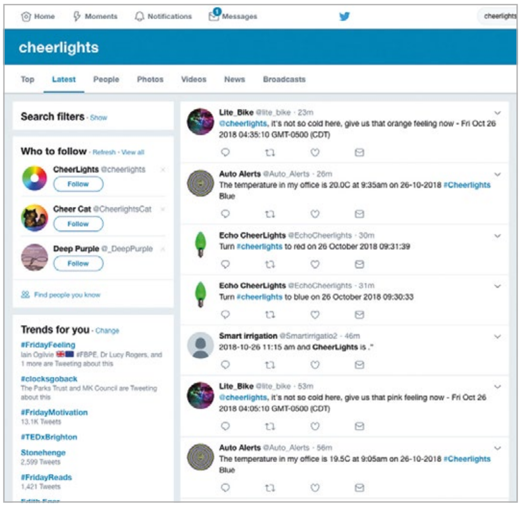
We can also monitor the state of Trådfri devices, including lights. The pytradfri library supports requests on the state of devices, including whether they are on or off and their current brightness level. A separate thread monitors the gateway for changes and keeps our variables up to date.

Download this script: **magpi.cc/qDqTXf**. Change the IP address and run it in the same directory as the .conf file. Now watch the output while you adjust the brightness using the remote control. As the Trådfri range also includes PIR sensors, you've got the beginnings of a home security system.

Top Tip

Use the GPIO for more fun

Connect any number of buttons or sensors to your GPIO pins and have your house lights react.



Lite_Bike @lite_bike · 23m
@cheerlights, it's not so cold here, give us that orange feeling now - Fri Oct 26 2018 04:35:10 GMT-0500 (CDT)

Auto Alerts @Auto_Alerts · 26m
The temperature in my office is 20.0C at 9:35am on 26-10-2018 #Cheerlights Blue

Echo CheerLights @EchoCheerlights · 30m
Turn #cheerlights to red on 26 October 2018 09:31:39

Echo CheerLights @EchoCheerlights · 31m
Turn #cheerlights to blue on 26 October 2018 09:30:33

Smart irrigation @Smartirrigatio2 · 46m
2018-10-26 11:15 am and CheerLights is ."

Lite_Bike @lite_bike · 53m
@cheerlights, it's not so cold here, give us that pink feeling now - Fri Oct 26 2018 04:05:10 GMT-0500 (CDT)

Auto Alerts @Auto_Alerts · 56m
The temperature in my office is 19.5C at 9:05am on 26-10-2018 #Cheerlights Blue

Connect your lights to the outside world and learn about APIs with the fun Cheerlights community

10 Some Christmas cheer

You can link your light to anything you like. As it is Christmas, we're going use Cheerlights (**cheerlights.com**), a service that allows Twitter users to set the colour on thousands of light devices around the world. It's also a great way to learn about APIs, thanks to its simplistic design.

Download the code from **magpi.cc/ettkzk** to the same directory as the .conf file from earlier. Change the IP address, run the script, and watch the light change colour. The code requests the current colour from the Cheerlights API at regular intervals and sets the light to match. Tweet a colour to **@cheerlights** and change it yourself!

11 Let the light flow

Most actions in home automation come down to basic cause and effect: 'if this happens, do this thing'. So it's no surprise that one of the most popular automation services is called IFTTT ('If This Then That'). IFTTT supports a huge range of internet services and IoT devices, acting as a 'middleware' to connect things together. Using its 'Webhooks' service, you can trigger any of IFTTT's actions. A Webhook is a unique URL that, when called, triggers an action, whether that's sending an email or switching on a coffee machine. Here's some example code to call an IFTTT Webhook: **magpi.cc/CSCdNK**.

12 Going further

Trådfri plus Raspberry Pi plus IFTTT opens up a world of possibilities. Trådfri isn't just for lighting: there are PIR sensors now available and smart sockets are on the way. The lights are also available in a range of sizes, perfect for projects involving bedside lamps or kitchen lighting. Whether you want to make a mood detector, ambient lighting system or add home security, this is a great low-cost way to start. IKEA is openly supporting the community and encouraging people to make use of the Trådfri range. Of course, there's plenty of scope for fun and mischief too!

brighten.py

Language: **Python 3**

DOWNLOAD THE FULL CODE:

magpi.cc/ntxpZk

```python
from pytradfri import Gateway
from pytradfri.api.libcoap_api import APIFactory
from pytradfri.util import load_json, save_json
from time import sleep

# Change this IP address to your gateway
IP_ADDRESS = '192.168.0.158'

# Make sure you're in the same directory as this file
CONFIG_FILE = 'tradfri_standalone_psk.conf'

# Load in the file, get our password for the gateway and create an API
conf = load_json(CONFIG_FILE)
identity = conf[IP_ADDRESS].get('identity')
psk = conf[IP_ADDRESS].get('key')
api_factory = APIFactory(host=IP_ADDRESS, psk_id=identity, psk=psk)

# This section connects to the gateway and gets information on devices
api = api_factory.request
gateway = Gateway()
devices_command = gateway.get_devices()
devices_commands = api(devices_command)
devices = api(devices_commands)

# Create an array of objects that are lights
lights = [dev for dev in devices if dev.has_light_control]

# Brightness ranges from 0 to 254, so lets set the light in stages of 5
for brightness in range(0, 255, 5):

    # Set the first light to 'brightness' and then pause for a moment
    api(lights[0].light_control.set_dimmer(brightness))
    sleep(0.1)
```

Start and stop at your command

AUTHOR

Richard Smedley

Having often found words better than pointing at things, Richard stuck with the command line whence all around had fled.

@RichardSmedley

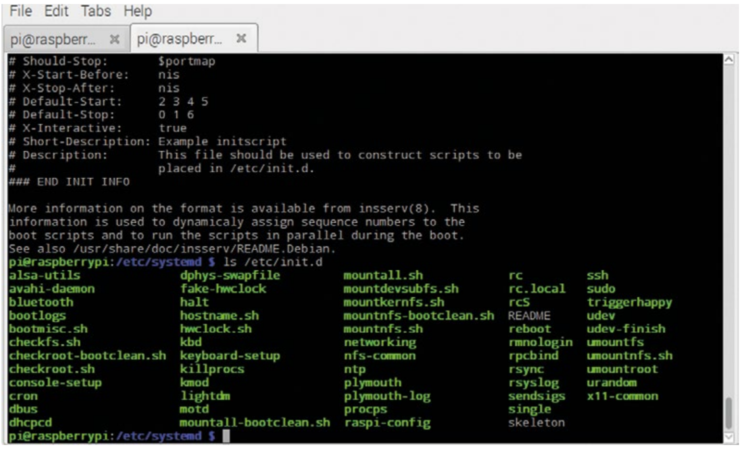
Use the command line to start and stop services, and auto-run them at startup

Over the last few years, every major GNU/Linux distribution – Raspbian included – has changed the way that it starts up. This means that much of the older literature on bookshelves and websites, dealing with where to put files in **rc.local** to get them to automatically run on startup, is no longer correct – unless you've yet to upgrade from Raspbian Wheezy.

One thing remains the same. Although the first process the kernel starts is not **/sbin/init**, but **/lib/systemd/systemd** (still with a PID of 1), it is still the parent process of everything that happens in user space once the Linux kernel has finished initialising devices and drivers, and mounted the file system.

The init process gets everything else started, and usually ends with the prompt inviting you to log in to your Pi. Recent versions of Raspbian hide most of the messages that this startup process generates, but you can see them by typing `dmesg`. They're also stored in **/var/log/kern.log**.

Startup – init – is the start of user space; this is the place where you can put your own programs to affect how the Pi runs, without having to modify the code of your Linux kernel! Traditionally, GNU/Linux distributions implemented a version of the UNIX System V init, which had a well-defined startup process with run levels that would indicate whether the system was at startup, 'single user mode' (rescue mode – a handy way to get back in when you've lost your password, or a security headache), console mode, the GUI, or heading for shutdown.



```
# Should-Stop:         $portmap
# X-Start-Before:      nis
# X-Stop-After:        nis
# Default-Start:       2 3 4 5
# Default-Stop:        0 1 6
# X-Interactive:       true
# Short-Description: Example initscript
# Description:         This file should be used to construct scripts to be
#                      placed in /etc/init.d.
### END INIT INFO

More information on the format is available from insserv(8).  This
information is used to dynamicaly assign sequence numbers to the
boot scripts and to run the scripts in parallel during the boot.
See also /usr/share/doc/insserv/README.Debian.
pi@raspberrypi:/etc/systemd $ ls /etc/init.d
alsa-utils              dphys-swapfile         mountall.sh            rc         ssh
avahi-daemon            fake-hwclock           mountdevsubfs.sh       rc.local   sudo
bluetooth               halt                   mountkernfs.sh         rcS        triggerhappy
bootlogs                hostname.sh            mountnfs-bootclean.sh  README     udev
bootmisc.sh             hwclock.sh             mountnfs.sh            reboot     udev-finish
checkfs.sh              kbd                    networking             rmnologin  umountfs
checkroot-bootclean.sh  keyboard-setup         nfs-common             rpcbind    umountnfs.sh
checkroot.sh            killprocs              ntp                    rsync      umountroot
console-setup           kmod                   plymouth               rsyslog    urandom
cron                    lightdm                plymouth-log           sendsigs   x11-common
dbus                    motd                   procps                 single
dhcpcd                  mountall-bootclean.sh  raspi-config           skeleton
pi@raspberrypi:/etc/systemd $
```

Backwards compatibility with init.d scripts – and even run levels – is maintained by systemd

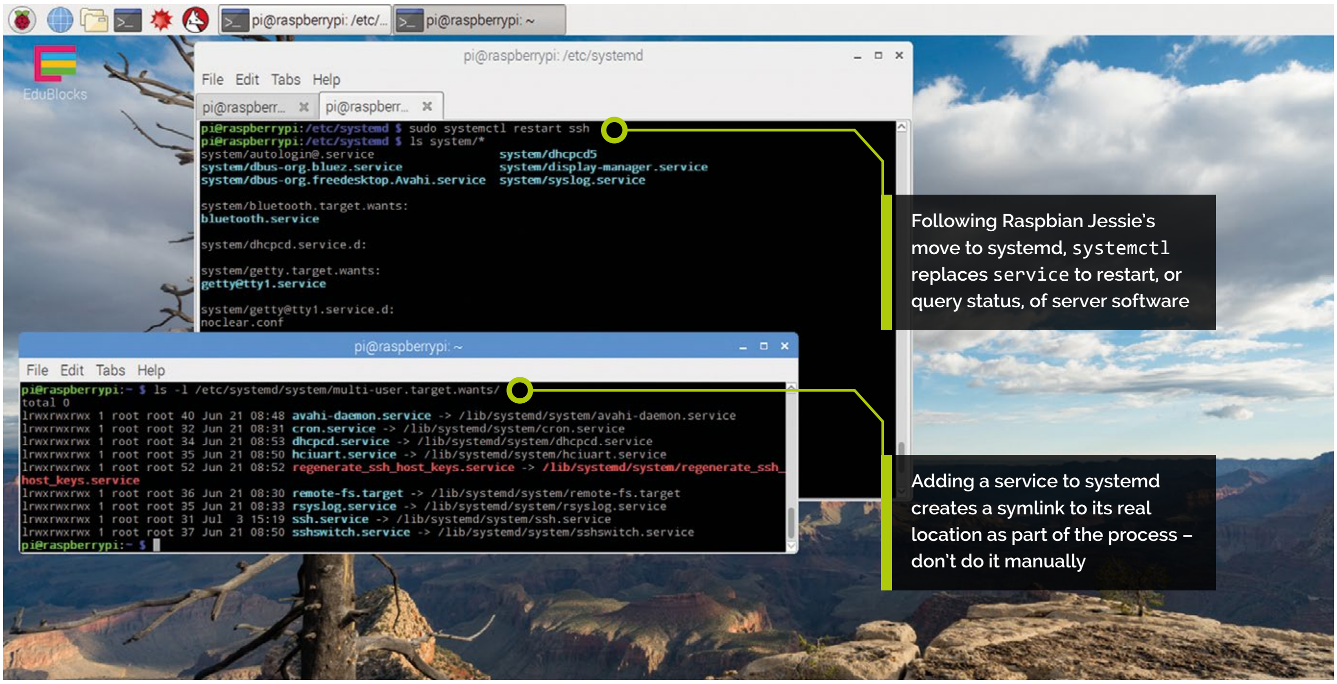
Along the way, files in **/etc** with names beginning rc0 through rc6 get called – running startup scripts in **/etc/init.d** – and **/etc/rcS.d**, which contains files always called at startup, regardless of run level. Those **/etc/init.d** scripts can be called directly to start or stop your databases, web servers, or anything else that needs intervention. Many support further commands such as `status`, to check a service is running properly, and `reload` – the latter is useful if you want a service to take a look at fresh config settings without doing a full restart.

```
sudo /etc/init.d/couchbase-server status
```

Although the regular and predictable scripted startup of Sys-V init makes it easy to place your own programs in the startup process – particularly useful on an unattended Pi – the performance of a purely sequential startup process is poor, even when booting from a solid-state drive. Enter systemd…

Systemd

Systemd can start services in parallel, and can defer service starts until they are needed. Rather than many scripts for individual components, a target is set, and systemd resolves the dependencies until it reaches that target,



```
pi@raspberrypi:/etc/systemd $ sudo systemctl restart ssh
pi@raspberrypi:/etc/systemd $ ls system/*
system/autologin@.service                  system/dhcpcd5
system/dbus-org.bluez.service              system/display-manager.service
system/dbus-org.freedesktop.Avahi.service  system/syslog.service

system/bluetooth.target.wants:
bluetooth.service

system/dhcpcd.service.d:

system/getty.target.wants:
getty@tty1.service

system/getty@tty1.service.d:
noclear.conf
```

```
pi@raspberrypi:~ $ ls -l /etc/systemd/system/multi-user.target.wants/
total 0
lrwxrwxrwx 1 root root 40 Jun 21 08:48 avahi-daemon.service -> /lib/systemd/system/avahi-daemon.service
lrwxrwxrwx 1 root root 32 Jun 21 08:31 cron.service -> /lib/systemd/system/cron.service
lrwxrwxrwx 1 root root 34 Jun 21 08:53 dhcpcd.service -> /lib/systemd/system/dhcpcd.service
lrwxrwxrwx 1 root root 35 Jun 21 08:50 hciuart.service -> /lib/systemd/system/hciuart.service
lrwxrwxrwx 1 root root 52 Jun 21 08:52 regenerate_ssh_host_keys.service -> /lib/systemd/system/regenerate_ssh_
host_keys.service
lrwxrwxrwx 1 root root 36 Jun 21 08:30 remote-fs.target -> /lib/systemd/system/remote-fs.target
lrwxrwxrwx 1 root root 35 Jun 21 08:33 rsyslog.service -> /lib/systemd/system/rsyslog.service
lrwxrwxrwx 1 root root 31 Jul  3 15:19 ssh.service -> /lib/systemd/system/ssh.service
lrwxrwxrwx 1 root root 37 Jun 21 08:50 sshswitch.service -> /lib/systemd/system/sshswitch.service
pi@raspberrypi:~ $
```

Following Raspbian Jessie's move to systemd, `systemctl` replaces `service` to restart, or query status, of server software

Adding a service to systemd creates a symlink to its real location as part of the process – don't do it manually

avoiding any fixed startup sequence along the way. Files are found under **/etc/systemd/system** and there's a lot to learn, but as Raspbian now starts up far more quickly, at least we've created extra time for all that learning. The one thing to remember for any user is that systemd and its service manager are controlled with the `systemctl` command:

```
sudo systemctl restart ssh
```

...will restart the SSH server – something you'll need to do if you change the port it listens on, for example. For compatibility, as well as **/etc/init.d** scripts to start and stop services, the system of service commands that worked on older versions of Raspbian, such as `sudo service apache2 reload`, still works (here we ask a running Apache 2 web server to reread its configuration files).

In **/etc/systemd/system/multi-user.target.wants** you'll find files like **openvpn.service** which, when examined closely with `ls -l`, you'll see are links to files of the same name in **/lib/systemd/system** (other GNU/Linux distributions may place the files under **/usr/lib**). Don't worry if what's inside these files looks confusing; there's a logic to them with their conditional dependencies, but you can safely forget about them until you need to get some software working automatically on every system restart for your Pi project. Even then, we'll show you another way with crontab – otherwise you'll need to be aware of the following, as those links aren't created manually:

```
sudo systemctl enable postgresql.service
```

...will create the link, and means PostgreSQL will be enabled upon startup. To control the service before the next restart:

```
sudo systemctl daemon-reload
```

...will make systemd aware of the changes.

Linked In

Systemd makes links between files automatically, but there will be times you'll want a file to appear to be in a local directory when it is elsewhere, with a handy little command we have not so far had a chance to show you: `ln`.

`ln` makes a link which allows a file to effectively exist in two places at once. In the example:

```
ln -s /backupdisk/important/essay12.txt
important-essay.txt
```

...(where `/backupdisk/important` should be replaced by the actual path to a directory on your backup disk or eleswhere, and `essay12.txt` by the name of a file there) a file will appear in your current working directory. But `ls -l` and you'll see that it's a special type of file, a link pointing to the actual file. Edit **important-essay.txt** and you'll find that **essay12.txt** on the backup disk will be edited. Soft, or 'symbolic' links, are created

Top Tip

Who / Where

Although run levels are no longer particularly meaningful under systemd, you can still check which run level you are in with `who -r`.

with the -s switch – you don't even need the file to which you're linking to actually exist, which makes it handy if linking across a network, or to a removable drive.

It's called a symbolic link because it works by linking to the name of the target file, rather than to the file data itself. Create a hard link:

```
 ln /etc/apaches2/apache2.conf myapache2.conf
```

> " Remove the original file in the hard link case, and the link still points to the data "

...and you have two names (and locations) for the same file – sounds like the same thing? Not exactly: if you delete the original file in the first example, you can replace it with a new file of the same name, but different contents, and the symbolic link will point to the new file. Remove the original file in the hard link case, and the link still points to the data.

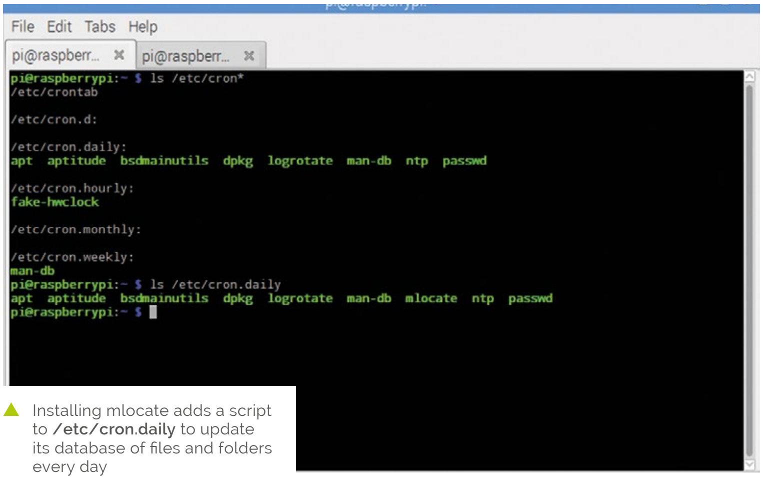
Top Tip

Systemd dot

Systemd is a complex system, with many components, from systemd.unit to systemd.slice – but they all have their own man pages, and it's worth reading through to help get the overall concept.

Location, location

The startup scripts – whether init.d or systemd – are generally for daemons: processes you want running all of the time, like web servers and databases. There are plenty of programs which do housekeeping that need to be run periodically – hourly, daily, weekly. For this purpose, the cron software utility is ideal for scheduling the running of such programs and tasks. Cron searches its configuration directories and runs through the scripts it finds there – have a look at the various folders in **/etc** with names beginning with cron.



```
pi@raspberrypi:~ $ ls /etc/cron*
/etc/crontab

/etc/cron.d:

/etc/cron.daily:
apt  aptitude  bsdmainutils  dpkg  logrotate  man-db  ntp  passwd

/etc/cron.hourly:
fake-hwclock

/etc/cron.monthly:

/etc/cron.weekly:
man-db
pi@raspberrypi:~ $ ls /etc/cron.daily
apt  aptitude  bsdmainutils  dpkg  logrotate  man-db  mlocate  ntp  passwd
pi@raspberrypi:~ $ 
```

Installing mlocate adds a script to **/etc/cron.daily** to update its database of files and folders every day

The easiest way to get to know where things like this are on your system is to search with `locate` – which is not installed by default on Raspbian. Enter `apt-get install mlocate`, to get the best version, then `locate cron` – which will find you every file or directory with cron as its name or as part of its name.

The `locate` tool maintains a database of every file on the system, which itself is updated daily by cron. If you've made a lot of changes, or want to find out where some software you've just installed has put its config file, get locate to update its database with `sudo updatedb`.

