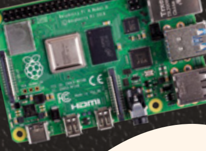


BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

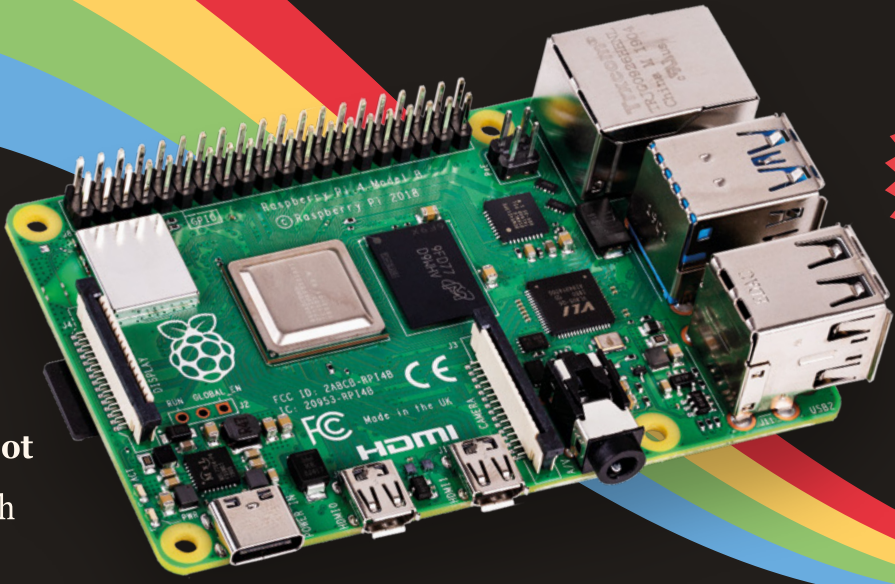
The MagPi

Issue 88 | December 2019 | magpi.cc

The official Raspberry Pi magazine

RETRO COMPUTING
WITH RASPBERRY PI 4

Upcycle a classic computer | Play retro video games | **Rediscover classic coding**



ZX Spectrum Next

British icon reborn

Plus!

- Build the best Christmas projects
- **Assemble a low-cost camera bot**
- Capture the stars with astrophotography

Thermal Testing

Raspberry Pi firmware turns down the heat

£5.99

43 PAGES OF PROJECTS & TUTORIALS

GLOBAL DELIVERY
magpi.cc/store

Raspberry Pi PRESS

Over 10,000 Different Boards and Modules In Stock

NO ONE KNOWS BOARDS LIKE DIGI-KEY!

- Open Source Community Boards
- Evaluation and Development Boards
- Wireless Modules
- Single Board Computers
- Reference Designs
- Programmers, Emulators and Debuggers

DESIGNS START HERE!
DIGIKEY.COM/BOARDS

Digi-Key® ELECTRONICS

MAKER.IO POWERED BY DIGI-KEY

#MAKEWITHDIGIKEY

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2019 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

WELCOME
to *The MagPi* 88

Retro Computing is always close to our hearts here at Raspberry Pi Towers. We grew up with all the greats: BBC Micro, Sinclair Spectrum, Commodore 64 and Amiga.

No matter which classic system you prefer, we can all agree that retro computing on Raspberry Pi has never been better.

With its faster processor and increased RAM, Raspberry Pi 4 can become virtually any 8- or 16-bit computer, enabling you to relive those glory years, rediscover classic programs, and learn computer science from the source. See our wonderful Retro Computing feature (page 24).

Raspberry Pi 4 is a powerful computer – more power needs more energy, and that means more heat. The engineering team have been hard at work bringing down the heat, through clever tweaks that keep the speed levels up, but the heat demands down.

We've expanded our heat tests for Raspberry Pi 4 and the result is a sublime Thermal Testing the Raspberry Pi 4 feature (page 66). It's packed with insider information about how computers are made.

Finally: Merry Christmas! Rob starts Christmas around October and has been working up to a big Christmas Lights tutorial (page 40) and the Top 10 Christmas Projects (page 80). Both are cracking features.

I hope you all have a great holiday season, and make lots of amazing things. Don't forget to share your projects with us!

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Lucy is Editor of *The MagPi*. Her first computer was a ZX Spectrum, but it was Commodore that stole her heart. First with the C64, then the Amiga.

magpi.cc

GET A RASPBERRY ZERO W KIT
WITH A SUBSCRIPTION!
PAGE 22

WIN! RASPBERRY PI ROBOT ARM KITS
The MagPi
The official Raspberry Pi magazine
RETRO COMPUTING WITH RASPBERRY PI 4
Plus!
43 PAGES OF PROJECTS & TUTORIALS

Contents

Issue 88 December 2019

Cover Feature

24

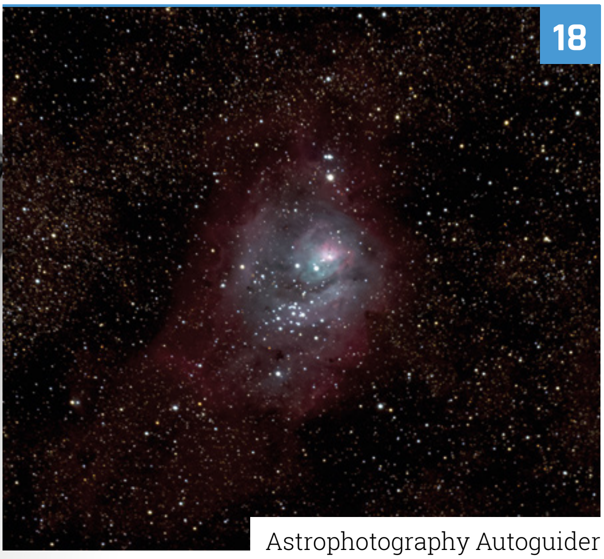
Regulars

Project Showcases

10

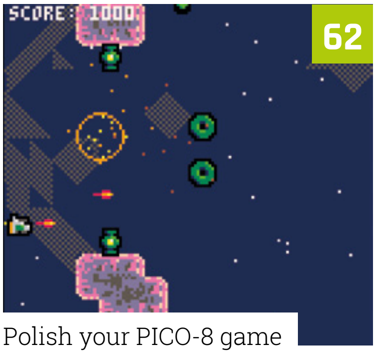
ZX Spectrum Next

18


Astrophotography Autoguider

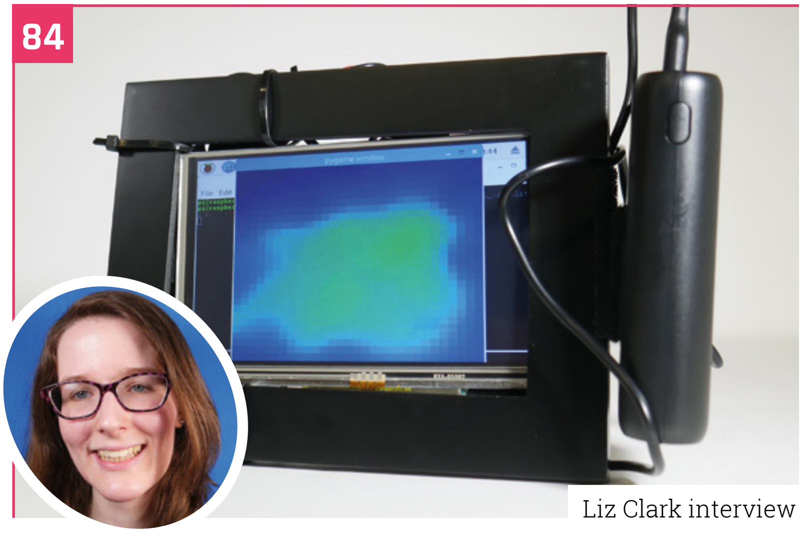
DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and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

Smart Christmas lights



Polish your PICO-8 game

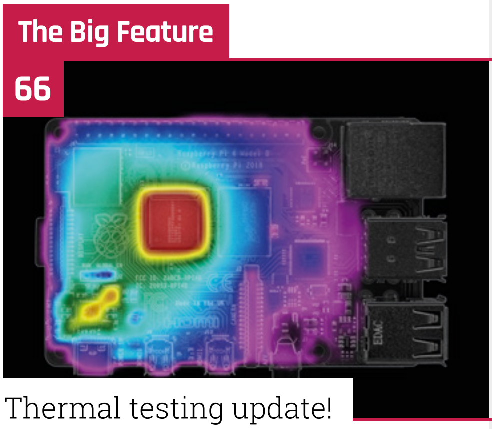
GPIO Xmas Tree



Liz Clark interview

Tutorials

The Big Feature



Thermal testing update!

Reviews

Community

WIN PIARM ROBOT ARM KIT!

Gender Balance in Computing programme opens

Raspberry Pi Foundation rolls out scheme to study why young women don't choose to study computing. By **Rosie Hattersley**

The Raspberry Pi Foundation has announced a UK-wide rollout of the Gender Balance in Computing (GBIC) programme. The GBIC programme launched in April to understand why many young women don't choose to study computing-related subjects. "We are working with schools to understand which particular initiatives to address gender balance are most effective," says Sue Sentance, Chief Learning Officer at the Raspberry Pi Foundation.

"We are drawing on existing research which points to a variety of influencing factors, including girls feeling like they don't belong in the subject or its community, a lack of sustained encouragement, and a lack of role models in computing when making career choices," she explains.

The series of rigorously evaluated pilot interventions involving 15,000 pupils and 550 schools will be the largest national research effort to tackle gender balance in computing to date.

You can read more about the Gender Balance in Computing programme on the Raspberry Pi blog (**magpi.cc/QbBrQ6**). Schools can register for the scheme at **magpi.cc/gbic**.

▼ Many young women don't choose to study computing-related subjects

▲ The Department for Education has funded the Raspberry Pi' Foundation's Gender Balance in Computing (GBIC) research programme

3 ISSUES FOR £5

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Issue 86 | October 2019 | magpi.cc
10 RETRO GAMING KITS
The best arcade controllers, kits and consoles
POWER RASPBERRY PI PROJECTS
25 rewarding masterful
SPOOKIFY YOUR HALLOWEEN COSTUME
BUILD AN INTERACTIVE ROLE-PLAYING TABLE
MAKE 8-BIT RETRO MUSIC
Issue 87 | November 2019 | magpi.cc
RASPBERRY PI 4
DIGITAL MEDIA
High definition & p
MAKE A CLUSTER COMPUTER
Learn to code a supercomputer
BUILD A ROBOT WITH SENSORS
DESIGN RETRO GAME LEVELS
HACK A MARBLE RUN GAME
40 PAGES OF PROJECTS
WIN! RASPBERRY PI ROBOT ARM KITS
The MagPi
The official Raspberry Pi magazine
Issue 88 | December 2019 | magpi.cc
RETRO COMPUTING
WITH RASPBERRY PI 4
Upcycle a classic computer
Play retro video games
Rediscover classic coding
ZX Spectrum Next
British icon reborn
Thermal Testing
Raspberry Pi firmware turns down the heat
Plus!
Build the best Christmas projects
Assemble a low-cost camera bot
Capture the stars with astrophotography
43 PAGES OF PROJECTS & TUTORIALS

☎ Subscribe by phone: **01293 312193**

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

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

BLACK
FRIDAY
www.OKdo.com

WWW.CANAKIT.COM
CanaKit
CanaKit™ Raspberry Pi 3 Model B+ Retro Gaming Kit - 32 GB EVO+ Edition
Grove Starter Kit for IoT based on Raspberry Pi
for Microsoft Windows 10 IoT Core
This kit consists of selective Grove modules designed to simplify innovation on Raspberry Pi 3
Bare Conductive
Electric Paint Lamp Kit
Draw, paint and create with electronics. Everything you need to transform paper into light with Electric Paint.
THE THINGS NETWORK
okdo™
DESIGN THE WORLD

ZX Spectrum Next

An iconic 8-bit computer is set to enjoy a retro revival with the help of a Raspberry Pi Zero, as **David Crookes** explains

MAKER

Jim Bagley

Jim has been programming games professionally for the past 31 years and he's a key member of the team dedicated to bringing back the ZX Spectrum.

specnext.com

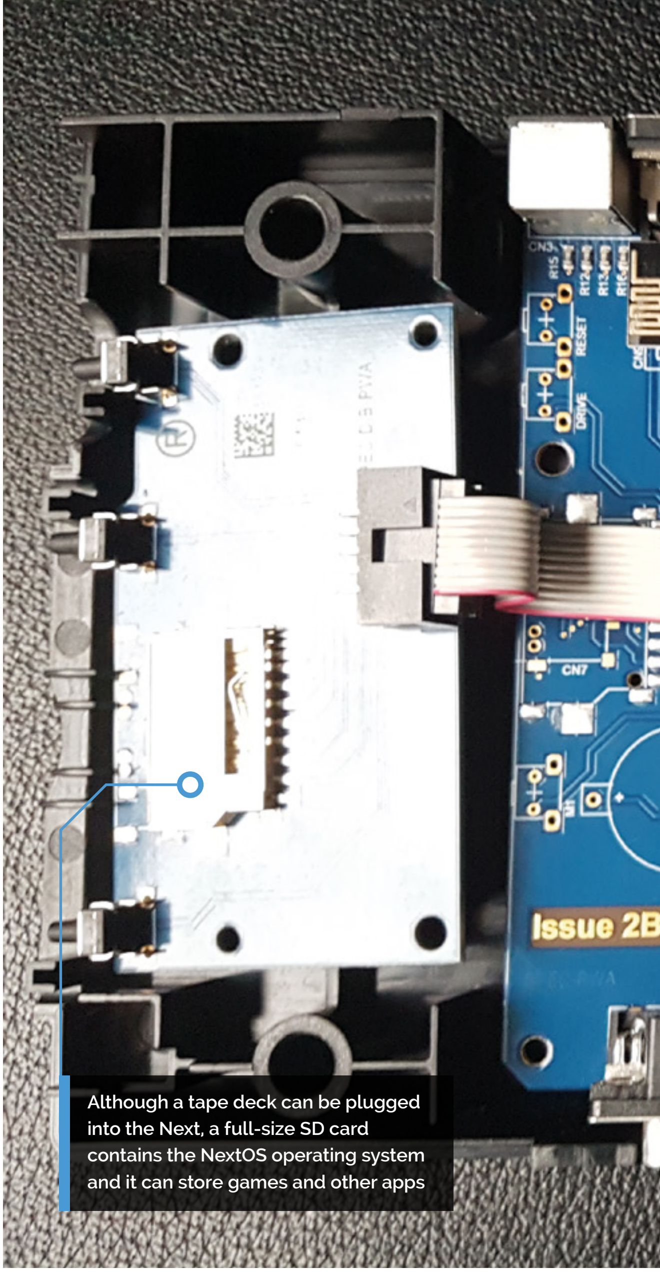
In the 1980s, the ZX Spectrum range of 8-bit computers quickly became legendary. It boasted no more than 128kB of RAM (16kB on the original model), a Zilog Z80 CPU running at just 3.5MHz, and a palette of up to 16 colours, two of which were black. Yet the humble 'Speccy' encouraged a generation of bedroom programmers and underpinned a flourishing British video games industry.

Now it's returning in a new, enhanced form called the ZX Spectrum Next, and Raspberry Pi Zero has proven invaluable in its development. "For me, the goal has been to encourage a new generation of bedroom coders," says legendary games developer Jim Bagley, who is among a group of Spectrum fans behind the project. With 3113 Kickstarter backers handing over £723,390 in cash, many others potentially share that dream.

Work has certainly progressed well. "When the Next was first announced three years ago, it was intended to be a normal Spectrum with an SD card and a Raspberry Pi Zero as an extender," Jim explains. "Raspberry Pi was going to be used to add extra features such as hardware sprites and hardware scrolling so that newcomers would find it easier to code the computer and get the wow factor of getting something running instantly on the screen."

During the Kickstarter campaign, however, a large field-programmable gate array (FPGA) was announced for the Next – a configurable integrated circuit which allowed the hardware sprites, scrolling, and other advanced features to be incorporated within the machine itself. "It freed up Raspberry Pi Zero to do something else," Jim says. So the developers began to play around.

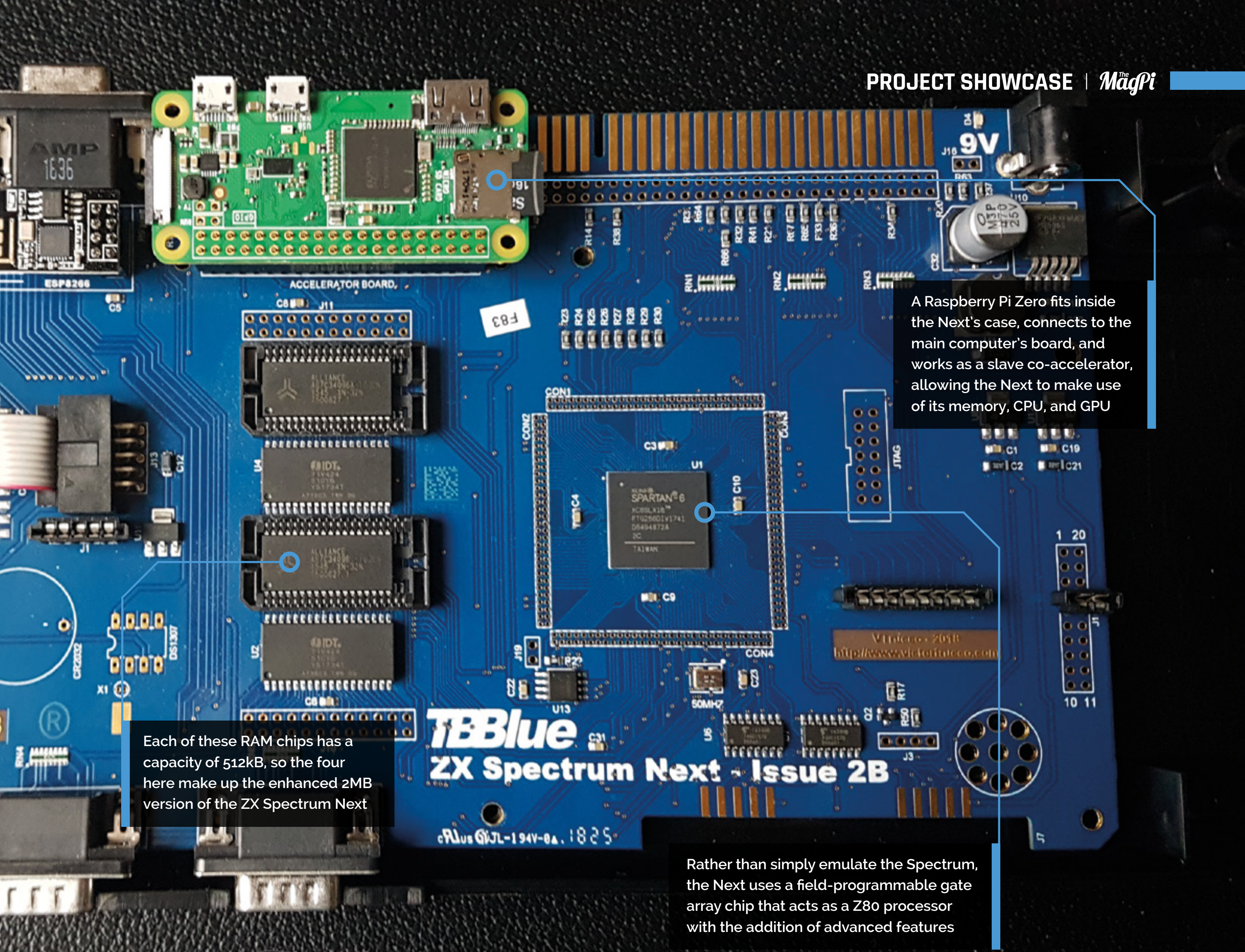
▲ The ZX Spectrum Next contains a Z80 processor on an FPGA, 1MB of RAM expandable to 2MB, hardware sprites, 256 colours, RGB/VGA/HDMI video output, and three AY-3-8912 audio chips



Although a tape deck can be plugged into the Next, a full-size SD card contains the NextOS operating system and it can store games and other apps

Load and run

One of the initial ideas was to recreate the feel of loading a game from tape – generating the series of scratchy, beepy, high-pitched noises which could be heard as a program was ingested into



A Raspberry Pi Zero fits inside the Next's case, connects to the main computer's board, and works as a slave co-accelerator, allowing the Next to make use of its memory, CPU, and GPU

Each of these RAM chips has a capacity of 512kB, so the four here make up the enhanced 2MB version of the ZX Spectrum Next

Rather than simply emulate the Spectrum, the Next uses a field-programmable gate array chip that acts as a Z80 processor with the addition of advanced features

sinclair
ZX Spectrum Next

As well as playing classic games, the ZX Spectrum Next packs advanced features

> The goal has been to encourage a new generation of bedroom coders

It will work with CRT and VGA monitors, as well as more modern screens, thanks to the support of a HDMI output

Games made specially for the Next are saved as .NEX files which load directly to the computer via a SD card. Since the games are 768kB in size (or 1792kB for the 2MB models), it would take too long to route them via Raspberry Pi Zero

the computer. On the original hardware, this was accompanied by a loading image slowly building on the screen and animated stripy borders.

Rather than have Next users connect a cassette deck, the developers had a cunning plan. "We thought it would be a good idea to have commands sent back and forth between the Next and Raspberry Pi Zero," Jim says.

This developed into a tool allowing a TZX file (a format that stores an exact copy of a ZX Spectrum tape) to be uploaded from an SD card to Raspberry Pi Zero. By sending it back down to the new-gen Speccy through an audio-in pin, the Next would be fooled into thinking it's loading a cassette. "It gives the original feeling of loading from tape," Jim affirms.

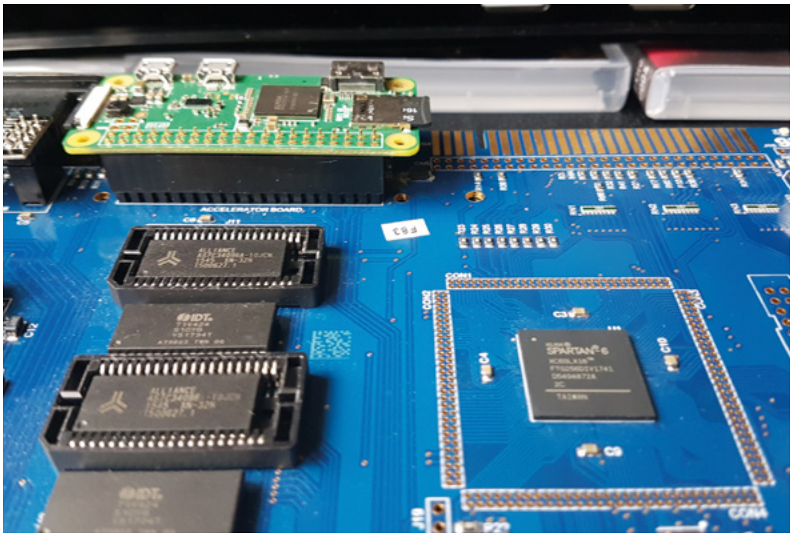
> Having a Raspberry Pi Zero accompanying the Next has also enabled other audio delights

Are you there, SID?

Having a Raspberry Pi Zero accompanying the Next has also enabled other audio delights. A Sound Interface Device (SID) emulator has been developed that allows audio created for Commodore's SID programmable sound generator chip to be played on the Next.

It will also be possible to enjoy Atari ST audio files and tunes created using music trackers on the Commodore Amiga. "We can send the audio through a GPIO pin and it goes straight to the FPGA, where it's mixed with the audio of the Next," Jim explains.

To achieve all of this, the developers have used DietPi to create a new OS called NextPi. "It



A Raspberry Pi Zero was chosen as an accelerator board for the ZX Spectrum Next because of its power and low cost

remains a fully functioning Raspberry Pi Zero that is running at the same time as the Next, but we wanted the Next to be more in control."

The most recent extra use for the Next's Raspberry Pi Zero is the ability to connect the latter to its own display. "It's possible to send screens to Raspberry Pi Zero so that you can have a dual screen," Jim says. "This means you could have a game being played on the Next, with a global map or stats shown via Raspberry Pi Zero."

The team is now looking to get USB controllers to work via Raspberry Pi Zero, allowing them to be read by the Next. "We'd also like Raspberry Pi Zero to help with 3D maths so you can take vertices, have them rotated, and passed back," Jim says.

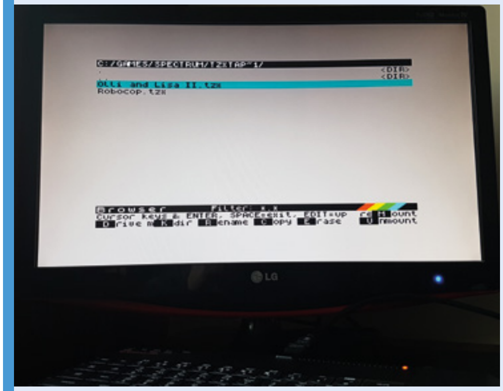
In the meantime, the team is readying the new computer for a January release and a new Kickstarter is being planned for those who didn't pledge the first time around. We're certainly looking forward to seeing where it – and Raspberry Pi – goes next.

sinclair
ZX Spectrum Next
Drive
SD
M1

The Next's keyboard/case design is based on the ZX Spectrum 128

Raspberry Pi loading

01 Games in the file format TZX – a tape format used for preservation purposes – need to be saved on to an SD card and inserted into the ZX Spectrum Next.



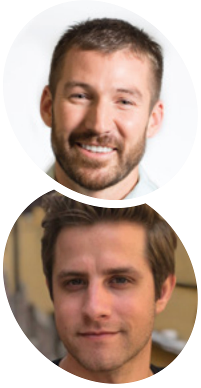
```
C:/GAMES/SPECTRUM/TZXTAP~1/
.                                                  <DIR>
..                                                 <DIR>
Olli and Lisa II.tzx
Robocop.tzx

Browser              Filter: *.*
Cursor keys & ENTER, SPACE=exit, EDIT=up     reMount
Drive mKdir Rename Copy Erase                Umount
```

02 A game can then be selected via the Next computer's built-in Browser mode. Raspberry Pi Zero will be instructed to load the game's data from the SD card.

03 Raspberry Pi Zero sends the data back to the ZX Spectrum Next as audio and this generates the once-familiar loading noise and loading screen ahead of the game running.

The Swirl Machine

Make fantastic art with wine using The Swirl Machine. **Rob Zwetsloot** picks a nice merlot and proceeds to 'swirl his style'



MAKER

Rob Gaedtke, Jonathan Rutheiser

KPS3 is an integrated marketing agency that works with companies across the globe to find insights in data and bring them to life.

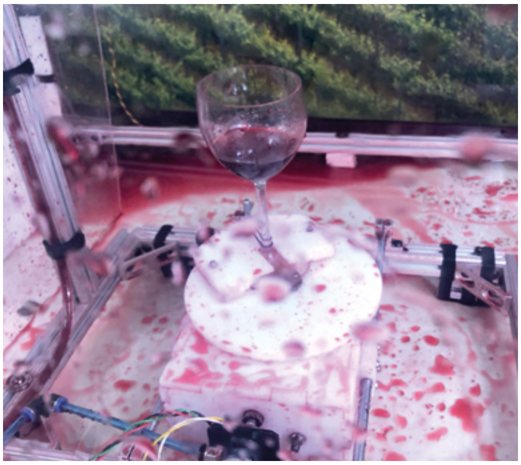
magpi.cc/9yrkJW

Warning! Spinning glass

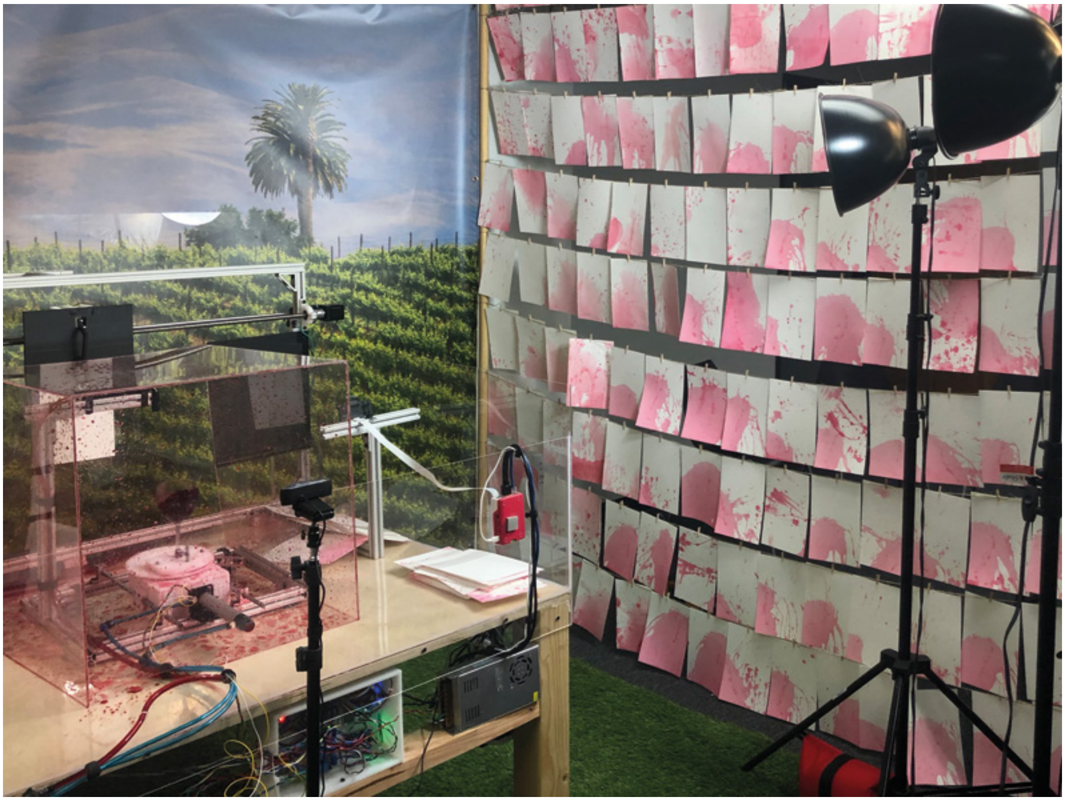
If you choose to recreate this project, please fix glass on a spinning turntable carefully (and at your own risk).

Art comes in many forms, and for this project it takes the form of an interactive wine art creator called The Swirl Machine. It's a lot more complicated than it might sound, though.

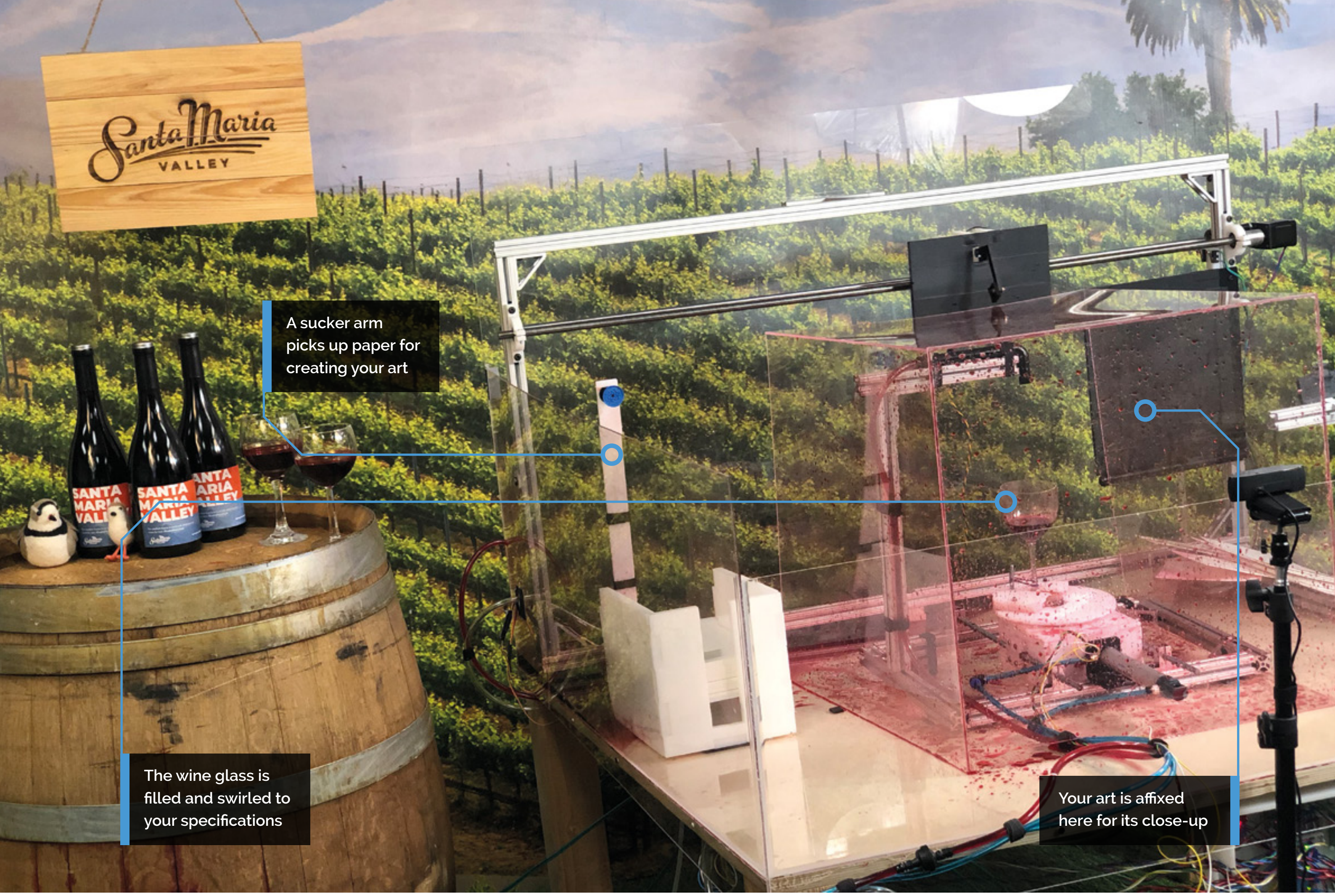
"The Swirl Machine is an interactive, digital-meets-the-real-world machine that swirls Santa Maria Valley wine and turns it into a digital piece of art," say the team from KPS3, the marketing agency behind The Swirl Machine. "It was created and developed by KPS3 for Visit Santa Maria Valley. The machine allows users to select their 'Fill Level' and 'Swirl Speed'. In real-time, the user can watch the glass fill, swirl, and splash the wine onto a piece of paper, creating an original Santa Maria-style spill artwork. Every swirl and spill will be unique to each individual user."



A lot of art creates a lot of mess



Thousands of pieces of art have been created – is your piece on display?



A sucker arm picks up paper for creating your art

The wine glass is filled and swirled to your specifications

Your art is affixed here for its close-up

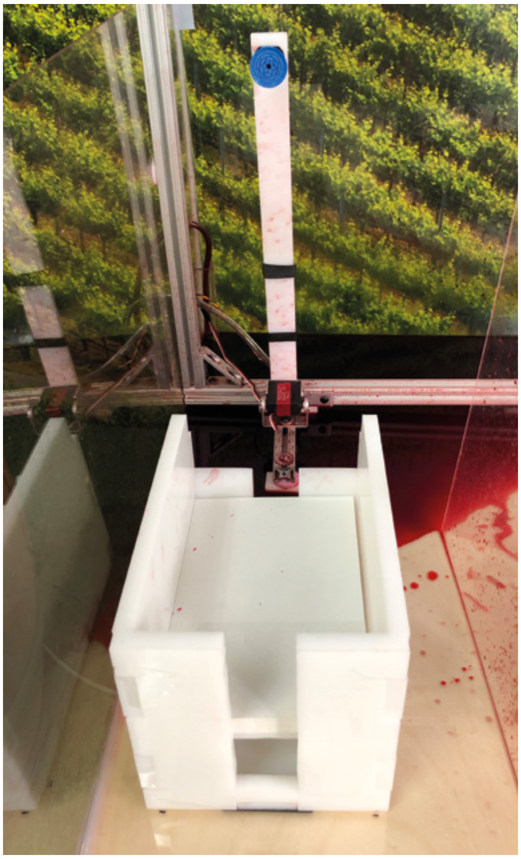
It's a bit like a Rube Goldberg device, albeit without the *Powerhouse* music playing. The result of your swirl is sent to you as a photo, and there's even a fun personality result that comes with it.

"The idea came on a car ride to Santa Maria Valley," the KPS3 team explain. "The team was playing around with the idea of capturing slow-motion swirls, which led to how interesting wine spills are, and that led to The Swirl Machine. The fact that it was a spill fit perfectly into the vibe of Santa Maria Valley's wine experience… if you spill a little, no one really cares. And because of who KPS3 is as a company, it clearly had to push technical boundaries."

> " In real-time, the user can watch the glass fill, swirl, and splash the wine onto a piece of paper "

Making a splash

At the time of writing, there have been around 2000 'swirls' on The Swirl Machine's website (**magpi.cc/9yrkJW**), and the machine has been made possible thanks to a Raspberry Pi.

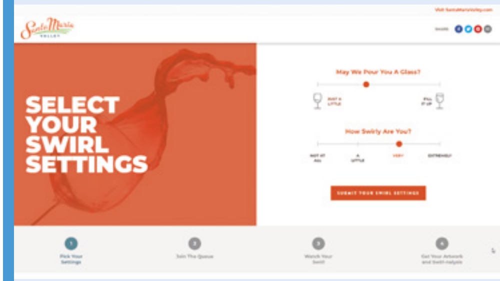


We love the arm with an air compressor that sucks up paper

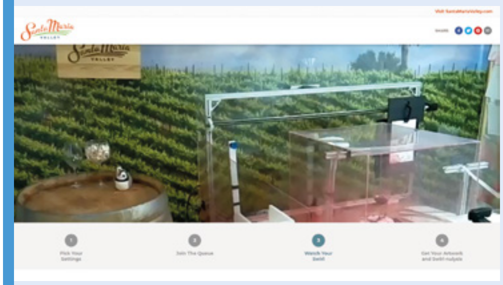
Quick FACTS

- The video of a user's swirl can be replayed
- The Swirl-nalysis takes into account the final art and your settings
- The queue lets you watch the swirls ahead of you
- More than 1600 people have created about 2000 swirls
- Art is very cool, however you make it

Swirl your style



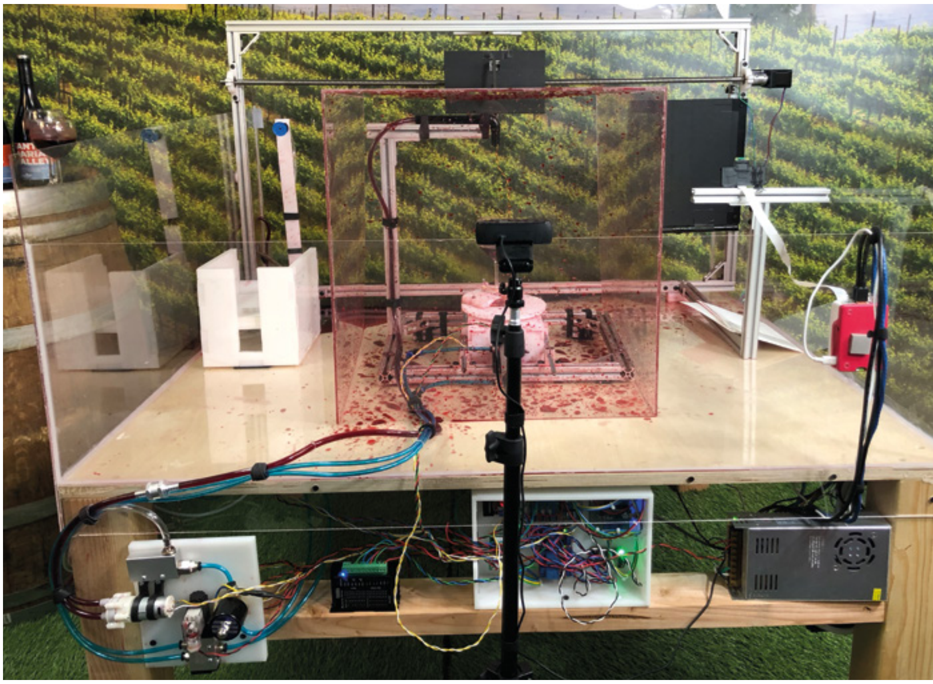
01 First, you need to select how full the wine glass is. This tells the pump how much wine to dispense into the glass. You can also choose the amount of 'swirl' for your glass, which sets a target speed for the spinning wine plate.



02 A gripper arm with an air-compressor-powered suction cup grabs your piece of special paper. It brings it to the paper holder, where the paper is then moved into position. Wine is dispensed, the glass is spun, and art happens.

SWIRL
YOUR
STYLE

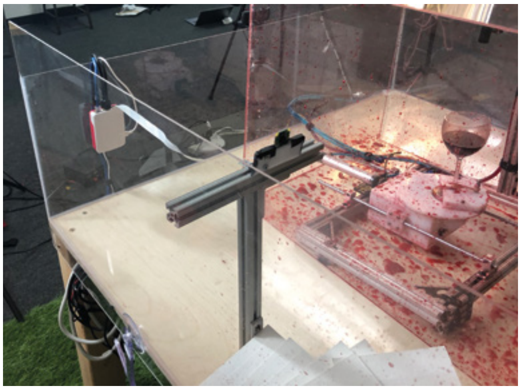
03 A camera records the footage of the swirl, and a different camera takes a picture of the final piece. As the wet paper isn't perfectly straight, the image is processed to make it look perfect before being shared with the creator.



▲ The complete machine is a masterpiece of messy art and many electronics

"We like Raspberry Pi because it has a very low barrier to entry," the KPS team say. "For only $35 you get a fully functioning computer with I/O capabilities. The community behind Raspberry Pi is also active and helpful, which means you get software packages that are thoroughly tested and you never spend too much time figuring out solutions to problems. We've used Raspberry Pi on many different projects in the past [...]. Whether it's powering a media server, emulator, live-streaming client, or The Swirl Machine, we know it's more than powerful enough to handle the tasks we've thrown at it."

The machine uses a number of technologies. The website is hosted in AWS. A robot arm grabs paper and moves it along the swirl production line. Arduinos control the wine glass, and a spinning plate swirls the glass. Each piece of wine art is unique, including the corner splash that we managed to make during our go with it.



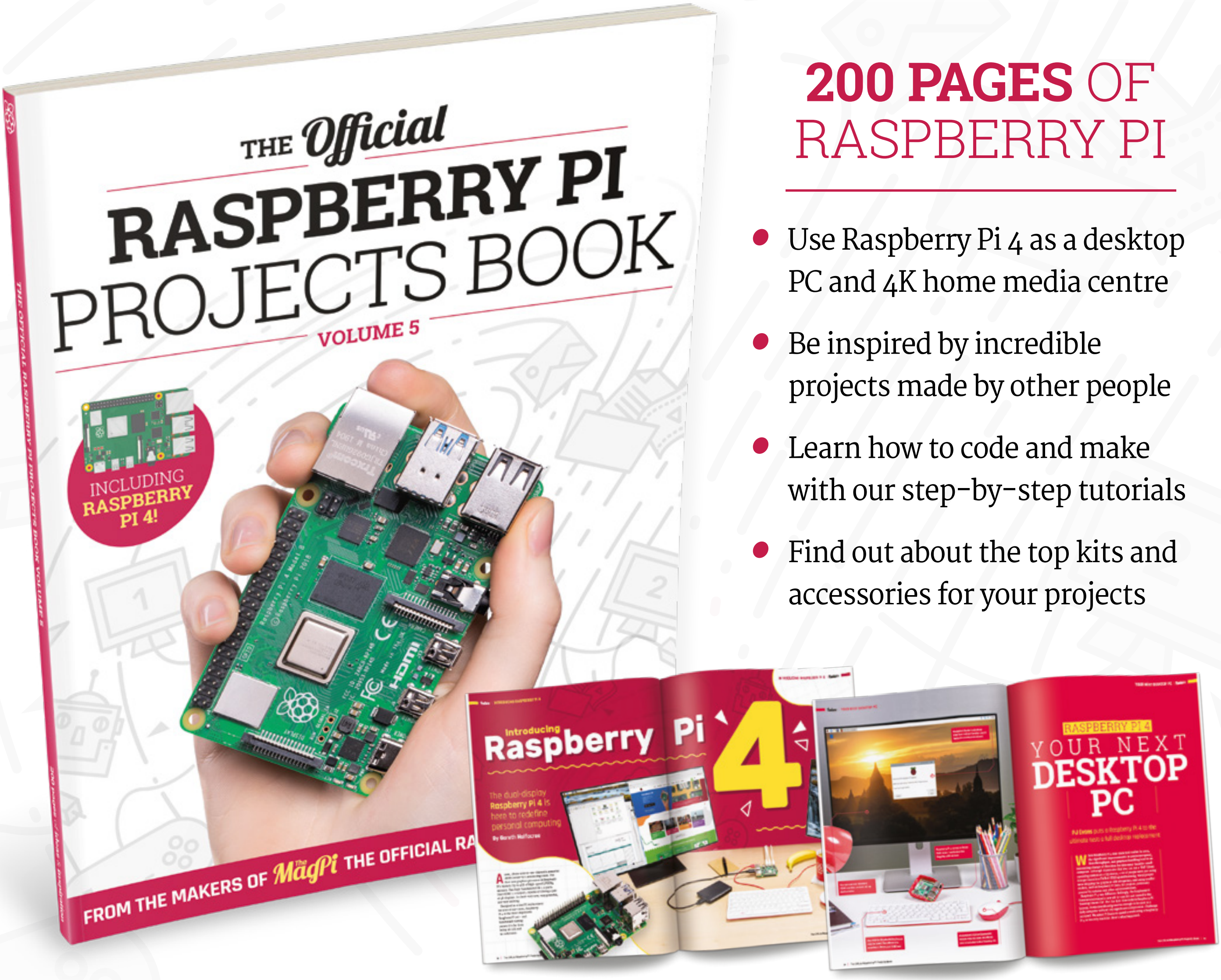
▶ A Raspberry Pi takes photos of the final piece

THE *Official*

RASPBERRY PI

PROJECTS BOOK

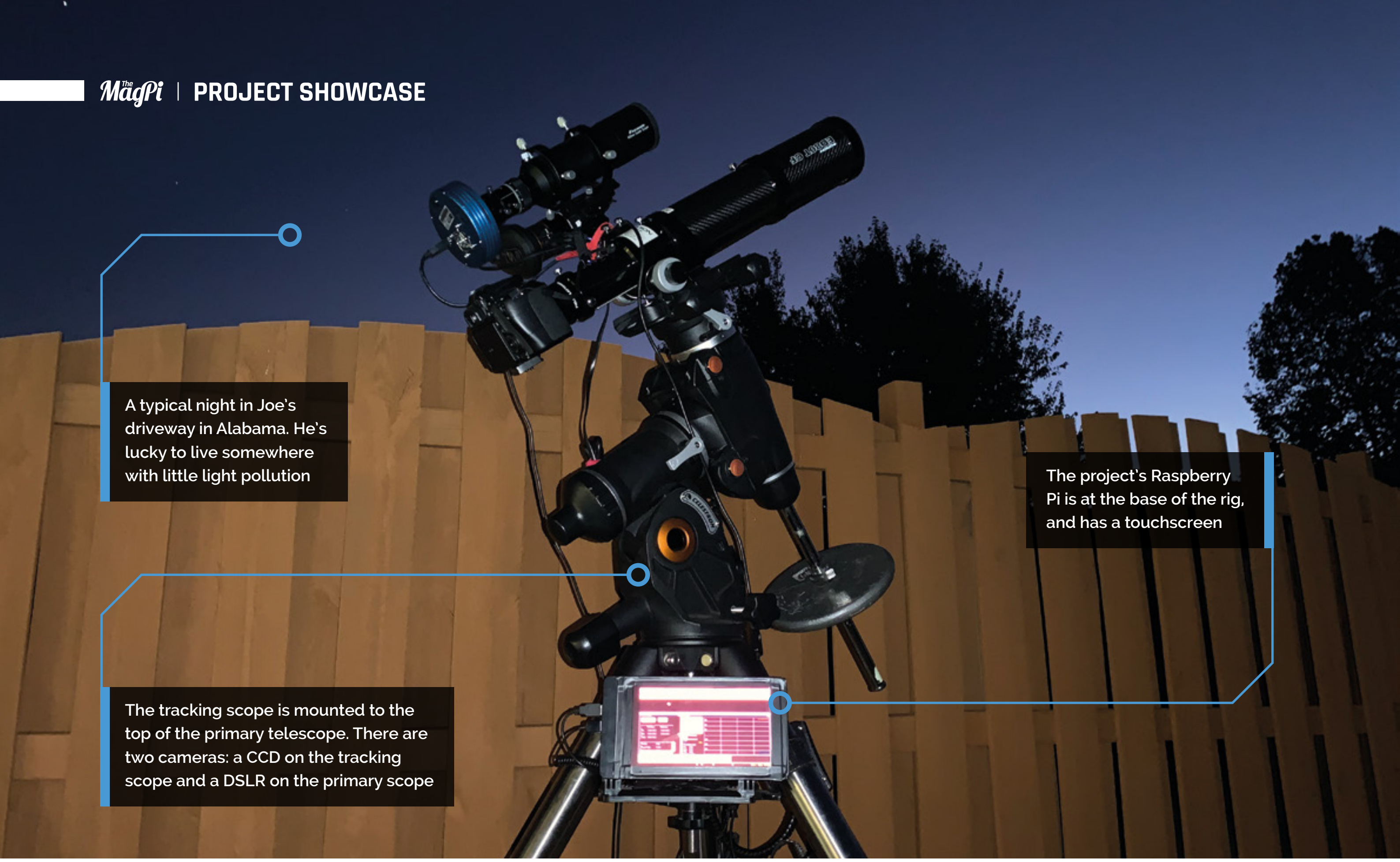
VOLUME 5


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 5
INCLUDING RASPBERRY PI 4!
FROM THE MAKERS OF MagPi THE OFFICIAL RA
Introducing Raspberry Pi 4
RASPBERRY PI 4
YOUR NEXT DESKTOP PC

200 PAGES OF
RASPBERRY PI

- Use Raspberry Pi 4 as a desktop PC and 4K home media centre
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store



A typical night in Joe's driveway in Alabama. He's lucky to live somewhere with little light pollution

The project's Raspberry Pi is at the base of the rig, and has a touchscreen

The tracking scope is mounted to the top of the primary telescope. There are two cameras: a CCD on the tracking scope and a DSLR on the primary scope

MAKER

Joe Kutner

Joe Kutner (aka Codefinger) is a software architect at **Salesforce.com**, where he works primarily with Java and other open-source technologies. He's published several books about programming with Ruby and Java. He enjoys amateur astronomy, mostly observationally, but he also dabbles in astrophotography.

@codefinger

Astrophotography Autoguider

The awe-inspiring wonders of the night sky encouraged a keen stargazer to find a better way to capture their beauty. **Rosie Hattersley** gets inspired

Creating stunning photographs of the night sky requires planning, patience, and reliable star-tracking equipment. A desire to travel a little lighter led keen amateur astronomer Joe Kutner to embark on his first Raspberry Pi project.

Joe says there's nothing worse than taking hours of astrophotography images only to find out your telescope was drifting, causing the stars to look more like lines than points. To protect against this kind of misalignment, he needed an autoguider: a computer and camera that track a star in the telescope's field of view to ensure that it stays in the same position throughout the session.

"The main goal of my project was to get rid of the laptop," Joe tells us. "I needed to control my telescope in the field. I spend enough time in front of a computer at work, and the laptop took the fun out of observing. It served an important function, though: controlling both my camera and my mount. Without it I would only be able to take very short exposures of the moon and planets."

Joe considered using an iPad or a Microsoft Surface instead, but both were far too expensive. He wanted to keep the build cost below $100, and neither worked well with his chosen software.

Instead Joe picked up a Raspberry Pi, a case, and a touchscreen for less than $100, and added a red plastic cover so he was still able to use the setup in night-vision mode. These work alongside the various bits of astronomy kit Joe uses regularly on his stargazing missions.

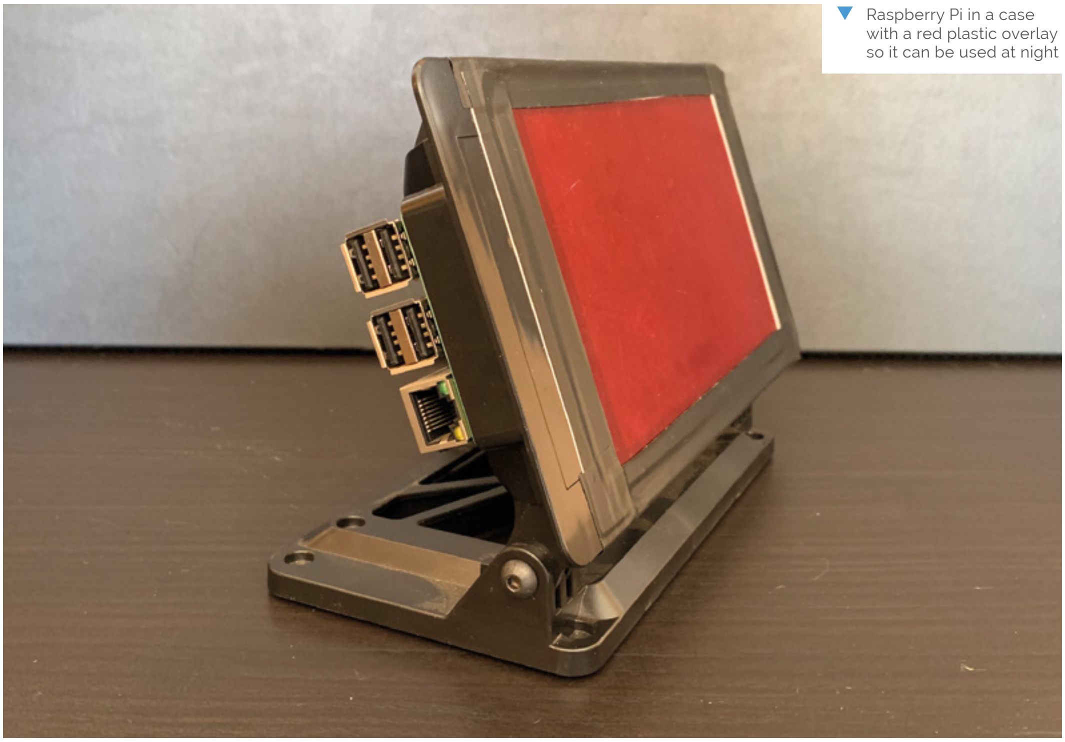
Under open skies

Joe made extensive use of general purpose open-source software such as Raspbian Stretch and Git,

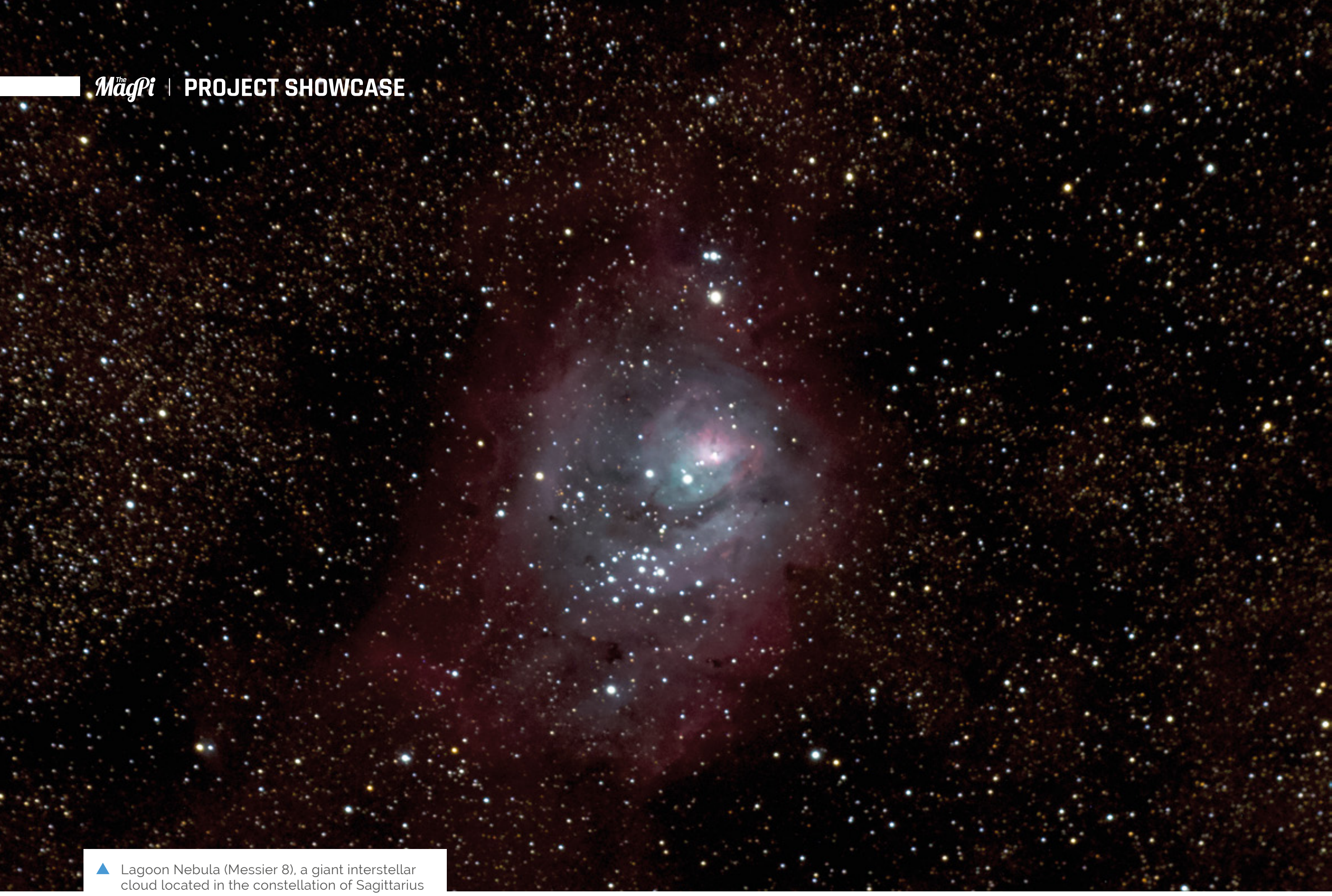
The Horsehead Nebula and Flame Nebula, photographed from Joe's Alabama driveway with the help of his Raspberry Pi-controlled autoguider

Quick FACTS

- Raspberry Pi records over several hours while Joe sleeps in his tent
- Joe recommends Lacerta MGEN II (**magpi.cc/3XGdSW**) if you don't want to build your own
- Many of his astronomy photos are taken from Huntsville, Alabama
- He's using Raspberry Pi to attempt to image an exoplanet transit
- He also blogs about coders' fitness and nutrition at **healthyprog.com**



Raspberry Pi in a case with a red plastic overlay so it can be used at night



Lagoon Nebula (Messier 8), a giant interstellar cloud located in the constellation of Sagittarius

Whirlpool Galaxy (Messier 51) taken at the Texas Star Party 2019 in Fort Davis

plus astronomy-specific open-source tools Libnova (**magpi.cc/libnova**), INDI (**indilib.org**), and PHD2 (**openphdguiding.org**) telescope guiding software.

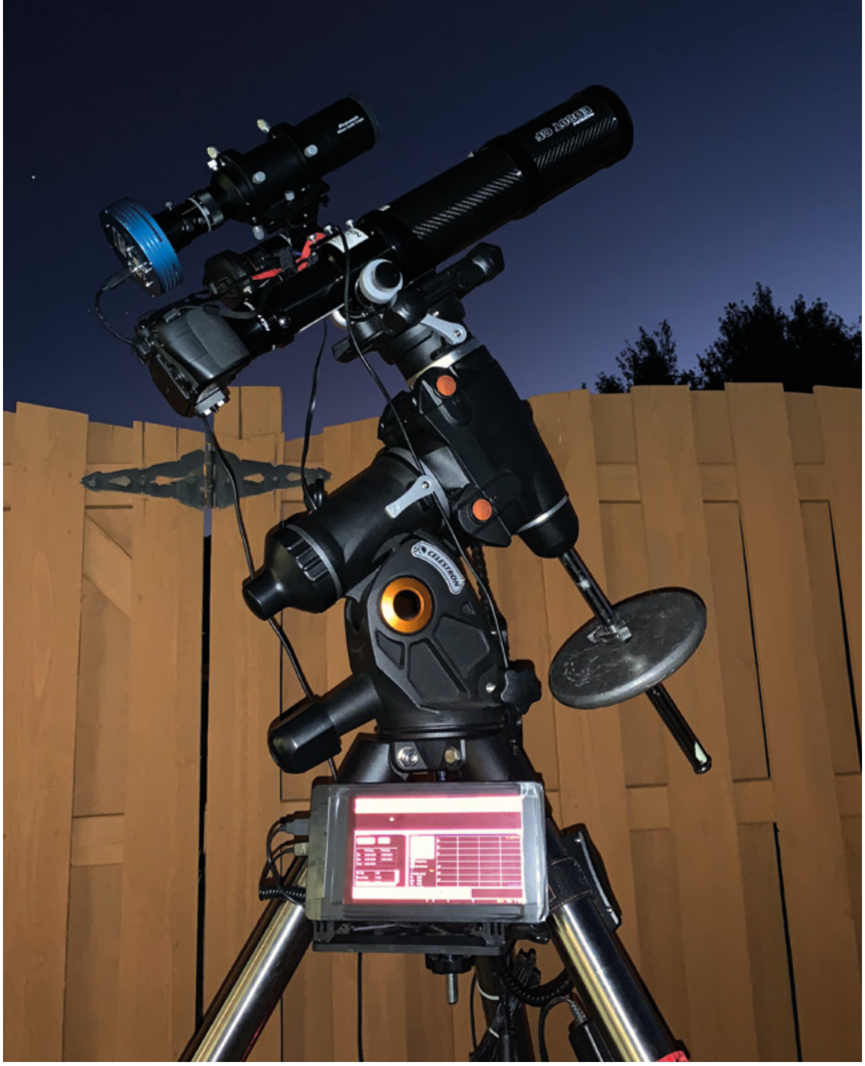
He wrote scripts to automate the software so he could just use the touchscreen, without a mouse or keyboard. But for the most part, things worked without customisation.

> " I can roll it onto my driveway and start imaging in just a few minutes "

"Every step in the process had its challenges," Joe recalls. "I would install one piece of software and then find out it wasn't compatible with some version of another piece of software I needed. When I finally got everything running, it wouldn't talk to my telescope until I installed yet another version of the software. There were dozens of these little paper-cuts, but in the end it was worth working through them."