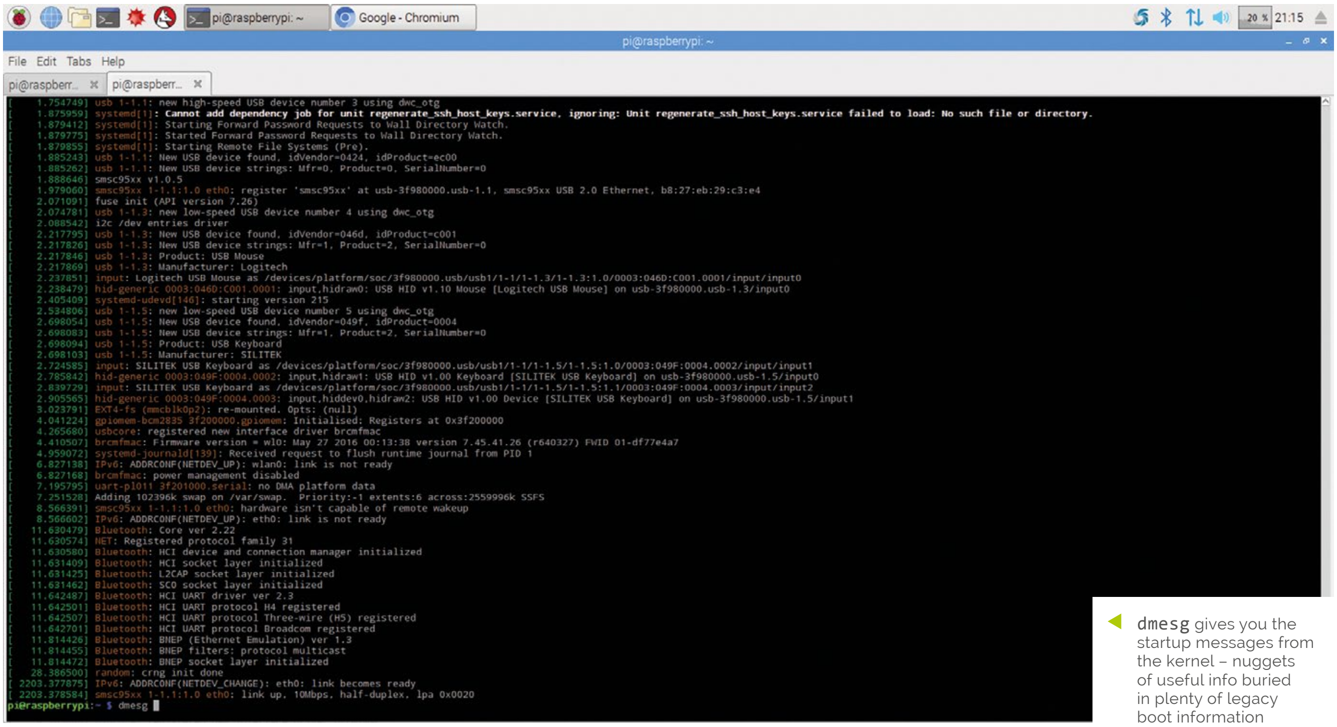
The built-in alternative is `find`, a powerful utility which enables you to search particular directories – or the whole file system – by name or name fragment, size of file, how long ago they were modified, or whether they're bigger than another file. Because it searches the file system, rather than a cached listing, it takes longer than a locate, but it is always up to date, and has search options not found in locate (see 'RegExp' box). To replicate our `locate cron` command with `find`:

```
 find / -name '*cron*'
```

If you were looking for cron or crontab, but not anacron, you could search for `'cron*'` instead. There will be more output than you want, so pipe it through a pager, or perhaps a grep. Back to using cron – the easiest way is via crontab, which maintains a table where each row specifies a command and how often it is to run. You edit the crontab file not directly, but with `crontab -e`, which calls up the default text editor to do the job. Add an entry in the form:

```
 45 05 * * 1-5 calendar | mail -s 'Your calendar' me@myemailaddress.com
```

...to take an example from Michael Stutz's *Linux Cookbook* (No Starch Press); this will grab the output from the venerable UNIX calendar program and email it to you every morning. The first five crontab fields cover minute, hour, day of month, month, day of the week, and can all be replaced with a single special value, like `@daily` or `@hourly`. While `man crontab` tells you a little about crontab, `man 5 crontab` is far more useful



```
[    1.754749] usb 1-1.1: new high-speed USB device number 3 using dwc_otg
[    1.875959] systemd[1]: Cannot add dependency job for unit regenerate_ssh_host_keys.service, ignoring: Unit regenerate_ssh_host_keys.service failed to load: No such file or directory.
[    1.879412] systemd[1]: Starting Forward Password Requests to Wall Directory Watch.
[    1.879775] systemd[1]: Started Forward Password Requests to Wall Directory Watch.
[    1.879855] systemd[1]: Starting Remote File Systems (Pre).
[    1.885243] usb 1-1.1: New USB device found, idVendor=0424, idProduct=ec00
[    1.885262] usb 1-1.1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    1.888646] smsc95xx v1.0.5
[    1.979060] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at usb-3f980000.usb-1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:29:c3:e4
[    2.071091] fuse init (API version 7.26)
[    2.074781] usb 1-1.3: new low-speed USB device number 4 using dwc_otg
[    2.088542] i2c /dev entries driver
[    2.217795] usb 1-1.3: New USB device found, idVendor=046d, idProduct=c001
[    2.217826] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[    2.217846] usb 1-1.3: Product: USB Mouse
[    2.217869] usb 1-1.3: Manufacturer: Logitech
[    2.237851] input: Logitech USB Mouse as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.3/1-1.3:1.0/0003:046D:C001.0001/input/input0
[    2.238479] hid-generic 0003:046D:C001.0001: input,hidraw0: USB HID v1.10 Mouse [Logitech USB Mouse] on usb-3f980000.usb-1.3/input0
[    2.405409] systemd-udevd[146]: starting version 215
[    2.534806] usb 1-1.5: new low-speed USB device number 5 using dwc_otg
[    2.698054] usb 1-1.5: New USB device found, idVendor=049f, idProduct=0004
[    2.698083] usb 1-1.5: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[    2.698094] usb 1-1.5: Product: USB Keyboard
[    2.698103] usb 1-1.5: Manufacturer: SILITEK
[    2.724585] input: SILITEK USB Keyboard as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.5/1-1.5:1.0/0003:049F:0004.0002/input/input1
[    2.785842] hid-generic 0003:049F:0004.0002: input,hidraw1: USB HID v1.00 Keyboard [SILITEK USB Keyboard] on usb-3f980000.usb-1.5/input0
[    2.839729] input: SILITEK USB Keyboard as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.5/1-1.5:1.1/0003:049F:0004.0003/input/input2
[    2.905565] hid-generic 0003:049F:0004.0003: input,hiddev0,hidraw2: USB HID v1.00 Device [SILITEK USB Keyboard] on usb-3f980000.usb-1.5/input1
[    3.023791] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[    4.041224] gpiomem-bcm2835 3f200000.gpiomem: Initialised: Registers at 0x3f200000
[    4.265680] usbcore: registered new interface driver brcmfmac
[    4.410507] brcmfmac: Firmware version = wl0: May 27 2016 00:13:38 version 7.45.41.26 (r640327) FWID 01-df77e4a7
[    4.959072] systemd-journald[139]: Received request to flush runtime journal from PID 1
[    6.827138] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[    6.827168] brcmfmac: power management disabled
[    7.195795] uart-pl011 3f201000.serial: no DMA platform data
[    7.251528] Adding 102396k swap on /var/swap.  Priority:-1 extents:6 across:2559996k SSFS
[    8.566391] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[    8.566602] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[   11.630479] Bluetooth: Core ver 2.22
[   11.630574] NET: Registered protocol family 31
[   11.630580] Bluetooth: HCI device and connection manager initialized
[   11.631409] Bluetooth: HCI socket layer initialized
[   11.631425] Bluetooth: L2CAP socket layer initialized
[   11.631462] Bluetooth: SCO socket layer initialized
[   11.642487] Bluetooth: HCI UART driver ver 2.3
[   11.642501] Bluetooth: HCI UART protocol H4 registered
[   11.642507] Bluetooth: HCI UART protocol Three-wire (H5) registered
[   11.642701] Bluetooth: HCI UART protocol Broadcom registered
[   11.814426] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[   11.814455] Bluetooth: BNEP filters: protocol multicast
[   11.814472] Bluetooth: BNEP socket layer initialized
[   28.386500] random: crng init done
[ 2203.377875] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 2203.378584] smsc95xx 1-1.1:1.0 eth0: link up, 10Mbps, half-duplex, lpa 0x0020
pi@raspberrypi:~ $ dmesg
```

dmesg gives you the startup messages from the kernel – nuggets of useful info buried in plenty of legacy boot information

as it covers the layout of the file, with examples. Run `man man` for more on the numbered sections available with man commands.

Note that traditional UNIX command-line mail is not installed by default on the Pi, so if you wanted to follow the example, you'd need to install a simple mailer – we recommend ssmtp, and the Raspberry Pi forums contain plenty of tips on command-line mail, as it can be used in all sorts of projects.

A fresh startup

Using the value `@reboot`, we can easily run our own scripts on startup, without messing about with system startup scripts. There are times when a full systemd startup script will be more appropriate, but for quickly getting something tested, put the script into crontab.

There are two things that may catch you out. Firstly, you might be running scripts out of a directory that you have in your $PATH, which defines where Bash looks for commands. As $PATH is only set once you log in and your personalised **.bashrc** file is read, scripts running from crontab which are run immediately upon startup will not be aware of your $PATH setting. So, you will need to express all commands by their full paths, such as **/home/pi/bin/test.sh** – as well as making sure that the permissions are sufficient.

Secondly, systemd's parallel service starts also mean that some services, such as the network, as well as environment variables, may not be ready when your `@reboot` commands are called. If you have problems, try giving a short pause first. It's ten seconds in this example crontab entry, but you could use the smallest time that consistently works on testing:

```
@reboot sleep 10; /usr/bin/python3 /home/pi/Documents/Python_Projects/hello_gpio.py
```

One of the scripts that you'll see called by cron is to run anacron, designed to periodically run tasks on machines that were not always switched on – so it is very useful on laptops, too – with tasks specified, in anacrontab, to run after so many days have passed since they last ran. You can also use the automation of crontab or anacron to run your own backup scripts.

Top Tip

RegExp

The `find` tool can search by Regular Expressions, as well as (part) name. They're a whole book topic in themselves, but well worth investigating once you've got command-line basics under your belt, as regexps can be used with many commands.

Conquer the Command Line

For more terminal tutorials, take a look at our Essentials book, *Conquer the Command Line*. Discover the power of the command line to simplify complex tasks, or instantly carry out simple ones.
magpi.cc/Essentials-Bash

Part 04!

Pi Bakery: The Matrix

The Matrix is an undedicated array of switches and lights that can be put to any number of uses – animated light displays, games, controlling synthesizers, and more

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/kLYEZs

With the hardware finished, and tested, it is now time to see how we can use it with the minimum of user complexity. By providing a simple user interface, we can begin to explore all the potential of The Matrix, without having to worry about the registers in the matrix driver.

01 Why a class library?

Last month we published some code to get the matrix working. There was a bunch of functions you could use to do things like look at switches, transfer data into the MAX7219, and light LEDs. The aim was just to produce a quick and dirty test to try out the hardware and spot any wiring faults. The functions were not designed to be easy to use and were too close to the hardware. With a class library, we can present a clean useful interface to the user without them needing to know the dirty bits of the hardware. This abstraction is known as an API (application programming interface).

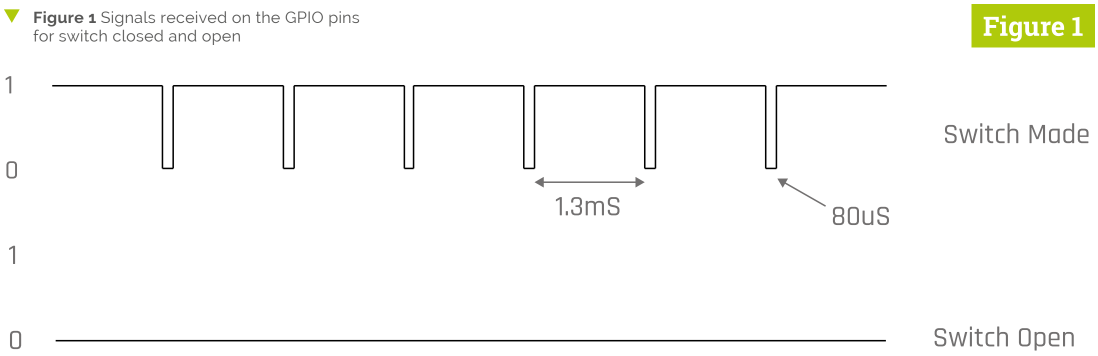
02 The switches

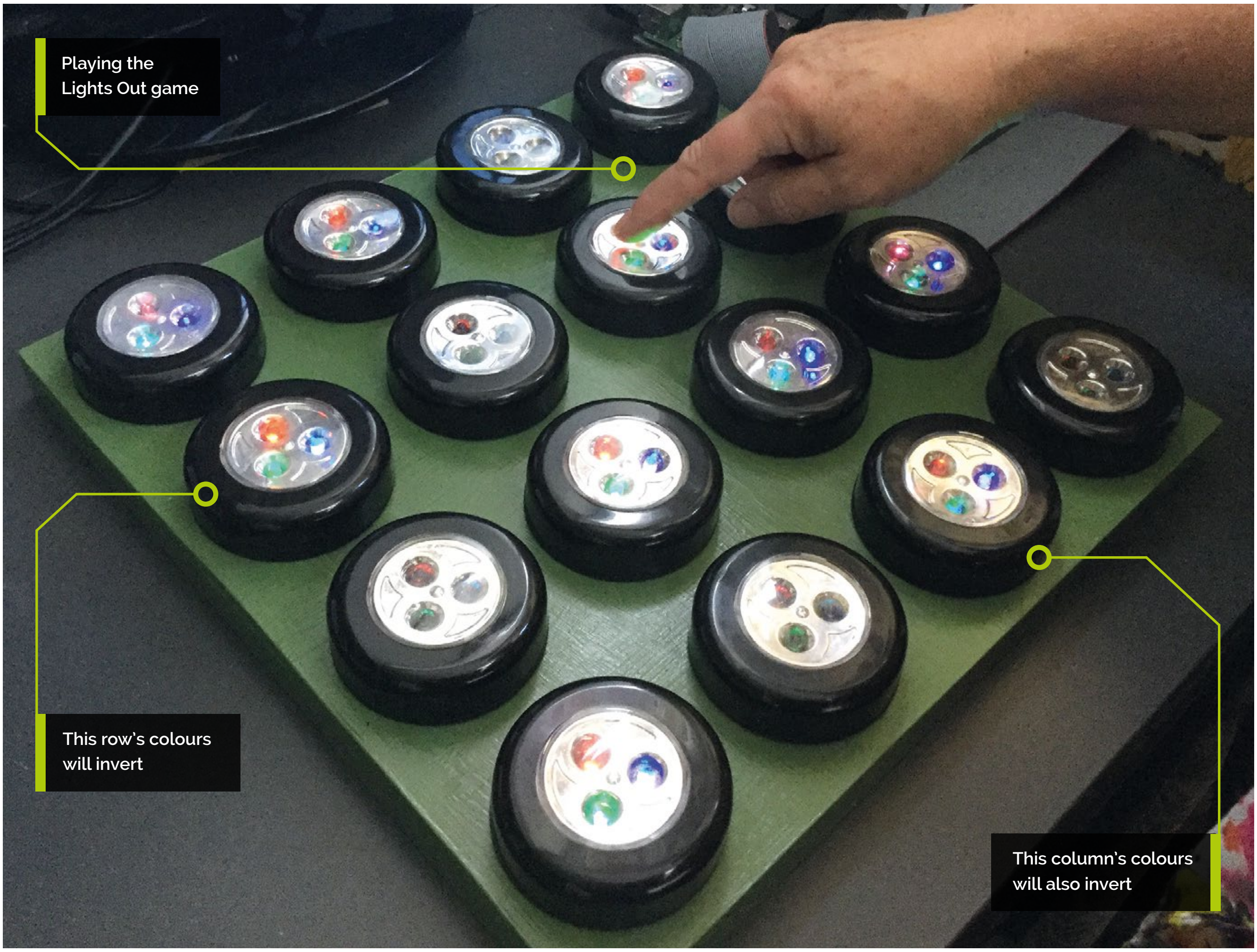
The latching switches on the lights simply connect the multiplexing cathodes in the matrix to a GPIO pin. When connected, you get pulses on the GPIO pins; when not connected, the pins would float, so we arrange for the pins to have the internal pull-down resistors enabled. The signals we get are shown in **Figure 1**. This presents a bit of a problem for a Linux-based computer such as the Raspberry Pi, as these pulses are fast and you can't guarantee to always see them – or indeed, as we want, to miss them. So, following advice on the Raspberry Pi forums, we turned to the pigpio library.

03 What is pigpio?

Pigpio is a GPIO interface, but this one has a new feature: a glitch filter. That is, we can set it to ignore any small changes in the signal level and only report longer-term steady levels. This is exactly what we want here. We can set the

Figure 1 Signals received on the GPIO pins for switch closed and open





Playing the Lights Out game

This row's colours will invert

This column's colours will also invert

glitch filter to ignore the logic low pulses and use a callback function when the level changes from a one to zero, or zero to one. Both transitions tell us a switch has changed, and we can use the callback function to set a Boolean flag in a switch list.

04 The switch API

So what we want to do to read the switches is to call a method in our class library which will return what switch number has been pressed. If nothing has been pressed since the last time we asked the API, then the value -1 is returned. This is handled by the `getSwitch` method of our class library. Note that if more than one switch has been pressed, this method will only return the lowest switch number; if this situation is possible, the user will need to make successive calls to this method until you see a -1 returned.

Figure 2

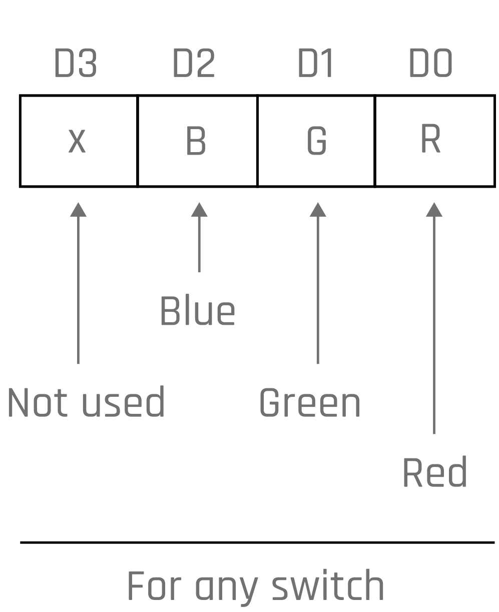


Figure 2 User's view of bits for the LED controls

Top Tip

More examples

We have a few more examples of using this class library in the GitHub repo (**magpi.cc/1NqJjmV**).

05 The LEDs

Translating the LEDs that need to be on, in any one switch, into the register settings in the MAX7219 is a bit more complex. This is because each register actually controls two switches' worth of LEDs. **Figure 2** (previous page) shows how we'd like the user to think of the LEDs: a simple 4-bit value that'll set which LEDs are on for each switch. The `setLed` method does this. It is called with the switch number, and a number corresponding to the required bit pattern. This overwrites any previous value, but you might want to add a colour to a previous pattern, so the `addRed`, `addGreen`, and `addBlue` methods are provided as well.

06 More LED methods

To make things even simpler, we have added a few more methods. The `setRed`, `setGreen`, and `setBlue` methods just set that LED to be lit in a switch, and the `clrLEDs` turns off all the LEDs on the matrix. Also, if you want to find out what LEDs are lit on a switch, the `getLed` method will return the colour number bit pattern for an LED, in the same format used by the `setLed` method. Finally, the `setBrightness` method takes a number between 0 and 15 to control the brightness. Note that a value of zero is not off, but the minimum brightness of 1/32 of the maximum brightness.

07 Housekeeping methods

The two final methods are what we call housekeeping methods: those to start up and close down the class library. The `__init__` function is called automatically when you make an instance of this class. It is where you set what GPIO pins to use for the clock, data, and load pins used to bit-bang

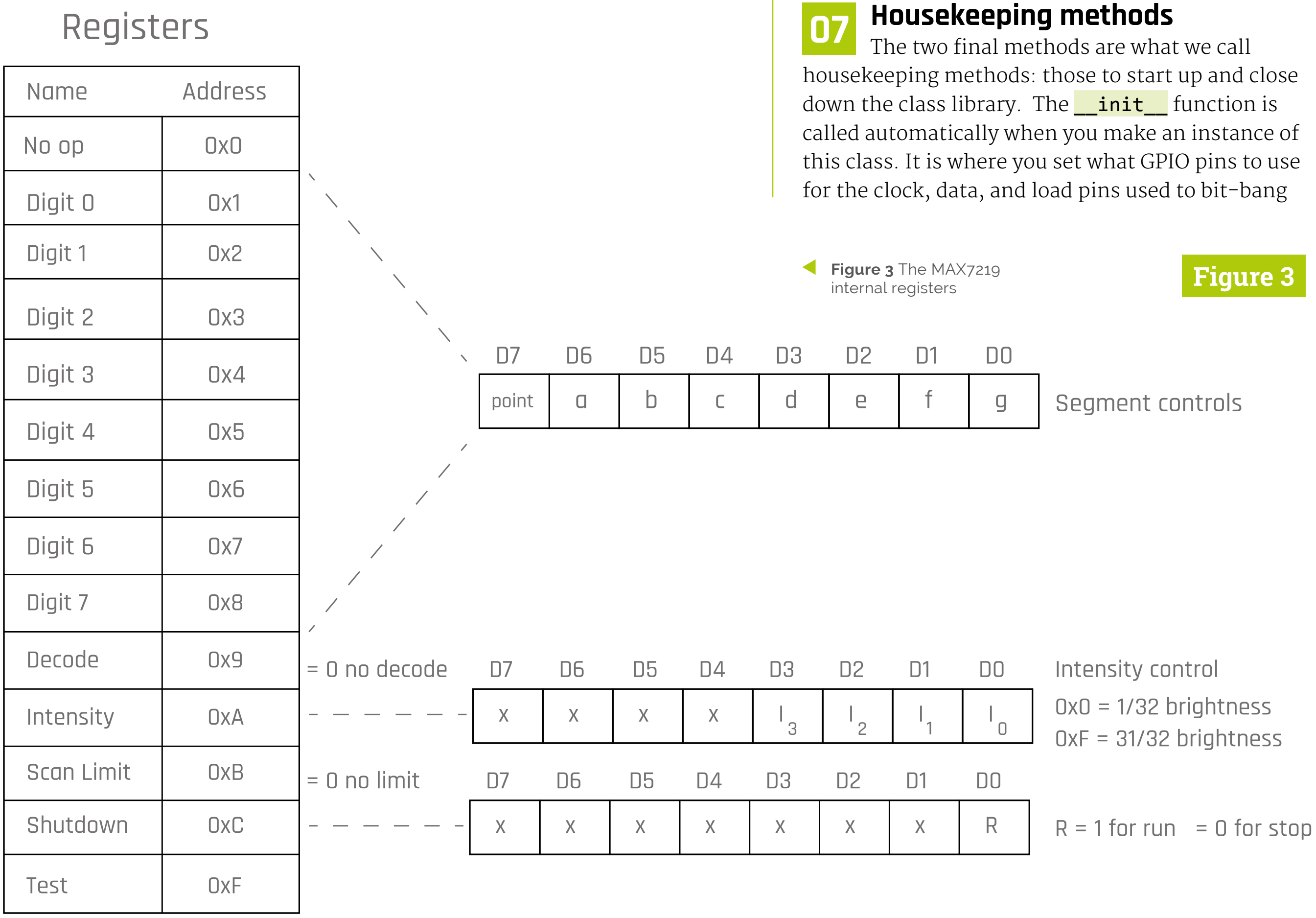


Figure 3 The MAX7219 internal registers

Figure 3

Registers

Name	Address
No op	0x0
Digit 0	0x1
Digit 1	0x2
Digit 2	0x3
Digit 3	0x4
Digit 4	0x5
Digit 5	0x6
Digit 6	0x7
Digit 7	0x8
Decode	0x9
Intensity	0xA
Scan Limit	0xB
Shutdown	0xC
Test	0xF

D7	D6	D5	D4	D3	D2	D1	D0
point	a	b	c	d	e	f	g

Segment controls

= 0 no decode

D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	I_3	I_2	I_1	I_0

Intensity control
0x0 = 1/32 brightness
0xF = 31/32 brightness

= 0 no limit

D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	x	x	x	R

R = 1 for run   = 0 for stop

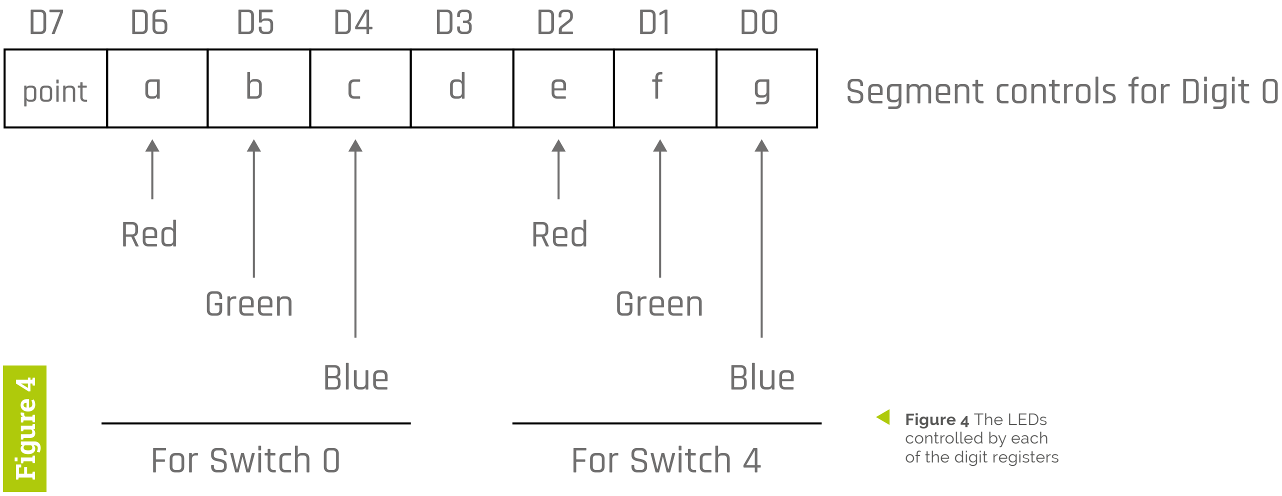


Figure 4

Figure 4 The LEDs controlled by each of the digit registers

the MAX7219 multiplexer chip. You also set the initial brightness with this call. The `cleanUp` method puts the MAX chip into shutdown so it is not multiplexing, and also frees up the callback vectors, clears the glitch filter, and stops pigpio from running. We recommend calling this method before the code terminates.

08 More about the MAX7219 registers

Figure 3 shows the register structure for the MAX7219 chip. Most of them are concerned with the LED segments, which are the coloured LEDs in our project. The chip's data sheet (**magpi.cc/JPjHhi**) tells the full story; **Figure 3** shows a simplified version for our project. We're not using any feature of the scan limit or decode functions, so these must be set to zero. The intensity control is used to set the brightness, and the shutdown for the start/stop function. The scan limit register will allow you increase refresh rates if you are using fewer than the maximum number of digits (8), so we don't need anything here either.

09 Segments to LEDs

For each digit, there is a segment control register; the circuit we used means that each of these controls the LEDs for two switches – as shown in **Figure 4**. As you can see, it is nothing like the simple API model we used in the methods to control the LEDs. The translation from one to the other is done in the library code, and so this complication is hidden from the user of the class library. Registers for digits 0 to 3 control the LEDs on switches 0 to 7, and registers digits 4 to 8 control switches 8 to 15.

matrix_demo2.py

Language: **Python**

```python
import time, os
os.system("sudo pigpiod") # enable pigpio system
from max7219bang import Max7219bang

brightness = 8
dataPin = 14 ; clockPin = 15 ; loadPin = 18 # matrix wiring
matrix = Max7219bang(dataPin,clockPin,loadPin,brightness)

def main():
   print("Matrix demo - Ctrl C to stop")
   print("Read switches and light up all LEDs on the pushed one")
   matrix.clrLEDs() # turn all LEDs off
   while True:
     pressed = matrix.getSwitch() # return pressed switch or -1 for none
     if pressed != -1 : # if switch has been pressed
        print("Switch", pressed, "pressed", end=" ")
        if matrix.getLed(pressed) == 0: # if LEDs off
           matrix.setLed(pressed,7)     # turn them on
           print("lights on")
        else:
           matrix.setLed(pressed,0)     # turn them off
           print("lights off")

# Main program logic:
if __name__ == '__main__':
  try:
    main()
  except:
    matrix.cleanUp()
# Note the use of the code disables any error output from the code
# when developing code comment out the lines:- try: and except:
```

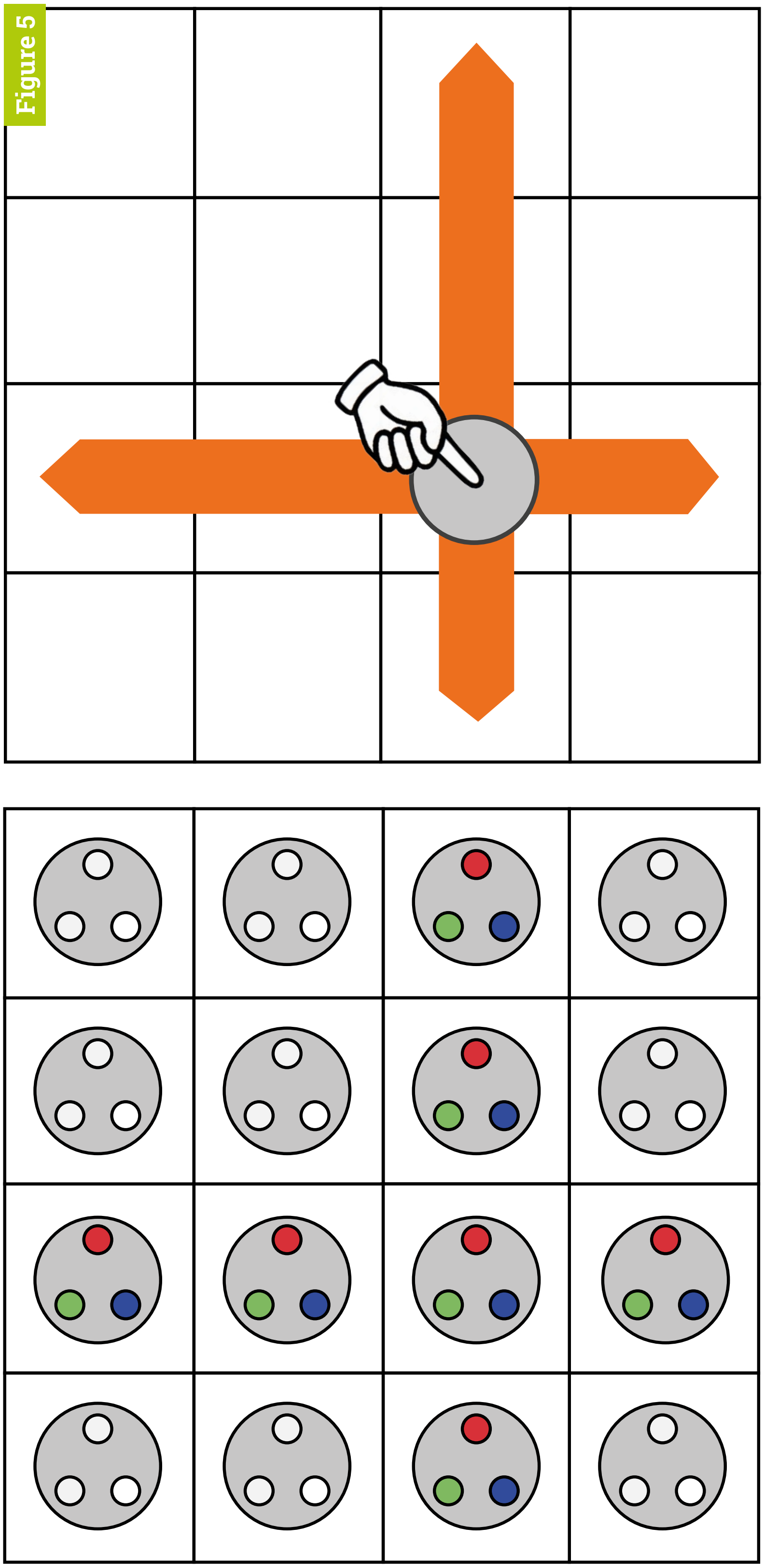


Figure 5 Pressing a switch inverts its column and row

10 The class library code

The code for the 'max7219bang' class library is shown in the **max7219bang.py** listing; it should be stored in the same folder as the programs that use it. You could install it permanently into the Python 3 system by altering Python's search path or storing it into a folder which is in Python's path, but we recommend that you simply have it in the same folder as programs that use it. The first time it is run, a **__pycache__** directory will be created to save the semi-compiled version of the class library. This makes it run faster.

11 Using the class library

The code in the **matrix_demo2.py** listing shows an example of reading the switches, and turning on or off all the LEDs on that switch. It shows the use of many of the class library functions. One important thing to note is that you must enable the pigpio system before loading the class library. This is only needed once per session and could be done from the command line, but we include it in the code for simplicity. When a switch is pressed, the state of the LEDs is retrieved for that switch, and all the LEDs are changed to the opposite state.

12 The Lights Out game

The **matrix_lightsOut.py** listing is a simple Lights Out game. The object is to turn out all the LEDs on the matrix board by pressing switches. Each switch inverts all the LEDs in the column and row of the switch pressed – illustrated in **Figure 5**. The game starts out with a set of lights that are on and you have to turn them all out with two presses. Once you get down to just needing one press, it is obvious where you need to press. If you press incorrectly, you can always press the same switch again and negate the press. **Figure 6** (overleaf) shows a typical puzzle.

13 Customising the Lights Out game

The call to the `setup` function determines the number of key presses you need to complete the game. We found going to three was a bit of a

matrix_lightsOut.py

▶ Language: **Python**

```python
import time, os, random
os.system("sudo pigpiod") # enable pigpio system
from max7219bang import Max7219bang

brightness = 8
dataPin = 14 ; clockPin = 15 ; loadPin = 18 # matrix wiring
matrix = Max7219bang(dataPin,clockPin,loadPin,brightness)
speed = 0.2 ; random.seed()

def main():
   global done
   print("Simple lights out game - Ctrl C to stop")
   print("Press buttons until all lights are out")
   matrix.clrLEDs() # turn all LEDs off
   while True:
     flash(4)
     done = False
     emptyPresses() # remove any bounce
     setup(2) # level of scrambling
     print("Try this one then")
     while not done:
        pressed = matrix.getSwitch() # return pressed switch or -1 for none
        if pressed != -1 : # if switch has been pressed
          matrix.setLed(pressed, matrix.getLed(pressed) ^ 7)
          invertRow(pressed)
          invertCol(pressed)
          done = checkOut()
          emptyPresses() # remove any bounce

def invertRow(switch) :
   row = switch // 4
   for i in range(row*4,row*4+4):
      matrix.setLed(i,matrix.getLed(i) ^ 7)
      time.sleep(speed)

def invertCol(switch) :
   col = switch % 4
   for i in range(col,16,4):
      matrix.setLed(i,matrix.getLed(i) ^ 7)
      time.sleep(speed)

def checkOut(): # check LEDs are all off
   off = True
   for switch in range(0,16):
      if matrix.getLed(switch) != 0:
         off = False
   return off

def setup(depth): # inital position of LEDs
   global speed
   speed = 0.001 # make setup quick
   matrix.clrLEDs() # turn all LEDs off
   for i in range(0,depth):
      target = random.randint(0,15)
      #print(target) # uncomment for testing
      matrix.setLed(target, matrix.getLed(target) ^ 7)
      invertRow(target)
      invertCol(target)
   speed = 0.2 # normal display speed

def flash(times):
   state = 7
   for flash in range(0,times*2):
      for i in range(0,16):
         matrix.setLed(i,state)
      time.sleep(0.3)
      state ^= 7

def emptyPresses(): # remove all bounce from key switches
    while matrix.getSwitch() != -1:
       pass
       print("extra bounce")

# Main program logic:
if __name__ == '__main__':
  try:
    main()
  except:
    matrix.cleanUp()
# Note the use of the code disables any error output from the code
# when developing code comment out the lines:- try: and except:
```

CoderDojo

Can you encourage young people to become digital creators?

CoderDojo is a global network of free, volunteer-led, project-based programming clubs for young people aged 7–17. Here they can learn to code, build a website, create an app or a game, and explore technology in an informal, creative, and social environment.

In order to run locally we need volunteers like you with a wide range of skills. You don't have to be a coder to mentor. Dojos benefit from a mix of mentors providing support and encouragement to CoderDojo attendees.

Dojo champions, who manage each club, will be able to discuss how you can best support attendees, and we'll provide you with any resources and advice you might need.

Join us at coderdojo.com/volunteer

The CoderDojo Foundation is Irish Registered Charity (524255) CHY20812
and is part of the Raspberry Pi Foundation, UK registered charity 1129409

Part of Raspberry Pi

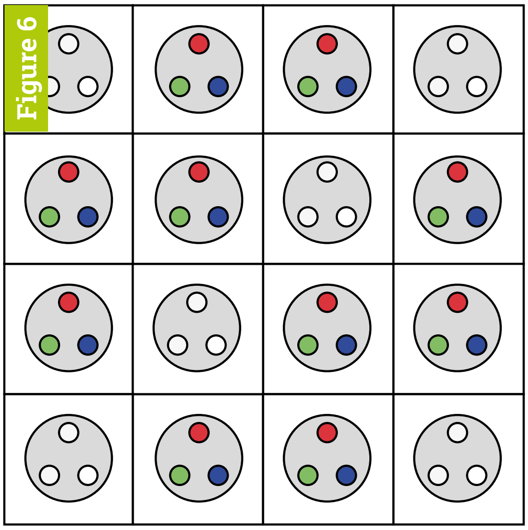


Figure 6

Figure 6 What switch do you have to press to get the Figure 5 pattern?

challenge, but it could be done by trying a key and seeing if you could solve the problem as presented in the original two-press version. If it couldn't be done, then you press that key again to undo its action and try another. Taking this to even more levels might be too hard unless you leave a trail of breadcrumbs – that is, print out the switches you have pressed, when you press one.

Your turn

You should now have all the tools you need to write your own programs, so here is the challenge we promised at the start of this project. Write a program to display a user-defined sequence of lights on the matrix that you can store and load from a file. You should be able to display the sequence at any speed, forwards or backwards, continuously. We will publish our efforts for this program on our GitHub page in two months; good luck with yours.

Start with #73

This is the latest instalment in a series of Mike's Pi Bakery tutorials. You can download digital editions of previous tutorials for free. Start with *The MagPi* #73.

magpi.cc/73

max7219bang.py

Language: **Python**

```python
import time
import pigpio

class Max7219bang():

    def __init__(self,data,clock,load,bright):
       self.da = data
       self.ck = clock
       self.ld = load
       if bright >-1 and bright < 16:
          self.br = bright
       else :
          self.br = 8
       # copy of the Max chip's registers
       self.registers = [0]*9
       #0=off,1=red,2=green,4=blue
       self.ledsState  = [0]*16 # all off
       self.ledCol     = [0,4,2,6,1,5,3,7] # LED colour numbers RGB
       # colour offset per switch
       self.colOff     = [0,0,0,0, 4,4,4,4, 0,0,0,0, 4,4,4,4]
       # switch number to register address
       self.address    = [1,3,5,7, 1,3,5,7, 2,4,6,8, 2,4,6,8]
       self.buttonPins = [12,25,24,23,6,13,19,26,16,20,21,4,17,27,22,5] # rows
       self.switches = [False] * 16

       self.pi = pigpio.pi()
       if not self.pi.connected:
          print("Pi not connected")
       self.pi.set_mode(self.ck,pigpio.OUTPUT)
       self.pi.set_mode(self.da,pigpio.OUTPUT)
       self.pi.set_mode(self.ld,pigpio.OUTPUT)
       for i in self.buttonPins:
          self.pi.set_mode(i,pigpio.INPUT)
          self.pi.set_pull_up_down(i, pigpio.PUD_DOWN)
       self.pi.write(self.ck, 0)
       self.pi.write(self.da, 0)
       self.pi.write(self.ld, 1)

       self.cb = [None] *32
       for i in range (0, len(self.buttonPins)):
         self.pi.set_glitch_filter(self.buttonPins[i], 200) # microseconds filter
         self.cb[i] = self.pi.callback(self.buttonPins[i], pigpio.EITHER_EDGE, self.cbf)
       self.sendMax(0x9,0)    # no decode
       self.sendMax(0xB,7)    # scan all digits
       self.sendMax(0xA,self.br) # intensity
```

DOWNLOAD THE FULL CODE:

magpi.cc/1NqJjmV

```
       self.sendMax(0xC,1)    # take out of shutdown mode
       self.clrLEDs()          # start with all off
       for i in range(0,16): # clear out any glitches
         self.switches[i] = False

    def setBrightness(self,brightness):
        if brightness > 15:
           brightness = 15
        if brightness < 0:
           brightness = 0
        self.br = brightness
        self.sendMax(0xA,self.br)

    def clrLEDs(self):
        for add in range(1,9): # all display registers
          self.registers[add] = 0
          self.sendMax(add,0)
        for i in range(0,16): # all display leds
            self.ledsState[i] = 0

    def getLed(self,switch):
        return self.ledCol[self.ledsState[switch]]

    def setRed(self,switch):
        self.setLed(switch,1)
    def setGreen(self,switch):
        self.setLed(switch,2)
    def setBlue(self,switch):
        self.setLed(switch,4)

    def addRed(self,switch):
        col = self.getLed(switch) | 1 # add red
        self.setLed(switch,col)
    def addGreen(self,switch):
        col = self.getLed(switch) | 2 # add green
        self.setLed(switch,col)
    def addBlue(self,switch):
        col = self.getLed(switch) | 4 # add blue
        self.setLed(switch,col)


    def setLed(self,switch,col): # set what LEDs are on for switch
        if switch > 15 or switch < 0 :
            switch = 0           #default for switch out of range
        col &= 0x7               # restrict to 0 to 7
        col = self.ledCol[col]   #convert to register data
        self.ledsState[switch] = col
        reg = self.address[switch]
        if self.colOff[switch] == 0 :
           self.registers[reg] = (self.ledsState[switch] << 4) | (self.ledsState[switch+4])
        else:
           self.registers[reg] = (self.ledsState[switch-4] << 4) | (self.ledsState[switch])
        self.sendMax(self.address[switch], self.registers[reg])

    def cbf(self, gpio, level, tick): # call back function
       place = [i for i,x in enumerate(self.buttonPins) if x == gpio]
       #print("GPIO",gpio,"switch",place[0], "steady level",level)
       self.switches[place[0]] = True # indicate switch has changed

    def getSwitch(self):
        pressed = -1 ; i = 0
        while pressed == -1 and i<16:
           if self.switches[i]: # key press found
              self.switches[i] = False
              pressed = i
           i+=1
        return pressed

    def sendMax(self,add,data): # send a byte to mux chip
        package = (add << 8) | data # join into one bit pattern
        self.pi.write(self.ld, 0) # lower load
        for i in range(15,-1,-1): # from 15 to 0
           self.pi.write(self.da, 1 & (package >> i)) # send MSB first
           self.pi.write(self.ck, 1) # toggel clock
           self.pi.write(self.ck, 0)
        self.pi.write(self.ld, 1) # latch value

    def cleanUp(self):
       self.sendMax(0xC,0)     # put into shutdown mode
       for i in range (0, 16): # remove callback vectors
          self.cb[i].cancel()
          self.pi.set_glitch_filter(self.buttonPins[i], 0)
       self.pi.stop() # stop pigpio
```

MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
December 2018
Issue #13
BUILD A MAKER BUSINESS
Turn your hobby into your job
ALSO INSIDE
ELECTRIC PLAYDOUGH
Make circuits without wires
HACKADAY PRIZE
Discover the best in open hardware
DIY ROBOT
Build your first Arduino bot
SCREWS BRUSHES TRANSISTORS PSU

Raspberry Pi 3A+

Gone too long, the A+ is back with the power of the Pi 3 behind it

The last time the Raspberry Pi family had a new A-form-factor design, more compact and lightweight than the full-size B and B+ ranges, it was the Raspberry Pi A+ released four years ago. Since then, the family has grown considerably: we've seen the Pi 2, Pi 3, Pi Zero, Pi Zero W and WH, and most recently the 3B+.