Joe also says the hardware he chose worked perfectly from day one. Any tweaks he made were "mostly minor issues like figuring out how to install the correct version of a particular camera driver".

His Raspberry Pi now has an on-board autoguiding system for his astrophotography rig.

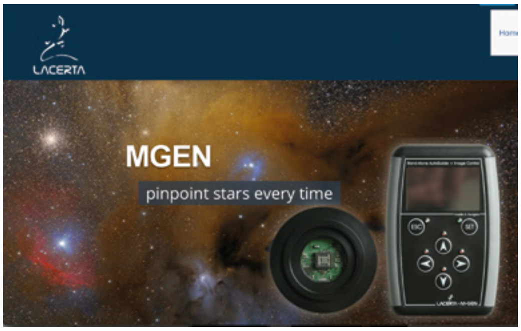


The lightweight nature of Joe's astrophotography setup makes it easy to move around

Because Raspberry Pi attaches to the base of the mount, it's easily accessible. Unlike using a laptop, there's no need for an extra table or complicated wiring. Joe says the setup is perfect for his needs: "I can roll my telescope onto my driveway and start imaging in just a few minutes."

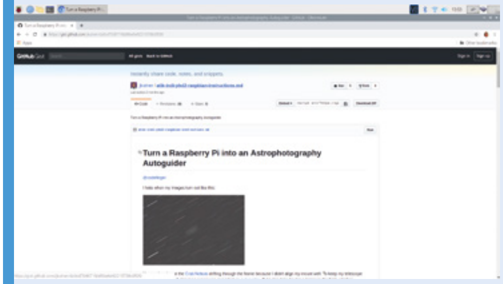
International expansion

Now that Joe has successfully built a fairly portable astrophotography rig, he sees its potential for explorations further afield. He's keen to try out his autoguider with other types of astrophotography kit such as the ultra-compact Sky-Watcher Star Adventurer series of mounts. "When combined with my Raspberry Pi," he says, "I could take the whole rig on an airplane as carry-on. That would give me access to some very dark skies."



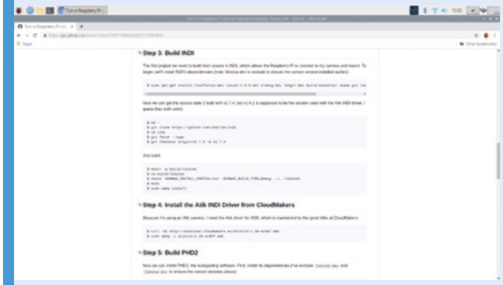
Lacerta MGEN II is a good alternative if you don't want to build your own autoguider

Build an autoguider



Turn a Raspberry Pi into an Astrophotography Autoguider

01 Start with a fresh installation of Raspbian and download the package for libnova 0.14. You can find the install instructions at Joe's GitHub page (**magpi.cc/Xn4JCE**).



Step 3: Build INDI

Step 4: Install the Atik INDI Driver from CloudMakers

Step 5: Build PHD2

02 Use the GitHub instructions to build INDI, the software to connect Raspberry Pi to your digital camera and mount. Install the Atik camera driver if needed.

03 Install and build PHD2 autoguiding software, then start the INDI server so it looks for the camera and mount. Save the profile for future reference.

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

WIN! RASPBERRY PI ROBOT ARM KITS

The MagPi

The official Raspberry Pi magazine

Issue 88 | December 2019 | magpi.cc

RETRO COMPUTING WITH RASPBERRY PI 4

Upcycle a classic computer | Play retro video games | Rediscover classic coding

ZX Spectrum Next British icon reborn

Thermal Testing Raspberry Pi firmware turns down the heat

Plus!
- Build the best Christmas projects
- Assemble a low-cost camera bot
- Capture the stars with astrophotography

43 PAGES OF PROJECTS & TUTORIALS

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 & Rest of World)
£80 (EU)

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

Subscribe by phone: 01293 312193

Subscribe online: magpi.cc/subscribe

Email: magpi@subscriptionhelpline.co.uk

JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

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

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

Offer subject to change or withdrawal at any time

WORTH £20

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
RASPBERRY PI 4K
DIGITAL MEDIA HUB
High definition video streaming & playback made easy
MAKE A CLUSTER COMPUTER
TOP 10 AI PROJECTS
HACK YOUR KITCHEN TOOLS
BUILD A ROBOT WITH SENSORS
DESIGN RETRO GAME LEVELS
HACK A MARBLE RUN GAME
40 PAGES OF PROJECTS & TUTORIALS
POWERFUL RASPBERRY PI PROJECTS

Buy now: magpi.cc/subscribe

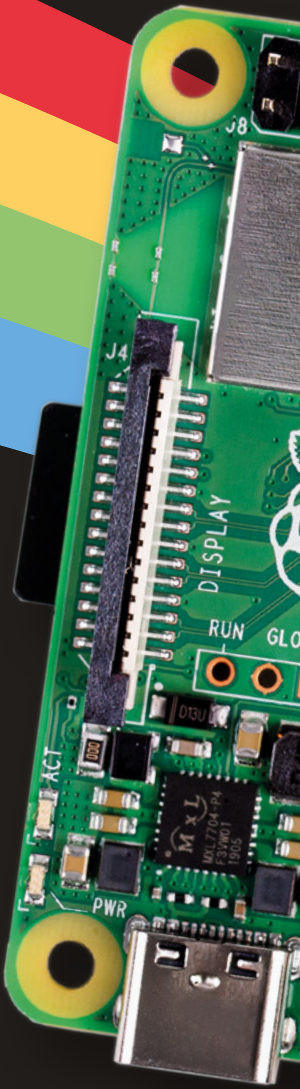
SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

RETRO COMPUTING with RASPBERRY PI 4



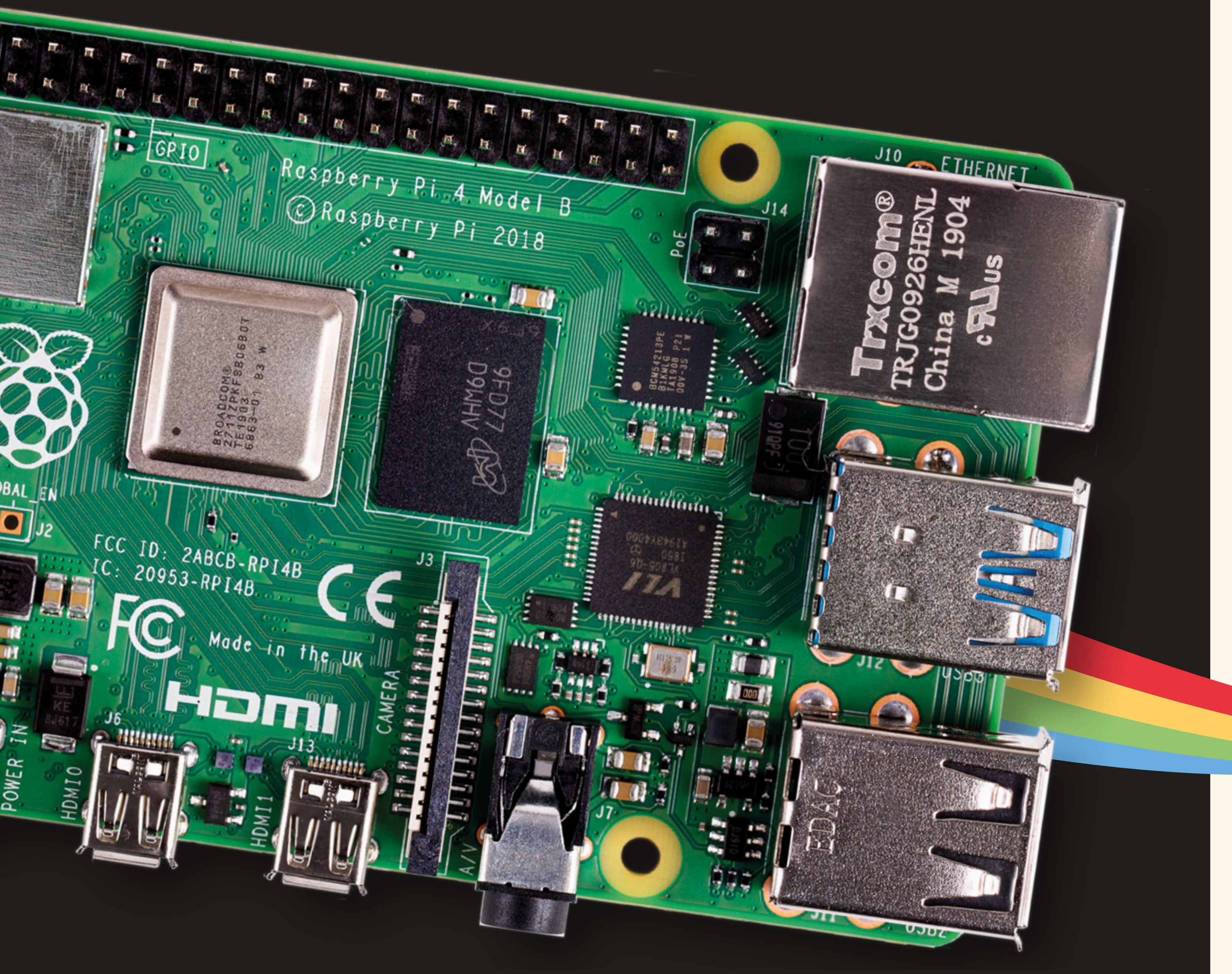
Want to rediscover the golden years of computing and enjoy a bit of classic gaming? Go old-school with **PJ Evans**

Modern computers like Raspberry Pi 4 are amazing, but sometimes we yearn for an earlier time with 8-bit processors, and either BASIC language or machine code. A time when computers were stripped back, with simpler architecture that required deep understanding.

Quite possibly it is this, rather than straightforward nostalgia, that fuels the healthy retro computing scene.

Retro computing is also a playground for makers, providing endless inspiration for projects and supported by a wealth of open-source software. Over the next few pages we'll fire your imagination by looking at building your own retro computer, whether it's revitalising old tech or coming up with a new design. We'll also try to answer that age-old question: what to do next?

In this feature, we're going to look at upcycling vintage computers, building retro games consoles, playing retro games, and classic programming. Let's go back in time.



26 Upcycle a vintage computer

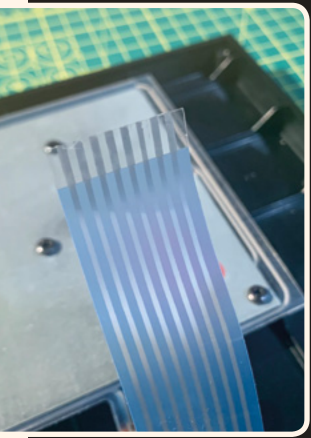
28 **Build a DIY arcade console**

30 Great gaming essentials

32 **Get hold of games**

UPCYCLE A *VINTAGE COMPUTER*

Don’t bin an old computer – give it a new lease of life with Raspberry Pi



The delicate ribbon connectors form part of the matrix system that detects key presses

Sadly, not all tech is built to last and it’s quite easy to pick up an expired 1980s home computer. You may not be able to save the circuitry, but if the case and keyboard are good, you might just have a great project for a Raspberry Pi computer. Add some emulation software and you’ve got a modern take on a classic.

01 Find a willing donor

Whichever old computer you elect to upcycle, the biggest task will be wiring in the keyboard so it can be used by Raspberry Pi. You may find that the keyboard has more inputs than Raspberry Pi’s GPIO can handle, so you’ll need a microcontroller such as an Arduino Leonardo (**magpi.cc/jbuEh9**) to act as a USB interface. Please do not destroy a working computer. We’ve chosen a mid-eighties ZX Spectrum+.

02 Map the keyboard

Requiring a GPIO pin for every key would be unmanageable. Luckily, most vintage home computers use a simple row-and-column matrix system to reduce the number of pins used. Electrical pulses are sent down each column in sequence and if a key is being pressed, the column and one of the rows will form a circuit. Now we can calculate which key has been pressed. Using a bit of Python, we can replicate this and feed the results into Raspberry Pi as if it were the keyboard. Most computers, including our ZX Spectrum (**magpi.cc/MwGP33**), have their matrices detailed online.

03 Make a keyboard adapter

Most keyboards use two ribbon cables, one for the row and the other for the columns. Select a spare GPIO pin for each and build a simple adapter to securely connect the keyboard to Raspberry Pi’s GPIO interface. For a ZX Spectrum, this requires the sourcing of two Molex connectors (relatively easy to source online – see **magpi.cc/molex**) and some pieces of stripboard to solder in the connectors, and some wiring to connect to the GPIO. The column side (KB2) will need diodes on each line to avoid short-circuiting.

04 Prepare your computer

If you haven’t done so already, now is the time to get your operating system set up. We’re using Raspbian Lite as we can run our emulator of choice using SDL (Simple DirectMedia Layer, **libsdl.org**), which is a fancy way of saying we don’t need the weight of a full desktop install. On the command line, make sure everything is up to date using `sudo apt -y update && sudo apt -y upgrade`, then install the dependencies:

```
sudo apt install python3-pip
pip install pynput
```

05 Add some Python

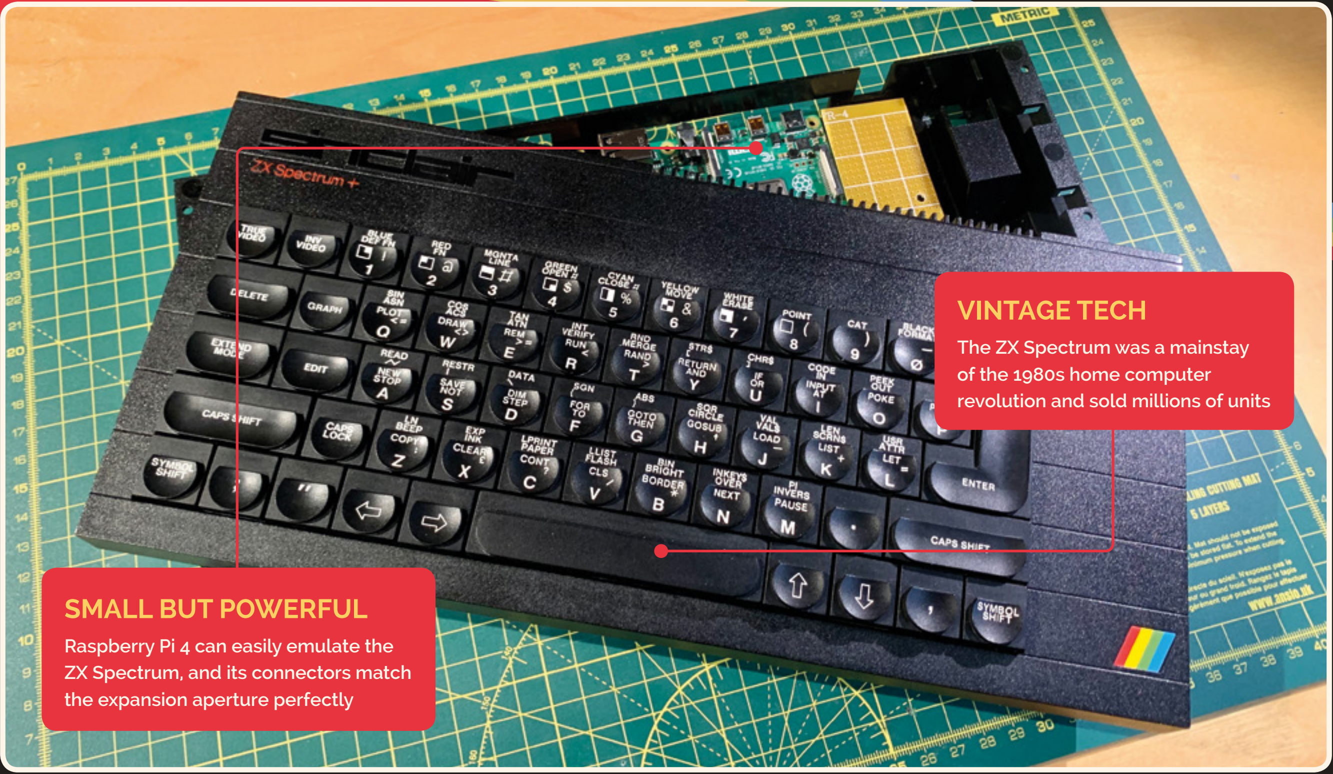
Now the keyboard is connected to the GPIO, it’s time to test it out and make sure we can get input into Raspbian. Download the Python script from **magpi.cc/YwTFgr** by issuing the following commands:

```
git clone https://github.com/mrpjevans/
zxscanner2.git
cd zxscanner2
```

Now run a simple test by running this command:

```
sudo python3 test_keyboard.py
```

Press a few letters on the keyboard. Do you see results? If not, check all your wiring. If you’re happy, edit the file **keyboard_scanner.py** following the instructions to get the correct ‘map’ for your chosen computer’s keyboard.



VINTAGE TECH

The ZX Spectrum was a mainstay of the 1980s home computer revolution and sold millions of units

SMALL BUT POWERFUL

Raspberry Pi 4 can easily emulate the ZX Spectrum, and its connectors match the expansion aperture perfectly

06 Background

To make best use of the keyboard, we need to have the keyboard scanner run at startup and sit in the background. To do this, create a system service:

```
sudo nano /usr/lib/systemd/zxscanner2.service
```

Add the following text:

```
[Unit]
Description=ZX Scanner
After=multi-user.target

[Service]
Type=idle
ExecStart=/usr/bin/python3 /home/pi/zxscanner2/keyboard_scanner.py

[Install]
WantedBy=multi-user.target
```

Now enable the service so that it always runs on boot:

```
sudo systemctl enable /usr/lib/systemd/zxscanner2.service
sudo systemctl start zxscanner2.service
sudo systemctl daemon-reload
```

07 Install an emulator

Once happy that your keyboard is working as expected, get your emulator running. For the ZX Spectrum there are a few options, including the easy-to-install Lakka. And a standalone favourite is FUSE – installed directly from the command line:

```
sudo apt install unclutter fuse-emulator-sdl spectrum-roms
```

If you're using Raspbian Lite, however, you may need to build a version of FUSE that doesn't need a graphical user interface. See the website for full instructions (**fuse-emulator.sourceforge.net**). You set FUSE to run on boot for an authentic experience.

08 Make it your own

Just because it's a ZX Spectrum, doesn't mean we're restricted to just one emulator. Annoy pedants by running VICE64, a Commodore 64 emulator, or even arcade emulators like MAME. Why not add a Bluetooth controller to play joystick games? For a more authentic experience, tap into the composite video connector on Raspberry Pi 4B to give slightly fuzzier output – modern displays are just too precise to be authentic for the ZX Spectrum. Whatever you decide to do, have fun with your newly revived vintage computer.

BUILD A DIY *ARCADE CONSOLE*

How one young maker built a retro games console

Jamie Harris

Jamie is 14, from Milton Keynes, and a STEM enthusiast.

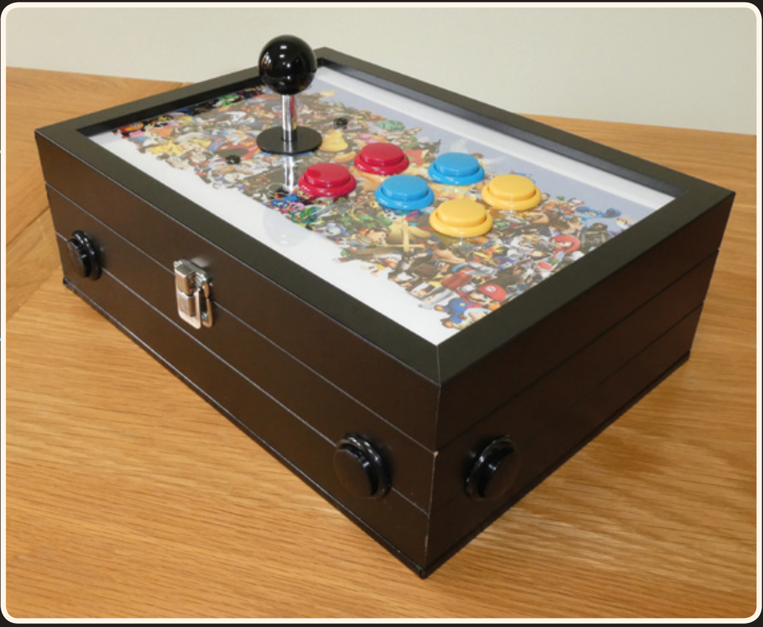
Some amazing projects come out of after-school clubs. Some even win awards. When Jamie, a young maker, brought this gorgeous retro gaming console to the Milton Keynes Raspberry Jam, we had to know more. So we asked him a few questions…

How did you come to build this project?

Last school year, when I was attending our school's science club, we were asked to go for the bronze STEM (science, technology, engineering, and mathematics) award. I was considering several project options but, in the end, I decided to build an object using Raspberry Pi and apply most of the disciplines of STEM.

Why did you choose retro gaming?

When I searched the internet for Raspberry Pi projects, retro games consoles seemed to be the most popular. I discovered RetroPie! I watched a few people on YouTube running RetroPie and it looked like good fun. I wanted the project to appeal to adults (who remember those games) and to kids so they can see what gaming was like before Xbox and PlayStation.



▼ Some clever thinking and attention to detail makes for a beautiful retro gaming console

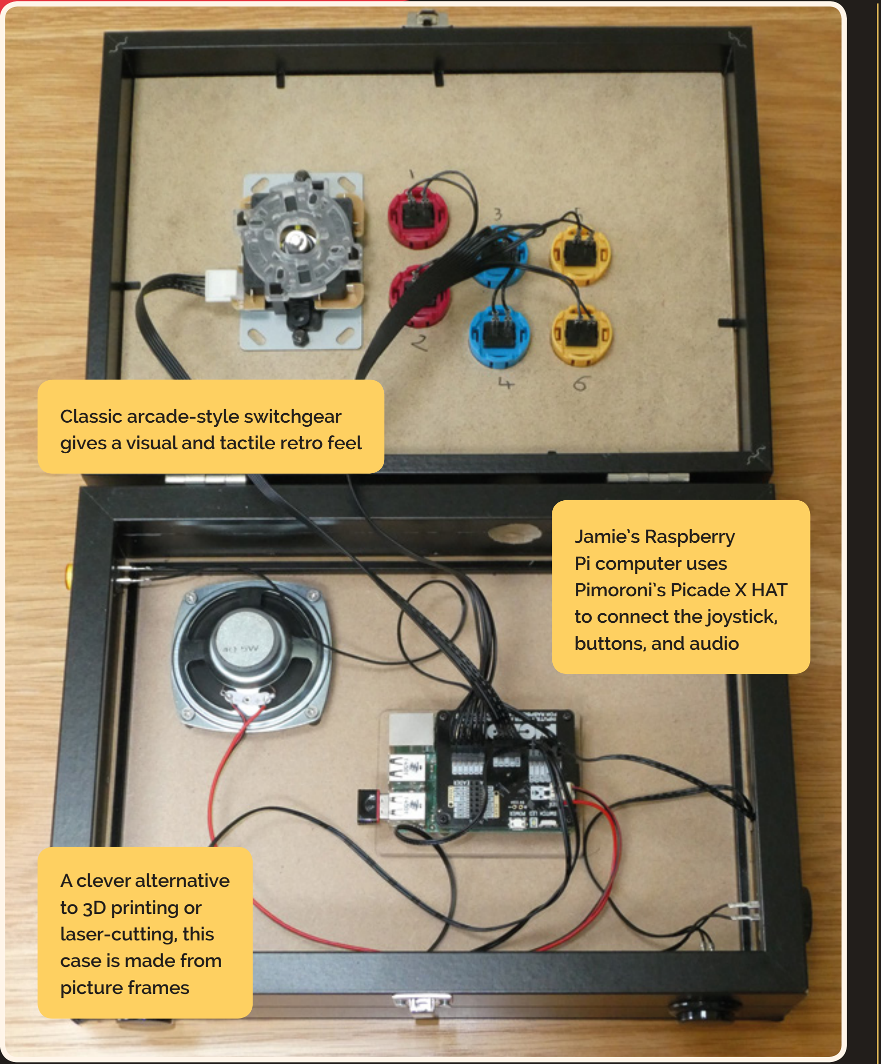
▲ The case may be chunky, but it's a great example of building a unit with arcade joysticks and buttons

What are you most happy about?

I built my own games console. It was an achievement; I was well chuffed! The external design of the console is unique to me. I have assembled several advanced Lego / Meccano models over the years, but this was my first build using Raspberry Pi. I was also very pleased about the reaction and the attention it received from my classmates and friends.

What was particularly challenging about the project?

The biggest challenge in this project was cutting the holes for the control buttons on the top panel and the side of the case. We had to do the top panel twice as, at the first attempt, the acrylic cracked. The second biggest challenge, to my surprise (and



Classic arcade-style switchgear gives a visual and tactile retro feel

Jamie's Raspberry Pi computer uses Pimoroni's Picade X HAT to connect the joystick, buttons, and audio

A clever alternative to 3D printing or laser-cutting, this case is made from picture frames

Emulate retro games with Lakka

We look at the hardware here, but what about software? We've chosen Lakka, an extension of the RetroArch emulation software – which provides a single point of configuration for multiple emulation systems, rather than having to deal with multiple diverse applications. Not only does Lakka run well on Raspberry Pi 4, but it's available for Windows and Mac too. Check out our Lakka tutorial on page 52 for how to set it up.

lakka.tv

> "THE BIGGEST CHALLENGE WAS CUTTING THE HOLES FOR THE CONTROL BUTTONS"

others), was finding the correct driver software for the Picade X HAT. Assigning the joystick and buttons was also tricky.

Do you have a next make planned?

My next idea is to produce a Raspberry Pi cluster made using Raspberry Pi 4 with a repurposed digital photo frame for a screen. I am also looking into upgrading my retro games console with a Raspberry Pi 4 to see what difference it will make.

READ CLASSIC *MAGAZINES*

Vintage computer magazines often carried code alongside news and reviews of the latest technology. You can read many classic magazines on **archive.org** and other websites. Here are a few of our favourites:

Your Spectrum (and Your Sinclair)

There's some great code to play around with in Your Spectrum magazine and the early issues of Your Sinclair.

magpi.cc/yoursinclair

Ahoy!

Focused mainly on Commodore computers, Ahoy! had a range of high-quality tutorials for Commodore 64 and Amiga machines.

magpi.cc/ahoy

Acorn User

This magazine launched alongside the BBC Micro and offered a wealth of classic computing articles and type-in programs.

magpi.cc/acornuser

TEN GREAT GAMING ESSENTIALS

For the proper experience you need the proper equipment. Here's some of our favourite retro computing kit

Cool Retro Cases

£8 / $8 | **magpi.cc/ga1VQp**

If you would prefer something a bit more plug-and-play when giving your Raspberry Pi retro gaming machine an authentic look, you can skip upcycling an old machine case and get a dedicated case off-the-shelf. Many replicate classic machines from the eighties, even wiring up the buttons to allow you to shut down cleanly, and other features. If you can't find one you like and you have access to a 3D printer, there are countless designs available on sites such as Thingiverse, from retro replicas to mini arcade cabinets and some amazing hand-held units. Many come with detailed instructions for displays and switches.

8BitDo joysticks *From £35* | magpi.cc/8bitdo

Controllers are another facet of retro where the old and modern collide. You want to use a genuine joystick from the era of course, but many are fragile or, well, a bit rubbish by today's standards. Also, why not take advantage of Raspberry Pi Bluetooth and go wireless? 8BitDo's range of retro controllers is an elegant solution to this dilemma: all the advantages of modern engineering, available with Bluetooth, in a range of classic stylings to match your case. You can even use them with your PC, as more recent features like rumble are supported.

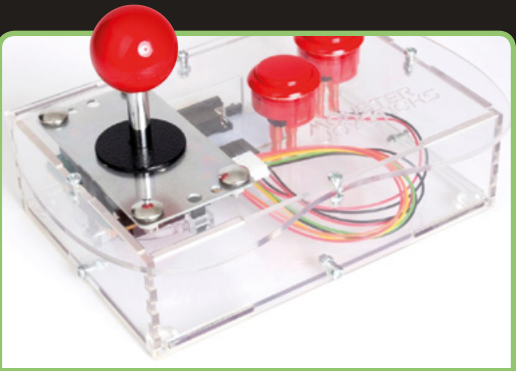
RGBPi *€30 (£30)* | rgb-pi.com

There's a lot of debate over display choices for retro gaming. Many prefer modern technology and razor-sharp pixels. The fact is that a lot of 1980s and 1990s games were designed to be viewed on a CRT display, with 'fuzziness' taken into account. So, to see the games as the designer intended without having to compromise too much, add an RGBPi to your system. This add-on creates an old-school SCART RGB signal from Raspberry Pi's GPIO, making it compatible with most CRT colour TVs – and there's plenty of the latter on auction sites.

Picade

From £150 | **magpi.cc/picade**

For the ultimate retro experience, this bar-top cabinet is hard to beat. Picade was one of the original flagship Raspberry Pi projects, and this revised unit is everything you could want for your own personal arcade. A suite of professional-grade switches and joystick, LCD monitor, Picade X HAT, and a cool customisable cabinet, all in one kit. If you don't need a keyboard (and you could always use a wireless one), Picade is a great choice.



Clear Deluxe Arcade Controller Kit

£100 | **monsterjoysticks.com**

This arcade-stick-style kit allows you to build an arcade machine right into the stick, and the clear version enables you to see Raspberry Pi at work inside. It comes with high-quality Sanwa arcade parts and a Monster Joysticks GPIO Interface to hook them up to Raspberry Pi. There are a few different design options, in case you don't want the clear version, and a slightly cheaper plywood variation.

Wireless USB Gamepad

£14 | **magpi.cc/wirelessgamepad**

If you've ever attended a big Raspberry Pi robotics event such as Pi Wars, then chances are you'll already be familiar with this little device being used to drive things about. Pi Hut's budget controller plays well above its price league, featuring a D-pad, shoulder buttons, and two analogue controls in a sturdy package. As it uses an RF USB connection rather than Bluetooth, setup is a matter of plug-and-play. It's well supported by all the popular gaming environments, such as RetroPie and Lakka.

Joysticks and buttons

From £2 | **magpi.cc/Jx6AKs**

The popularity of retro gaming has led to a vibrant market in components, and that means lower prices. If you're planning on a console or cabinet build, you're going to want sturdy and accurate input devices. Kits and components are widely available, many coming from original moulds and tooling used to build arcade equipment but with improved design and more durable electronics. A full set is surprisingly affordable and will also do for restoring any original equipment you've managed to acquire. Regardless, for any DIY project, you're going to want that original button-mashing sensation.

LACK Table *£6* | **magpi.cc/UAtraY**

It's one of IKEA's most famous pieces of furniture. The LACK table is cheap, cheerful, and available in a range of colours. Best of all, it's hollow and that means we can put stuff in it. It wasn't long after the original Raspberry Pi devices appeared that various projects started to centre around this ubiquitous table. This particular project, from Matt (aka Raspberry Pi Spy), is a beauty, featuring two-player support and an attractive bezel. LACK is perfect if you're considering a 'cocktail'-style build. Read more at **magpi.cc/9RG6SD**.

JAMMA board *€50 (£50)* | **rgb-pi.com**

If you can find a genuine cabinet from the golden era of arcades, you're a very lucky person indeed. Don't despair if it's not working. If screen and controls are intact, why not add Raspberry Pi! Most 1980s arcade machines used a standard interface, JAMMA, to connect the logic board to inputs and outputs. The Raspberry Pi JAMMA board adapter enables anyone to bring an old cabinet back to life, hooking up the display and controls to the GPIO. Add RetroPie and you're running thousands of games on original equipment.

FreePlay Zero Kit

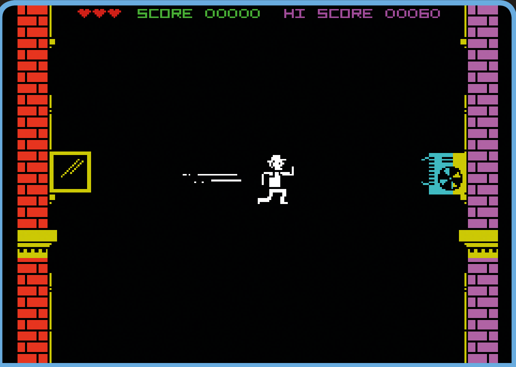
$70 (£55) | **magpi.cc/freeplayzero**

There's a selection of portable console kits based on Raspberry Pi Zero, but the FreePlay Zero is a real looker. A custom PCB makes assembly easy and provides a D-pad, four buttons, shoulder buttons, and audio. You need to provide a few things and source a case, but the result looks impressive. Alternatives include the Adafruit PiGRRRL Zero, a very compact unit available as a kit that just needs a Raspberry Pi Zero and 3D-printed case.

GET HOLD *OF GAMES*

You've got the kit, but what to play? Welcome to the world of homebrew

If you think retro gaming is just about old games, then we've got some great news. Thanks to the 'homebrew' scene, new games for old systems are appearing all the time. Original games rub shoulders with 'demakes', modern games on old systems. We've picked a few of our favourites and provided links for finding out more.

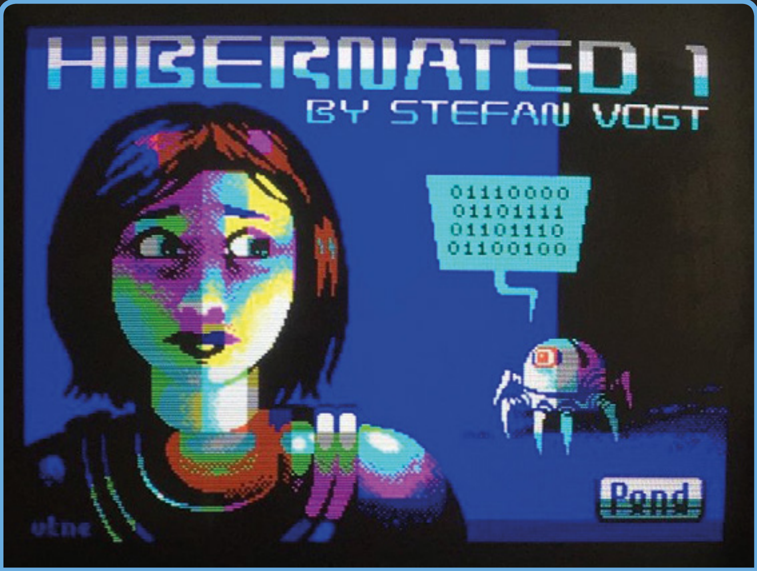


Nohzdyve

Free | **magpi.cc/nohzdyve**

PLATFORM: **ZX Spectrum**

Charlie Brooker's *Black Mirror* episode 'Bandersnatch' made headlines with its innovative 'choose your own adventure' format. It's riddled with references to the home computer scene of the 1980s,'Tuckersoft' being based on Liverpool powerhouse Imagine. ZX Spectrum enthusiast Matt Westcott was commissioned to bring one of the featured games to life.

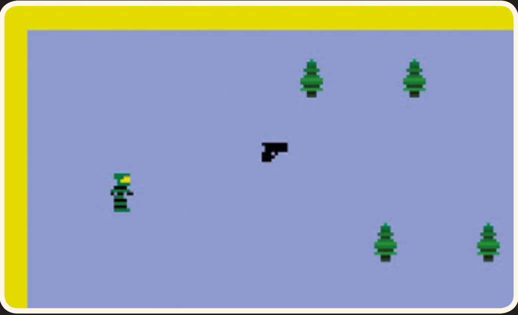


Hibernated One

Free / Name your price | **magpi.cc/hibernated1**

PLATFORM: **ZX Spectrum / Commodore 64 / Amstrad CPC**

In Stefan Vogt's adventure you play Olivia, awakened from hibernation when an alien spacecraft traps her ship, Polaris-7, in a tractor beam. With no communication from the other craft and surrounded by death and decay, can she escape? This is a text adventure in the classic style and the opening chapter in an ongoing series.

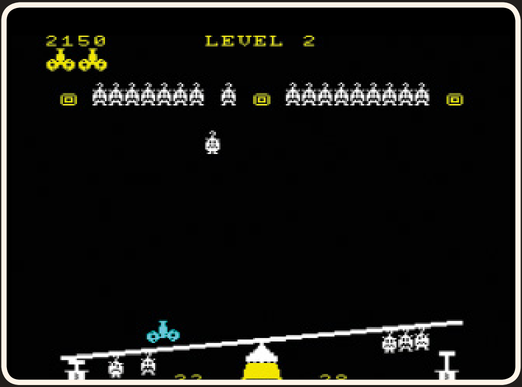


Halo 2600

Free | **magpi.cc/halo2600**

PLATFORM: **Atari 2600**

Yes, you read that correctly and yes we are talking about Microsoft's legendary Halo franchise. Remarkably, unlike many demakes that can infringe copyright, this version was written by Ed Fries, leader of the original Xbox project and has been given Microsoft's blessing. Some cartridges were manufactured by AtariAge, although you can download the game for free.

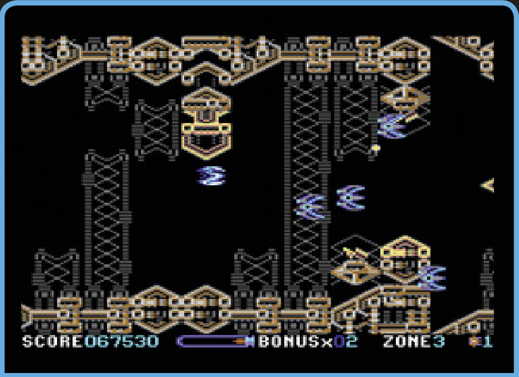


Teeter Torture

Free | **magpi.cc/teetertorture**

PLATFORM: **MAME**

This game is an original from 1982 which has now been released free of charge for non-commercial use by Exidy. It has mysterious origins and only one cabinet is known to exist, which luckily still works! You control a cannon on a trolley that balances on a barrel of TNT. Shoot the aliens or they'll topple you over, triggering the detonator.



Relentless 64

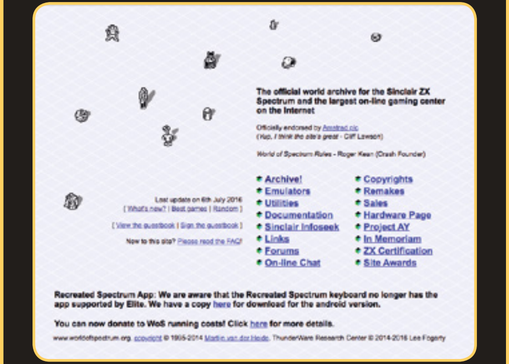
Free | **magpi.cc/relentless64**

PLATFORM: **Commodore 64**

Relentless 64 is a homebrew of a homebrew. Originally, the Amstrad CPC version was the winner of a 16kB cartridge competition in 2013 and was so well received, it was commercially released on cassette. The game is a classic shoot-'em-up and lives up to its name as there's no pause and no bosses: it just gets harder and harder.

HOMEBREW *DESTINATIONS*

Need more games for your console? These sites are full of homebrew and legal downloads



The official world archive for the Sinclair ZX Spectrum and the largest on-line gaming center on the Internet

Officially endorsed by Amstrad plc
(Yup, *I think the site's great* - Cliff Lawson)

World of Spectrum Rules - Roger Kean (Crash Founder)

Last update on 6th July 2016
[What's new? | Best games | Random]

[View the guestbook | Sign the guestbook]

New to this site? Please read the FAQ!

- Archive!
- Emulators
- Utilities
- Documentation
- Sinclair Infoseek
- Links
- Forums
- On-line Chat
- Copyrights
- Remakes
- Sales
- Hardware Page
- Project AY
- In Memoriam
- ZX Certification
- Site Awards

Recreated Spectrum App: We are aware that the Recreated Spectrum keyboard no longer has the app supported by Elite. We have a copy here for download for the android version.

You can now donate to WoS running costs! Click here for more details.

www.worldofspectrum.org. copyright © 1995-2014 Martijn van der Heide. ThunderWare Research Center © 2014-2016 Lee Fogarty

World of Spectrum

The admins of this site have been thorough in getting clearance to host the many thousands of games available.

worldofspectrum.org

MAME Official Site

A selection of legal ROM downloads of classic arcade machines.

mamedev.org/roms

TFW8B releases Crazy Blaster for the C64

The Future Was 8 Bit (TFW8B) is pleased to announce the physical cartridge release of Crazy Blaster, the latest Commodore 64 title developed by Mika 'Misfit' Keranen. Added to the TFW8B 9.99 range of cartridges, Crazy Blaster...

Vintage Is The New Old

A massive collection of homebrew software for many different platforms.

vintageisthenewold.com

Homebrew Legends

A community-focused site that is essential for keeping up with the latest releases.

homebrewlegends.com

Warning!

It is illegal to download copyrighted ROMs from the internet. *The MagPi* does not endorse video game piracy and strongly recommends you stick to emulators that do not use any protected software, such as BIOS files, and stick to game downloads that are offered with the consent of the rights holder.

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
RETRO GAMING WITH RASPBERRY PI
164 PAGES OF VIDEO GAME PROJECTS
+ BUILD AN ARCADE MACHINE
PLAY & CODE GAMES!
AMAZE BALLS!
RETRO GAMING WITH RASPBERRY PI

RETRO GAMING WITH RASPBERRY PI

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games. Build your own portable console, full-size arcade cabinet, and pinball machine with our step-by-step guides. And learn how to program your own games, using Python and Pygame Zero.

- ***Set up your Raspberry Pi for retro gaming***
- *Emulate classic computers and consoles*
- ***Learn to program retro-style games***
- *Build a portable console, arcade cabinet, and pinball machine*

BUY ONLINE: ***magpi.cc/store***

Part 05

Add a camera to your low-cost robot

See the world from a robot's eye view by adding a camera to it

MAKER

Danny Staple

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

orionrobots.co.uk

Warning!
Drill safely

Please use safety goggles and a desk clamp when drilling

You'll Need

- Raspberry Pi Camera Module **magpi.cc/h5XSDp**
- Pi Zero Camera Cable adapter (10–15 cm length)
- Hand drill with 8 mm and 2.5 mm bits
- 2.5 mm nylon bolts and nuts
- Needle files (up to 8 mm)

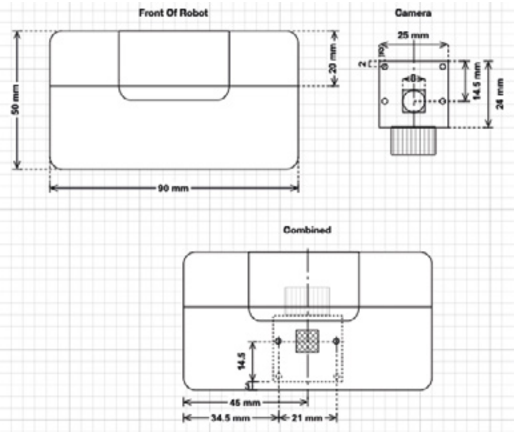
A camera is an exciting accessory to give to a robot. First, you will be able to see the world from the point of view from the robot, creating a mobile periscope or debugging your code from this view. However, a camera also opens up the possibility of processing an image and using it as a sensor in the robot.

In this part, a camera will be securely mounted onto the Lunchbox robot, attached to the Raspberry Pi, and you will get the first images from it.

01 Planning out

Find a sketch of the camera and make a cardboard replica to test fit this in the robot. The camera is upside-down, so its cable is clear of Raspberry Pi.

Make a drawing of the front of the robot with dimensions to add camera holes. Make a vertical dashed line for the screw holes and sensor at 14.5 mm plus clearance from the base. 17.5 mm works with 3 mm clearance.



Make sketches of the front of the robot, the camera (using the net), and dimensions to position any holes. Shade where holes will be

At the horizontal middle of this line, draw the 8 mm square hole for the lens, with mounting screws either side by 10 mm. These have a 2 mm diameter. Two screws are enough for the camera.

02 Some disassembly required

A space behind the front of the robot is needed to make the camera holes safely without breaking anything. Remove at least one battery from the robot to prevent short circuits.

To keep wires tidy, use tape to group bundles of sensor grounds, sensor Vcc, sensor signals, and battery power. Mark them to help. Unplug these from the breadboard.

Unscrew the battery leads and remove the top of the robot. Now unscrew its Raspberry Pi, keeping the screws for later. Lift Raspberry Pi and breadboard out and to the side of the robot.

03 Marking out

Use the sketches, a ruler, and set square to measure and mark out where to drill the holes.

Start with the height first and carefully use a sharp tool, like a maths set compass, to scratch a horizontal line for the height of holes, and another for the camera.

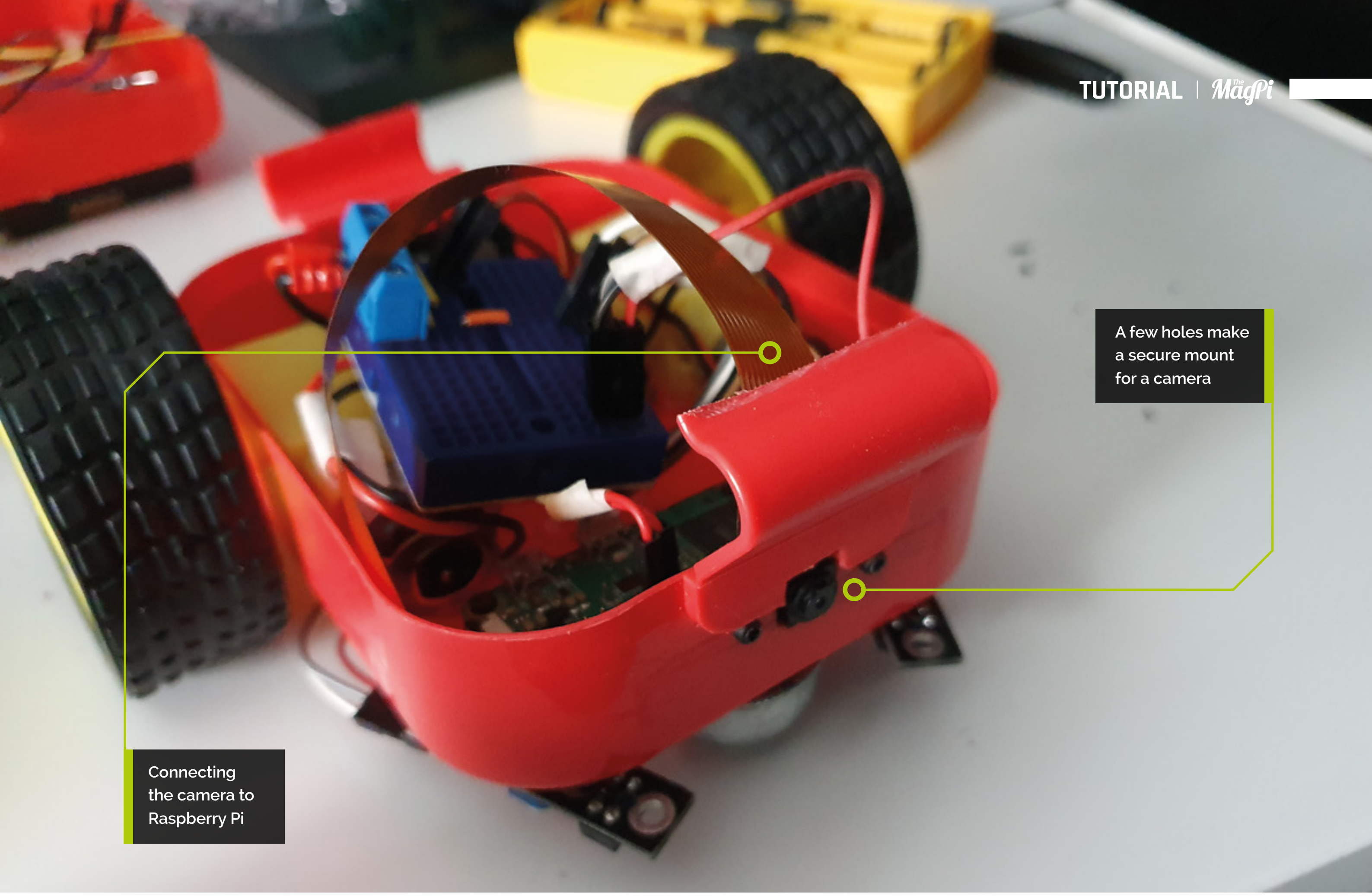
Use a ruler across the horizontal; scratch marks for the 8 mm camera hole centre, and then 10.5 mm either side of this for screw holes.

The front panel should now have three crosses scored into it, one for each hole.

04 Drilling the holes

Wear goggles for this step

With the 2 mm holes, use the 2 mm drill bit resting in the cross. Start slowly and speed up as the tool



A few holes make a secure mount for a camera

Connecting the camera to Raspberry Pi

goes through. Don't drill too deep and into the electronics behind. Support the plastic, but never put fingers where the drill bit is going. Robot parts are more replaceable than fingers.

For the 8 mm hole, start with the 2 mm bit, but then work up through sizes to the 8 mm bit. If the holes are rough, let the drill bit spin and move it back and forth through the hole to clean it up.

> "Not every robotics task is glamorous, and filing away plastic can take a while"

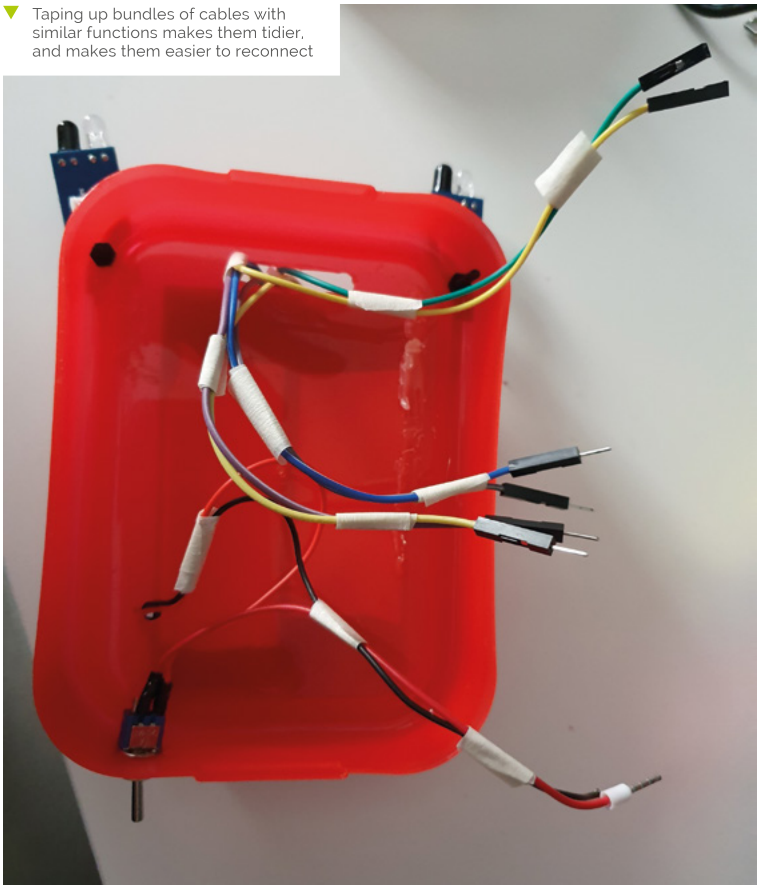
05 Making a square hole

The 8 mm hole is round, and Raspberry Pi Camera Modules need a square hole. Not every robotics task is glamorous, and filing away plastic can take a while.

Push a flat needle file gently through the middle of the 8 mm hole, work it back and forth until it goes through smoothly, then file one side to make the shoulders of the square. Repeat for other sides.

Measure and repeat until it's square and 8.5 mm each way. When it looks right, try fitting the camera – remove to scrape more if needed. The screw holes should line up around the lens.

▼ Taping up bundles of cables with similar functions makes them tidier, and makes them easier to reconnect



Top Tip

Letting the drill glide

Do not push or force a drill. It should cut its way through and glide into a hole. Go slowly to avoid snags.

06 Attaching the camera cable

The camera has a connector for a cable to attach it to Raspberry Pi. The adapter cable is flat, with a wide end for the camera and a narrow end for Raspberry Pi Zero's connector.

The cable is less fiddly to attach before bolting the camera in place. Slide the plastic securing tab on the camera connector into the out position on the camera.

Ensure the contacts (metal bits) at the wide end of the cable face into the camera board, then push gently into the connector, ensuring it is straight. Slide the plastic tab in to secure it.

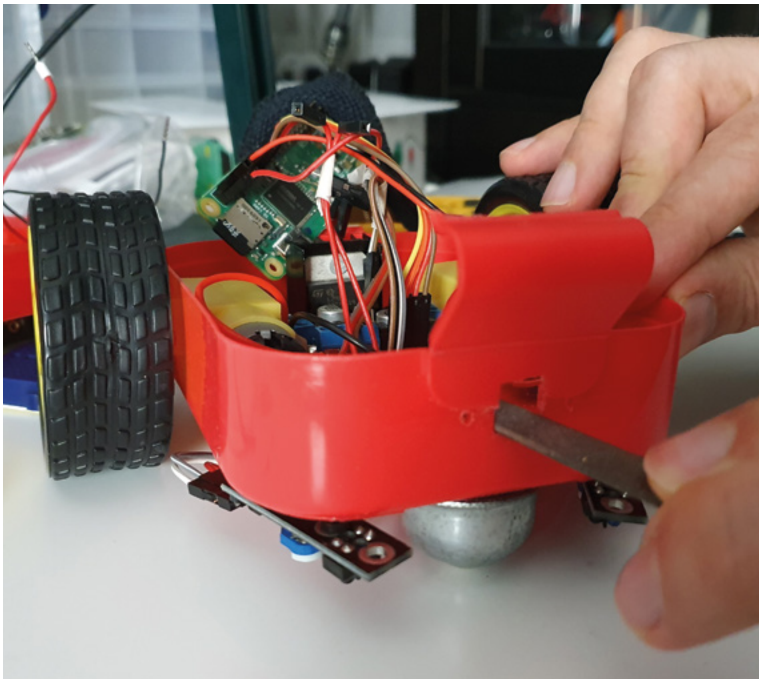
07 Fitting the camera

With the camera removed, push the 2.5 mm screws through from the front. Now line the camera up behind the screws and push it on, with the cable facing upwards.

If the lens and screws don't align, take note of where the overlap is and adjust the lens hole by filing more. Adjusting and fitting may take a few attempts.

Once the camera fits well, put the nuts in from behind and gently turn them. Tighten them up only a little, taking great care not to over-tighten against the robot. The camera should be firmly attached, but not crushed.

The large hole for the camera should be filed square for it to poke through. This will need patience. A rotary tool could help



08 Plugging the camera in

The narrow end of the cable goes into a Raspberry Pi Zero. Partially manoeuvre your Raspberry Pi back into the robot, but leave space to work and get the to the camera connector.

Raspberry Pi Zero has a somewhat fragile retaining clip on the camera connector, so take care when loosening this.

Make gentle bends (no kinks) to comfortably get the camera ribbon cable to go face-up into the connector on Raspberry Pi.

Gently push the cable into the connector with the contacts (metal parts) facing the top of the clip. Push the retaining clip back in carefully.

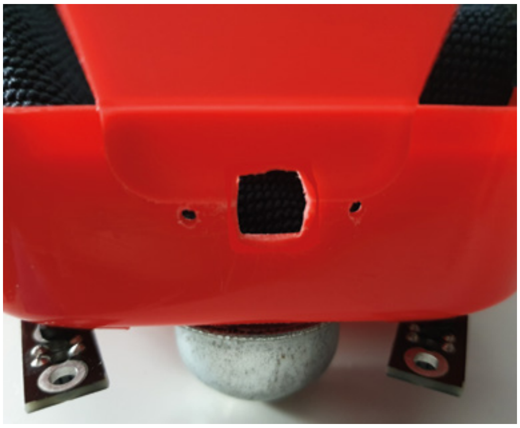
> “Gently push the cable into the connector with the contacts (metal parts) facing the top of the clip”

09 Reassembling the robot

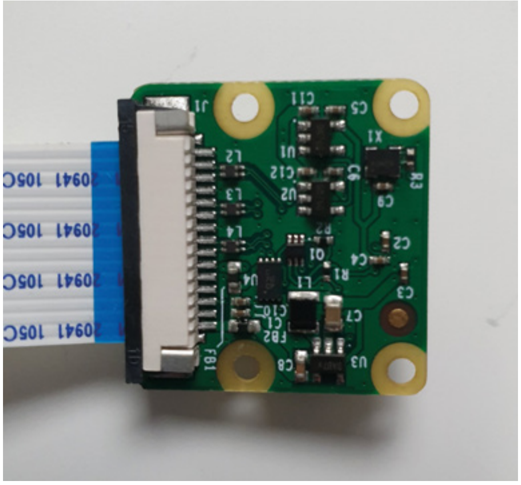
At this point, the robot's Raspberry Pi Zero can be put into place and bolted back into the robot. Above this, the breadboard is lowered in. Make good any electrical connections on this layer that were disconnected or came loose.

Reconnect the top layer sensor and power cables to Raspberry Pi's GPIO header and breadboard. The previous code used BCM pins 13 and 26 for the top sensors. Verify the polarity of reconnected power cables.

Place the top of the robot back on, ensuring not to catch any wires and that the camera cable is not being distressed.



After drilling, the camera mount area should look like a bit like this. It may need filing to remove rough edges



Gently slide the clip from the camera. Gently push the cable in straight. The contacts should be facing into the camera board

10 Configuring the camera

Turn on the robot and connect to it. Once connected, type the following:

```
sudo raspi-config
```

Typing this starts a configuration menu. Using the up and down arrows on your keyboard, move to ‘Interfacing options’. Press **ENTER** to select this. Choose ‘Camera’ on the next screen and select ‘yes’, then ‘Ok’. Press the **ESC** button on your keyboard to exit.

Enter `sudo reboot` to restart this Raspberry Pi, so the camera is ready to use. Wait a couple of minutes for it to reboot, and then use SSH to connect to it again.

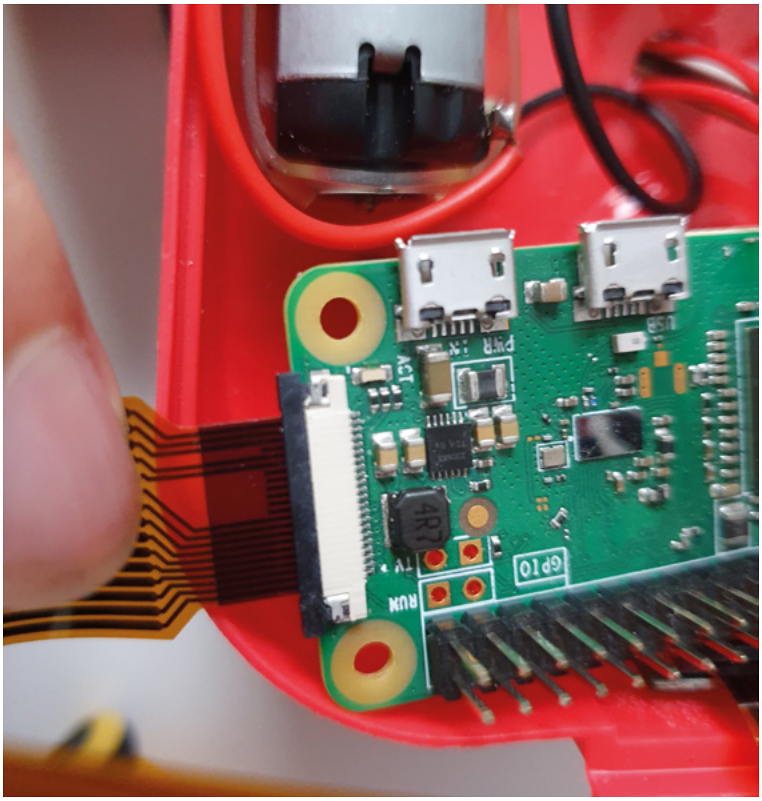
11 Taking a first picture

The raspistill tool can be used to get the first picture and ensure the camera is correctly installed. From Raspberry Pi SSH connection, enter the following:

```
raspistill -o test_image.jpg --vflip
```

This asks the camera to take a picture. The `-o` option says to store the output in the file **test_image.jpg**”. The `--vflip` option flips it vertically, as the camera is mounted upside down.

Depending on light conditions, it should take a few seconds and return without an error. You can use SFTP software, as shown at **magpi.cc/6PK7Bd**, to copy the picture back to view it.



Unclip Raspberry Pi’s camera connector very carefully (it is fragile). Slide in the connector with the contacts facing up. Gently push in the clip

12 Troubleshooting

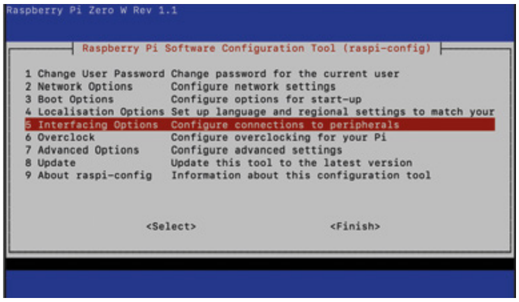
If raspistill fails to capture an image, read any error messages carefully for information.

If cables need to checked, turn off the robot with `sudo poweroff`. Check the camera connections, ensure they are the correct way up on both ends, then try retaking a photo.

Fresh batteries are needed if the robot loses power taking a photo.

A common problem is a tiny ‘sunny’ connector on the front of the camera. Check if it’s loose and if so, carefully push it back in.

Also, ensure you have used the raspiconfig menu in Step 10 set up and rebooted before trying to obtain an image. M



```
Raspberry Pi Zero W Rev 1.1

         Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

                  <Select>                       <Finish>
```

In raspi-config, select Interfacing options, then Camera, and select Yes to enable it

Top Tip

Help, I'm running out of space!

Replacing breadboard with soldered stripboard, the L298N with a DRV8833 or motor HAT, and putting motors outboard are ideas to free up space in the robot.

Smart Christmas tree lights

Bored of normal Christmas lights? Hack your Christmas tree with a Raspberry Pi and make it festively voice-controlled!

MAKER

Rob Zwetsloot

Rob is amazing. He's also the Features Editor of *The MagPi*, a hobbyist maker, cosplayer, comic book writer, and extremely modest.

magpi.cc

Christmas is here, and that means it's time to decorate. Over the years, we've made Christmas tree stars, normal tree lights, and even a light-up card you can make out of the cover of issue 52!