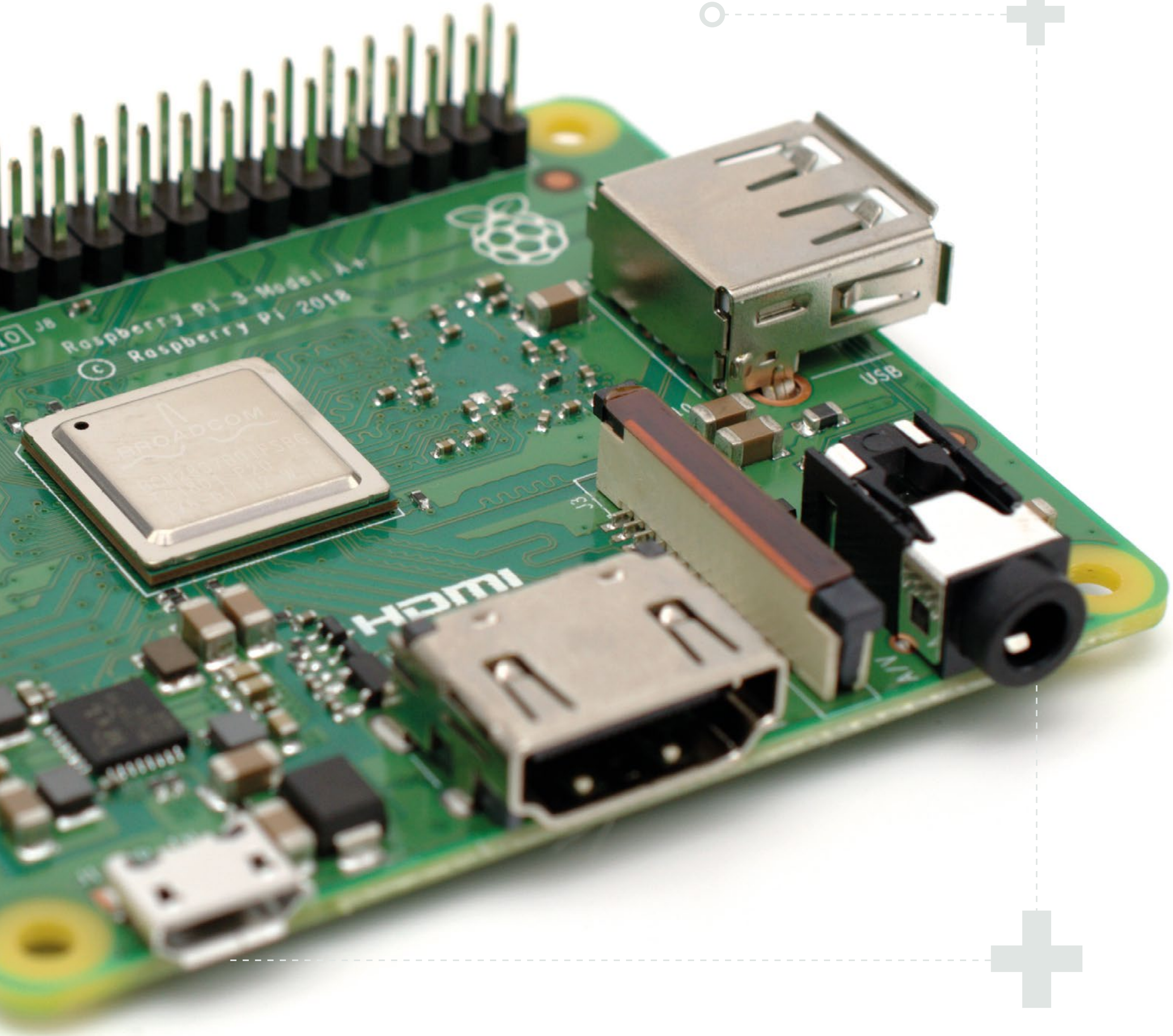
> Packing the power of the 3B+ into the same footprint as the original Pi A+

Now, the Raspberry Pi 3A+ is here, packing the power of the 3B+ into the same footprint as the original Pi A+. Read on for the lowdown on what Eben Upton describes as "the best product we can make."

UK suppliers:

The Pi Hut
magpi.cc/zVTpyC

Pimoroni
magpi.cc/LhuKdS



CPC
magpi.cc/algkOF

ModMyPi
magpi.cc/rMXCsW

RS Components
magpi.cc/nDkZGZ

For a complete list of resellers globally where the new Raspberry Pi 3 Model A+ can be purchased, please see rpf.io/products

Meet the **Raspberry Pi 3A+**

The 3A+ squeezes a lot into its compact footprint

SPECS

SoC:
Broadcom BCM2837B0 quad-core A54 (ARMv8) 64-bit @ 1.4GHz

GPU:
Broadcom VideoCore IV

NETWORKING:
2.4GHz and 5GHz 802.11b/g/n/ac wireless LAN

RAM:
512MB LPDDR2 SDRAM

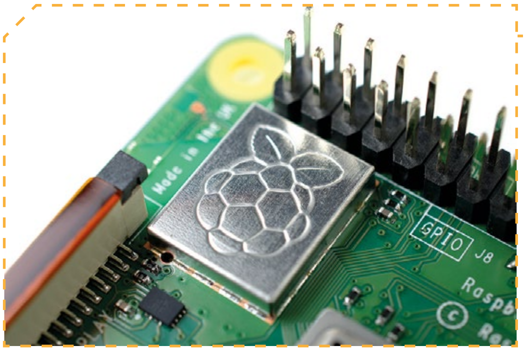
BLUETOOTH:
Bluetooth 4.2, Bluetooth Low Energy (BLE)

GPIO:
40-pin GPIO header, populated

STORAGE:
microSD

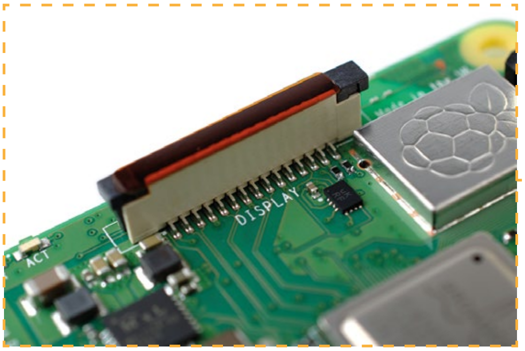
PORTS:
HDMI, 3.5 mm analogue audio-video jack, 1 × USB 2.0, Camera Serial Interface (CSI), Display Serial Interface (DSI)

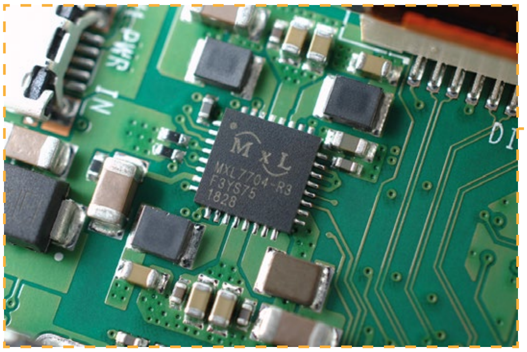
DIMENSIONS:
67×56×11.5 mm



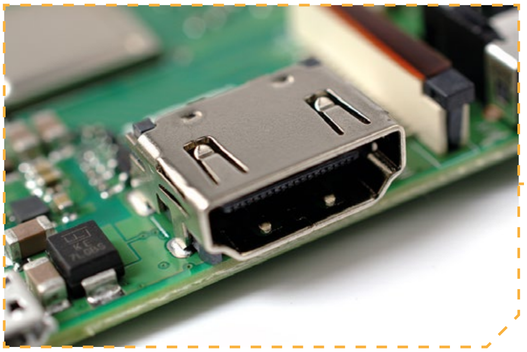
▲ The radio, originally developed for the 3B+, makes the 3A+ the first A-model to feature on-board networking

▼ Full-size display (DSI) and camera (CSI) ports make connecting peripherals a snap





◀ The 3A+ benefits from the same clever power management chip (PMIC) as the 3B+



▲ Unlike the Zero family, the 3A+ includes both full-size HDMI and analogue audio-video outputs

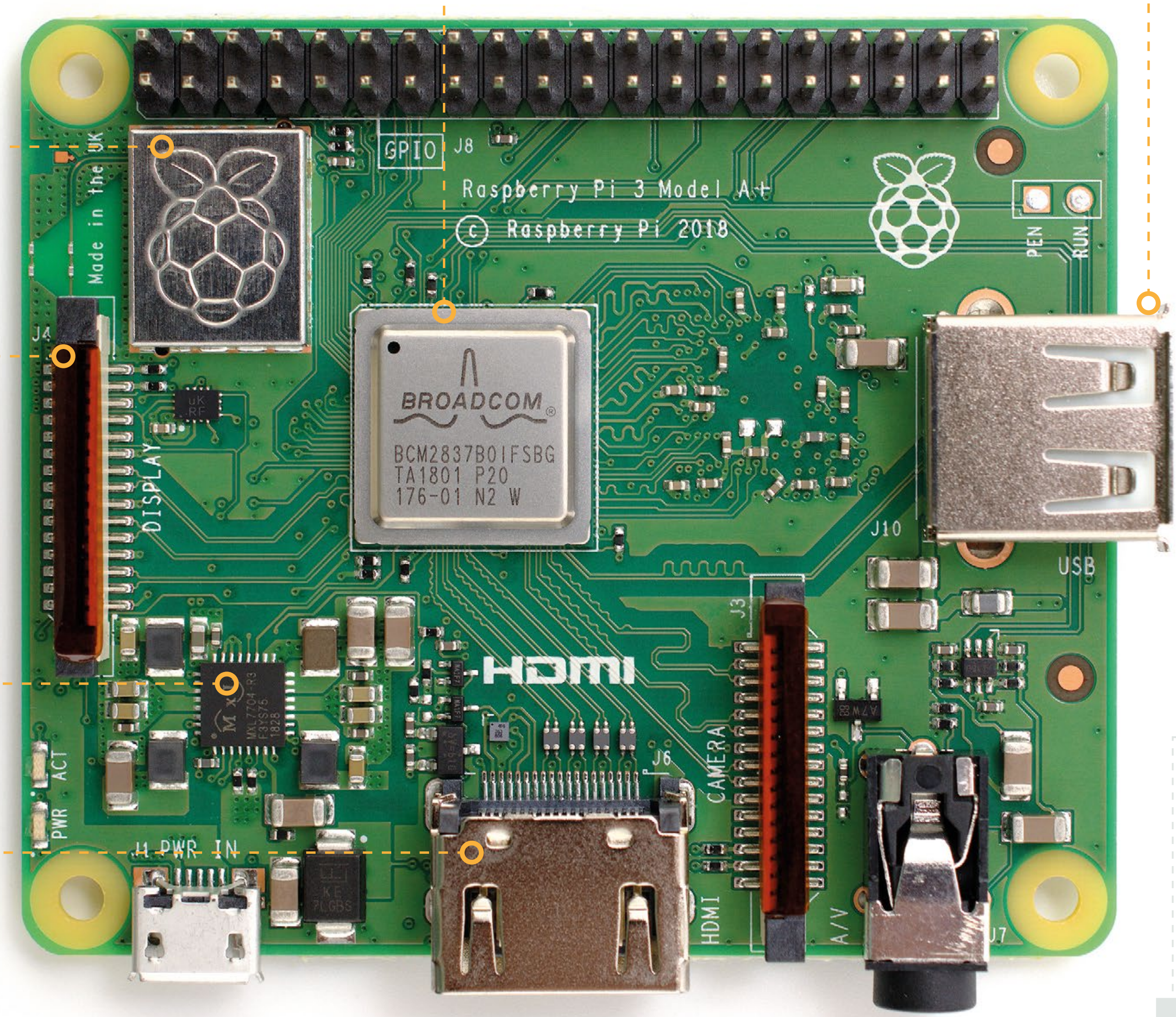
QuickStart guide

The Raspberry Pi 3A+ is directly compatible with the 3B+ and all other Raspberry Pi models. If you've already got a power supply and microSD card with a new version of Raspbian installed (grab the latest image from **rpf.io/downloads**), you're ready to go. If the 3A+ is your first Raspberry Pi, see our online QuickStart guide to get up and running fast.

magpi.cc/quickstart

The 3A+ has the same 1.4GHz system-on-chip (SoC) as the 3B+, with no loss of performance

The biggest change from the 3B+: the reduction from four to one USB 2.0 ports



Benchmarking the Raspberry Pi 3A+

Small but mighty, the reduction in size doesn't mean the 3A+ costs you any performance over its bulkier brother

For projects where a full-size Pi 3B+ is simply too bulky, there hasn't been much choice: aside from the Compute Module 3, which is targeted at industrial users, the choices were the four-year-old Pi A+ or ultra-compact Pi Zero family – and in either case it meant a dramatic drop in performance.

The Pi 3A+ comes with the promise of full-size performance in a small, lightweight, and less power-hungry package, but the only way to see if that is true or not is to put it through its paces in a range of benchmarks.

Spec comparison

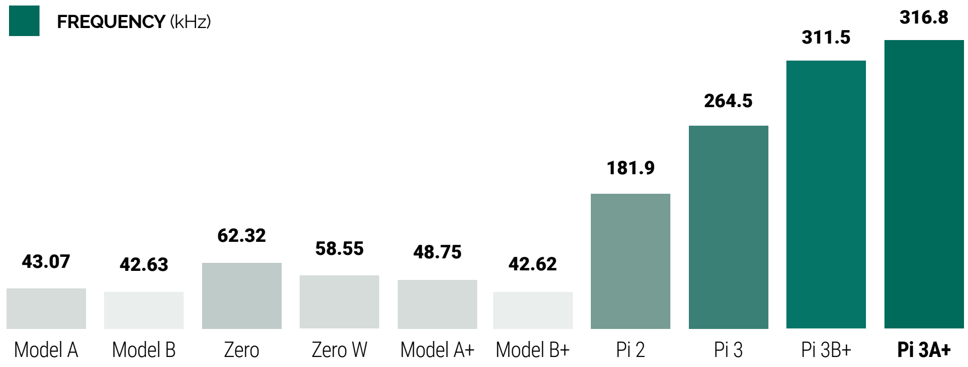
At its heart, the Pi 3A+ is literally a cut-down version of the Pi 3B+. It has the same Broadcom BCM2837B0 system-on-chip (SoC) roughly in the centre of the board, hidden under a metal heat-spreader, which runs at the same 1.4GHz frequency. While 512MB of RAM is less than the 1GB of a Pi 3B+, the smaller 3A+ can certainly hold its own.

Looking back at the original Pi A+, it's hard to imagine they're from the same family: from a single-core 32-bit 700MHz processor and no networking to a quad-core 64-bit 1.4GHz processor with built-in wireless LAN and Bluetooth, the 3A+ should prove a serious upgrade for users of its predecessor.

Python GPIO

Higher is better

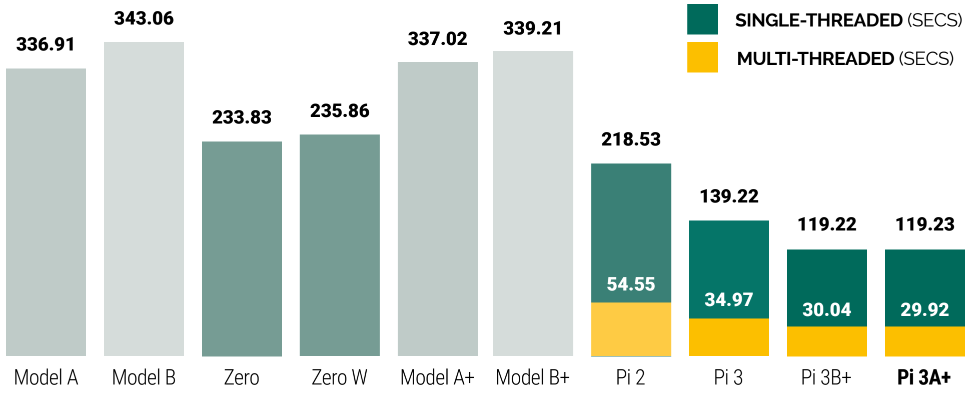
Designed to highlight a real-world bottleneck, the Python GPIO benchmark switches a single pin on and off while a frequency counter measures how quickly the pin is toggled. The faster a Pi's processor, the faster the pin can be toggled before the processor hits its limit.



SysBench CPU

Lower is better

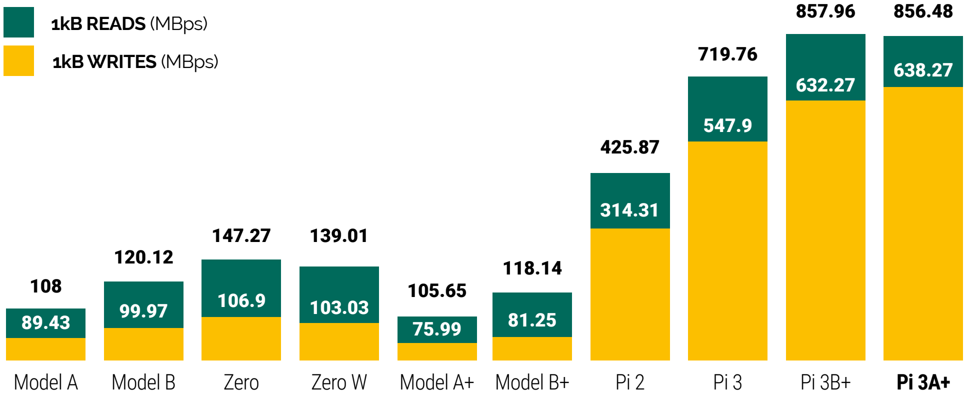
Designed to focus on the central processor's performance, the SysBench CPU benchmark tests how quickly a Pi can perform prime number calculations. For Pi models with quad-core processors, the test is run twice: once with a single thread using only one of the cores, and again with four threads using all four cores.



SysBench Memory Throughput

Higher is better

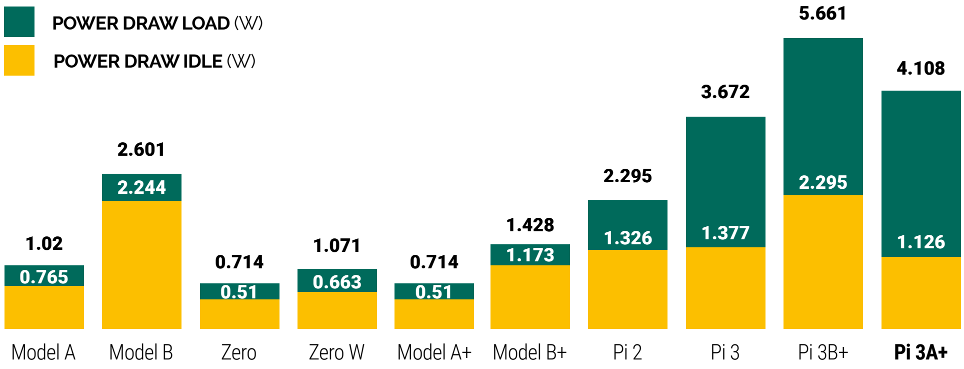
Processor performance is only part of the puzzle when it comes to overall system performance: in the SysBench Memory Throughput test, measurements are taken to show how quickly a Pi can read and write to the random-access memory (RAM) in 1kB chunks, reported in megabytes per second (MBps).



Power Draw

Lower is better

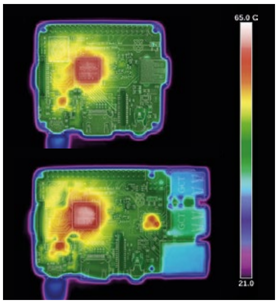
More performance typically means more power used, and here each Pi is connected to an HDMI display, wireless keyboard, and, where applicable, a WiFi or wired Ethernet network before two measurements are taken: the power used, in watts, while the Pi is sat idle at the desktop, and again while running a CPU-heavy application.



Thermal Performance

Cooler is better

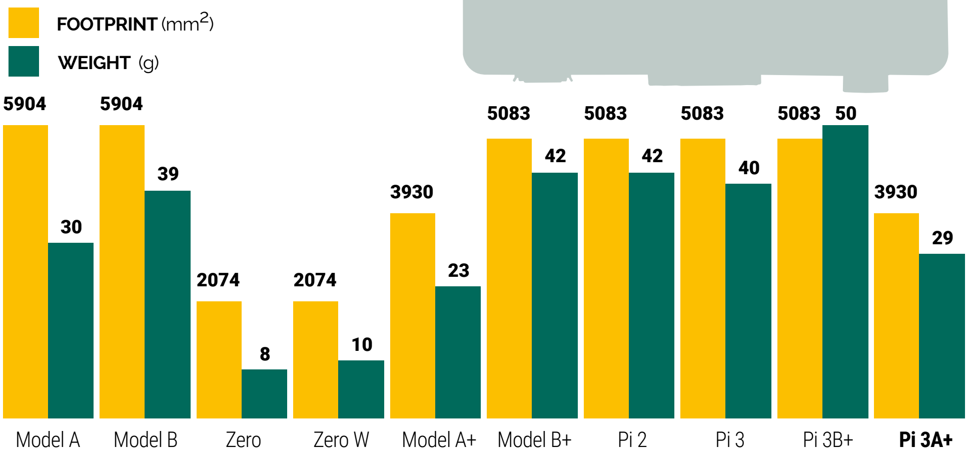
The Pi 3B+ benefited from a change to the way the system-on-chip (SoC) is attached to the circuit board, allowing it to better dissipate heat. With the 3A+ having a smaller board, this test captures thermal images under heavy CPU load to show how well the two designs cope.



Size & Weight

Lower is better

The biggest difference between the Pi 3B+ and the 3A+ is their respective sizes. Here, the footprint of each Pi model is measured from its widest points – to include the size of the ports which sit proud of the board – and its weight measured, both important aspects for embedded and robotics applications.



Eben Upton and Roger Thornton on the Pi 3 A+

Eben and Roger talk about the Pi 3 A+'s inspiration, launch timing, and hint at the Raspberry Pi 4

"We describe our product design process as resembling a bunch of trains driving along, because we're dealing with a bunch of subsystems," Eben Upton, co-founder of the Raspberry Pi Foundation, explains of the development process that has delivered the Raspberry Pi 3 Model A+. "Some subset of the trains will arrive at the station, and when enough of them arrive then we've got a new product we can make. We don't have a big team of marketing guys who make PowerPoints about our roadmap. It's much more coffee-table-oriented than that."