This year, we're heading back to the tree lights to give them an extra ability: voice-controlled lights! We'll be doing this using a slightly easier method than other audio services, so you won't need to sign up to any Amazon or Google developer accounts. Let's get festive!

01 The right lights

For our specific build, we're using 3 m of RGB NeoPixels with 30 LEDs per metre. You can find the ones we used here: **magpi.cc/iP97nc**.

These may not be the right lights for you, though. While we think 144 lights per metre is a bit much, the 60 lights per metre version would work well wrapped around a tree or installed on some other furniture. You may also consider getting RGBW LEDs for a brighter (and easier to program) white light effect if you desire.

02 Install the software

For the voice control and NeoPixels, we need to install a few extra Python libraries to Raspbian. First, install SpeechRecognition by opening the Terminal and typing:

```
sudo pip3 install SpeechRecognition
```

Then we need to install PyAudio and a FLAC encoder so that Raspberry Pi can hear what you're saying and record it. Do it with:

```
sudo pip3 install pyaudio
sudo apt install flac
```

Plug a microphone in, and make sure that Raspbian is using it as an input device (you may need to right-click on the volume icon in the top right to do so). Then run the test script:

```
python3 -m speech_recognition
```

It will ask you to speak a word and it should return what you said.

Finally, install the NeoPixel libraries with:

```
sudo pip3 install rpi_ws281x
adafruit-circuitpython-neopixel
```

03 Build the circuit

Build the circuit using the Fritzing diagram as a guide (**Figure 1**, overleaf). A Raspberry Pi can only power a handful of NeoPixel LEDs safely at a time, which is why you need an external power supply to power the LED strip. Four rechargeable AA batteries will get you the roughly 5 V you need. However, if you have a spare 5 V power supply and a jack to use, that will work just as well, if not better.

The button is installed so we can activate the voice commands – it's much easier than adding a trigger word like on a home voice assistant.

04 Test your lights

Once the circuit is all wired up, test your NeoPixels using the **neopixel_rpi_simpletest.py** script (if you can't find it on your Raspberry Pi, head here: **magpi.cc/Pci4iN**).

You'll Need

- Flexible NeoPixel lights **magpi.cc/iP97nc**
- USB microphone
- Python SpeechRecognition library **magpi.cc/5b2vkB**
- Push-button
- 5 V power supply (batteries or PSU)
- 220 Ω resistor

Make sure to test your setup while you're building, so you can discover any issues early on

Change the number of NeoPixels to your requirements and run the script. If the colours seem off, you may need to change the `ORDER =` line from `neopixel.GRB` to `neopixel.RGB` – or `neopixel.GRBW` or `neopixel.RGBW` if you have an RGBW strip of NeoPixels.

This is also a good chance to make sure your circuit is properly wired up – check the grounds, the specific GPIO pins used, and whether or not it's powered properly.

> A Raspberry Pi can only power a handful of NeoPixel LEDs safely at a time

05 Tweak the code

Download or type out the code listing for this tutorial – **smartlights.py**. This code was specifically tailored to our build, so you may need to make some changes to the code: specifically, the number of NeoPixels in your circuits, the type of NeoPixels (RGB/GRB/RGBW/GRBW), and the GPIO pins you're using for the lights and button if you modified our circuit.

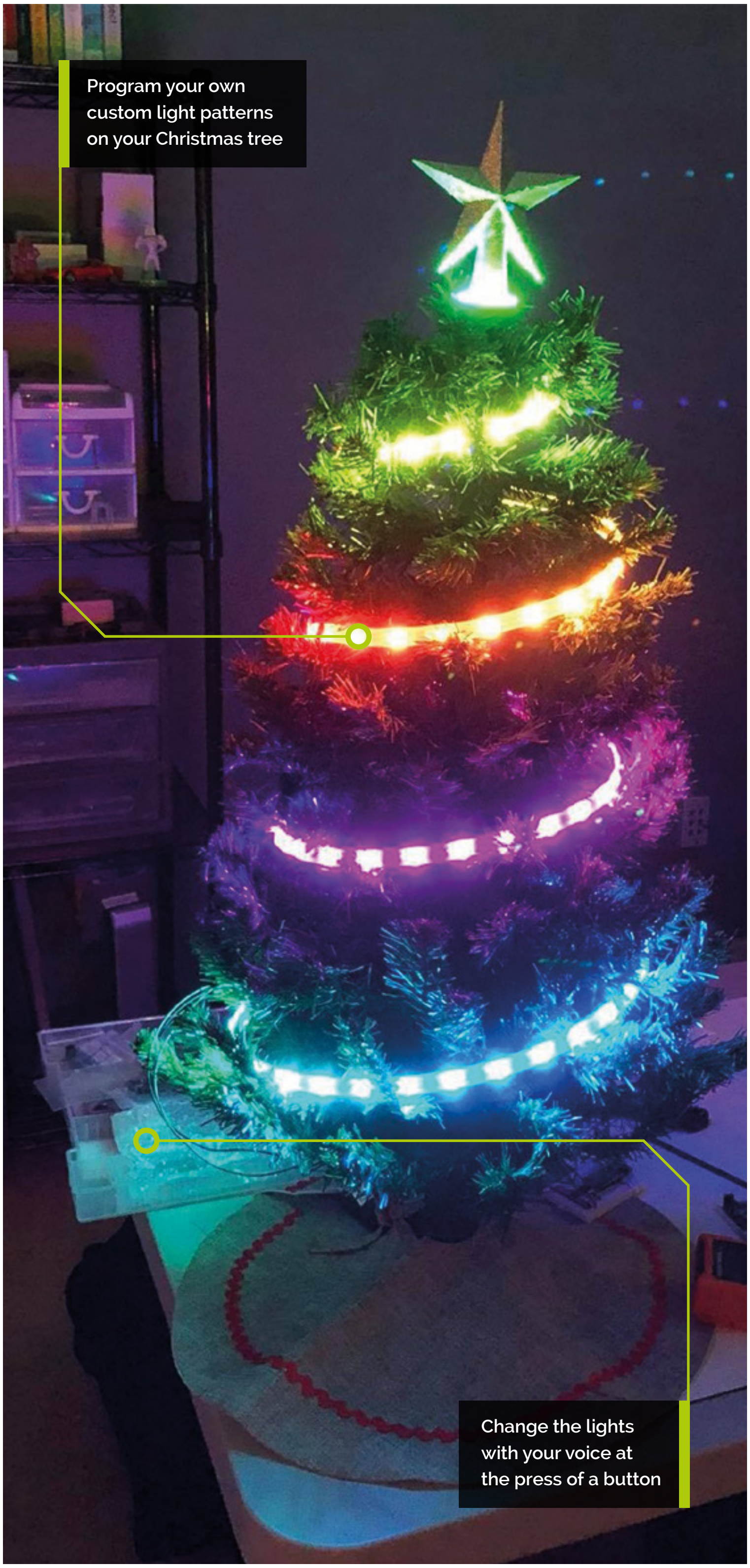
Save the file, open up the Terminal, and enter:

```
sudo nano /etc/profile
```

Add this line to the end of the file:

```
sudo python3 /home/pi/smartlights.py &&
```

This will start the script automatically whenever you turn on your Raspberry Pi.



Program your own custom light patterns on your Christmas tree

Change the lights with your voice at the press of a button

Figure 1

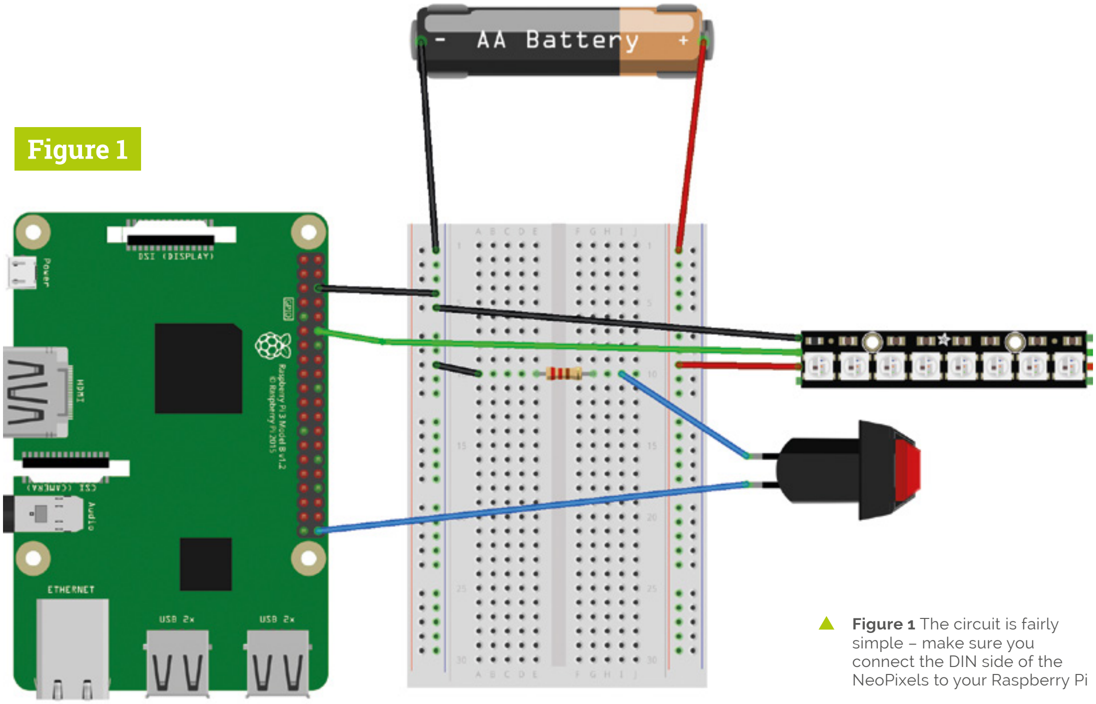


Figure 1 The circuit is fairly simple – make sure you connect the DIN side of the NeoPixels to your Raspberry Pi

Top Tip

Speech recognition troubleshooting

If you're having trouble with the test for the SpeechRecognition library, head here to troubleshoot: **magpi.cc/RgpLyc**.

06 Explaining speech recognition

The speech recognition part of the code is remarkably simple, and is mostly handled by the speech recognition library. It's imported as `sr` in the script, and we use it to listen in on the microphone when required. Once a phrase has been said, it sends an audio file to Google Speech Recognition to be analysed, which is the `value` variable in our code.

We use this `value` variable to check against our `if` statements – you'll notice we haven't capitalised 'lights on' or 'lights off', but we have done for 'Merry Christmas' as it returns the value in that way. If you plan to add more phrases, you may need to experiment with them.

07 Adding trigger phrases

We've tried to keep this very simple, including trigger words as part of `if` statements. To add one, all you need to do is add an extra `if` statement to the loop. Say you want to add the term 'happy holidays', you'd do so like this:

```
if value == 'happy holidays':
    strip.fill((255,0,0)
    strip.show()
```

In this example, we make the lights turn bright red.

08 Custom light patterns

The basics of setting LED colours with NeoPixels involves telling the NeoPixels which red, green, and blue (RGB) values to use, from 0 (off) to 255 (maximum). In the previous step, we set red/R to 255 for full red. You can make all three 255 for white. Check out this colour chart to find colour values: **magpi.cc/Ey77xh**.

For more complex lights (alternating colours, rainbow effects, etc.) you'll need to create a specific function that executes the effect. Some of these can be quite complicated, so we suggest checking the examples and docs for the NeoPixel library if you have an idea you'd like to try: **magpi.cc/SQynFx**.

09 Install your lights

Once you've perfected your lights and voice control, it's time to put the lights up! We like to wrap ours around the tree, making sure there's easy access to change any batteries. You may need to add some clips to the tree as well.

If you plan to stick them to other furniture, you may need to make sure you have a temporary solution, like Velcro with sticky backs, so you can easily add and remove them at the start and end of the festive season.

Top Tip

RGBW code

With RGB NeoPixels, you only need to program three variables: red, green, and blue. With RGBW ones, you'll need to add a fourth, e.g. (255,0,255,255).

▲ We've seen people install lights in bookcases for fun effects

10 Light up your tree!

It's time to do the final test of your lights and turn them on! Test out the voice control, and maybe think of moving the microphone position around. With some really long wires, you can also put the button to activate the speech recognition in a place that's easy to reach.

At the end of the day, you don't have to turn the lights off, either: you just need to say 'lights off'.

> " The speech recognition part of the code is remarkably simple "

11 Adding sound and more

One extra feature you might consider is sound output, which can be handled with Pygame. For some voice keywords, you could have lights sync up to classic Christmas songs, carols, or whatever you choose.

We suggest making it so the song only plays through once before going back to normal, though, to avoid incurring the wrath of the people you live with. Last year we create a tree-topper star – you can easily add some NeoPixels to a 3D-printed star like this and have the system control it as well!

12 Happy Holidays!

We really hope you enjoy making this project and, even if you don't celebrate Christmas, we hope you think of it as a great introduction to voice control with Python.

From all of us at *The MagPi*, Happy Holidays and a Happy New Year!

smartlights.py

DOWNLOAD THE FULL CODE: magpi.cc/SmartXmas

▶ Language: **Python**

```python
import speech_recognition as sr
import time

from gpiozero import Button

import board
import neopixel

button = Button(21)

# speech recognition settings
r = sr.Recognizer()
m = sr.Microphone()

# LED strip configuration:
LED_COUNT    = 90      # Number of LED pixels.
LED_PIN      = board.D18      # GPIO pin
LED_BRIGHTNESS = 0.2  # LED brightness
LED_ORDER = neopixel.GRB # order of LED colours. May also be GRB, GRBW, or RGBW

# Create NeoPixel object with appropriate configuration.
strip = neopixel.NeoPixel(LED_PIN, LED_COUNT, brightness = LED_BRIGHTNESS, auto_write=False, pixel_order = LED_ORDER)

# Setting variables for a specific sequence
red = (255,0,0)
green = (0,255,0)
flip = 0

# Function to make an alternating series of lights
def merrychristmas():
    global flip
    for i in range(90):
        if flip == 0:
            strip[i] = red
            flip = 1
        else:
            strip[i] = green
            flip = 0
    strip.show()

while True:
    button.wait_for_press()
    button.wait_for_release()

    # set the audio source

    with m as source: audio = r.listen(source)

    # recognize speech using Google Speech Recognition
    value = r.recognize_google(audio)

    # check the speech against trigger words
    if value == 'lights on':
        strip.fill((255,255,255))
        strip.show()

    if value == 'lights off':
        strip.fill((0,0,0))
        strip.show()

    if value == 'Merry Christmas':
        merrychristmas()
```

Part 02

Hack GraviTrax with Raspberry Pi

Make your GraviTrax layout trigger LEDs for a dazzling light show to enhance your layouts

MAKER

Mike Cook

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

magpi.cc/TPaUfT

You'll Need

- Pimoroni LED SHIM **magpi.cc/9962nm**
- Pimoroni 11×7 LED Matrix Breakout **magpi.cc/JdMBNi**
- WS2812 LED strip, 144 per metre **magpi.cc/BfaifB**

Last month we saw several ways of detecting the balls as they passed by in a GraviTrax layout. We used this to trigger sound samples to accompany your ball's run through your layout. This month we look at how to use those triggers to fire LED patterns. We are going to use both I^{2}C-controlled displays and the ubiquitous WS2812B LEDs, sometimes called NeoPixels.

01 What sort of LED?

We will look at two sorts of LED here: the WS2812 and the I^{2}C-controlled type, based on the 31FL3731 LED matrix driver. This last type has become more popular lately as it lends itself to controlling a lot of LEDs from a very small package. As the I^{2}C protocol is a bus, each 31FL3731 you have on a bus needs to have a different address. This chip is capable of being set to four different addresses, depending on what signal you connect to an external address line pin on the chip. There are techniques you can use with the I^{2}C bus to get more, but we won't go into these here.

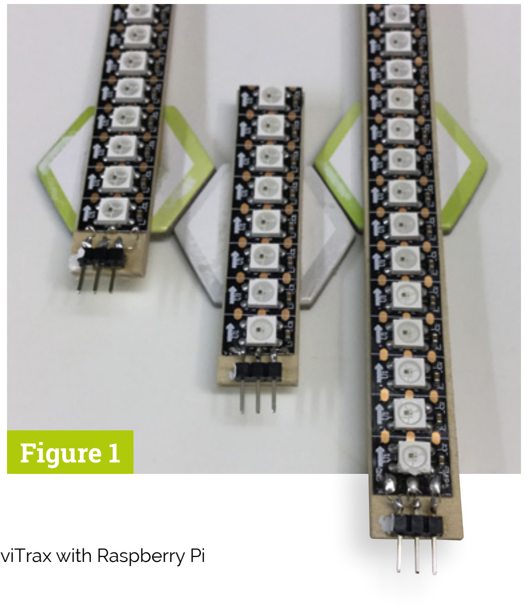


Figure 1

Figure 1 Three lengths of LED strip

02 The WS2812B

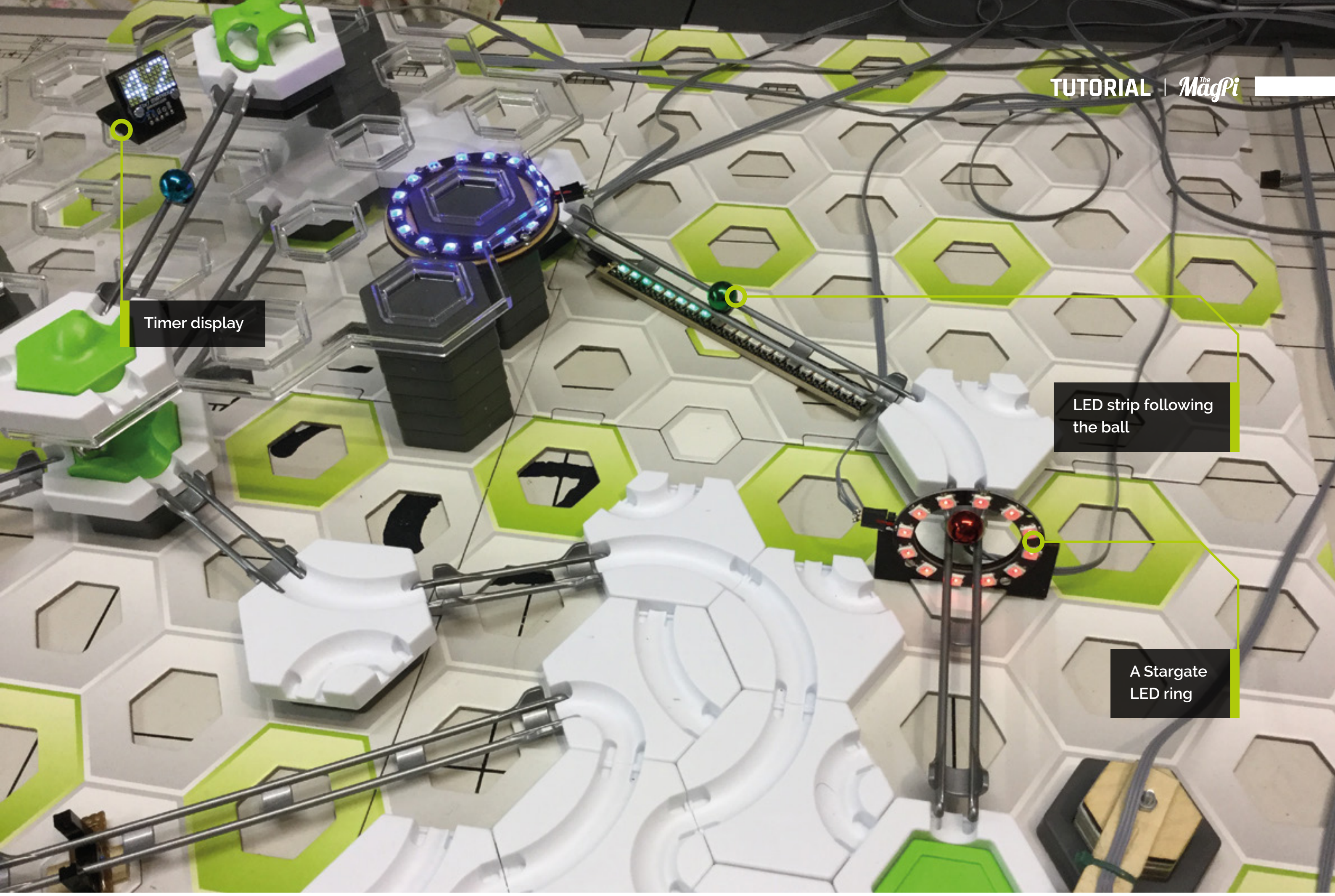
The WS2812B (aka NeoPixel) LED is available in a number of different configurations, ranging from individual LEDs to arrays. We will concentrate here on just using the LED strips and LED rings. The aim is to make them fit into the GraviTrax system like an extension component, as we did with the detectors last month. For the strips, we used the highest LED density we could find at 144 LEDs per metre, although you could use others. Making a strip to run under a track is simple. We mounted a small section of LED strip on 13 mm wide, 3 mm plywood. The length of track determines how many LEDs you can fit underneath them.

03 LED strips

For the three lengths of track used in the starter pack, we made 8-, 14-, and 20-LED strips. We glued a three-pin connector on the end and wired it up to the strip. Then we glued a cardboard hexagon to the middle of the strip, so the strip went across the flats of the hexagon. For the 8- and 20-LED strips, the hexagon was glued in the middle of the strip. But for the 14-LED strip, we glued the hexagon 10 mm from one end because it only spans two hexagon positions – see **Figure 1**.

04 LED rings

There are two sizes of LED ring we found useful, the 12-LED ring and the 16-LED ring. We mounted the 16-LED ring horizontally on a 71 mm round disc of 3 mm plywood, and cut a 6 mm hexagonal hole in, to push-fit a large height tile into it. You can do this with a fret-saw if you haven't got access to a laser cutter. If your cutting is not so accurate and the tile pushes through, you can always fix it with a dab of hot-melt glue.



A vertically mounted ring, where the ball runs through it, just has to be known as a Stargate. See **Figure 2** for the dimensions of the mounting for each type.

05 Horizontally mounted ring

After cutting out the disc with a hexagonal hole in it, we used the existing holes in the ring to act as a marker. One hole at a time, we drilled through the holes in the ring with a 2 mm drill. After each hole, we screwed the plate and ring together with an M2 screw and nut so the assembly would not move while we did the next one. When all the holes were drilled, we fixed the ring to the mount just using 5 mm-long M2 screws and nuts to act as a spacer. Note that this holds the ring firmly in place without resorting to a nut on the other end of the wood. See **Figure 3**.

06 The Stargate

We used a 12-LED ring for this, mounted on a half plate. This time we used only two of the mounting holes to fix the ring on the plate, including nuts on the back of the board. We cut one cardboard hexagon in half across the points,

Figure 2

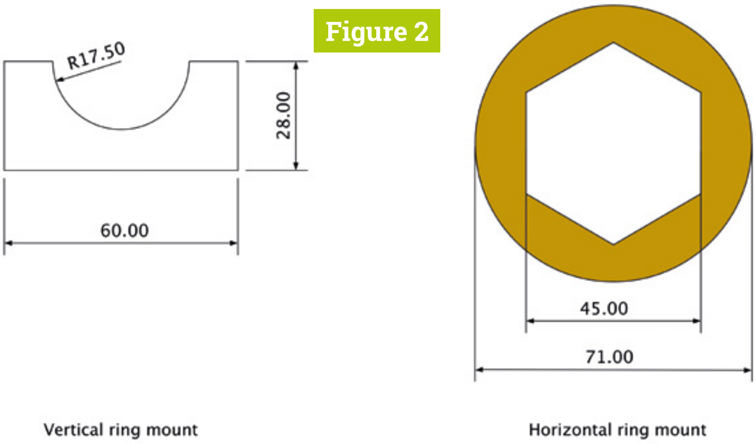
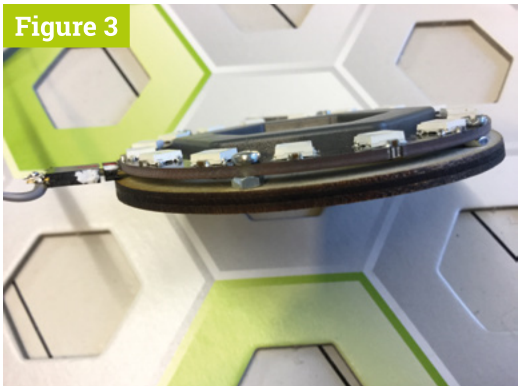


Figure 3



▲ **Figure 2** Mountings for horizontal and vertical rings

◄ **Figure 3** Detail of horizontal ring

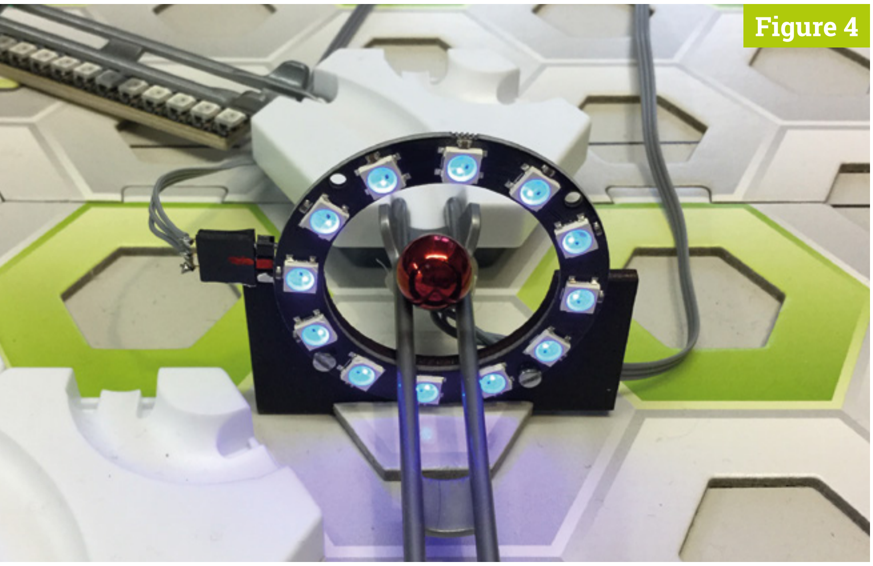


Figure 4

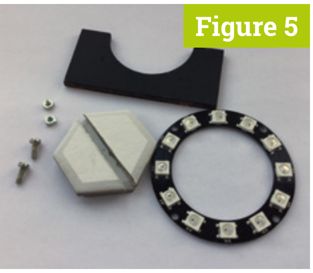


Figure 5

▲ **Figure 4** Parts for the Stargate ring

▲ **Figure 5** The finished Stargate ring

▼ **Figure 6** Schematic of WS2812 driver board

glued this to another hexagon, and sandwiched the mounting plate between the two half hexagons – see **Figures 4** and **5**. The rings we got had the order of connections different to the strips: the 5 V power was in the middle. To correct this, we made a crossover lead and, to be sure we didn't mix them up with the others, we marked then with a blob of red paint in the middle.

07 Driving the LEDs

Normally when you use WS2812B LEDs, you chain them – that is, connect the data output from each strip to the data input on the next. While this works well most of the time, it doesn't work here with the modular mix-and-match arrangement we would like. So we decided to only use the data input to a strip and use a bit of electronics to steer the data signal to one of eight outlets. We also need a bit of level shifting to boost the 3.3 V signal from Raspberry Pi to the 5 V level the LEDs need. This circuit is shown in **Figure 6**.

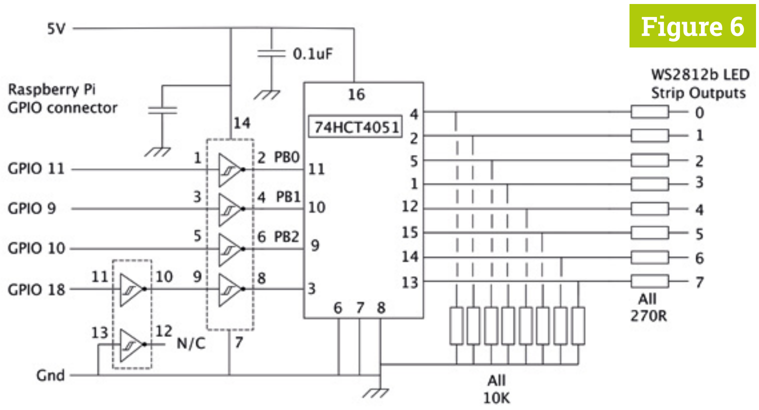


Figure 6

08 LED driver circuit

The circuit works using a 74HCT4051 analogue multiplexer to direct the data from GPIO 18 to one of the outputs. This signal is boosted to 5 V first by passing it through two inverters – two so we don't change the signal polarity. Each of the three select lines is also boosted to 5 V and we cope with the signal inversion by simply relabelling the output pins. The circuit was made on a piece of stripboard 37 holes long and 13 strips high, and the physical layout is shown in **Figure 7**, with the cut tracks on the back shown in **Figure 8**.

09 Using 31FL3731 LEDs

We used two types of 31FL3731 LED boards from Pimoroni: the colourful LED SHIM with 28 tiny LEDs, and the 11×7 white matrix. They both use the I²C bus to communicate and, disturbingly, are at the same address of 0x75. However, the address of the matrix can be changed by cutting a link on the back marked ADDR-2 to change it to 0x77, and a look at the data sheet shows that it can be set to two more addresses by connecting the top pad to either the I²C data pin (0x76) or to ground (0x74). Whit enables you to use three matrix boards with a SHIM, or four without.

> "The SHIM is cleverly designed to be directly pushed over the GPIO pins"

10 LED SHIM

The SHIM is cleverly designed to be directly pushed over the GPIO pins, so we had to modify it in order to use it with the GraviTrax system. First, we soldered a 4-pin header onto even pins 32 to 38 – this was purely for mechanical support. Then we soldered another on the odd-numbered pins 1 to 7, and connected pin 1 to pad for pin 4 for the power, and pin 7 to the pad for pin 20 for the ground with thin wire. Then we cut a piece of 16 mm by 69 mm plywood and pushed the header pins into the wood. The small indentations were marked with pencil and 1 mm holes drilled, so the SHIM pushed into the board. Painted black and glued onto a stack of two cardboard hexagons, the results are shown in **Figures 9** and **10**.

11 11×7 matrix

This is a white LED matrix and we can mount it in the GraviTrax system in two ways – horizontally or vertically. Horizontal mounting uses 45 mm by 24 mm piece of 3 mm plywood, mounted on a stack of three cardboard hexagons (**Figure 11**). Vertically uses a 25 mm length of 32 mm right-angle wood moulding on a single hexagon. We used a long header and pushed the pins to one end of the plastic strip to give us an even longer header length of 9 mm. By using M2.5 screws and 3 mm nylon stand-off tubes, we mounted the matrix with the holes provided on it, as can be seen in **Figure 12**.

12 LED animations

The sorts of animations you can get the LEDs to do are almost limitless. The long LED strips can programmed so a single LED follows the path of the ball underneath it, with careful timing. Or a strip can be placed across the track so that the ball triggers an effect like it is breaking through a barrier. LED rings can be 'kicked' by a passing ball so an LED makes a complete circuit; or they can be made to flash when a dramatic event triggers, like the ball dropping to a lower level happens. Likewise, a matrix can show an animation or a number, like a time (**Figure 13**, overleaf).

13 WS2812 software

Most animations can be used with a large number of variation of colour, speed, and effects, so we need to have software that will allow animations to be called with a large number of variations. We have chosen to use threads to make this simpler. **neo_thread.py** shows what we came up with for the WS2812 strips. We appreciate that a thread can be timed out at any time and, if this happens between the setting of the strip address and the showing of that strip, the animations might be wrong. In practice this rarely happens and is not very noticeable when it does.