Bringing back the A+

"The A+ form factor was always a good form factor," says Roger Thornton, Principal Hardware Engineer. "A lot of people have asked us for an A+ again, and we were able to make that, so we did."

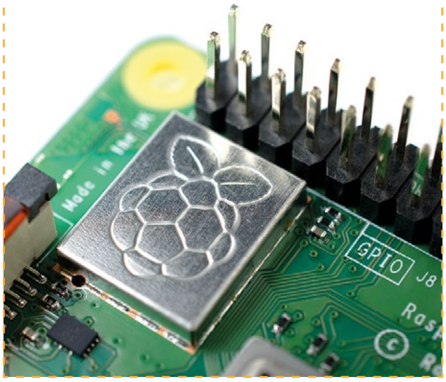
With both the Raspberry Pi 2 and 3 having launched exclusively in larger B-variant form factors, the A+ has been absent from the line-up for a full two generations. "It was skipped for Raspberry Pi 2 because it simply wasn't around

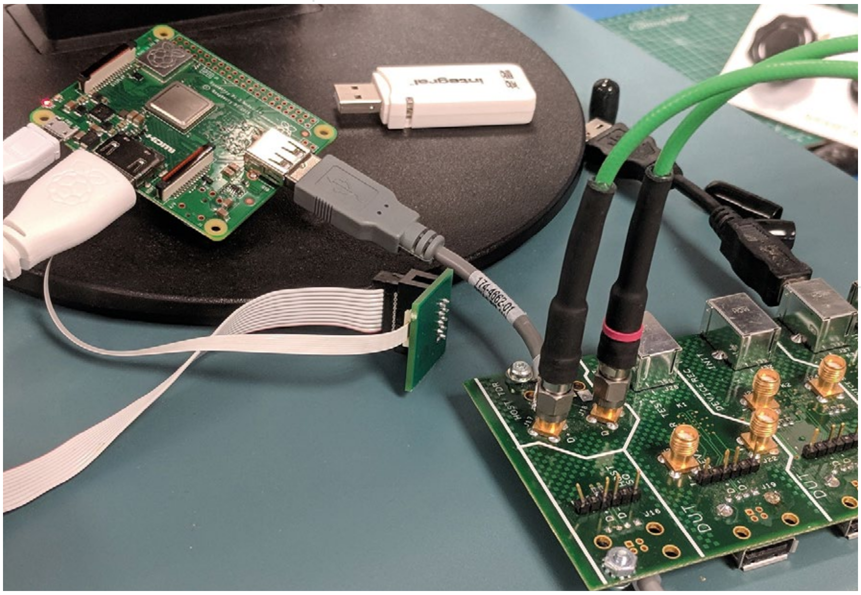
Simplifying certification

"What's the cost to do a certification campaign," Eben asks. "Couple of hundred thousand?"

"For a five-gig one, yeah, it's about 250… 300 000," answers Roger – the cost of clearing a radio-containing product like the Raspberry Pi 3 for general sale.

The Pi 3B+ moved to a modular radio certification, meaning that the existing compliance work remains largely valid for the 3A+. "A lot of the work we'd done to modularise the board meant that we could copy over pretty much all of our compliance work," Roger explains, "so thankfully this product's been just a bit of paperwork. There's definitely been an improvement in time-to-market."





▼ As with any Raspberry Pi, the 3A+ undergoes in-depth testing before it's released to the public

"If we have an opportunity to bring our best tech down a bit, then we do," explains Eben Upton

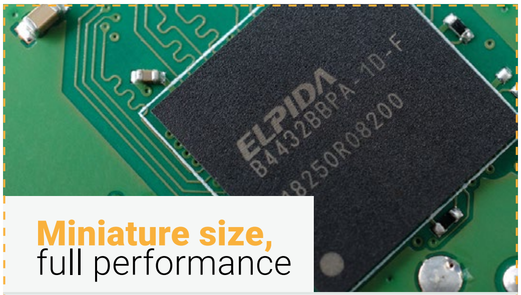
"A lot of people have asked us for an A+ again, and we were able to make that," says Roger Thornton of the launch

for long enough," recalls Eben. "It normally takes six to nine months for us to get our feet under ourselves, and by that time it was very clear that the 3B was imminent."

"The Zero W happened the year after the 3B," Roger adds, "and to some extent addressed the gap that an A+ might have occupied."

The Zero family isn't a complete replacement for the A+, though, as the launch of the Pi 3B+ earlier this year hammered home.

Miniature size, full performance



"We're confident that the performance you saw on a 3B+ you'll be able to get on a 3A+," says Roger. "There's no reduction because of the board size change; you should still be able to run the processor at the same speed. We have had to reduce the memory to hit the price point we're after, but it still runs nicely."

"When we launched the original A+, we launched it with half the RAM of the original B+," Eben adds. "At the time that meant 256MB; now halving the RAM still leaves you with half a gig, which is kind of fun."

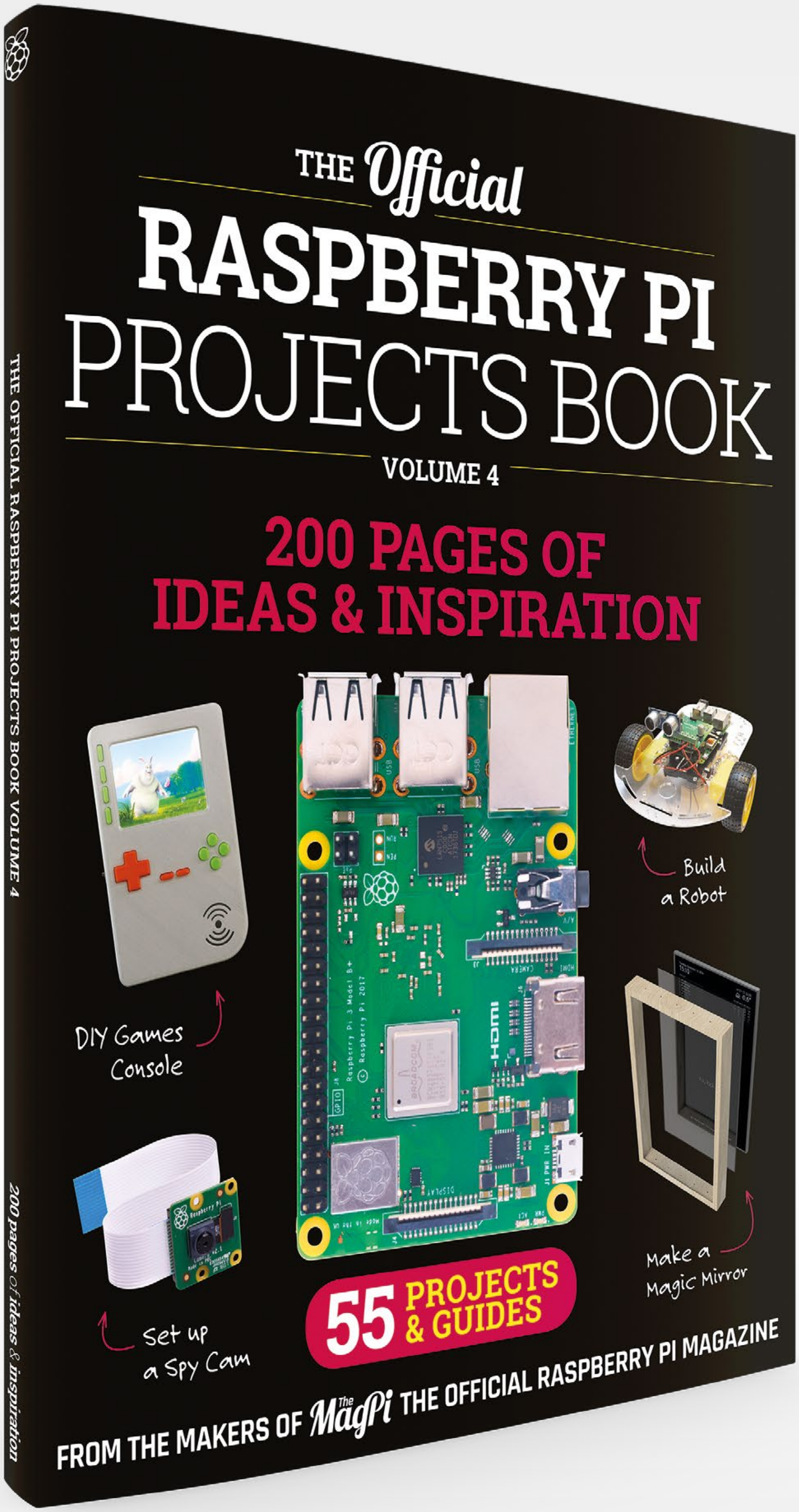
Filling a hole

"There's a massive performance gulf between the 3B+ and Zero," explains Eben. "3A+ represents us bringing our best technology to the lowest possible price point. We can't bring this technology down to the Zero price point, at the moment, but we can bring it down a bit.

"We don't just like to sit there at $35 feeling pleased with ourselves. If we have an opportunity to bring our best tech down a bit, then we do. I mean, this is the best product we can make. We don't know how to make anything better at the moment. It's lovely, actually. We like to make physically beautiful objects. They sometimes cost a bit of money to make, or cost a bit of blood, I think, and this… I mean, for me, once we knew we could make it, I think we kind of had to.

> This is us bringing our best technology to the lowest possible price point

"There may be another motivation for doing it: it's kind of tidying up 'classic' Raspberry Pi," Eben admits. "We're not about to launch a Raspberry Pi 4, but our attention is turning to what might go into a Raspberry Pi 4; we're starting the research phase of figuring out what we might do next. Tying up loose ends is an important part of that, and the lack of an A+ form-factor product is a loose end."



THE *Official* RASPBERRY PI PROJECTS BOOK

VOLUME 4

200 PAGES OF IDEAS & INSPIRATION

55 PROJECTS & GUIDES

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

THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 4

200 pages *of* ideas & inspiration

GFX HAT

▶ Pimoroni ▶ magpi.cc/ZWvcLG ▶ £20 / $25

Rob Zwetsloot looks at a slightly simpler, but much more practical, Raspberry Pi display from Pimoroni

We get asked a lot on *The MagPi* about practical projects – in essence, builds that are less experimental party pieces, and something you can use day-to-day to automate or improve your life in some way. There are plenty of fun Raspberry Pi products you can get for creating fun stuff, but there are also the lesser-known add-ons which can truly help you make a more useful project.

This is the sort of category we see the GFX HAT sitting in. That's definitely not a bad thing, though, and if you fancy using it to make a Dalek or a tweeting cat flap, we're definitely not going to stop you. However, the functionality of the GFX HAT makes it perfect for using it in the kind of low-power, headless Raspberry Pi solution that takes up little space.

Useful functionality

What makes the GFX HAT so good for practical projects is that it contains a monochrome, 128×64 display and six capacitive touch buttons. While the display is very simple compared to modern Pi display HATs, it does mean it's easier to program custom interfaces, while still having very basic options for readable text and number displays.

The buttons hook straight into Python, allowing you to control the Pi and the display. They have

SPECS

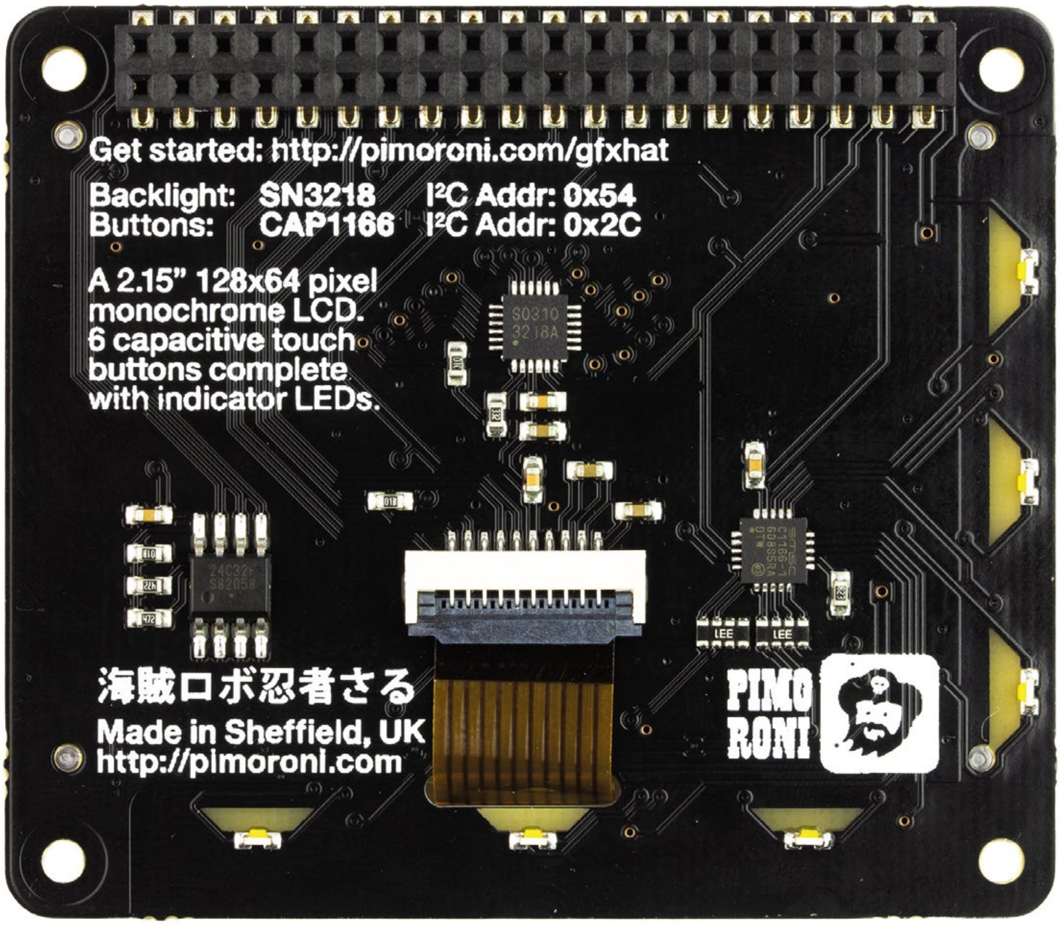
DISPLAY SIZE: 2.15 inches

DISPLAY RESOLUTION: 128×64 monochrome

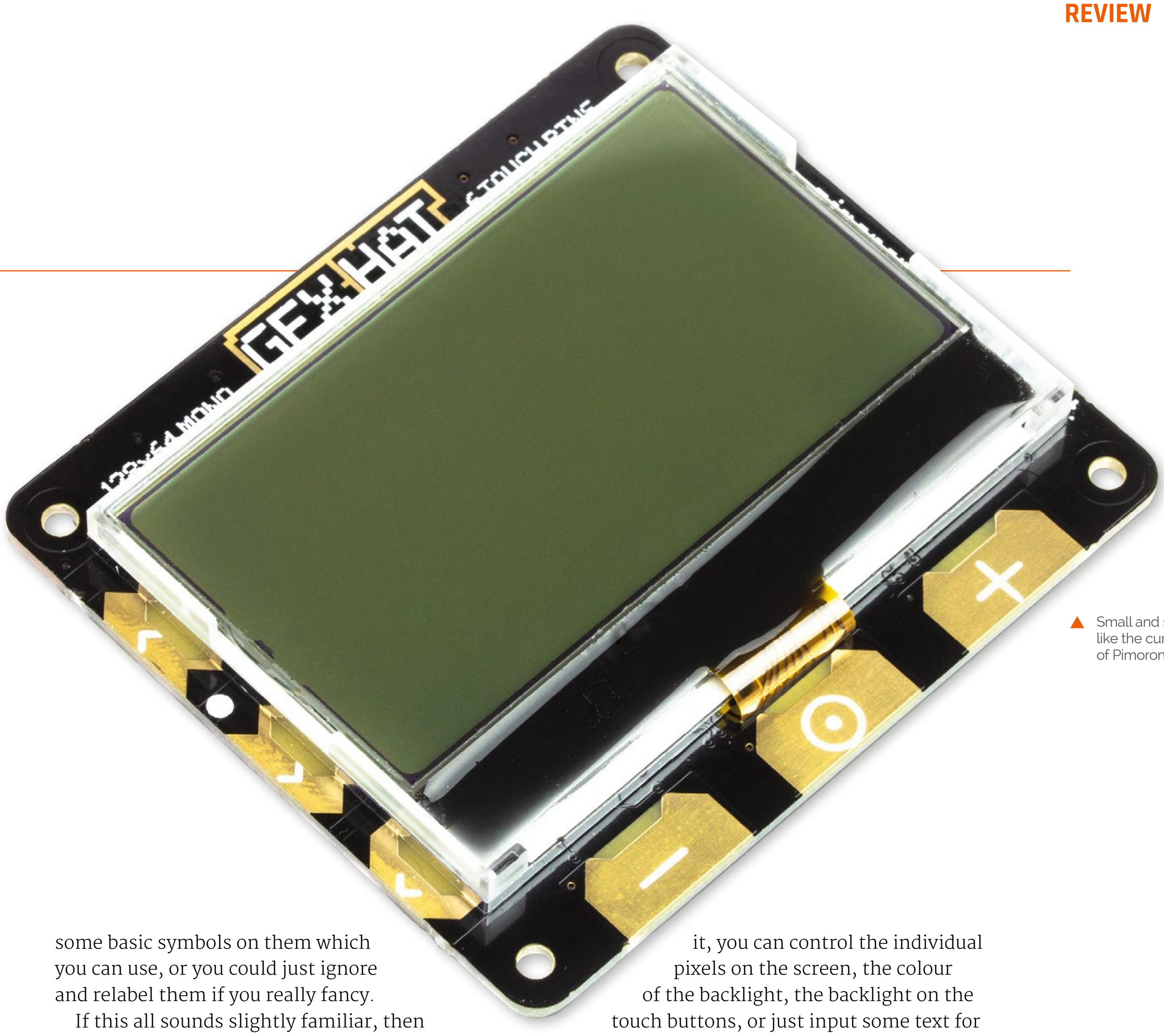
INPUT: 6 × capacitive touch buttons

CODING: Programmable in Python

COMPATIBILITY: All 40-pin GPIO Pi boards


Get started: http://pimoroni.com/gfxhat
Backlight: SN3218 I²C Addr: 0x54
Buttons: CAP1166 I²C Addr: 0x2C
A 2.15" 128x64 pixel monochrome LCD. 6 capacitive touch buttons complete with indicator LEDs.
海賊ロボ忍者さる
Made in Sheffield, UK
http://pimoroni.com
PIMO RONI

▶ All the info you need is on the back, although you can't see that when it's on the Pi



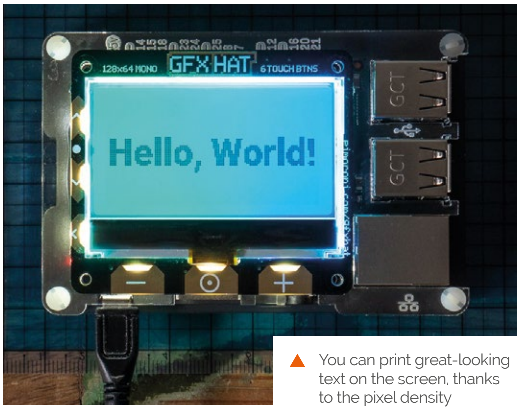
Small and sleek, much like the current range of Pimoroni HATs

some basic symbols on them which you can use, or you could just ignore and relabel them if you really fancy.

If this all sounds slightly familiar, then you may remember an early Pimoroni product called the Display-O-Tron, which had a few buttons and a scrolling display. The GFX HAT is the evolution of this concept, which we regularly used at home in practical projects ourselves.

Programmable and hackable

As with a lot of Pimoroni products, a custom Python library is available for the GFX HAT. With it, you can control the individual pixels on the screen, the colour of the backlight, the backlight on the touch buttons, or just input some text for it to display.



You can print great-looking text on the screen, thanks to the pixel density

> “The programmable backlights really make this something else as well ”

It’s not quite as straightforward as GPIO Zero, because it’s a fair bit more complex than just lighting an LED; however, with the provided examples and some testing, you’ll quickly be able to figure out how to work it.

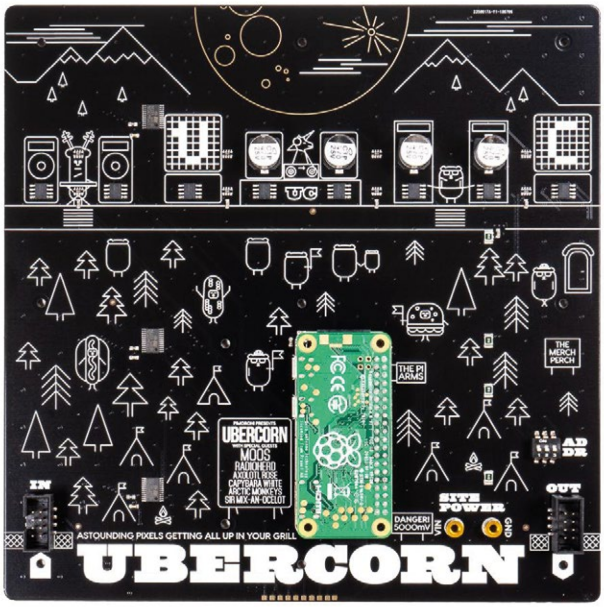
The programmable backlights really make this something else as well – to get visual feedback from your press on the buttons from a UX perspective – and really opens the doors to the kind of projects you could use it for. Simon game, anyone?

We also think it works great, with quick responses, and a very smart look when you use the right case. Hopefully it will prove useful in your next Raspberry Pi project.

Verdict

A great upgrade over the older Display-O-Tron, the GFX HAT is very customisable for use in many practical Pi projects.

9/10



A Pi Zero (or other model) can be securely mounted on the rear – we love the amusing illustrations, too

Ubercorn

▶ Pimoroni ▶ **magpi.cc/eebBWo** ▶ £45 / $49

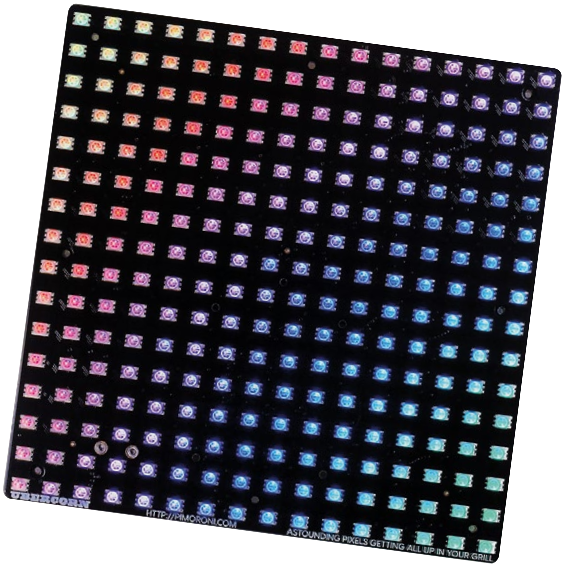
While bigger doesn't always mean better, this super-sized display HAT dazzles **Phil King**

Dwarfing the Unicorn HAT HD, the Ubercorn is even bigger than we anticipated. It comes pre-assembled, so you can just plug a Pi Zero into the female GPIO header on the rear, or use the supplied header extension and metal stand-offs to raise the Ubercorn above the USB ports on a larger Pi model.

Since it is essentially a super-sized version of the Unicorn HAT HD 16×16 matrix display, you can make use of the latter's Python library (**magpi.cc/EETRAi**) to program it, including functions to set individual pixels and scroll text. Standard RGB LEDs are used, so they can be controlled individually, making the Ubercorn suitable for animations. Several code examples are supplied, including a rainbow, graphical effects demo, and slideshow of sprite PNGs.

> The graphical effects look much better when diffused, increasing the perceived size of the pixels

Measuring 19cm square, the Ubercorn's 16×16 RGB LED matrix can produce impressive visual displays



Blurred lines

The graphical effects look much better when diffused, increasing the perceived size of the 5 mm pixels. While no diffuser is supplied, you can make your own with frosted Perspex or even blank pieces of white paper.

As the Ubercorn draws around 1 A of current, a connected Pi will easily power it. If you want to daisy-chain (up to eight) Ubercorns using the eight-pin data in/out headers, however, you'll need to power them via their screw terminals.

As the current flow is limited by three on-board 560 Ω resistors to avoid overheating, the display is a little dim in daylight – particularly when diffused – at the default 0.5 brightness, but turn it up to max and it's plenty bright enough.

SPECS

DIMENSIONS:
19×19 cm

PIXELS:
256 (16×16) RGB LEDs

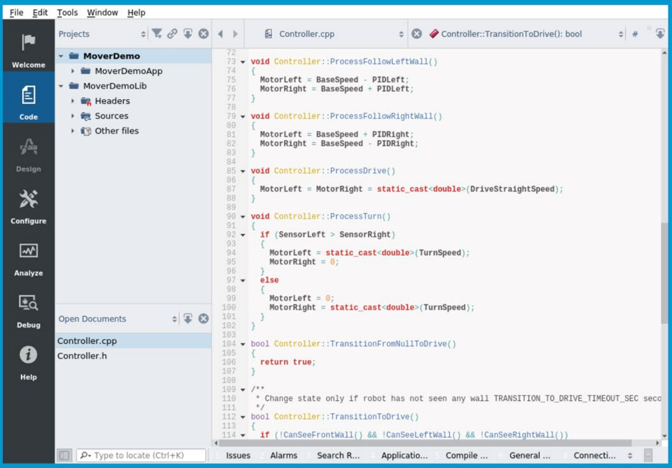
DRIVER CHIP:
ARM STM32F

INTERFACE:
SPI

Verdict

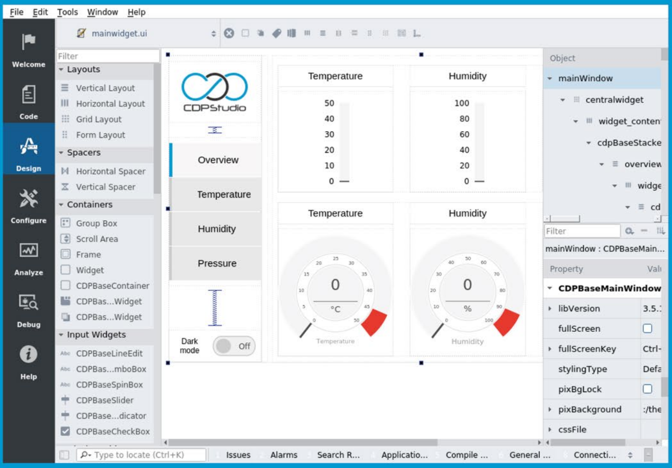
Easy to program, the Ubercorn is a super-impressive large display. Potential projects include a weather display, IoT dashboard, and 'Game Frame' (see **magpi.cc/nbHkow** for an example).

9/10

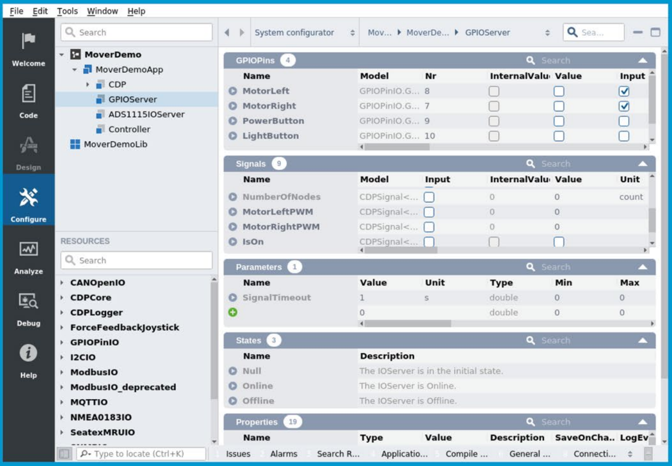


```
void Controller::ProcessFollowLeftWall()
{
  MotorLeft = BaseSpeed - PIDLeft;
  MotorRight = BaseSpeed + PIDLeft;
}

void Controller::ProcessFollowRightWall()
{
  MotorLeft = BaseSpeed + PIDRight;
  MotorRight = BaseSpeed - PIDRight;
}

void Controller::ProcessDrive()
{
  MotorLeft = MotorRight = static_cast<double>(DriveStraightSpeed);
}

void Controller::ProcessTurn()
{
  if (SensorLeft > SensorRight)
  {
    MotorLeft = static_cast<double>(TurnSpeed);
    MotorRight = 0;
  }
  else
  {
    MotorLeft = 0;
    MotorRight = static_cast<double>(TurnSpeed);
  }
}

bool Controller::TransitionFromNullToDrive()
{
  return true;
}

/**
 * Change state only if robot has not seen any wall TRANSITION_TO_DRIVE_TIMEOUT_SEC seco
 */
bool Controller::TransitionToDrive()
{
  if (!CanSeeFrontWall() && !CanSeeLeftWall() && !CanSeeRightWall())
```

Code



Design



GPIOPins 4

Name	Model	Nr	InternalValu	Value	Input
MotorLeft	GPIOPinIO.G...	8	□	□	☑
MotorRight	GPIOPinIO.G...	7	□	□	☑
PowerButton	GPIOPinIO.G...	9	□	□	□
LightButton	GPIOPinIO.G...	10	□	□	□

Signals 9

Name	Model	Input	InternalValu	Value	Unit
NumberOfNodes	CDPSignal<...	□	0	0	count
MotorLeftPWM	CDPSignal<...	□	0	0	
MotorRightPWM	CDPSignal<...	□	0	0	
IsOn	CDPSignal<...	□	□	□	

Parameters 1

Name	Value	Unit	Type	Min	Max
SignalTimeout	1	s	double	0	0
	0		double	0	0

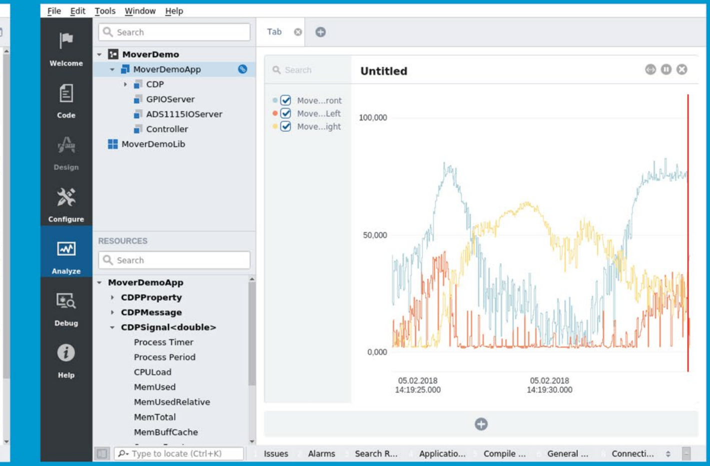
States 3

Name	Description
Null	The IOServer is in the initial state.
Online	The IOServer is Online.
Offline	The IOServer is Offline.

Properties 19

Name	Type	Value	Description	SaveOnCha..	LogEv

Configure



Analyze

Now free for home projects

A professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

Free download on www.cdpstudio.com

CDP Technologies AS
Nedre Strandgate 29
P.O. Box 144
NO-6001 Ålesund, Norway

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

CDPStudio

10 Best:

Raspberry Pi Christmas gifts

Fill some stockings with these Pi-powered presents

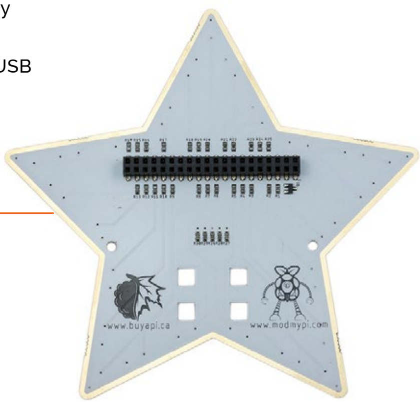
As this issue comes out at the start of December, hopefully this little list arrives just in time to help you get something for the maker in your life. From all of us at *The MagPi*, have a Merry Christmas and a Happy Holiday!

Christmas Tree Star

TYPE: Pi Zero tree star

This Pi Zero HAT is also a star you can attach to your tree! Upgrading your decorations with a Raspberry Pi is always cool. You'll need to make sure you have a very long USB power cable, though.

- £13 / $17
- **magpi.cc/woQGiF**



3D Xmas Tree

TYPE: Three-dimensional light-up kit

Although there are a few Christmas tree kits available for the Raspberry Pi, none of them is three-dimensional like this one! It comes in solder-yourself and pre-soldered versions..

- £15 / $19
- **magpi.cc/2GewFwe**

Mood Light

TYPE: Fun little light project

This perfect stocking-sized gift is part of the excellent range of Pimoroni Pi Zero W project kits. Keep it by your bed or in your lounge and use it to show off your big moods with this highly customisable build.

- £30 / $33
- **magpi.cc/xqySvs**

SnowPi

TYPE: The GPIO snowman

We love this classic project kit, so we're more than happy to bring it back for this year's Christmas guide. It fits on top of a Raspberry Pi, and comes as an adorable little kit that's perfect for practising a bit of soldering and coding on Christmas Day.

- £6 / $8
- **magpi.cc/2GdupFA**

Christmas Tree Solder Kit

TYPE: For makers or programmers

There are two versions of this kit: the one for makers which you just need to solder together and run, and the version for programmers that you can build and then hack and modify. Choices, choices.

- £6 / $8
- **magpi.cc/efhJHh**

InsPiRing

TYPE: Customisable light arrays

Want to build the ultimate Christmas light display? Look no further than the RasPiO InsPiRing range for creating weird and wonderful shapes that are sure to make the neighbours jealous.

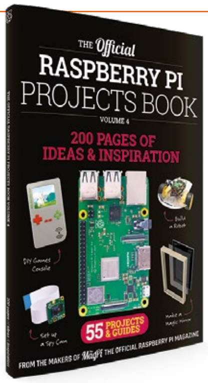
- £10 / $13
- **rasp.io/inspiring**

The Official Raspberry Pi Projects Book Volume 4

TYPE: Handy book of projects

Knowledge is power, and our latest Projects Book has 200 pages of projects and tutorials to inspire and guide you in your journey to becoming a maker. It also slots nicely into a stocking.

- £13 / $17
- **magpi.cc/projects4**

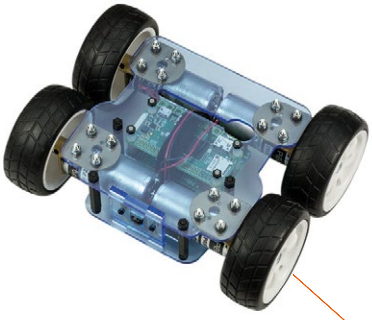


OctoCam

TYPE: Stick to the camera

This cute little kit includes a tiny 5MP camera. You can stick its suckers to any window for a peek outside, or – using the supplied desk stand – put it on a shelf to keep an eye on your all-important gifts under the tree.

- £40 / $43
- **magpi.cc/RqFLZW**



YetiBorg v2

TYPE: The ultimate tiny robot

This tiny robot kit can just about squeeze into a stocking, and is the ultimate little toy to play around with on Christmas Day – after some adult-supervised building, that is. Once you're done playing around with it, you can start hacking it to do even more.

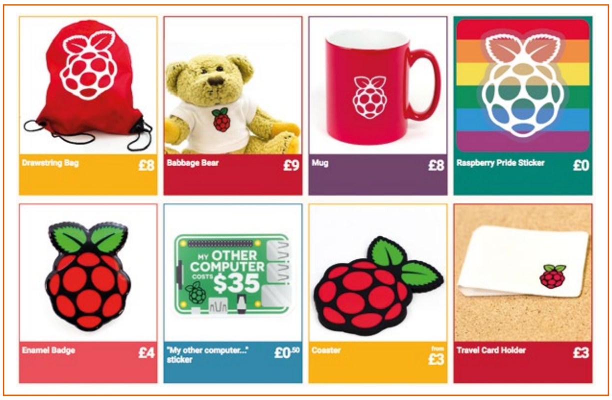
- £160 / $218
- **magpi.cc/TADbiy**

Raspberry Pi Swag

TYPE: Stuff for fans

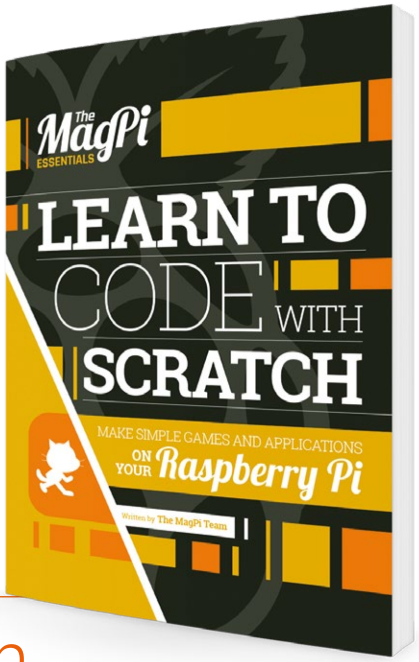
Pins, bears, mugs, stickers, and coasters for the Raspberry Pi fan in your life. And the best part is, any profits go straight to the Raspberry Pi's charitable mission! We especially like the enamel badges; they're very smart.

- Various
- **rpf.io/swag**



CHRISTMAS PROJECTS

Have a Christmas project you'd like to show us? Make sure to tag us in social media posts (Twitter **@TheMagPi** and Facebook **TheMagPiMagazine**) and maybe you'll be featured next issue…


The MagPi
ESSENTIALS
LEARN TO CODE WITH SCRATCH
MAKE SIMPLE GAMES AND APPLICATIONS ON YOUR Raspberry Pi
Written by The MagPi Team

Learn Scratch with Raspberry Pi

Scratch is the perfect first programming language.
Sean McManus rounds up some of the best resources

Learn to Code with Scratch

AUTHOR **The MagPi Team**

Price:
£4 / $5 (free download)

magpi.cc/learnscratch

Our own Essentials book brings together some of our best Scratch articles, with content that's exclusive to this book. Chapters are typically three or four pages, and each one explains a separate project.

Following an introduction to making your first Scratch program, there are chapters on creating games called Bouncy Hedgehog (a catching game), Boat Race (a steering game), and a multiple-choice quiz. Non-game projects include a simple chatbot you can extend, a poetry generator, and LED traffic lights. The traffic lights use the Raspberry Pi's GPIO pins and components from the CamJam EduKit #1 (available separately).

There are two chapters to help you customise your own Scratch games: one shows you how to add a title screen to your game to give it a professional feel, and the other shows you how to add a high-score table to keep players coming back for more.

The book builds up to a ten-page project to build a 3D space shooter and concludes with a reference guide to the Scratch blocks.

Scratch Books

Learn Scratch with these books

COOL SCRATCH PROJECTS IN EASY STEPS

Covering both Scratch 1.4 and Scratch 2.0, this book includes anaglyph glasses for eye-popping 3D effects, music and art projects, maze games, and a Raspberry Pi stop-motion movie creator.

▶ magpi.cc/rFxcha

LIFELONG KINDERGARTEN

Explore the educational philosophy behind Scratch's design with this book by Mitchel Resnick, who leads the Scratch development team. Includes tips for teaching with Scratch.

▶ magpi.cc/XkAYXE

SUPER SCRATCH PROGRAMMING ADVENTURE

A fun comic strip links the nine projects in this book, which gradually build your Scratch skills. Games include a soccer game, side-scrolling racing game, and a combat game.

▶ nostarch.com/scratch1

ScratchEd

AUTHOR

Lifelong Kindergarten, OHO Interactive

Price:
Free

magpi.cc/vTfPtH

ScratchEd is a community for teachers and other educators using Scratch, created by the team behind Scratch itself. At the heart of the site is a directory of Scratch resources, making it a great destination for learners too.

The resources are organised by education level, content type, curricular area, and language. The curricular areas include engineering, maths, music, visual arts, and social studies. At the time of writing, there are 603 English resources, so it's a great place to go if you're looking for inspiration, tutorials, or lesson plans.

ScratchEd has a forum too, although it is a bit quiet there. If you're looking for help with scripts or other issues unrelated to teaching, try the much busier main Scratch forum first (**magpi.cc/kucJpY**).

Scratch Coding Cards

AUTHOR

Natalie Rusk, Lifelong Kindergarten Group, MIT

Price:
£20 / $25
(free download)

nostarch.com/scratchcards

This pack of 75 cards for Scratch 2.0 is ideal for younger readers who might feel overwhelmed by a book. The cards break some simple projects down into easy-to-follow steps. Each card contains instructions for adding one feature to the program with a short script.

The games included are Pong, a catching game, and a multiplayer race game. The cards go beyond games, though, to cover fashion (with a dress-up project), music, dance, a virtual pet, and storytelling. While some projects require readers to follow the cards in the correct order like a book, others give you the freedom to use the cards in any order.

High-quality cards are available to buy in a box from No Starch, or you can print your own (**magpi.cc/vkRjlA**).

Color Clicker
Spin
Play a Sound
Dancing Letter
Change Size
Press a Key
Glide Around
scratch.mit.edu/name
GET READY
Choose a backdrop.
Choose a letter from the Letters category.
ADD THIS CODE
Try different numbers.
TRY IT
Click your letter to start.
TIP
Drag your sprite to where you want it and then add a glide block to your script.
When you drag a sprite, its x and y position will update in the Blocks palette.

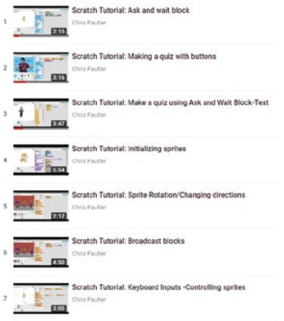
YouTube Videos

Discover Scratch 1.4 tips and tricks

SCRATCH MINI TUTORIALS

Chris Pautler provides short videos on topics such as keeping score, keyboard inputs, and using text the player types in. Well-paced, bite-size chunks of Scratch goodness.

▶ **magpi.cc/mmOWUn**


1 Scratch Tutorial: Ask and wait block
Chris Pautler
2:15

2 Scratch Tutorial: Making a quiz with buttons
Chris Pautler
3:16

3 Scratch Tutorial: Make a quiz using Ask and Wait Block-Text
Chris Pautler
3:47

4 Scratch Tutorial: Initializing sprites
Chris Pautler
5:54

5 Scratch Tutorial: Sprite Rotation/Changing directions
Chris Pautler
2:12

6 Scratch Tutorial: Broadcast blocks
Chris Pautler
4:50

7 Scratch Tutorial: Keyboard Inputs -Controlling sprites
Chris Pautler
3:05

MAKE A PLATFORM GAME

This six-minute video walks you through creating a platform game. It doesn't get into level design and characters, but it does show you how to simulate gravity and momentum.

▶ **magpi.cc/WxzDCa**

MAKE TWO-PLAYER PONG

Discover how to make a Pong game with two paddles, a bouncing ball, and a scoring system. This twelve-minute video is well-explained so you can easily follow along.