14 31FL3731 software

We separated this from the WS2812 software because it's easer to leave out if you don't have any of that sort of display, but it follows the same format as the other code (see **FL3731_thread.py**) You first need to install the libraries for the SHIM

Figure 7

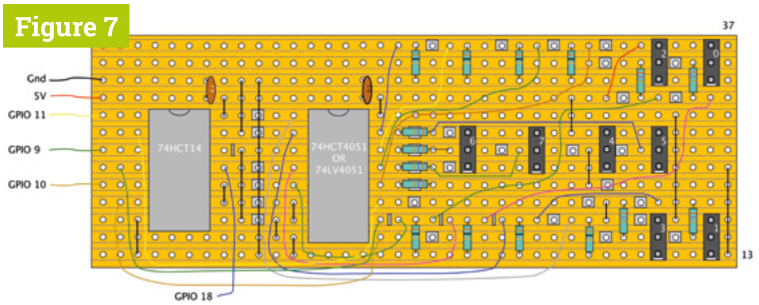


Figure 8

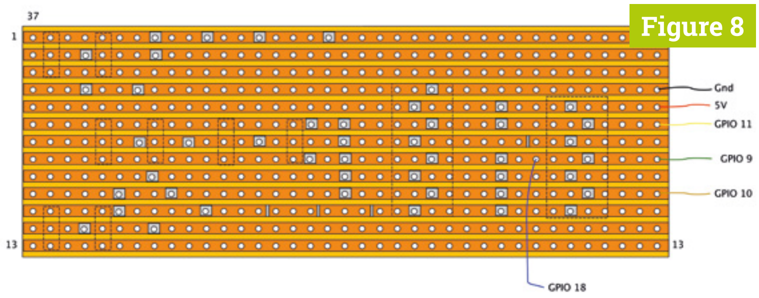


Figure 9

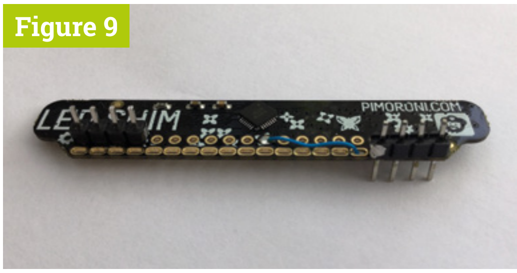


Figure 7 Physical layout of WS2812 driver board

Figure 8 Track cuts for the WS2812 driver board

Figure 9 Modifications to the LED shim

Figure 10

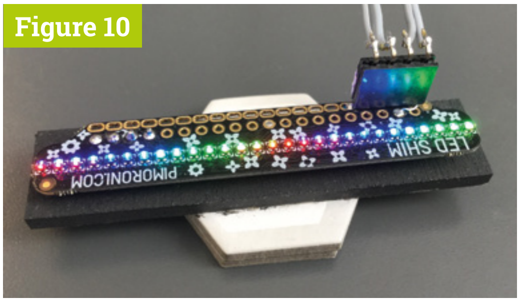


Figure 10 Mounted LED SHIM

Figure 11 Horizontally mounted matrix

Figure 12 Vertically mounted matrix

Figure 11

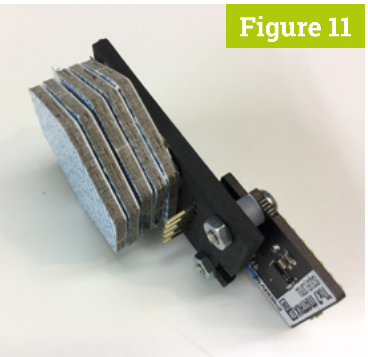
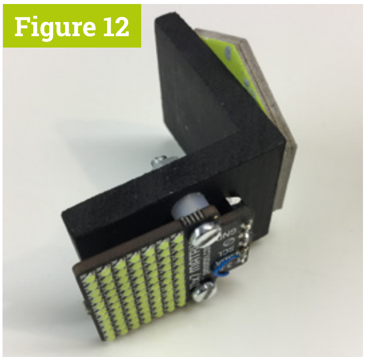


Figure 12



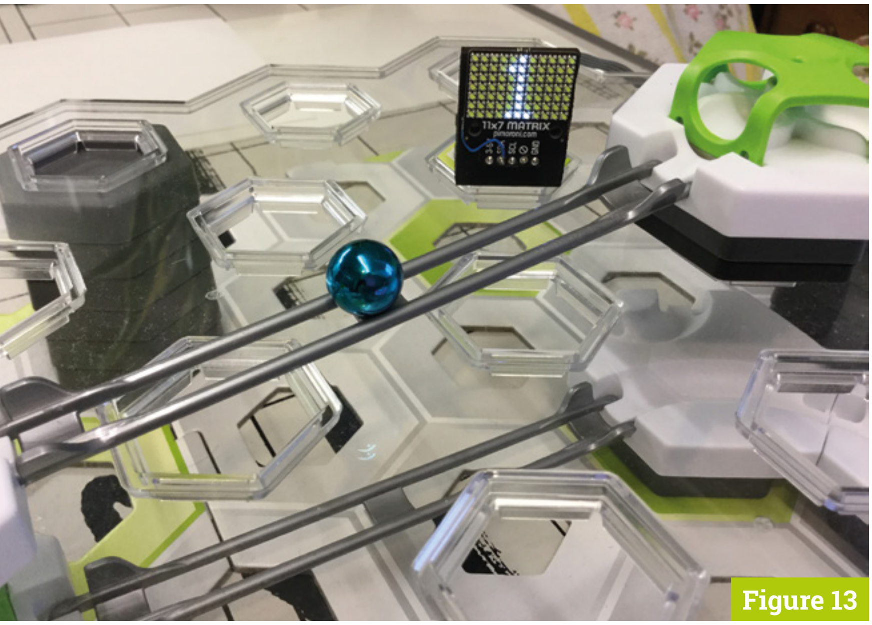


Figure 13

and matrix from the Pimoroni website. Feel free to write your own animation functions to call if you want something else. The SHIM patterns can consist of a pre ball passage animation with another animation for when the ball passes. In the listing, this is triggered on thread 6.

As a temporary measure, **pattern_trigger.py** is some simple code to trigger these effects from your keyboard for testing.

Next month, in the final part, we will look at how to bring this all together in terms of the hardware interface to Raspberry Pi's GPIO connector and having a single integrated software interface.

neo_thread.py

Language: **Python**

```python
#!/usr/bin/env python
# Neopixel patterns from a thread
# By Mike Cook November 2019

import threading
import time
import board
import neopixel
import RPi.GPIO as io

def startWs2812Thread(number):
    if number == 0:
        # Thread  Leds Dely Dir    front     back    wipe
        ws2812_thread0 = threading.Thread(target = push,
            args = (number, 12, 0.5, True,
                (30, 0, 0), (0, 30, 0), True))
        ws2812_thread0.start()
    elif number == 1:
        ws2812_thread1 = threading.Thread(target = push,
            args = (number, 12, 0.5, True,
                (30, 0, 0), (0, 30, 0), True))
        ws2812_thread1.start()
    elif number == 2:
        ws2812_thread2 = threading.Thread(target = push,
            args = (number, 12, 0.5, False,
                (30, 0, 0), (0, 30, 0), True))
        ws2812_thread2.start()
    elif number == 3:
        ws2812_thread3 = threading.Thread(target = push,
            args = (number,12,0.2, False,
                (30, 0, 0), (0, 30, 0),True))
        ws2812_thread3.start()
    elif number == 4:
        ws2812_thread4 = threading.Thread(target = seq,
            args = (number,3, 0.06, False,
                (0, 0, 30), (20, 20, 20), False))
        ws2812_thread4.start()
    elif number == 5:
        ws2812_thread5 = threading.Thread(target = flash,
            args = (number,4,0.12, False,
                (0, 30, 30), (30, 0, 0), False))
        ws2812_thread5.start()
    elif number == 6:
        ws2812_thread6 = threading.Thread(target = run,
            args = (number, 14, 0.13, False, (30, 0, 0),
            (0, 0, 0), False))
        ws2812_thread6.start()
    elif number == 7:
        ws2812_thread7 = threading.Thread(target=push,
            args = (number, 20, 0.03, False,
            (30, 30, 0), (0, 0, 30), False))
        ws2812_thread7.start()

def run(add, numLed, delay, direction, frontCol, backCol, wipe):
    strip[add].fill(backCol)
    for i in range(0, numLed):
        j = i
        if direction : j = numLed - i
        strip[add][j] = frontCol
        setAdd(add) ; strip[add].show()
        time.sleep(delay)
        if not wipe : strip[add].fill(backCol)
    setAdd(add) ; strip[add].show()
    time.sleep(delay)
    strip[add].fill((0, 0, 0)) # clear out
    setAdd(add) ; strip[add].show()

```

pattern_trigger.py

DOWNLOAD THE FULL CODE:

magpi.cc/Et6Qnt

▸ Language: **Python**

```python
#!/usr/bin/env python
# Testing Neopixel and FL3731 patterns from a thread
# By Mike Cook November 2019

import neo_thread as ws
import FL3731_thread as fl

ws.initIO()
fl.initI2C()
print("To quit just type return")
print("To trigger FL3731 threads add 10 to the number you type")
print("To stop animations and counts type 100")
try:
    while 1:
        thread = int(input("Thread to trigger "))
        if thread >=0 and thread <=7 :
            ws.startWs2812Thread(thread)
        if thread >=10 and thread <= 16 :
            fl.startFL3731Thread(thread - 10)
        if thread == 100 :
            fl.stopCount()
            fl.stopAnimations()
except :
     fl.stopCount()
     fl.stopAnimations()
```

```python
def flash(add, repeats, delay, direction, frontCol, backCol, wipe):
    strip[add].fill(backCol)
    for i in range(0, repeats):
        strip[add].fill(frontCol)
        setAdd(add) ; strip[add].show()
        time.sleep(delay)
        strip[add].fill(backCol)
        setAdd(add) ; strip[add].show()
        time.sleep(delay)
    if not wipe :
        strip[add].fill((0, 0, 0))
        setAdd(add) ; strip[add].show()

def push(add, numLed, delay, direction, frontCol, backCol, wipe):
    strip[add].fill(backCol)
    k = numLed // 2
    for i in range(numLed // 2, numLed):
        j = i ; k -= 1
        strip[add][j] = frontCol
        strip[add][k] = frontCol
        setAdd(add) ; strip[add].show()
        time.sleep(delay)
        if not wipe :
           strip[add][j] = backCol
           strip[add][k] = backCol
    setAdd(add) ; strip[add].show()

def seq(add, repeats, delay, direction, frontCol, backCol, wipe):
    seqToLight = ( [3], [2, 4], [1, 5], [0, 6],
            [11, 7], [10, 8], [9], [10, 8], [11, 7],
            [0, 6], [1, 5], [2, 4], [3])
    for j in range(0, repeats):
      strip[add].fill(backCol)
      for i in range(0, len(seqToLight)):
          for k in range(0, len(seqToLight[i])) :
              strip[add][seqToLight[i][k]] = frontCol
          setAdd(add) ; strip[add].show()
          time.sleep(delay)
          if not wipe :
              for k in range(0, len(seqToLight[i])) :
                  strip[add][seqToLight[i][k]] = backCol
      setAdd(add) ; strip[add].show()

def initIO():
    global muxAdd, strip
    io.setwarnings(False)
    io.setmode(io.BCM)
    muxAdd = [11, 9, 10]
    io.setup(muxAdd, io.OUT), # set pins as outputs
    pixel_pin = board.D18
    num_pixels = 20 # maximum LEDs in biggest strip
    ORDER = [neopixel.GRB] * 8
    strip = []
    for i in range(0, 8):
        pixels = neopixel.NeoPixel(
               pixel_pin, num_pixels, brightness = 0.1,
               auto_write = False, pixel_order = ORDER[i])
        strip.append(pixels)

def setAdd(add):
    for i in range(0, 3) : io.output(muxAdd[i], 0)
    if add & 1 : io.output(muxAdd[0], 1)
    if add & 2 : io.output(muxAdd[1], 1)
    if add & 4 : io.output(muxAdd[2], 1)
```

FL3731_thread.py

▶ Language: **Python**

```python
#!/usr/bin/env python
'''
FL3731 patterns from a thread
By Mike Cook November 2019
With some functions from Pimoroni examples
See Pimoroni site for how to install
the matrix11x7 and ledshim libiaries
'''
import threading
import time
from matrix11x7 import Matrix11x7
from matrix11x7.fonts import font5x7
import random
import colorsys
from sys import exit
import ledshim
try:
    import numpy as np
except ImportError:
    exit('This script requires the numpy module\nInstall with: sudo pip install numpy')
ledshim.set_clear_on_exit()

def startFL3731Thread(number):
    if number == 0:
        matrix_thread0 = threading.Thread(target =
                   spinC, args = (2, 0.008, 0.1, 0.01))
        matrix_thread0.start()
    elif number == 1:
        matrix_thread1 = threading.Thread(target =
                   spinA, args = (2, 0.008, 0.1, 0.01))
        matrix_thread1.start()
    elif number == 2:
        matrix_thread2 = threading.Thread(target =
                        timer, args = (2, 1.0))
        matrix_thread2.start()
    elif number == 3:
        matrix_thread3 = threading.Thread(target =
                         solid, args = ())
        matrix_thread3.start()
    elif number == 4:
        matrix_thread4 = threading.Thread(target =
                         pulse, args = ())
        matrix_thread4.start()
    elif number == 5:
        matrix_thread5 = threading.Thread(target =
                         larson, args = ())
        matrix_thread5.start()
    elif number == 6: # after shim background animate
        matrix_thread6 = threading.Thread(target =
                         breakup, args = ())
        matrix_thread6.start()

def stopCount():
    global stopTimer
    stopTimer = True

def stopAnimations():
    global animateStop
    animateStop = True

def initI2C():
    global matrixInstance
    stopCount()
    stopAnimations()
    matrixInstance = []
    for i in range(0,4):
        try :
            ins = Matrix11x7(i2c_address = i + 0x74)
        except :
            print("nothing", i, "at address",
                   hex(i+ 0x74))
            ins = 0
        matrixInstance.append(ins)

def spinC(device, speed, stopSpeed, retard):# clockwise
    pattern = 0
    while speed < stopSpeed:
        draw(pattern, device)
        pattern += 1
        if pattern > 3:
            pattern = 0
            speed += retard
        time.sleep(speed)

def spinA(device, speed, stopSpeed, retard):
    pattern = 3
    while speed <= stopSpeed:
        draw(pattern,device)
        pattern -= 1
        if pattern < 0:
            pattern = 3
            speed += retard
        time.sleep(speed)
    draw(3,device)

def timer(device, speed):  # show time
    global stopTimer
    stopTimer = False
    matrixInstance[device].set_brightness(0.5)
    matrixInstance[device].clear()
    startT = time.time()
    while not stopTimer :
         display = str( (int(time.time()-startT) * 10)
                         // 10 )
         matrixInstance[device].write_string(display,
                        x=0, y=0, font=font5x7)
         if len(display) == 1:
             matrixInstance[device].scroll(-3)
             if display == "1":
                 matrixInstance[device].scroll(-1)
         else :
             if display[0] == "1":
                 matrixInstance[device].scroll(-1)
         matrixInstance[device].show()
         matrixInstance[device].clear()
         time.sleep(1.0)
         if int(display) > 98 : stopTimer = True
    matrixInstance[device].set_brightness(1.0)

def wipeL(device, speed):  # wipe left
    for x in range(0,12):
        matrixInstance[device].clear()
        for y in range(0,7):
            matrixInstance[device].set_pixel(x,y,0.5)
        matrixInstance[device].show()
        time.sleep(speed)

def wipeR(device, speed):  # wipe right
    for x in range(11,-1,-1):
        matrixInstance[device].clear()
        for y in range(0,7):
            matrixInstance[device].set_pixel(x,y,0.5)
        matrixInstance[device].show()
        time.sleep(speed)
    matrixInstance[device].clear()
    matrixInstance[device].show()

def larson():
    global animateStop
    animateStop = False
    REDS = [0] * 56
    SCAN = [1, 2, 4, 8, 16, 32, 64, 128, 255]
    REDS[28 - len(SCAN):28
            + len(SCAN)] = SCAN + SCAN[::-1]
    start_time = time.time()
    while not animateStop:
        delta = (time.time() - start_time) * 56
        offset = int(abs((delta % len(REDS)) - 28))
        for i in range(0, 28):
            ledshim.set_pixel(i, REDS[offset + i],
                              0, 0)
        ledshim.show()
        time.sleep(0.05)

def solid():
    global animateStop
    animateStop = False ; step = 0
    while not animateStop :
        if step == 0 : ledshim.set_all(128, 0, 0)
        if step == 1 : ledshim.set_all(0, 128, 0)
        if step == 2 : ledshim.set_all(0, 0, 128)
        step += 1 ; step %= 3
        ledshim.show() ; time.sleep(0.5)

def pulse():
    global animateStop
    animateStop = False
    while not animateStop:
        for z in list(range(1, 10)[::-1]) + list(
                      range(1, 10)):
            fwhm = 15.0 / z
            gauss = make_gaussian(fwhm)
            y = 4 ; start = time.time()
            for x in range(ledshim.NUM_PIXELS):
                h = 0.5 ; s = 1.0 ; v = gauss[x, y]
                rgb = colorsys.hsv_to_rgb(h, s, v)
                r, g, b = [int(255.0 * i) for i in rgb]
                ledshim.set_pixel(x, r, g, b)
            ledshim.show()
            end = time.time()
            t = end - start
            if t < 0.04:
                time.sleep(0.04 - t)

def breakup():
    stopAnimations() # stop animation thread
    showTime = 2.0 # show for 2 seconds
    startTime = time.time()
    while time.time() - startTime < showTime :
        pixels = random.sample(
               range(ledshim.NUM_PIXELS),
               random.randint(1, 5))
        for i in range(ledshim.NUM_PIXELS):
            if i in pixels:
                ledshim.set_pixel(i, 255, 150, 0)
            else:
                ledshim.set_pixel(i, 0, 0, 0)
        ledshim.show() ; time.sleep(0.05)
    ledshim.clear() # turn off after showTime
    ledshim.show()

def draw(n,ins):
    matrixInstance[ins].clear()
    if n == 2:
       y=0
       for x in range(2, 9):
           matrixInstance[ins].set_pixel(x, y, 0.5)
           y +=1
    elif n == 3:
       x=5
       for y in range(0,7):
           matrixInstance[ins].set_pixel(x, y, 0.5)
    elif n == 0:
       y=6
       for x in range(2, 9):
           matrixInstance[ins].set_pixel(x, y, 0.5)
           y-=1
    elif n == 1:
        y=3
        for x in range(1, 10):
            matrixInstance[ins].set_pixel(x, y, 0.5)
    matrixInstance[ins].show()

def make_gaussian(fwhm):
    x = np.arange(0, ledshim.NUM_PIXELS, 1, float)
    y = x[:, np.newaxis]
    x0, y0 = 3.5, (ledshim.NUM_PIXELS / 2) - 0.5
    fwhm = fwhm
    gauss = np.exp(-4 * np.log(2) * ((x - x0) ** 2 +
(y - y0) ** 2) / fwhm ** 2)
    return gauss
```

Set up a Raspberry Pi retro games console

Lakka lets you relive the games of the past by enabling your Raspberry Pi to emulate a host of retro computers and consoles

MAKER

Lucy Hattersley

Lucy is editor of *The MagPi* magazine. She enjoys retro gaming; especially making retro games.

magpi.cc

You'll Need

- Raspberry Pi 4
- USB or wireless game controller, e.g. **magpi.cc/vilrosgamepad**
- Windows PC or Mac computer for setup
- Blank microSD card (8GB or larger)
- SD Formatter **magpi.cc/sdcardformatter**
- NOOBS image file **magpi.cc/downloads**
- A game ROM, e.g. **magpi.cc/bladebuster**

Whether you are nostalgic for the games of yesteryear or you're simply dying to discover gaming's rich history, all you ultimately need to get stuck in is a bunch of emulators and a stack of gaming ROMs.

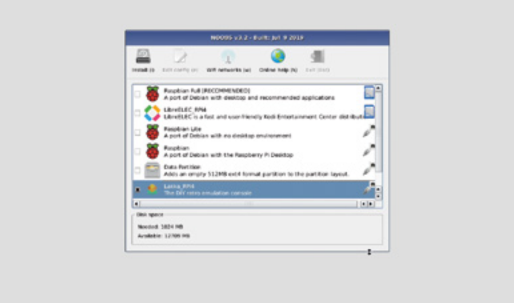
In the past, however, this has also entailed finding and downloading the BIOSes of various machines and a fair bit of configuration. Fortunately, with the software platform Lakka installed on your Raspberry Pi 4, the path to gaming glory is much smoother these days.

Lakka allows you to emulate arcade games as well as titles originally released on a host of 8-bit, 16-bit, and even 32- and 64-bit systems.

Lakka is a Linux operating system (OS) based on RetroArch (**retroarch.com**). Lakka is designed to run games, and it turns a Raspberry Pi into a powerful games system.

You can hook up a gamepad and even make use of wireless controllers (there's more about those at **magpi.cc/HpPSSV**). It has an interface that will be very familiar to anyone who has used modern games consoles and because it is open-source, it is constantly being improved.

You can run Lakka on any Raspberry Pi, although Raspberry Pi 4 enables smoother emulation of more recent consoles.



NOOBS v3.2 - Built: Jul 9 2019

□ Raspbian Full [RECOMMENDED]
A port of Debian with desktop and recommended applications

□ LibreELEC_RPi4
LibreELEC is a fast and user-friendly Kodi Entertainment Center distributi

□ Raspbian Lite
A port of Debian with no desktop environment

□ Raspbian
A port of Debian with the Raspberry Pi Desktop

□ Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

☒ Lakka_RPi4
The DIY retro emulation console

Disk space
Needed: 1024 MB
Available: 12709 MB

▲ NOOBS (New Out Of Box Software) is used to install operating systems such as Lakka on Raspberry Pi

Some features help you organise your growing gaming collection and take screenshots of the in-game action. For now, though, we're looking solely at getting you up and running with a classic homebrew video game.

01 Get SD Card Formatter

We're going to install Lakka RPI4 to a blank microSD card using the OS installer NOOBS (**magpi.cc/noobs**).

In this tutorial, we're using a Windows PC to format a microSD card and copy the NOOBS files to the card (the process is identical for Mac computers). We will then use the NOOBS card with our Raspberry Pi 4 and set up Lakka. From then on, our Raspberry Pi 4 will boot straight to Lakka and let us run games.

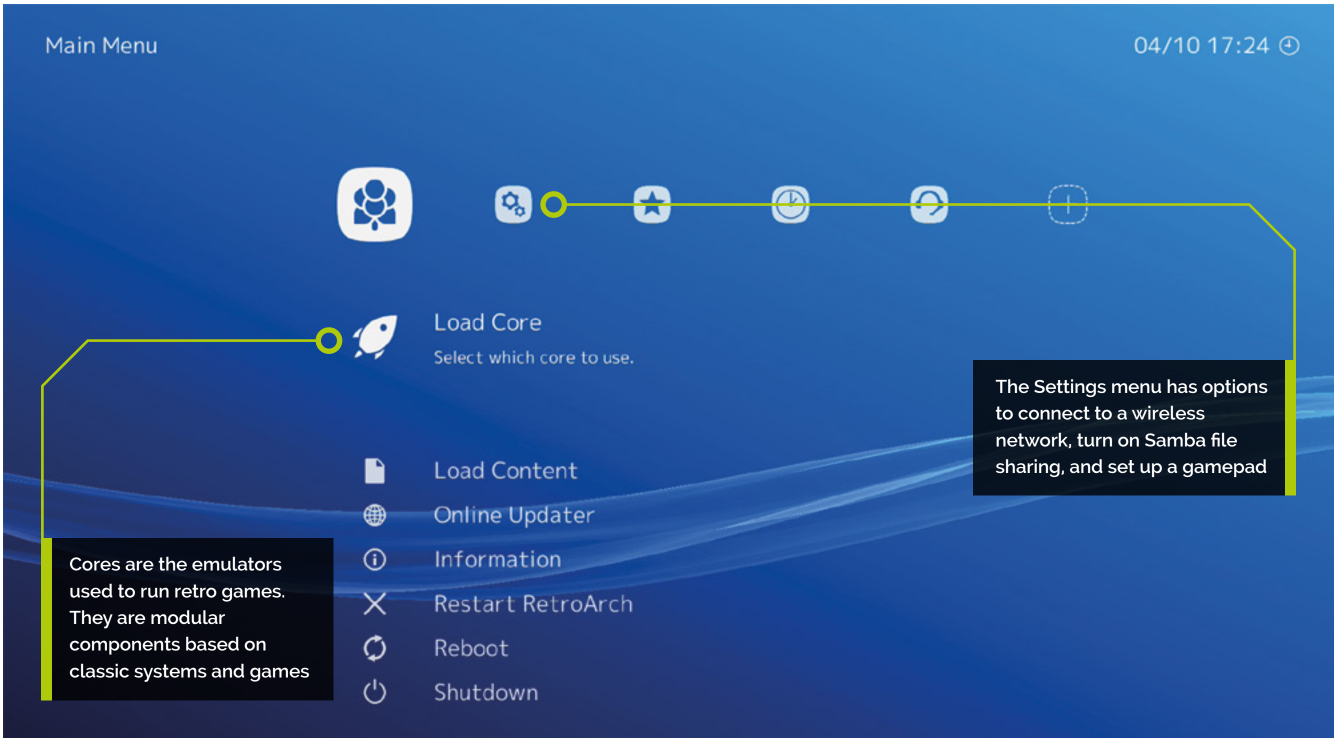
First, download SD Formatter on a computer from **magpi.cc/sdcardformatter**. Click 'For Windows' or 'For Mac' depending on your machine.

02 Format the card

We're now going to format the microSD card that you will use to boot Lakka on a Raspberry Pi. Note that this completely wipes the card, so make sure it contains nothing you need.

Insert the microSD card into your Windows or Mac computer. You will need to use either a USB SD card adapter or microSD card to SD card adapter.

Close any alert windows that appear, and open the SD Card Formatter app. Accept the terms and conditions and launch the program. On a Windows PC, click Yes to 'Do you want to allow this app to make changes to your device' (you won't see this on a Mac; the approval comes later).



Main Menu

04/10 17:24

The Settings menu has options to connect to a wireless network, turn on Samba file sharing, and set up a gamepad

Cores are the emulators used to run retro games. They are modular components based on classic systems and games

The card should be assigned a letter under Select Card. It is 'D' on our system. Check the Capacity and other details to ensure you have the correct card. Now click Format and Yes. On a Mac, you'll be asked to enter your Admin password.

03 Download NOOBS

Now visit **magpi.cc/downloads** and click the NOOBS icon. Select 'Download ZIP' next to NOOBS.

The latest version of the NOOBS zip file (currently **NOOBS_v3_2_1.zip**) will be saved to your **Downloads** folder.

Extract the files from the NOOBS zip file (right-click and choose Extract All and Extract). Now open the extracted NOOBS folder (it's important to ensure you are using the extracted files and not looking at the files inside the zip file. Make sure you have opened the **NOOBS_v3_2_1** folder and not the **NOOBS_v3_2_1.zip** file.

You should see three folders – **defaults**, **os**, and **overlays** – followed by many files beginning with 'bcm2708…'. It is these folders or files you need to copy to the microSD card.

Select all of the files inside the NOOBS folder and copy them to the microSD card. When the files have copied, eject and remove the microSD card from your PC or Mac.

04 Boot to NOOBS

Now set up your Raspberry Pi 4. You'll need to connect a USB keyboard and HDMI display for the installation process (you can remove the keyboard later and use just a game controller).

The display does not have to be the television you intend to use. It's best to use Raspberry Pi 4's HDMI 0 port. We're going to use a wireless LAN network to connect to the internet, but you can connect an Ethernet cable attached directly to your modem/router.

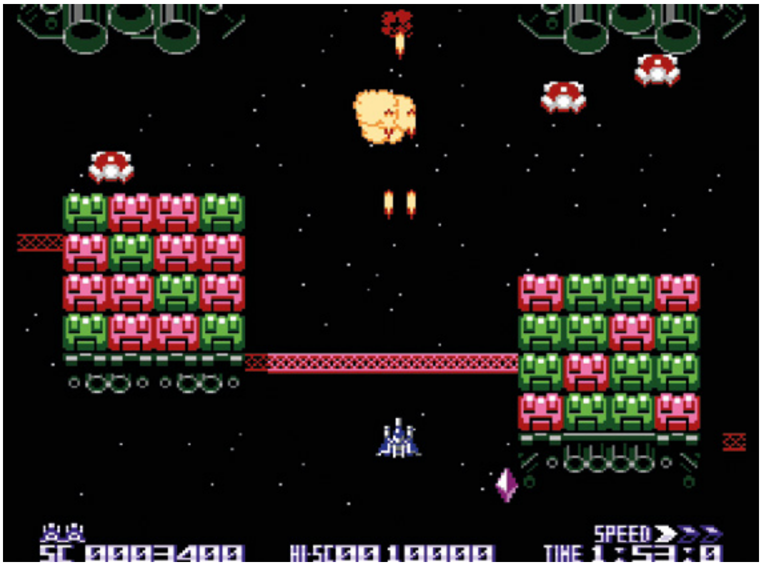
Insert the microSD card into your Raspberry Pi and attach the USB-C power supply to power up.

> “To get further installation options on the NOOBS screen, you will need to be connected to the internet”

05 Connect to wireless LAN

The NOOBS screen will appear, displaying two installation options: Raspbian Full and LibreELEC. To get further installation options, you will need to be connected to the internet.

Connect Raspberry Pi directly to your modem/router using an Ethernet cable; or click the 'Wifi networks (w)' icon. The WiFi network selection window appears; wait until it displays the local



Blade Buster, a homebrew shoot-'em-up, running on a Raspberry Pi 4

Warning!

It is illegal to download copyrighted game ROMs from the internet. Please respect the original maker and seek a legal source for retro gaming instead. We use homebrew ROMs made by modern makers for classic systems.

magpi.cc/legalroms

Top Tip

SSH

You can also use SSH to copy files from your computer to Raspberry Pi. In Lakka, enable SSH in Services. You can use a program such as FileZilla to copy files across. See **magpi.cc/ssh** for more information.

networks. Select your wireless network and enter the password for it in the Password field. Then click OK.

With Raspberry Pi connected to a network, you get a much broader range of installation options. Near the bottom will be Lakka_RPi4.

Use the arrow keys on your keyboard to select Lakka and press the **SPACE** bar to add a cross to its selection box (or use a connected mouse to select the Lakka option).

Click Install and answer Yes to the Confirm window. NOOBS will download and extract the Lakka file system to the microSD card. Sit back and wait for the system to be installed.

When it has finished, NOOBS will display 'OS(es) Installed Successfully'. Press **ENTER** on the keyboard (or click OK with the mouse).

06 Starting Lakka

Raspberry Pi will restart and this time it will boot into the Lakka operating system. You will see a blue screen with a series of windows and 'Load Core' will be highlighted. You can use the arrow keys on the keyboard to navigate the menu, and **X** to select a menu option, then **Z** to back up.

Highlight Load Core and press **X** to select it. Here you will find a list of 'cores'. These are the engines that emulate different retro consoles and computers.

To test the system is working, highlight 2048 and press **X** again. You'll be returned to the main menu, but this time you'll see 'Start Core'. Press **X** to start the core and you'll be presented with a classic game called 2048. Use the arrow keys to slide the blocks together. Matching numbers double in size, and the aim is to make a 2048 block. Press **ESC** and **ESC** again to return to the main Lakka menu.

07 Connect to the network

You need to connect Lakka to the network. Use your cursor keys to navigate Lakka's menus, and head to the Settings list. Press the down arrow and select 'Wi-Fi'. Wait for Lakka to scan the local networks.

Select your wireless LAN network and use the keyboard to enter the Passphrase. The Lakka interface will display the name of your wireless network with 'Online' next to it.

08 Get a game

Now it's time to find and play a game. Games are downloaded as ROM files and added to Lakka. These ROM files need a compatible core to run (most but not all ROM files will run correctly).

We'll use a Japanese homebrew ROM called Blade Buster. Download it on your PC or Mac from **magpi.cc/bladebuster** – click the 'Blade Buster Download' link.

A file called **BB_20120301.zip** will appear in your **Downloads** folder. Unlike NOOBS, you do not extract the contents of this file – ROMs are run as compressed zip files. You now need to transfer this file from your computer to your Raspberry Pi.

09 Turn on Samba

With your Raspberry Pi and PC on the same network, go to the Settings menu in Lakka on your Raspberry Pi and select Services. Highlight Samba and turn it on by pressing **X** (or using right arrow).

Samba is installed by default on macOS and used to be installed by default in Windows, but it has recently become an optional installation.

In Windows 10, click on the Search bar and type 'Control Panel'. Click on Control Panel in the search results. Now click 'Programs' and 'Turn Windows features on or off'. Scroll down to find 'SMB 1.0/CIFS File Sharing Support' and click the '+' expand icon to reveal its options. Place a check in the box marked 'SMB 1.0/CIFS Client'. Click OK. This will enable Samba client support on your Windows 10 PC so it can access Raspberry Pi.

10 Transfer the ROM

Lakka may appear in the left-hand column of your other computer's file browser (File Explorer on a PC or Finder on a Mac).

WIN! RASPBERRY PI ROBOT ARM KITS
The MagPi
The official Raspberry Pi magazine
Issue 88 | December 2019 | magpi.cc
RETRO COMPUTING
WITH RASPBERRY PI 4
Play retro games
Rediscover classic coding

If not, select Lakka's main menu on your Raspberry Pi, then choose Information and Network Information.

Take note of the IP address. Enter that into the File Explorer using the format:

\\insert.full.ip.address\

Ours, for example, is:

\\192.168.0.13\

Copy the Blade Buster zipped game to the **ROMS** folder on Lakka.

Back on your Raspberry Pi, go to Load Content > Start Directory in the Lakka menu and find the **BB_20120301.zip** file. Click it before selecting Load Archive. Choose FCEUmm as the core to play it on.

Press **ENTER** to start the game. Use the arrow keys to move and **X** to fire. Enjoy playing the game. Press **ESC** twice when you're done, to return to Lakka.

11 Set up a controller

Video game consoles rarely come with keyboards. And no doubt you'll want to attach a controller to your console.

If using a wireless gamepad, insert its dongle into one of Raspberry Pi's USB ports, insert the batteries, and turn it on. Press the Start button on the gamepad and it will light up.

Use the arrow keys to choose Input and User 1 Binds. If it is connected correctly, you will see 'RetroPad' next to User 1 Device Type. Scroll down and choose User 1 Bind All. Follow the on-screen instructions to press the buttons and move the analogue sticks on the gamepad. You may have to go through it a few times to get the process right.

You can also set each button individually using the options. Once everything is set up correctly, you'll be able to use the gamepad to control your Raspberry Pi console.

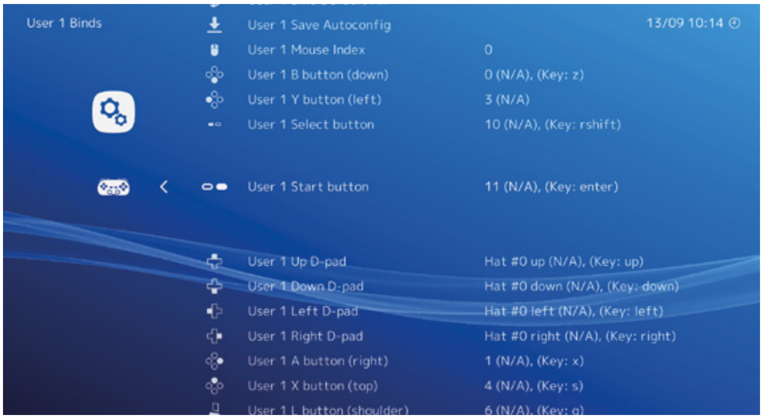
12 Move to the television

Your Raspberry Pi games console is now ready to be moved to your television. You will be able to control the games console using your USB or wireless controller and move ROM files directly to it from your Windows PC or Mac computer. There's a lot more to Lakka to discover, but for now we hope you enjoy playing retro games on your Raspberry Pi console. M

Top Tip

Ask for help

It's worth heading over the Lakka forums for friendly help and advice: **magpi.cc/lakkaforum**



User 1 Binds   13/09 10:14

User 1 Save Autoconfig	
User 1 Mouse Index	0
User 1 B button (down)	0 (N/A), (Key: z)
User 1 Y button (left)	3 (N/A)
User 1 Select button	10 (N/A), (Key: rshift)
User 1 Start button	11 (N/A), (Key: enter)
User 1 Up D-pad	Hat #0 up (N/A), (Key: up)
User 1 Down D-pad	Hat #0 down (N/A), (Key: down)
User 1 Left D-pad	Hat #0 left (N/A), (Key: left)
User 1 Right D-pad	Hat #0 right (N/A), (Key: right)
User 1 A button (right)	1 (N/A), (Key: x)
User 1 X button (top)	4 (N/A), (Key: s)
User 1 L button (shoulder)	6 (N/A), (Key: q)

Match the buttons and sticks on a gamepad to the controls used in each core

Part 08

Dialogs with C and GTK

Give users information and ask them questions using dialogs, including some ready-made ones for frequently used functions

MAKER

Simon Long

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

rpf.io

If we want to ask the user a question, or to inform them of something, the best way to do this is with a dialog box. GTK makes it easy to create dialogs – a GtkDialog can contain any GTK widgets, so can be as simple or as complex as you need.

The characteristic which distinguishes a dialog from a window in GTK is that a dialog interrupts the operation of the application – once a dialog is shown, the rest of the application waits until the dialog is closed.

Dialogs all have one or more buttons (typically with functions like 'OK' and 'Cancel') in what is called the action area at the bottom of the dialog; these close the dialog and return control to the main window. Each button in the action area generates a different return code, which is passed back from the dialog to give the user's response.

Here's the code for a simple dialog – connect this to the `clicked` signal on a button with `g_signal_connect`, and pass a pointer to the main window of your application as the general-purpose pointer in the `g_signal_connect` function:

```
void open_dialog (GtkWidget *wid,
    gpointer ptr)
{
  GtkWidget *dlg =
      gtk_dialog_new_with_buttons (
      "My dialog", GTK_WINDOW (ptr),
      GTK_DIALOG_MODAL |
      GTK_DIALOG_DESTROY_WITH_PARENT,
      "Cancel", 0, "OK", 1, NULL);

  int result = gtk_dialog_run (
      GTK_DIALOG (dlg));
  gtk_widget_destroy (dlg);
  printf ("Return code = %d\n, result");
}
```

The `gtk_dialog_new_with_buttons` function takes a number of arguments. The first is the text to be displayed on the title bar of the dialog. The second is a pointer to the main window of the application – this is why a pointer to the main window should be supplied to the `g_signal_connect` function which links this handler to a button's `clicked` signal.

The third argument is a set of flags which control the behaviour of the dialog; in this case we set it to be a *modal* dialog and set it to be destroyed along with its parent window. Making a dialog modal means that the parent window will be locked

Figure 1

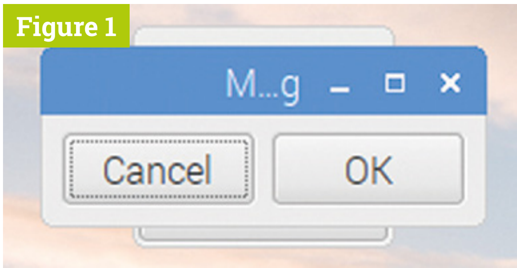
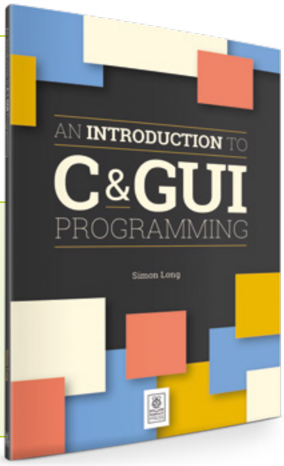


Figure 1 A simple GtkDialog, with two buttons in the action area

An Introduction to C & GUI Programming

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

magpi.cc/GUIbook



while the dialog is on display – it forces the user to close the dialog before they can continue, which is usually the desired behaviour. Making the dialog be destroyed along with its parent is just for tidiness – it means that if the main window of the application closes for some reason while the dialog is on display, then the dialog will also close. (It is quite rare to create a dialog without setting it to be modal and destroyed with its parent.)

The remaining arguments control which buttons are displayed in the action area of the dialog. This is a list of paired values; the first of each pair is the label to be displayed on a button, and the second is the return code which that button will generate when clicked. The list is terminated with a `NULL` value.

Once we have created our dialog, we call `gtk_dialog_run` – this displays the dialog and allows the user to interact with it. They will be able to operate any widgets on the dialog, but the main window will be locked while they do so, and execution of code waits inside the `gtk_dialog_run` function until a button in the action area is clicked. At this point, you need to call `gtk_widget_destroy` on the dialog to get rid of it – you might have expected clicking a button to remove the dialog, but this doesn't happen; the dialog stays on screen until it is explicitly removed.

> “You might have expected clicking a button to remove the dialog, but this doesn't happen”

Build and run the code – when you click the button which calls this handler, the dialog should be displayed (**Figure 1**).

While the dialog is on screen, clicking any other controls on the main window will have no effect. When you close the dialog by clicking either the 'OK' or 'Cancel' buttons, you will see a message showing the return code for the button you clicked in the terminal window from which you launched the application.

That's a basic dialog, but it's a bit small and bare! Let's add a label to it to ask the user a question. To do this, we have to access the dialog's *content* area – effectively, a dialog is a GtkWindow which holds a GtkVBox with two GtkContainers in it – one is the action area at the bottom which holds the buttons, and the other is the content area at the top. We add any additional content to the dialog in the content area, which can be accessed using the function `gtk_dialog_get_content_area`. Modify the handler above like this:

Figure 2

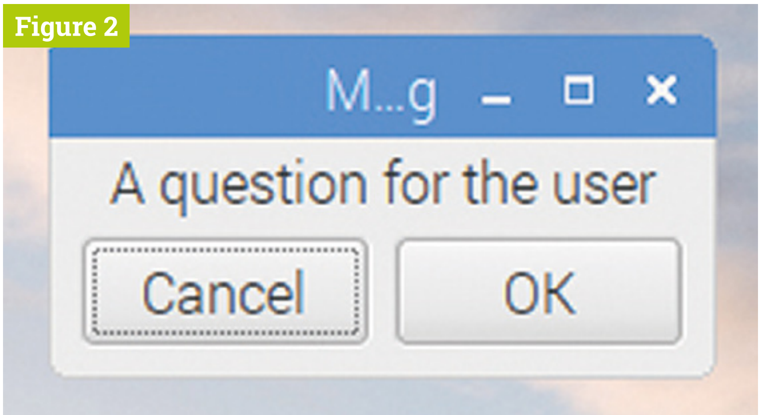


Figure 2 A GtkDialog with a label added to the content area

```
void open_dialog (GtkWidget *wid,
    gpointer ptr)
{
  GtkWidget *dlg =
      gtk_dialog_new_with_buttons (
      "My dialog", GTK_WINDOW (ptr),
      GTK_DIALOG_MODAL |
      GTK_DIALOG_DESTROY_WITH_PARENT,
      "Cancel", 0, "OK", 1, NULL);

  GtkWidget *lbl = gtk_label_new (
      "A question for the user");

  gtk_container_add (GTK_CONTAINER (
      gtk_dialog_get_content_area (
      GTK_DIALOG (dlg))), lbl);
  gtk_widget_show (lbl);

  int result = gtk_dialog_run (
      GTK_DIALOG (dlg));
  gtk_widget_destroy (dlg);
  printf ("Return code = %d\n", result);
}
```

We've used the familiar `gtk_container_add` function to add our label to the content area of the dialog. Note that we also needed to call `gtk_widget_show` on the label we added – `gtk_dialog_run` automatically shows the widgets which are a part of the dialog itself, like the background and the buttons, but you need to explicitly show everything else you add (**Figure 2**).

As with the main application window, if we want to add more than one widget to the content area, we first need to add a box or a table, and then to put the widgets into that.

Built-in dialogs

There are some common dialog boxes which are used in many desktop applications; for example, for choosing a file name to load or save, or for selecting a colour. GTK includes ready-made versions of these common dialog boxes which can easily be included in an application without needing to create every aspect of the dialog from scratch.

For most of these dialogs, there is a GTK button which launches the dialog; the easiest way to include the dialog is to include the appropriate button in your application, and everything is then done for you.

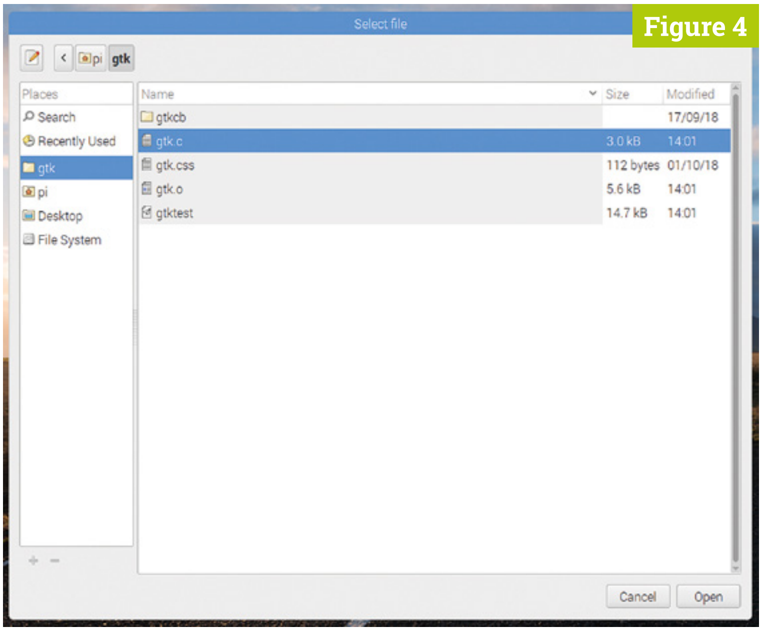
File chooser dialogs

Let's look at an example of a file chooser dialog, used to get the name and path of a file which can then be used in a subsequent file read operation.

```
static void file_selected (
    GtkFileChooserButton *btn, gpointer ptr)
{
  printf ("%s selected\n",
      gtk_file_chooser_get_filename (
      GTK_FILE_CHOOSER (btn)));
}

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

Figure 4 The dialog opened from a GtkFileChooserButton to select an existing file

Figure 4



Name	Size	Modified
gtkcb		17/09/18
gtk.c	3.0 kB	14:01
gtk.css	112 bytes	01/10/18
gtk.o	5.6 kB	14:01
gtktest	14.7 kB	14:01

Figure 3

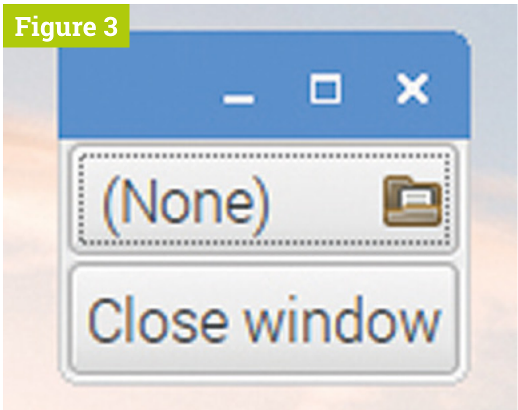


Figure 3 A GtkFileChooserButton

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

  GtkWidget *vbox = gtk_vbox_new (FALSE, 5);
  gtk_container_add (GTK_CONTAINER (win),
      vbox);

  GtkWidget *fc_btn =
      gtk_file_chooser_button_new (
      "Select file",
      GTK_FILE_CHOOSER_ACTION_OPEN);
  g_signal_connect (fc_btn, "file-set",
      G_CALLBACK (file_selected), NULL);

  gtk_box_pack_start (GTK_BOX (vbox),
      fc_btn, TRUE, TRUE, 0);
  gtk_box_pack_start (GTK_BOX (vbox), btn,
      TRUE, TRUE, 0);
  gtk_widget_show_all (win);
  gtk_main ();
}
```

A widget of type GtkFileChooserButton is created, and added to the window. The `gtk_file_chooser_button_new` function takes two arguments; the first is the title to apply to the file chooser window when it is opened, and the second determines what the file chooser window will do. In this case, we want to open an existing file, so `GTK_FILE_CHOOSER_ACTION_OPEN` is used. (The alternative is to select an existing folder, for which the argument would be `GTK_FILE_CHOOSER_ACTION_SELECT_FOLDER`.)

The `file-set` signal is connected to the button – this is called when the user makes a selection – and in the handler for that signal, the `gtk_file_chooser_get_filename` function is called to read back the name of the file selected.

When this code is run, you will see a window which looks like **Figure 3**.

The folder icon at the right-hand side of the top button indicates that it is a file chooser; the title of the button is the name of the currently selected file, which is '(None)' when the application is first run. If you click the button, a file chooser dialog will open (**Figure 4**).

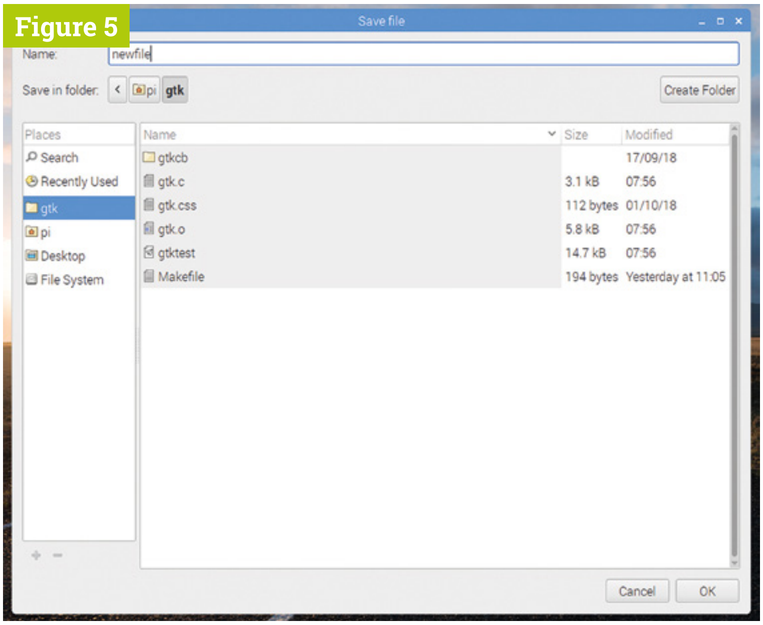
This provides a standard file browser; because we opened the window in 'file open' mode, it will only allow you to select a file that already exists on the file system. Use the browser to choose a file, and click 'Open'; when the window closes, the title of the button will update to show the selected file, and the full file path of the selected file will be printed to the terminal window from which you launched the application.

You can use a GtkFileChooserButton to open a browser which will select an existing file or folder, but the designers of GTK decided that you can't use this method to choose the location to which a new file can be saved. For that, you need to create a GtkFileChooser dialog yourself.

Modify the code above as follows:

```
static void save_file (GtkWidget *btn,
  gpointer ptr)
{
 GtkWidget *sch =
   gtk_file_chooser_dialog_new (
   "Save file", GTK_WINDOW (ptr),
   GTK_FILE_CHOOSER_ACTION_SAVE,
   "Cancel", 0, "OK", 1, NULL);
 if (gtk_dialog_run (
   GTK_DIALOG (sch)) == 1)
 {
  printf ("%s selected\n",
    gtk_file_chooser_get_filename (
    GTK_FILE_CHOOSER (sch)));
 }
 gtk_widget_destroy (sch);
}

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

Figure 5



Name: newfile

Name	Size	Modified
gtkcb		17/09/18
gtk.c	3.1 kB	07:56
gtk.css	112 bytes	01/10/18
gtk.o	5.8 kB	07:56
gtktest	14.7 kB	07:56
Makefile	194 bytes	Yesterday at 11:05

Figure 6

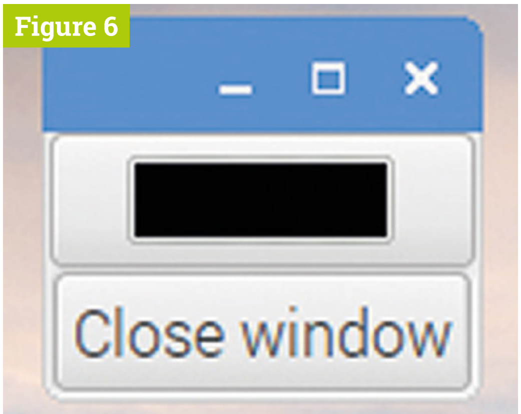


Figure 5 A GtkFileChooserDialog to select a new file

Figure 6 A GtkColorButton

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

GtkWidget *vbox = gtk_vbox_new (FALSE, 5);
gtk_container_add (GTK_CONTAINER (win),
  vbox);

GtkWidget *fc_btn =
  gtk_button_new_with_label (
  "Select file");
g_signal_connect (fc_btn, "clicked",
  G_CALLBACK (save_file), win);

gtk_box_pack_start (GTK_BOX (vbox),
```

Figure 7

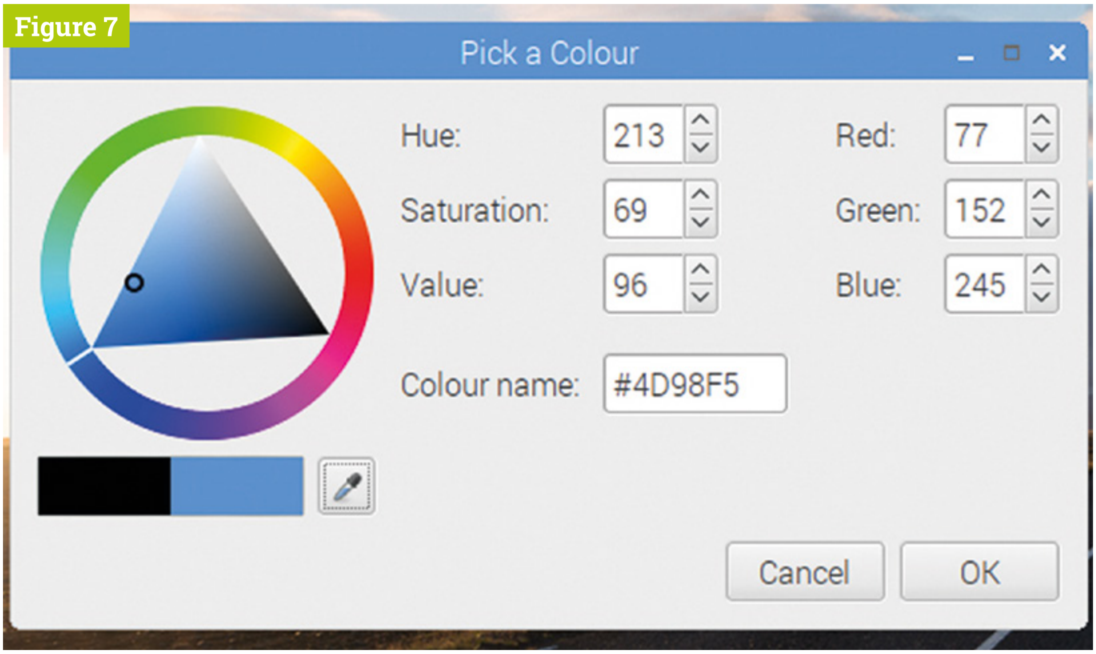


Figure 7 The colour picker dialog launched by a GtkColorButton

```
        fc_btn, TRUE, TRUE, 0);
    gtk_box_pack_start (GTK_BOX (vbox), btn,
        TRUE, TRUE, 0);
    gtk_widget_show_all (win);
    gtk_main ();
}
```

In this case, we create a button to open the dialog, and then have to manually create and open the dialog in the button handler.

The dialog is created by calling `gtk_file_chooser_dialog_new`, whose arguments are very similar to those for `gtk_dialog_new_with_buttons` that we saw in the previous chapter. First, the title of the dialog is provided, followed by a pointer to the parent window and then a flag which determines the behaviour of the dialog – in this case, it is set up to provide a file name and path to which a file can be saved. The remaining arguments are a `NULL`-terminated list of pairs of labels and return values for the buttons at the bottom of the dialog.

We create the dialog and call `gtk_dialog_run` when the button is pressed; the button handler then waits for the dialog to return. We then read the file path back from the file chooser using `gtk_file_chooser_get_filename`, as we did for the file open dialog.

If you run this code and press the 'Save file' button, you'll see that a file save chooser is slightly different from a file open chooser, in that it has a box to allow a new file name to be entered (**Figure 5**, previous page).

Enter a new file name and select a location for the new file; when you press 'OK', the path to the new file will be printed to the terminal.

There are several other predefined dialogs which can be added to an application just by including a button. Two of the more useful ones are the colour picker and the font chooser.

Figure 8

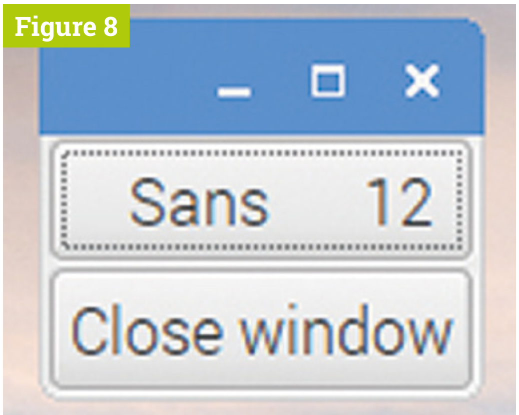


Figure 8 A GtkFontButton

Colour picker

Sometimes you need to allow the user to select a colour – to determine how things will be highlighted, for example. This is easy to do with the GtkColorButton. (Note that whenever GTK refers to 'colour', it does so using the US spelling, without the 'u' – it's a common source of compile-time errors for those of us on the East side of the Atlantic!)

Colours in GTK are stored as GdkColor data structures, which contain separate values for the red, green, and blue components of a colour. The GtkColorButton therefore operates on data stored as a GdkColor.

Here's an example of using a GtkColorButton:

```
static void col_selected (
    GtkColorButton *btn, gpointer ptr)
{
  GdkColor col;
  gtk_color_button_get_color (btn, &col);
  printf ("red = %d; green = %d; blue =
      %d\n", col.red, col.green, col.blue);
}

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

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

  GtkWidget *vbox = gtk_vbox_new (FALSE, 5);
  gtk_container_add (GTK_CONTAINER (win),
      vbox);

  GtkWidget *col_btn = gtk_color_button_new ();
  g_signal_connect (col_btn, "color-set",
      G_CALLBACK (col_selected), NULL);

  gtk_box_pack_start (GTK_BOX (vbox),
      col_btn, TRUE, TRUE, 0);
  gtk_box_pack_start (GTK_BOX (vbox), btn,
      TRUE, TRUE, 0);
  gtk_widget_show_all (win);
  gtk_main ();
}
```

We create the colour picker button in the same way as the file chooser button, and we connect to the `color-set` signal on it to detect when the user has made a selection. The handler for this signal calls `gtk_color_button_get_color` to read the selected value back into a GdkColor structure, and then prints out the red, green, and blue values.

If you run this, you'll see that the colour button shows a small rectangle of the currently selected colour. (**Figure 6**, previous page).

When clicked, the colour picker dialog is shown, which allows a colour to be set in various ways; either by clicking on the triangular colour map, by entering RGB or HSV values, or by picking a colour from somewhere on the screen using the dropper tool (**Figure 7**).

Figure 9

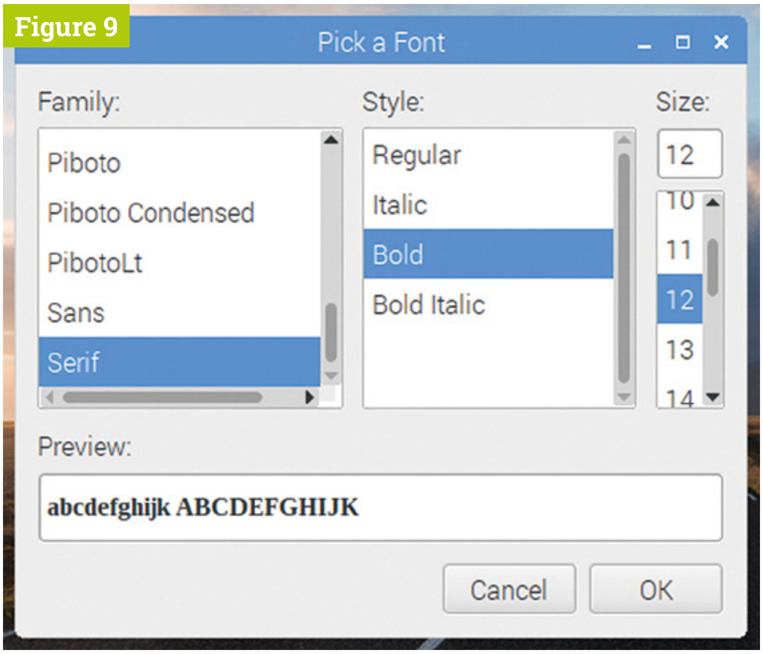


Figure 9 The font selector dialog launched by a GtkFontButton

When you click 'OK', the selected red, green, and blue values will be printed to the terminal.

Font selector

Another common operation is to select a font; this is used in many office applications, for example. GTK provides a font selector which works in the same way as the colour picker. Just change the GtkColorButton in the example above for a GtkFontButton:

```
  GtkWidget *fnt_btn = gtk_font_button_new ();
  g_signal_connect (fnt_btn, "font-set",
      G_CALLBACK (fnt_selected), NULL);
```

And create the handler for the `font-set` signal:

```
static void fnt_selected (
    GtkFontButton *btn, gpointer ptr)
{
  printf ("font = %s\n",
      gtk_font_button_get_font_name (btn));
}
```

When run, a font button is shown with the currently selected font name (**Figure 8**).

Pressing the font button opens the selector dialog, which allows you to pick any of the fonts currently installed on the system (**Figure 9**).

When you click 'OK', the selected font name and size will be printed to the terminal.

Part 06

Polish and release your PICO-8 game

Time to go gold! Let's apply some final finishing touches to our retro space shooter, and release it into the wild

MAKER

Dan Lambton-Howard

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

@danhowardgames

This final tutorial is about the most valuable skill in all of game development, actually finishing something. It's the skill that turns a million-dollar idea into a million dollars and like any skill, the more you do it the easier it gets. Helpfully, the fantastic feeling of polishing, packaging and publishing your own game is truly addictive. With that in mind, we'll be applying some final polish and 'game feel' to our plucky little shooter, creating an eye-catching title screen and covering how to actually release the darn thing. What a rush!