▶ **magpi.cc/OkEszB**

Randy Lynn

Helping kids learn to code in Mississippi

- Category **Educator**
- Day job **Partner and creative director, MWB**
- Website **kidscodems.org**
- Twitter **@RandyInMS**

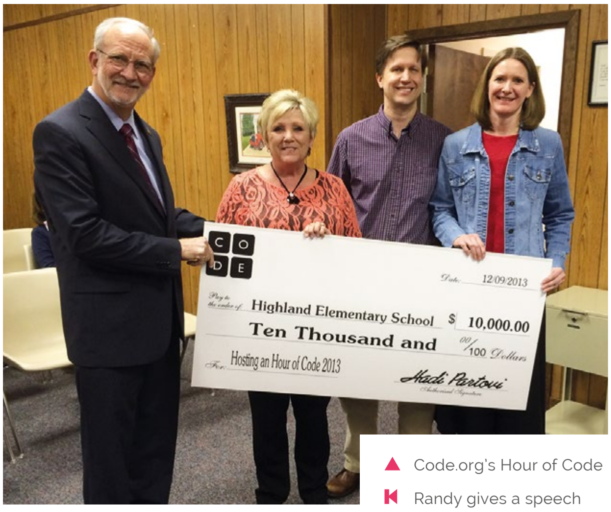
"I first developed an interest for computing in the 1980s after my mother purchased an Apple IIe computer for our family," Randy Lynn explains to us. His story isn't too different from all the kids that grew up with a BBC Micro in the UK. "I learned BASIC language programming on that computer and had lots of fun writing simple programs."

Unlike some other stories, Randy's experience of learning programming at college didn't go so well.

"I quickly realised that I was in over my head," says Randy. "I remember one of my instructors telling me that my code looked like spaghetti due to all the unnecessary branching… It probably didn't help that I had joined a punk rock band and was spending my weekends playing gigs instead of studying and learning to write better code. Eventually, I decided to change my major to focus on advertising."

It wasn't until 2013 that Randy came back to coding, introducing his kids to Scratch, and reintroducing himself to programming. He went on to help their school participate

Kids Code Mississippi holds regular workshops for kids and teachers


C O
D E
Date 12/09/2013
Pay to the order of Highland Elementary School
$ 10,000.00
Ten Thousand and 00/100 Dollars
For Hosting an Hour of Code 2013
Hadi Partovi
Authorized Signature

▲ Code.org's Hour of Code

Randy gives a speech

in Code.org's Hour of Code and, since then, has been a full advocate of computer science in schools, helping to co-found Kids Code Mississippi.

What is Kids Code Mississippi?

Kids Code Mississippi is an advocacy campaign that I co-founded with the president of my agency, Tim Mask. We devote our time to the initiative pro bono because we believe computer science education is vital to modernising our state's economy and making our workforce more competitive.

> A young Mississippian today could change the world in a similar way [to Elvis] with a Raspberry Pi

We've led several hackathons, teacher workshops, and policymaker forums over the past few years. I've also set up Raspberry Pi stations at events, where kids can try out Scratch, Sonic Pi, and the Minecraft Pi Edition. The kids have a great time, and it's a joy to see their eyes light up when taking their first steps with coding.

I've also remained involved in the CS4MS pilot, helping review the curriculum and serving on the steering committee. Most recently, I joined Mississippi's board for ECEP (Expanding Computing Education Pathways), a coalition of states working together to broaden participation in computing with a focus on equity and diversity.

What is your history with the Raspberry Pi?

I first became familiar with Raspberry Pi computers when my son signed up to represent his school in a Technology Student Association (TSA) competition. His team's idea involved a video display, animated characters, sound, button switches, and LEDs. I ordered a Raspberry Pi online, and it was the perfect tool for the job and a lot of fun for the team to use, too.

In 2017, I was at a CS education conference and happened to strike up a conversation with Matt Richardson, Raspberry Pi's Executive Director of North America, where I learned about the Raspberry Pi Foundation's Picademy training. I attended the Atlanta Picademy in 2018, and returned home with lots of ideas for holding local events.

What are your future plans?

I see Raspberry Pi computers as a game-changer for education in our state.

Mississippi has struggled, for decades, with issues like poverty and poor education outcomes. Despite some improvements, we still rank as the poorest state in the US and have the lowest rates of both computer ownership and access to broadband internet. But, at the same time, we Mississippians are known for our creativity. Where in the past, Mississippians like Elvis and Muddy Waters picked up a guitar at a young age and found inspiration, a young Mississippian today could change the world in a similar way with a Raspberry Pi.

Raspberry Pi's low price point means school districts can add a lot of computing power at a very low cost, which is critical for those areas where the needs are greatest. The computers are not only perfect for teaching kids how to code, but educators can incorporate physical computing projects into the curriculum, too, thanks to the GPIO pins. I believe we are ripe for a 'Raspberry Revolution'.

December: Computer Education Month

Last year, Kids Code Mississippi managed to bring their campaign to the Governor of the state, who was able to institute December (this month!) as 'Mississippi Computer Science Education Month'. You can join in wherever you live, though. Read more about it here: **magpi.cc/xgqjUR**

This Month in Raspberry Pi

Halloween 2018

The Raspberry Pi community gets spooky for October

Although this is the December edition of *The MagPi*, due to our schedule, Halloween occurred while we were writing this issue. Halloween is a great time in the community, as folks show off their scary projects and spooky builds.

Here are some of the best ones we saw! Feel like we overlooked yours? Drop us a line at **magpi@raspberrypi.org**!

A fun skull project we got sent on Twitter – this one plays a little Halloween message at the touch of the button. Hear what it has to say at the link… **magpi.cc/ZyXwNe**

This Pi Zero 'Scary Halloween Ghost' has a motion sensor to detect when someone is going by so that it can light up and shake and scare them. There are also full build instructions for it! **magpi.cc/WPLGnW**

Halloween has become popular in the Netherlands, according to maker Jaap Meihers: "Houses where candy is given out are decorated. It's not a competition (yet), but if it were, this year our house would have won. Using a projector and a Raspberry Pi running media player OSMC, we made it look like zombies were walking around in our house." **magpi.cc/gdAbxa**

Credit: Alasdair Allan – Babilim Light Industries

European Astro Pi Challenge update

Mission Space Lab: Phase One, complete

Listen up, cadets. We're pleased to tell you that the first phase of Mission Space Lab was a complete success. Here's a note from Astro Pi command:

"ESA Education and the Raspberry Pi Foundation are delighted to announce that Phase 2 of the European Astro Pi Challenge: Mission Space Lab has begun. During Phase 1, we received a record-breaking 471 entries from 24 countries! Now, the 378 selected teams will have the chance to write computer programs for the scientific experiments they want to send to the Astro Pi computers aboard the International Space Station (ISS)."

The selected teams will receive an European Astro Pi Challenge kit so that they can develop and test your experiments. The kit includes a Raspberry Pi, two Raspberry Pi Camera Modules, and the Sense HAT. You all have until 6 February to complete your experiments.

A new mission

Well done, cadets! For those who missed out, we have a new mission for you: Mission Zero. Here's Astro Pi command with further details:

"Young people up to the age of 14 still have to the opportunity to take part in our second challenge, Mission Zero. Teams will have until 20 March 2019 to write a simple program to display their personal message to the astronauts on board. You don't need any special equipment or prior coding skills, and all participants that follow the guidelines are guaranteed to have their programs run in space."

Head to the Mission Zero website to learn more: **magpi.cc/isLMBf**.

> "Young people up to the age of 14 still have to the opportunity to take part"

01

02

03

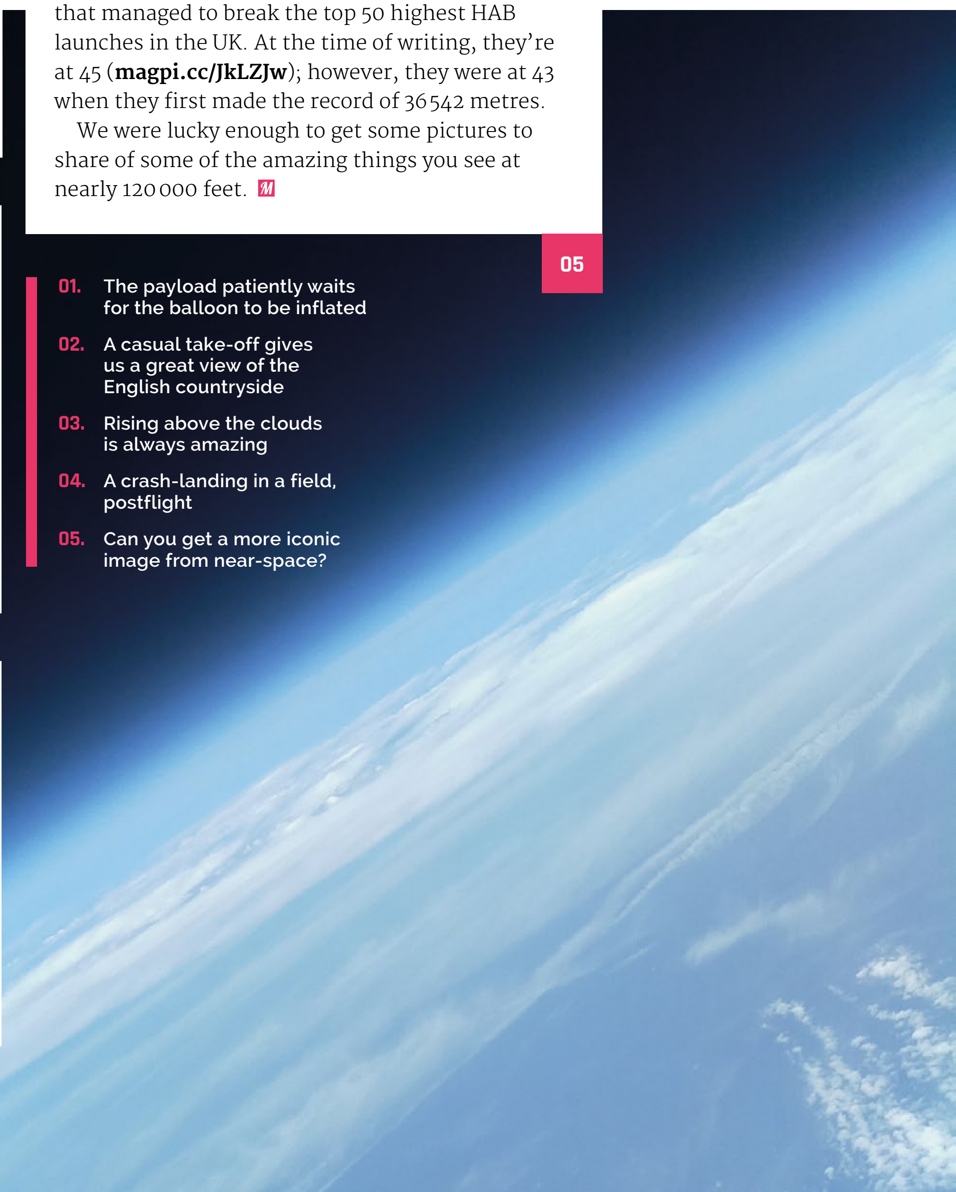
04

School HAB success

Reaching the top 50 highest launches in the UK, thanks to Skycademy

It's not every day we're told about successful HAB launches from schools, but we received an email about an incredible launch from Demeter House school, an SEN school in North Lincolnshire, that managed to break the top 50 highest HAB launches in the UK. At the time of writing, they're at 45 (**magpi.cc/JkLZJw**); however, they were at 43 when they first made the record of 36 542 metres.

We were lucky enough to get some pictures to share of some of the amazing things you see at nearly 120 000 feet.

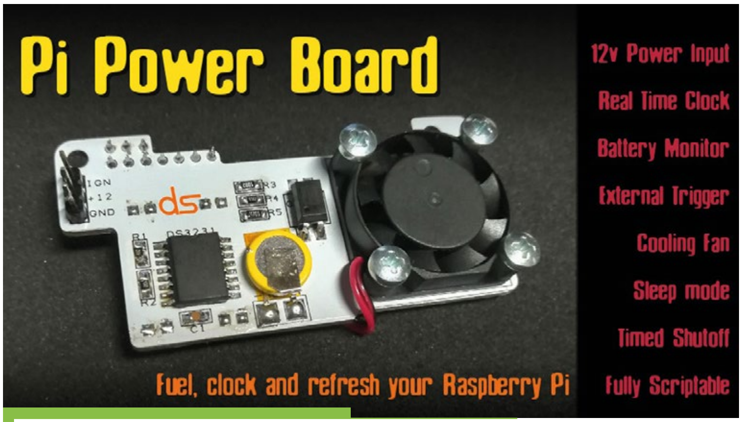


05

01. The payload patiently waits for the balloon to be inflated
02. A casual take-off gives us a great view of the English countryside
03. Rising above the clouds is always amazing
04. A crash-landing in a field, postflight
05. Can you get a more iconic image from near-space?

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month



Pi PowerBoard

The Pi PowerBoard is a programmable power board for the Raspberry Pi. It includes a real-time clock, 12 V power input, voltage monitor, and is hackable so you can set triggers and timings and such. The creator made it as he was looking for a similar power adapter for his Pi to be used in a car. Now you can get one too.

▶ **kck.st/2z4LBer**

CROWDFUNDING A PI PROJECT?

If you've launched an irresistible Pi-related project, let us know!

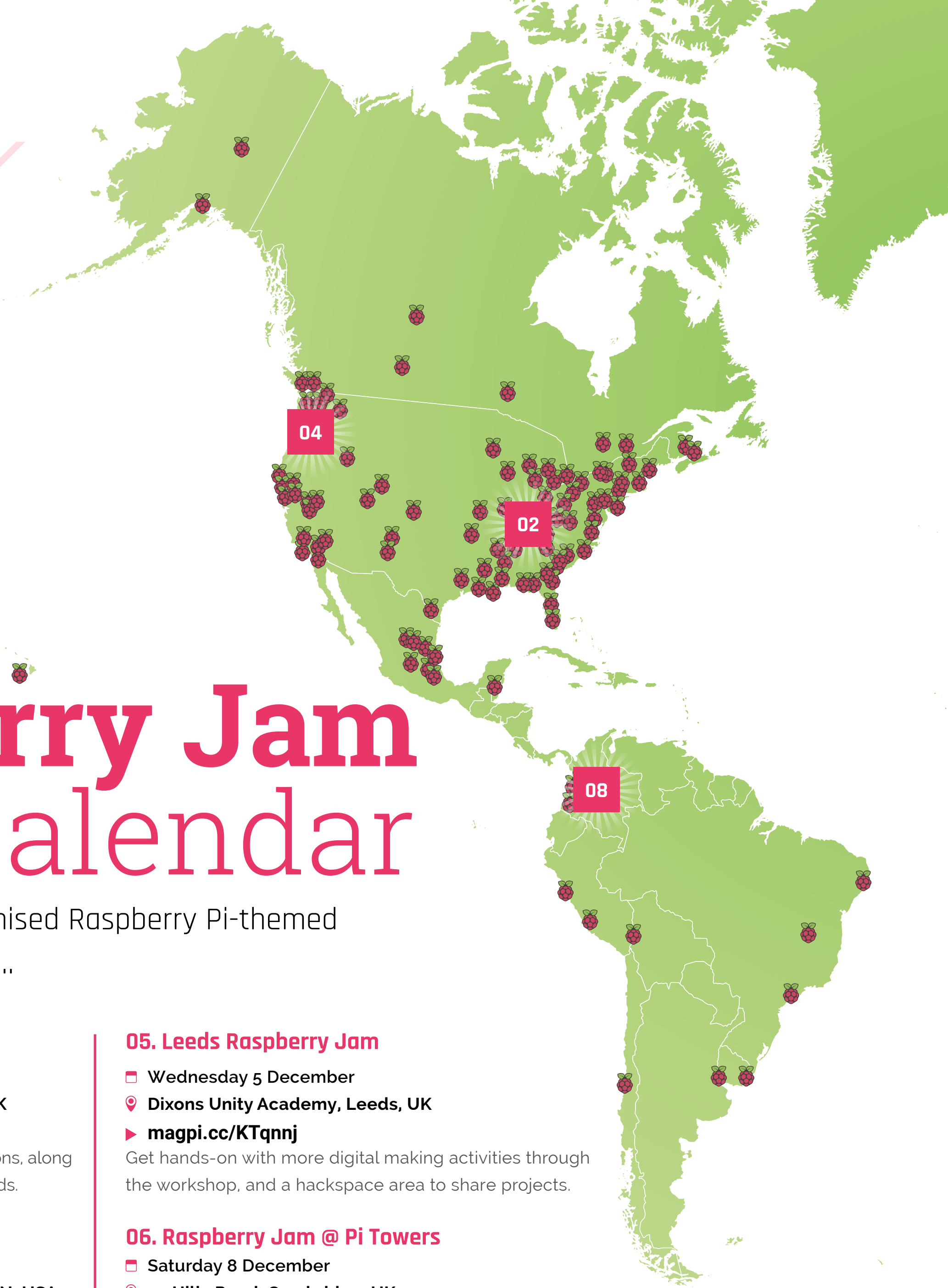
magpi@raspberrypi.org

WildCats3D High Resolution 3D Scanner MKII
New MKII Stand Alone Hardware - Raspberry PI 3B+ Based
STL Output - Exclusive Points Cloud Editing & Manipulation

High resolution 3D scanner MKII

This 'affordable 3D scanner' is being billed as an excellent Pi project, as it runs on a Raspberry Pi 3B+. 3D scanners can be very expensive, so the sub-£300 price point is actually pretty good for such a product, even if you do have to assemble it yourself.

▶ **kck.st/2pULP2Q**



Raspberry Jam Event Calendar

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

01. Bognor Regis Raspberry Jam

- Saturday 1 December
- **University of Chichester, Bognor Regis, UK**
- **magpi.cc/pVNTsb**

There will be a mix of show-and-tell presentations, along with tutorials and chances to tinker with Pi boards.

02. FSSD Evening of Code

- Thursday 6 December
- **Freedom Intermediate School, Franklin, TN, USA**
- **magpi.cc/bKFqKz**

The community is welcome to come and see what the students of Freedom Intermediate are learning about computational thinking.

03. Christmas Raspberry Jam

- Saturday 8 December
- **Worksop College, Worksop, UK**
- **magpi.cc/pZkWUP**

This Christmas-themed Raspberry Jam will have seasonal projects to see and play with.

04. Raspberry Pi Jam Session & Code Carnival

- Saturday 8 December
- **Third Place Commons, Seattle, WA, USA**
- **magpi.cc/sxSbGA**

Join fellow coders, builders, and Raspberry Pi makers in an eight-hour, all-day event.

05. Leeds Raspberry Jam

- Wednesday 5 December
- **Dixons Unity Academy, Leeds, UK**
- **magpi.cc/KTqnnj**

Get hands-on with more digital making activities through the workshop, and a hackspace area to share projects.

06. Raspberry Jam @ Pi Towers

- Saturday 8 December
- **37 Hills Road, Cambridge, UK**
- **magpi.cc/xpGiso**

A family-friendly event where everyone is welcome. Coding and digital making activities will be provided for participants.

07. Manchester Raspberry Jam

- Saturday 8 December
- **The Shed, Manchester, UK**
- **magpi.cc/EJzNAv**

Attend this monthly Jam to take part in taught workshops or to get help working on your own projects.

08. Yertx Raspberry Jam Colombia

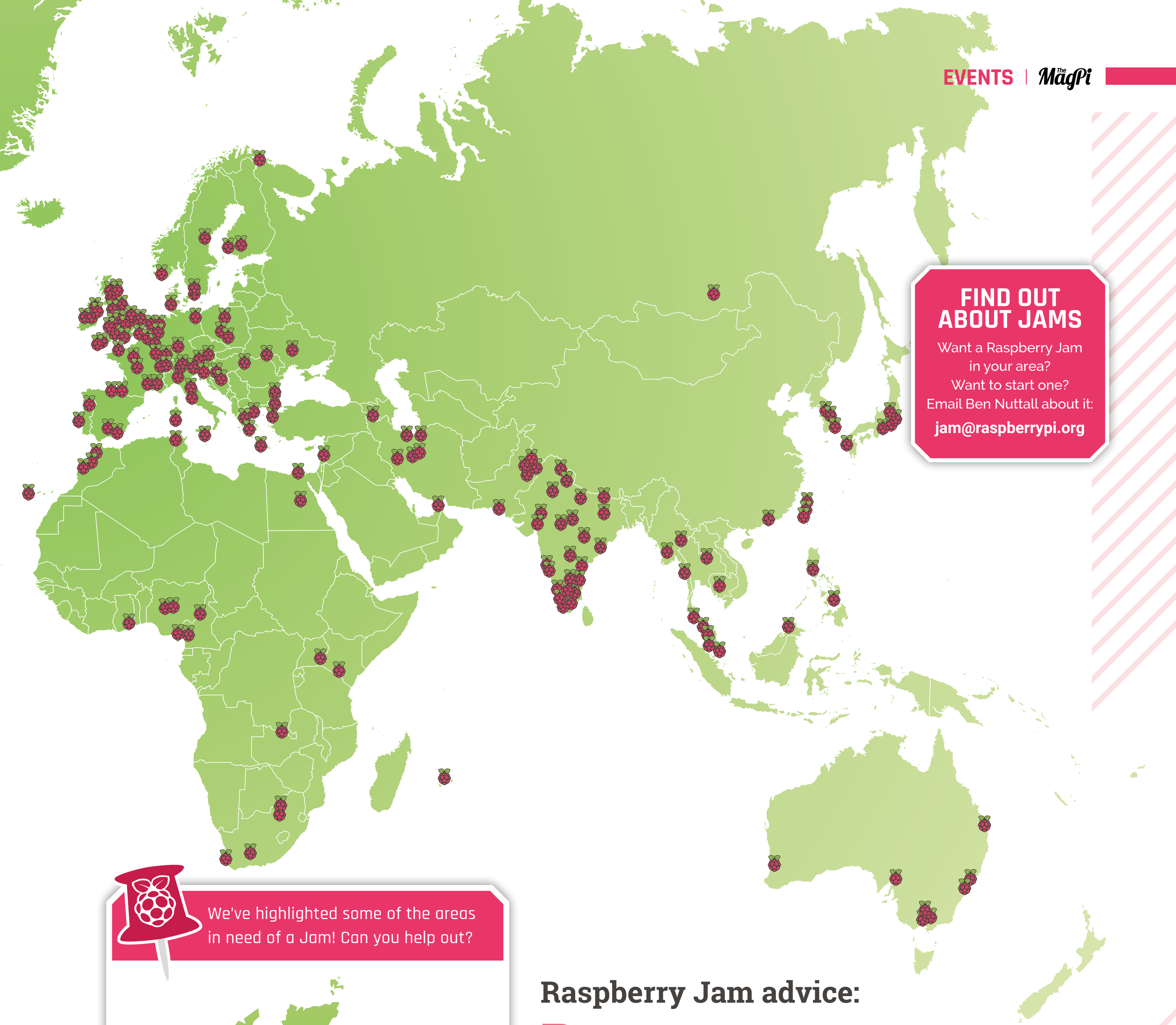
- Saturday 15 December
- **Biblioteca Pública Virgilio Barco, Bogotá, Colombia**
- **magpi.cc/UXPSnm**

Meet others in this Jam to share knowledge and your projects based on Raspberry Pi.

FULL CALENDAR

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

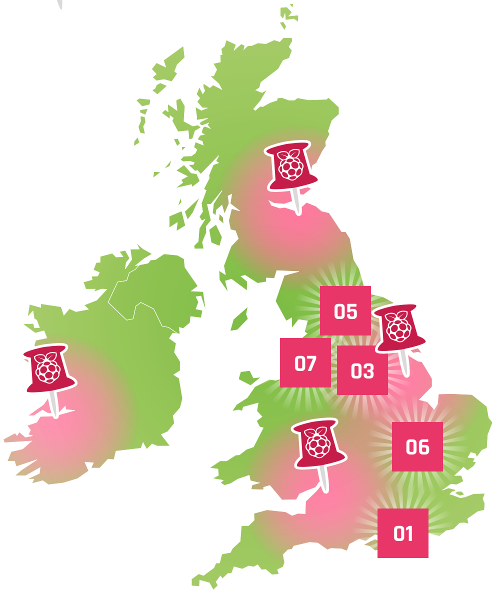
rpf.io/jam



FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area?
Want to start one?
Email Ben Nuttall about it:
jam@raspberrypi.org

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



Raspberry Jam advice:

Post event

> "I know I'll need some good ideas to move beginners on, or they won't return. I know I will have to advertise more widely to get the same number of people next time. People always come in greater numbers to the first of something new."

Anne Cahill – York 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

Get yourself a Raspberry Pi 3A+ when you subscribe to *The MagPi* for a year!

A on the cover

With the announcement of the brand new Raspberry Pi 3A+ I was wondering whether you'd be doing anything special like putting it on the cover?

I still use the Pi Zero I got with your magazine, and would love it if you could do it again! I'll also be able to prepare to grab an extra copy before everyone rushes to the shops.

Eve via email

The Pi 3A+ is not being given away this issue, we're afraid, and we don't usually announce our free give-aways ahead of time either. Sorry!

However, for this month only we have a very special subscription deal: if you take out a 12-month subscription to the magazine, you'll get a Raspberry Pi 3A+ instead of the Pi Zero W bundle! You can find out more details about it on page 26.

Educator discounts

Hi there, is there any sort of discount for schools to subscribe, by any chance?

Dan via Twitter

We're currently in the process of creating a new educators discount for *The MagPi*, and possibly our sister magazines. We had not finalised it before we sent this issue to print – however, you can email **rpipresshelp@raspberrypi.org** and they should be able to help you out in some way.

We're happy to offer the discounted rate to educators, but remember that every issue is also available as a completely free PDF, which might work better in some classroom situations!

THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS
(Hello World)
CHILDREN'S STORIES AND CODING
ETHICS, COMPUTING & THE CLASSROOM
ESCAPE ROOMS
INTRODUCING MICROBLOCKS
GET ON YOUR BIKE WITH MINECRAFT!

Educators should also check out Hello World, our sister magazine made specifically for teachers: **helloworld.cc**

Back issues

I'm looking to complete my collection of back issues of *The MagPi*. Do you have any printed copies left of issues 31 onwards? I'd happily pay for them, as my shelf is looking a little empty!

If not, when do you expect reprints to come in?

Jerry via Facebook

The best place to look for back issues is in our official store, **store.rpipress.cc**. Here you can find single issues of all our magazines. However, once they're sold out, they tend to be gone forever. We rarely do reprints due to the way the magazine industry works.

Also, issues 31 to 37 were never printed. However, with those and other issues, please feel free to download the PDFs and get them printed out for your collection. We only ask that you don't sell these printed-out copies, as that goes against our Creative Commons licence.

All available issues that are still in stock can be bought from our official store: **store.rpipress.cc**

Contact us!

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

Free issues

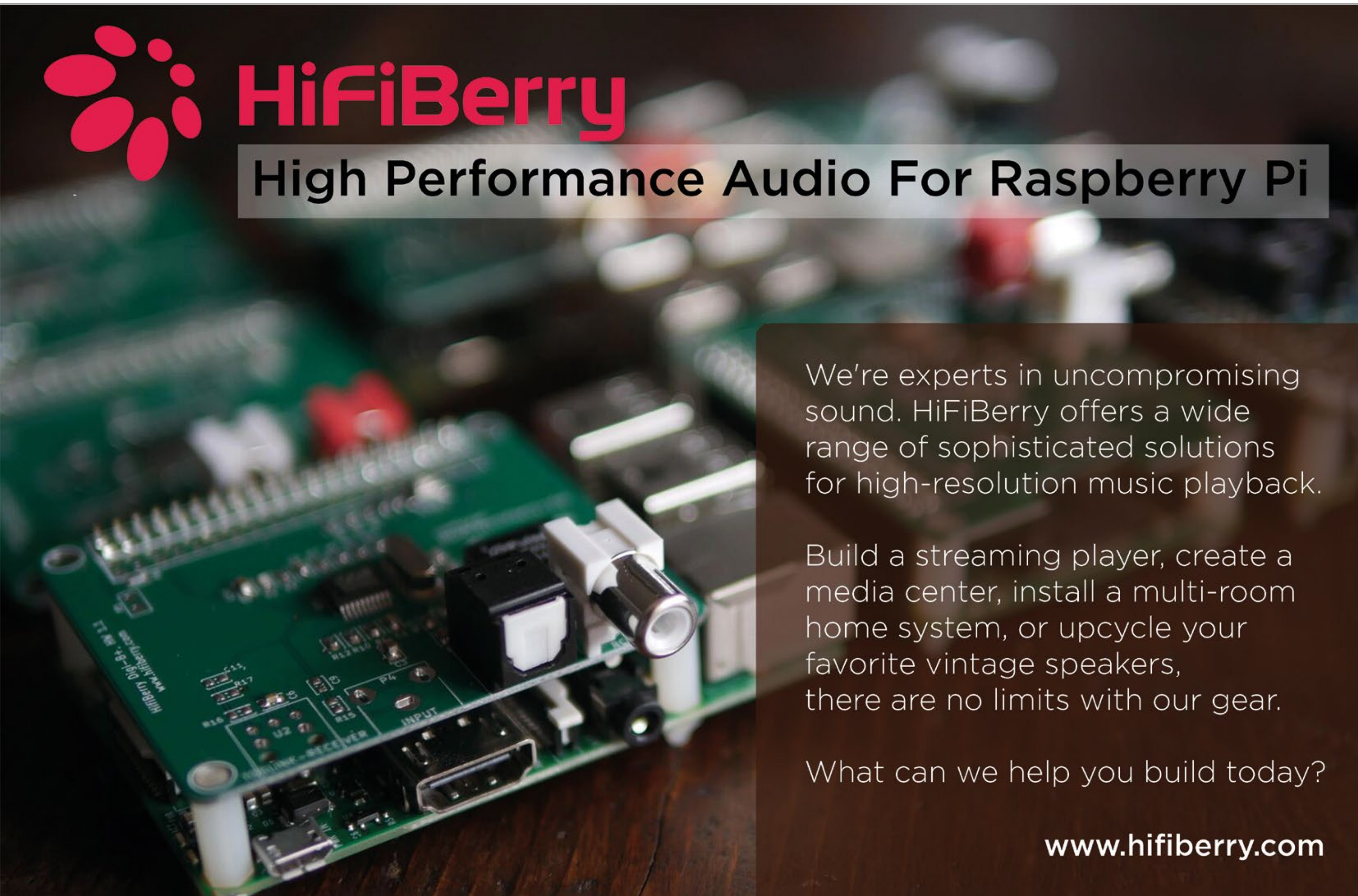
I have an almost complete set of *The MagPi* magazines from Christmas 2015 (issue 40) to September 2018 (issue 73), plus a 2018 *Pi Annual*. I will send them free to whoever wants them, but to one address in one parcel. Please email me if you would like them.

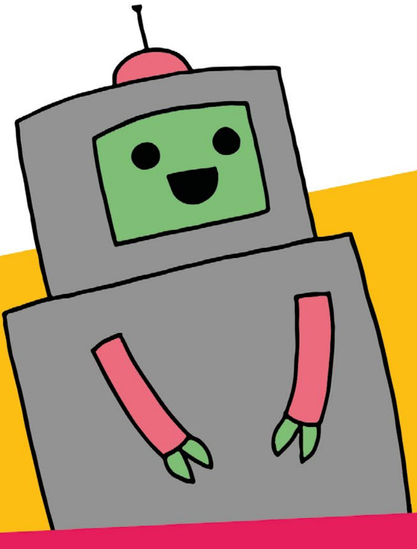
Matthew from the forums

Thanks for letting us know about this, Matthew – you can head to the Raspberry Pi forums, where he's posted his email address to take him up on his offer here: **magpi.cc/UxPzgg**.

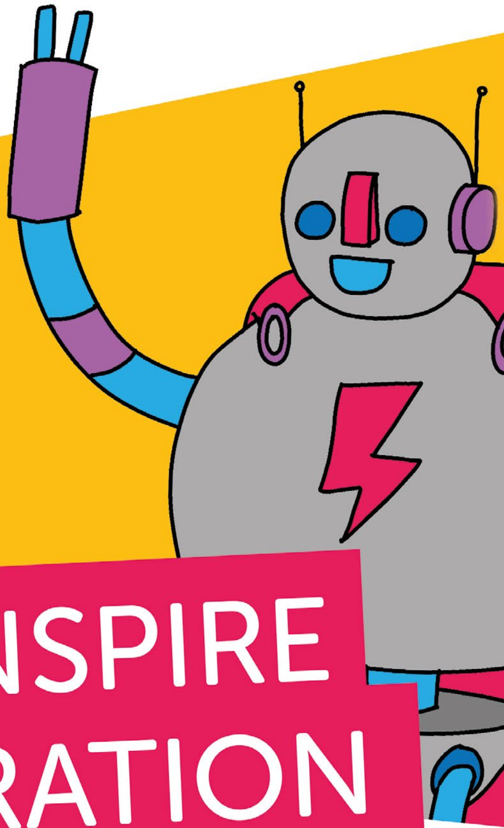
We also have a thread on *The MagPi* forum for listing any other magazines or books you might want to give away. You can find that thread here: **magpi.cc/DDeWqi**.

▲ Please be aware that Matthew will likely not have the free gifts from some of our previous issues, like the AIY Voice Kit





code club

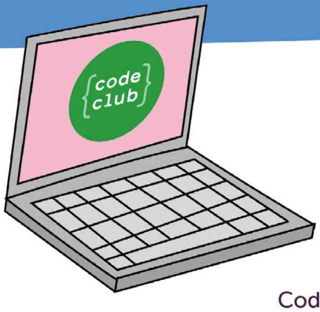


CAN YOU HELP INSPIRE THE NEXT GENERATION OF CODERS?

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.

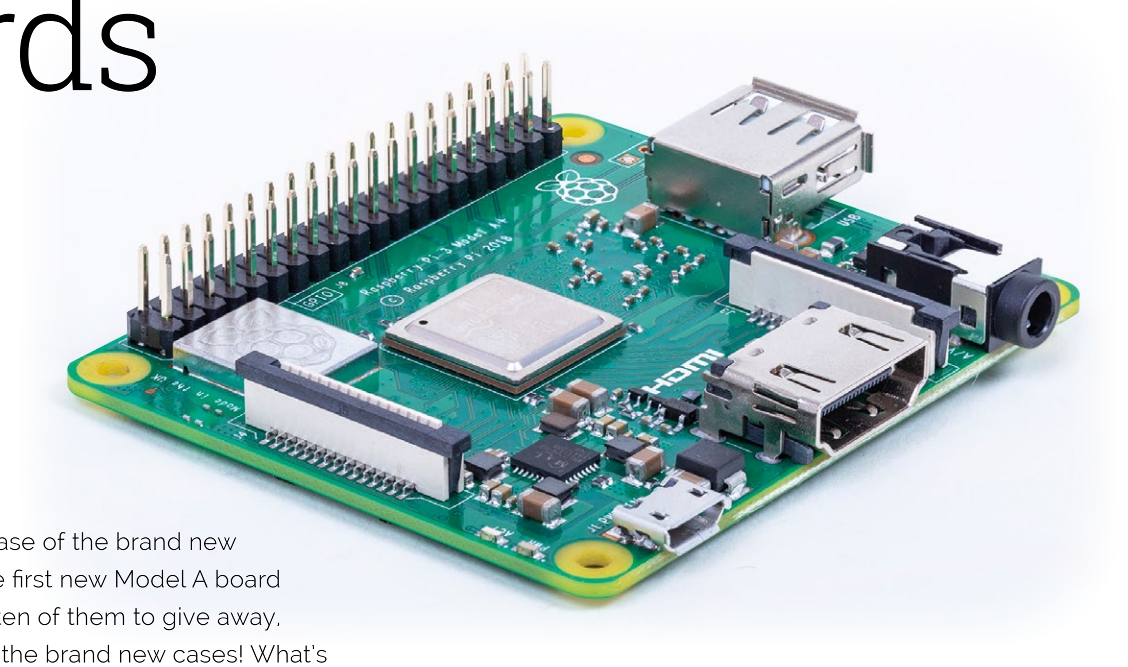


To find out more, join us at

www.codeclubworld.org

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

WIN 10 Signed Raspberry Pi 3A+ boards



To celebrate the release of the brand new **Raspberry Pi 3A+**, the first new Model A board in a while, we've got ten of them to give away, complete with ten of the brand new cases! What's more, you'll get it signed by **Eben Upton**. How cool is that?

Win one of ten 3A+ & cases, signed by Eben himself!

Head here to enter: magpi.cc/win | **Learn more:** magpi.cc/3Aplus

Terms & Conditions

Competition opens on **29 November 2018** and closes on **20 December 2018**. 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.

NEW
£3
ALL FORMATS
Wireframe
LIFTING THE LID ON VIDEO GAMES
Cyberpunk 2077
Adapting tabletop RPGs to video games
Mega Man
The changing face of a console icon
Solid Snake
Metal Gear creator Hideo Kojima profiled
THE BLACKOUT CLUB
CO-OP HORROR FROM
THE DIRECTOR OF BIOSHOCK 2
Issue 2 £3 wfmag.cc
EXCLUSIVE
Far Cry 4's Alex Hutchinson on his "louder, brasher" game
VIDEO

Stay Tuned

THE MAGPI #77 ON SALE 20 DEC

MAKE WITH CODE

Simple coding tricks that bring builds to life

Plus!

Accessorise your Pi

The best HATs, kits, and add-ons

Also

The SelfieBot robot

Build a smart door

Back up your precious data

Code an advanced Pac-Man

Create your own VDU display

And much, much more!

* Contents subject to change

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

DESIGN
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Mike Kay, Harriet Knight

Illustrator
Sam Alder

CONTRIBUTORS
Mike Cook, David Crookes, PJ Evans, Gareth Halfacree, Rosie Hattersley, Nicola King, Sean McManus, Ben Nuttall, Richard Smedley, Mark Vanstone

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org
+44 (0)7904 766523

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
Raspberry Pi Press
Mann Enterprises, Unit E,
Brocks Business Centre, Haverhill,
CB9 8QP

To subscribe
magpi.cc/subscribe

To get help:
rpipresshelp@raspberrypi.org

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., 30 Station Road, Cambridge, CB1 2JH. 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.

ISSN: 2051-9982.

Surfing the **learning curve**

Community Manager **Ben Nuttall** on removing barriers in digital making

The email said: "brain-picking, putting out feelers, other yucky metaphors." It arrived in my inbox on 8 October 2013. The sender: Liz Upton. The three-strong Raspberry Pi Foundation wanted to increase their staff numbers to four and apparently had a Ben Nuttall-shaped hole to fill.

I'll never forget that moment – I was sitting at my desk at my day job in Manchester, making website after website. It seemed Liz had noticed I was quite fond of the Raspberry Pi and thought my combination of web development skills, and experience setting up and running the Manchester Raspberry Jam, would come in handy at Pi Towers (then, a humble single-room office). I jumped at the chance, and became a resident of Cambridge in November, five years ago this month. In my time, I've seen us launch a dozen or so new hardware products, grow to over 100 staff, and we've just moved into our fourth office (where incidentally, I host the Pi Towers Raspberry Jam).

As we've grown and developed, so has my role. I helped launch Picademy; I revamped **raspberrypi.org**, and our first projects site; I ran the Creative Technologists programme; I met the Prime Minister of Estonia at a hackathon; I helped launch the new digital maker Scouts badge; I support Raspberry Jams worldwide; and I recently upgraded our Astro Pi units on the International Space Station to Raspbian Stretch.

Skills for life

Our partnership with The Scout Association is very close to my heart – I lived through the whole Scouting movement from age six to nineteen, and it helped me develop skills for life. I love the idea that today's Beavers, Cubs, and Scouts will get the chance to include digital making in their Scouting experience, and hope that they see the relevance of technology in their adventure-filled lives.

> “ I love the idea that today's Beavers, Cubs, and Scouts will get the chance to include digital making in their experience ”

Lots of people and organisations in tech talk about making learning coding and computing 'easy'. I don't necessarily think this is the right attitude to take. I believe in removing barriers to entry and smoothing the learning curve, making things more accessible, not just easy. GPIO Zero aims to make it easier to get started with physical computing, and helps make the code more approachable to read, write, and understand, but there's no end to where you can take it – you can do some really advanced stuff, and your code is beautifully succinct, not a plate of copy-pasted spaghetti code you cherry-picked from Stack Overflow. A few people have asked me recently about how they can use GPIO Zero to show and teach what's going on underneath and I've demonstrated how they can use the low-level classes in GPIO Zero to get the best of both worlds, which seems to have gone down well.

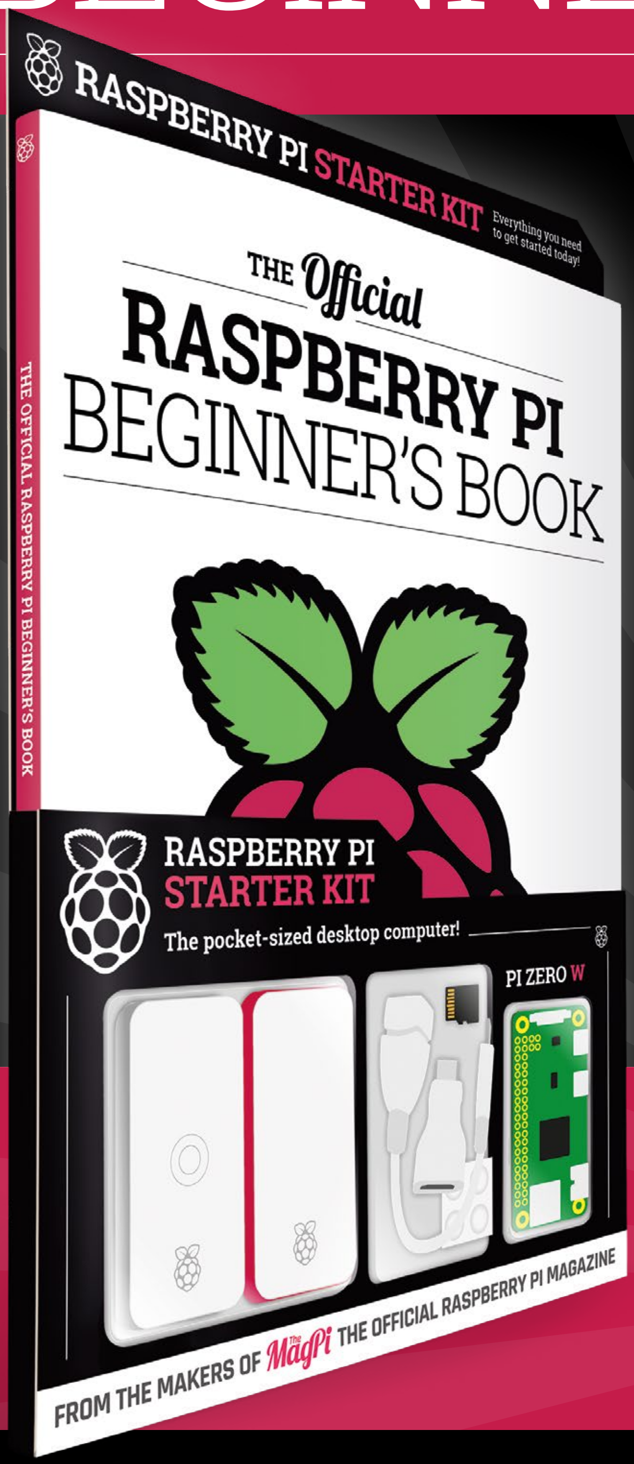
Another project I created to help remove barriers is piwheels, which makes `pip install` fast. The principle is simple – we automate compiling all Python packages on Raspberry Pi 3s (provided by the Mythic Beasts Pi Cloud), and provide them for download (on a Pi running a web server). This Pi currently serves over half a million packages per month, which in the last twelve months has saved Raspberry Pi users a total of 32 years of build time – that's more than my lifetime! This project also helped bring Google's machine-learning framework TensorFlow to the Raspberry Pi.

AUTHOR

Ben Nuttall

Raspberry Pi's Community Manager, creator of GPIO Zero and piwheels, Jam master, and the Foundation's resident Python expert!

bennuttall.com


RASPBERRY PI STARTER KIT
Everything you need to get started today!
THE Official
RASPBERRY PI
BEGINNER'S BOOK
THE OFFICIAL RASPBERRY PI BEGINNER'S BOOK
RASPBERRY PI
STARTER KIT
The pocket-sized desktop computer!
PI ZERO W
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

Available now