01 Game feel

The first thing we are going to polish, is the 'game feel'. Sometimes also called 'juice', game feel refers to the unique sensation of interacting with a game. It is created by everything from the tactile press of physical buttons, to graphical and audio feedback, to the movement and interaction of objects in a game. Good game feel is always satisfying. Think of Sonic the Hedgehog bouncing on robots, the roadie run in Gears of War, or the colourful spray of matching three in Candy Crush. Game feel is an important part of polishing a game.

02 Screen shake

One reliable way of creating satisfying game feel is to add a little bit of screen shake. Screen shake is a great way of communicating impact to a player. It works by jerking or shaking the game's camera in response to events within the game. You'll often see it used for explosions or big collisions in games, but indie developers Vlambeer have taken the art of screen shake to a new level with titles such as Nuclear Throne and Luftrausers. Luckily for us, it is a relatively simple effect and easy to implement.

03 Shake, rattle and roll

Declare two new variables at the start: `shkx,shky = 0,0`. Add these to the update camera coordinates call: `camera(camx+shkx,camy+shky)`. Next, declare two new functions, one to add screen shake and the other to update and reduce it. Check the **part6code.p8** listing (overleaf) for how we did this. Add `update_shake()` to your update loop and now you can insert `add_shake(x)` at any point in your code where you want a bit more impact. We've sprinkled some when an enemy or the player is destroyed. Play around with the amount, but remember: a little shake goes a long way!

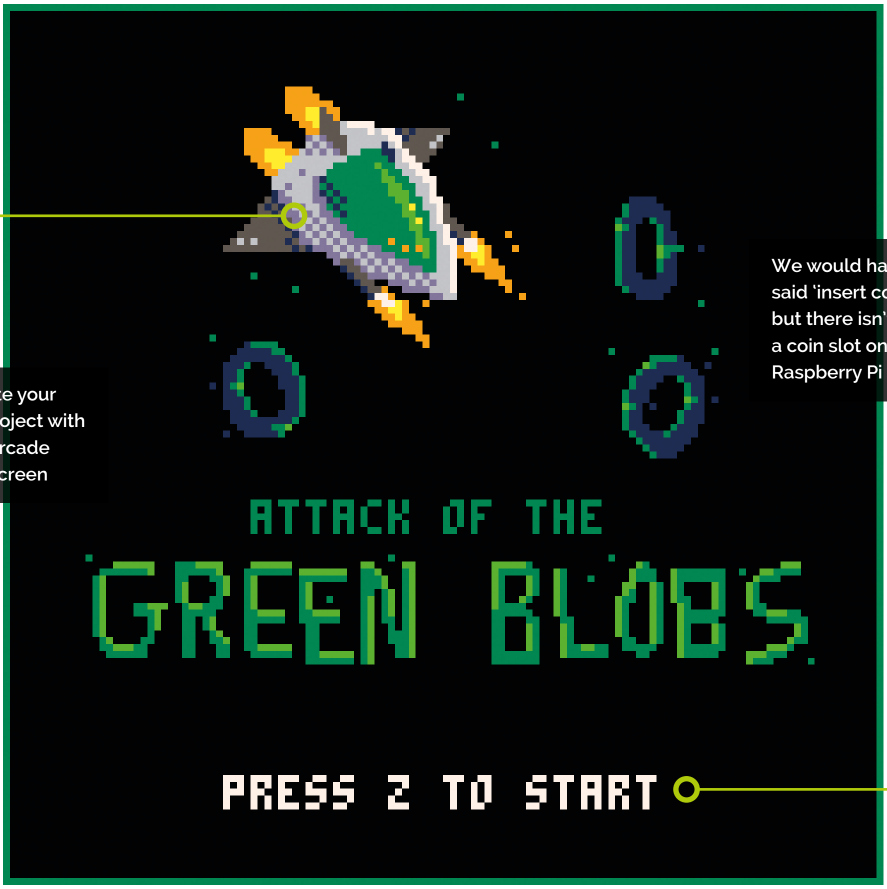
> "Screen shake is a great way of communicating impact to a player"

04 Little bits of dust

Next, we are going to add particle effects to our game. Particles are exactly what they sound like. They are small graphical elements that are used to represent things like dust, smoke, debris from explosions, and so on. As well as improving the look of the game, they are another way of giving feedback to the player to improve game feel. To add them to our game, we will be creating a simple particle engine to update them – and, much like we did for screen shake, adding new particles anywhere we deem cool enough.

You'll Need

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



Complete your game project with a retro arcade splash screen

We would have said 'insert coin', but there isn't a coin slot on Raspberry Pi (yet)

05 Particular particles

We need to create two new functions. See the code listing for details but essentially, we create new particles much like everything else, by adding it to a table. Is there anything tables in Lua can't do? Our update function loops through this table and advances each particle's age and colour so it looks darker the older it gets, until it eventually disappears. The real trick is where we add them to our game. We've created a particle trail from the player's ship and added a bit of code to our `create_explosion()` function that also spews out particles in random directions. Very cool!

06 Top and tail

The last thing we need to do to our game before release is to add an awesome retro title screen. Currently our game launches straight into battle, catching players by surprise and feeling a little unfinished. Let's fix this issue now so that it feels like a complete and polished game. To create our title screen, we can use all the unused room on our sprite sheet's second tab. Let's draw a cool sci-fi image that looks like something you would see blaring out of an arcade cabinet in the 1980s.

07 State of the game

We've gone for our iconic spaceship blasting through space, laser cannons flaring, and green blobs quivering in fear. It's a powerful and, dare we say it, moving image, and one that will definitely attract an eager line of kids queuing up to play it in our virtual arcade.

To implement it in game we create a new game state called title. Declare `title = true` in `_init()`. Then wrap the contents of `_update()` in a conditional so that it only runs if `title` is false. Do the same for `_draw()` as well.

08 Setting the screen

Now we need to actually add some code into our title state that will draw our image, and start the game if the player presses a button. Check the code listing for details, but it's nothing more complicated than drawing sprites and printing some text. We've also added a timer that makes it so the player has to wait a second before they can progress. We don't want them accidentally skipping our incredible artwork! To make it extra slick, we've added some new music and a chime SFX for the title screen.

Top Tip

Scratch an itch

As well as being a great place to host your game, **itch.io** also runs regular PICO-8 game jams.

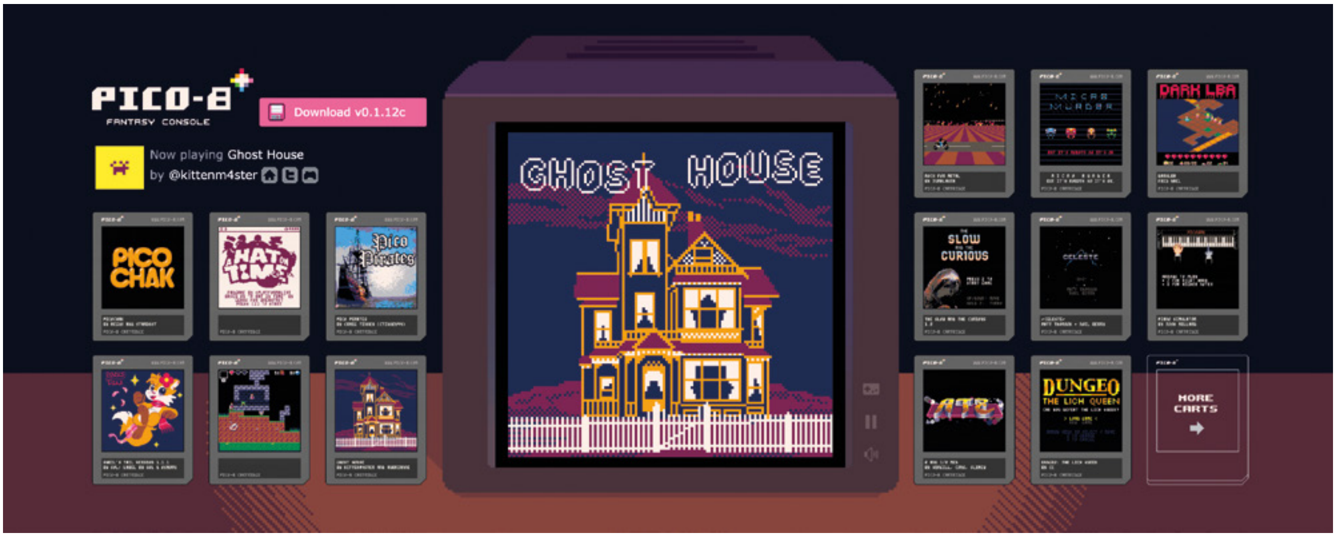
part6code.p8

DOWNLOAD THE FULL CODE:

magpi.cc/JbTKkP

Language: **Lua**

```lua
--new code reference
--init()
shkx,shky = 0,0 --screen shake variables
particles={} --table of particles
title = true --title state bool
timer = 0 --timer for states
music(16) --new title music

--update()
if title then
    timer+=1
    if btn(4) and timer>30 then
        timer=0
        title = false
        sfx(11)
        music(0)
    end
else
--reset of update
end

--in player update loop
if player.animtimer%2==0 then
    create_particle(player.x, player.y+3+rnd(2),-0.5,0)
end
--update screen shake
update_shake()

--sprinkle liberally...
add_shake(8)

--at end of if not gameover loop
if gameover then
    timer+=1
    if btn(4) and timer>30 then
        reload()--load all game data
        gameover=false
        _init() -- set game to initial state
    end
end

--draw()
if title then
    cls()
    camera(0,0)--reset camera
    --title gfx
    spr(64,32,12,6,4)
    spr(118,54,43)
    spr(72,86,28,2,2)
    spr(70,88,50,2,2)
    spr(104,30,48,2,2)
    print('attack of the',36,72,3)
    spr(74,12,80,6,2)
    spr(106,70,80,6,2)
    if timer>30 then
        print('press z to start',32,112,7)
    end
    rect(0,0,127,127,3)
else
--rest of draw code
end

--set camera shake
camera(camx+shkx,camy+shky)

--other functions
function add_shake(amount)
    local a=rnd(1)
    shkx+=amount*cos(a)
    shky+=amount*sin(a)
end

function update_shake()
    if abs(shkx)+abs(shky)<0.5 then
        shkx,shky=0,0
    else
        shkx*=-0.5-rnd(0.2)
        shky*=-0.5-rnd(0.2)
    end
end

function create_particle(x,y,vx,vy)
    local p={x=x,y=y,vx=vx,vy=vy,colour=10,age=0}
    add(particles,p)
end

function update_particles()
    for p in all(particles) do
        p.age+=1
        if p.age>15 then
            del(particles,p)
        elseif p.age>10 then
            p.colour=4
        elseif p.age>5 then
            p.colour=9
        end
        p.x+=p.vx
        p.y+=p.vy
        pset(p.x,p.y,p.colour)
    end
end
--in create_explosion()
--add particles to explosions
for i=1,10 do
    local a=rnd(1)
    local vx=cos(a)*rnd(2)
    local vy=sin(a)*rnd(2)
    create_particle(x,y,vx,vy)
end
```



▲ The Lexaloffle website is home of the cartverse, and the first place you should upload your finished PICO-8 game

09 Now for the tail

So that's the top done, but what about the tail? Currently when the player is defeated, they have to manually restart the game, which is not a good look. To polish this, we will add to our player update loop the option to restart the game by pressing a button. We do this by calling `_init()` and `reload()`, when the player presses a button during game over. We'll also reuse our timer from the title state to add a slight delay before players can input, so they don't miss the game-over message.

10 Enter the cartverse

Now that you have a pixel-polished piece of PICO-8 perfection, it's time to get it in front of the world and reap the adoration you deserve. The first port of call should always be uploading your game to the Lexaloffle website so that your game joins the cartverse and appears for everyone in SPLORE, PICO-8's built cartridge browser. To do this, navigate to the submit section (**magpi.cc/wwLy85**) and upload your .p8 file there. Now other PICO-8 peeps can play your game and comment on it!

▲ Particle effects from the ship and explosions give a sense of movement. The screen shake helps as well!

11 Pick a format

PICO-8 is like a Swiss Army knife when it comes to exporting. For example, create a standalone .bin for Raspberry Pi, Linux (64-bit), Windows and Mac. From the PICO-8 terminal, type `EXPORT YOURGAME.BIN` to create it. Another great place to upload is the indie game site **itch.io**, where you can post a web player version of your game. You can even set up to be paid through the site. To export for web, use `EXPORT YOURGAME.HTML` and you'll get a .html and a .js file. Rename the .html file to **index.html** and upload them both to **itch.io**.

> " PICO-8 is like a Swiss Army knife when it comes to exporting "

12 To a bright future

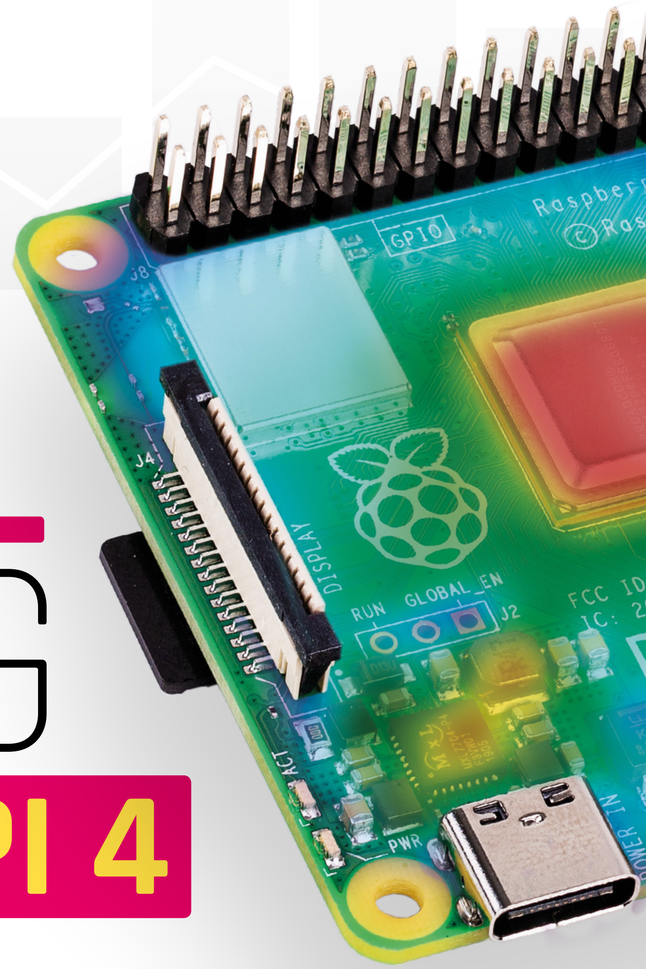
The quickest way to become a better game developer is to finish things and share them with people, and PICO-8 makes it easier than ever to do this. Raspberry Pi is the perfect platform for this little virtual console and has plenty of scope for expanding into arcade cabinets using the GPIO pins.

If you've followed these tutorials then let me know. I'd love to play your games and see what projects you have created. It's been an absolute pleasure to have been your guide over these last few tutorials. Keep in touch: ***@danhowardgames***

Top Tip

Tweet and greet

PICO-8 has a very active community on Twitter, use the hashtag #pico8 to post about your game.



THERMAL TESTING RASPBERRY PI 4

Raspberry Pi 4 just got a lot cooler! The last four months of firmware updates have taken over half a watt out of idle power and nearly a watt out of fully loaded power. By **Gareth Halfacree**

Raspberry Pi 4 launched with a wealth of new features to tempt users into upgrading: a more powerful CPU and GPU, more memory, Gigabit Ethernet, and USB 3.0 support.

More processing power means more electrical power, and Raspberry Pi 4 is the most power-hungry member of the family.

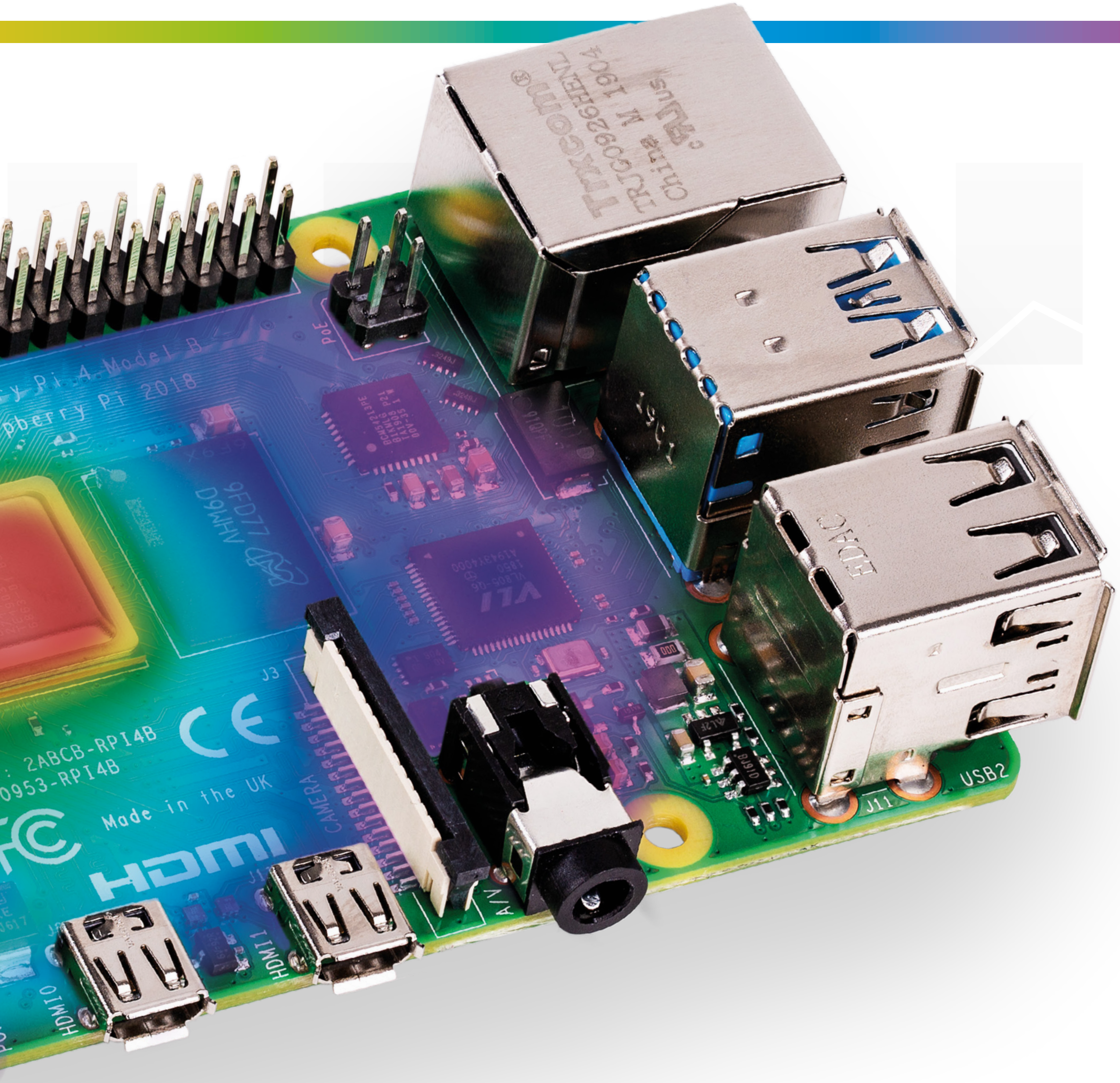
The launch of a new Raspberry Pi model is only the beginning of the story. Development is continuous, with new software and firmware improving each board long after it has rolled off the factory floor.

Raspberry Pi 4 is no exception: since launch, there have been a series of updates which have reduced its power needs and, in doing so, enabled it to run considerably cooler. These updates apply to any Raspberry Pi 4, whether you picked one up on launch day or are only just now making a purchase.

> “The launch of a new Raspberry Pi model is only the beginning of the story”

This feature takes a look at how each successive firmware release has improved Raspberry Pi 4, using a synthetic workload designed – unlike a real-world task – to make the system-on-chip (SoC) get as hot as possible in as short a time as possible.

Read on to see what wonders a simple firmware update can work.



How we tested

To test how well each firmware revision handles the heat, a power-hungry synthetic workload was devised to represent a worst-case scenario: the `stress-ng` CPU stress-testing utility places all four CPU cores under heavy and continuous load. Meanwhile, the `glxgears` tool exercises the GPU. Both tools can be installed by typing the following at the Terminal:

```
sudo apt install stress-ng mesa-utils
```

The CPU workload can be run with the following command:

```
stress-ng --cpu 0 --cpu-method fft
```

The command will run for a full day at default settings; to cancel, press **CTRL+C** on the keyboard.

To run the GPU workload, type:

```
glxgears -fullscreen
```

This will display a 3D animation of moving gears, filling the entire screen. To close it, press **ALT+F4** on the keyboard.

For more information on how both tools work, type:

```
man stress-ng
man glxgears
```

During the testing for this feature, both of the above workloads are run simultaneously for ten minutes. Afterwards, Raspberry Pi is allowed to cool for five minutes.

The thermal imagery was taken at idle, then again after 60 seconds of the `stress-ng` load alone.

Raspberry Pi firmware timeline

Launch firmware
(June 2019)

Finalised prior to production, the launch firmware is how every Raspberry Pi 4 rolled off the factory floor – fully functional, but not fully optimised.

VLI firmware
(July 2019)

Released as an early beta, then withdrawn following the discovery of a bug, the VLI firmware enabled power management in the USB 3.0 controller chip.

VLI, SDRAM firmware
(September 2019)

This release combines the benefits of the earlier, and now fixed, VLI firmware plus modifications to the way Raspberry Pi's LPDDR4 memory operates.

Clocking, and load-step firmware
(October 2019)

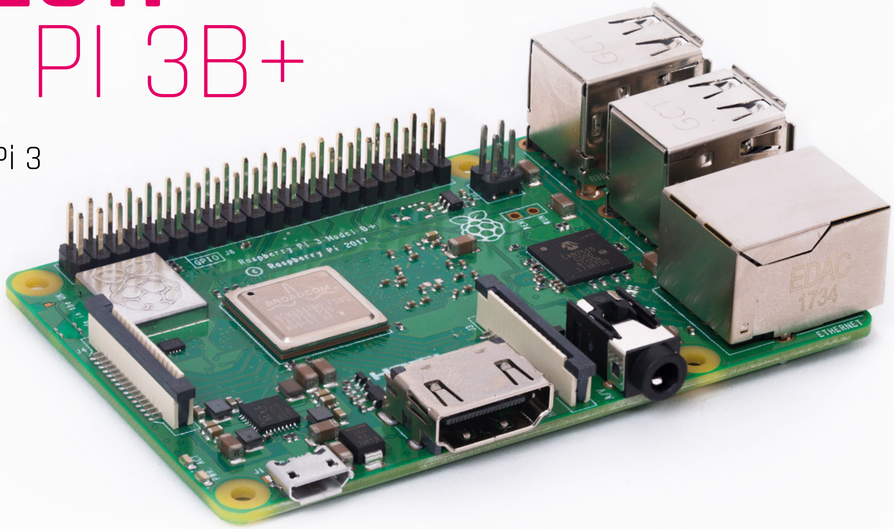
The latest public release at the time of writing, this firmware update improves how the system-on-chip can increase and decrease its clock speed and voltage.

Beta firmware
(out soon)

Due for public release soon, the beta firmware build includes power-saving tweaks to the system-on-chip operating voltage and the way HDMI video output works.

BASELINE TEST: RASPBERRY PI 3B+

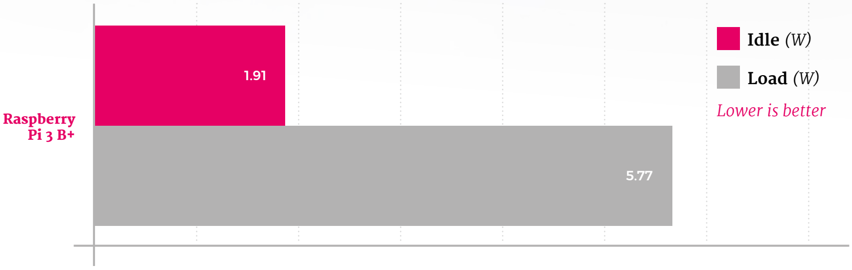
Already well established, Raspberry Pi 3 Model B+ was the device to beat



Before Raspberry Pi 4 came on the scene, Raspberry Pi 3 Model B+ was the must-have single-board computer. Benefiting from all the work that had gone into the earlier Raspberry Pi 3 Model B alongside improved hardware, Raspberry Pi 3B+ was – and still is – a popular device. Let's see how well it performs before testing Raspberry Pi 4.

Power Draw

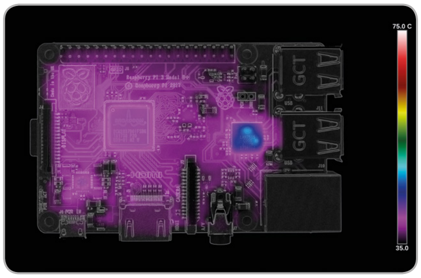
An efficient processor and an improved design for the power circuitry compared to its predecessor help keep Raspberry Pi 3B+ power draw down: at idle, the board draws just 1.91 W; when running the synthetic workload, that increases to 5.77 W.



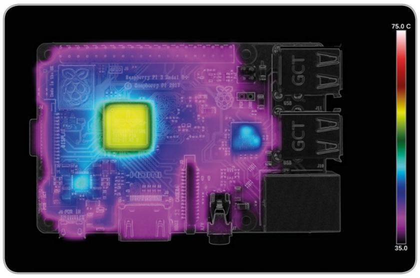
Thermal Imaging

A thermal camera shows where the power goes. At idle, the system-on-chip is relatively cool while the combined USB and Ethernet controller to the middle-right is a noticeable hot spot; at load, measured after 60 seconds of a CPU-intensive synthetic workload, the SoC is by far the hottest component at 58.1°C.

Raspberry Pi 3B+ heat distribution *(Idle)*

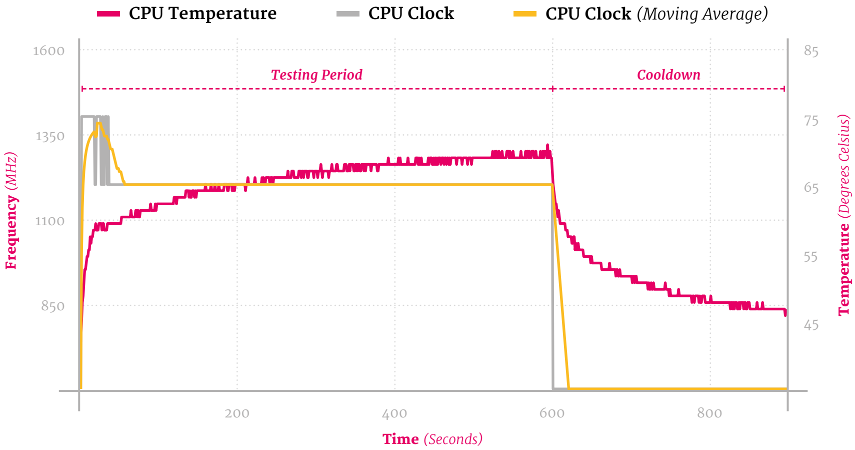


Raspberry Pi 3B+ heat distribution *(Load)*



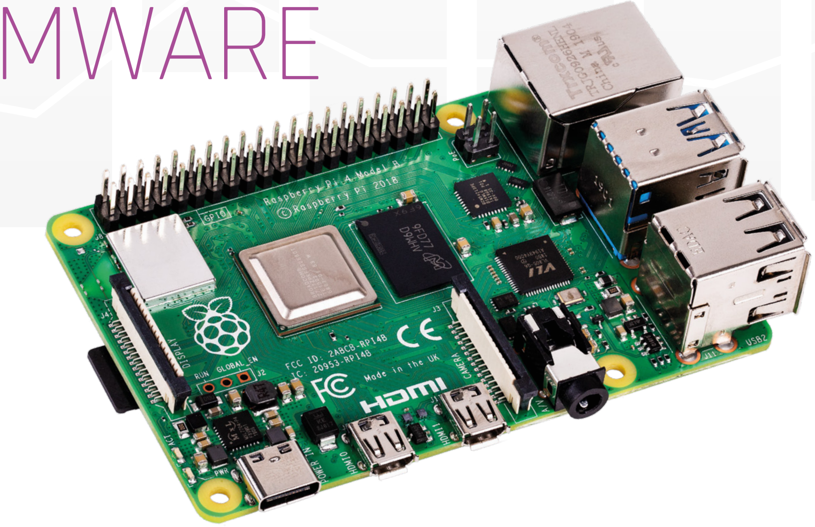
Thermal Throttling

This chart measures Raspberry Pi 3B+ CPU speed and temperature during a ten-minute power-intensive synthetic workload. The test runs on both the CPU and GPU, and is followed by a five-minute cooldown. Raspberry Pi 3B+ quickly reaches the 'soft throttle' point of 60°C, designed to prevent the SoC hitting the hard-throttle maximum limit of 80°C, and the CPU remains throttled at 1.2GHz for the duration of the benchmark run.



RASPBERRY PI 4 LAUNCH FIRMWARE

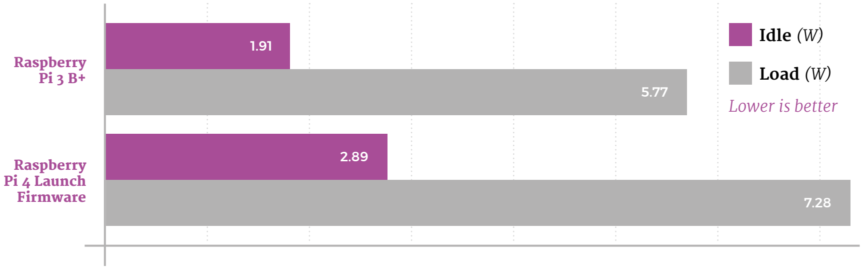
The fastest Raspberry Pi ever made demanded the most power



Raspberry Pi 4 Model B launched with a range of improvements over Raspberry Pi 3B+, including a considerably more powerful CPU, a new GPU, up to four times the memory, and USB 3.0 ports. All that new hardware came at a cost: higher power draw and heat output. So let's see how Raspberry Pi 4 performed at launch.

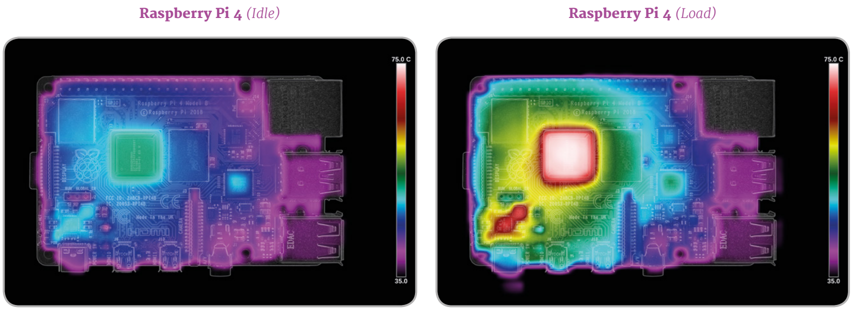
Power Draw

There's no denying it, Raspberry Pi 4 was a hungry beast at launch. Even idling at the Raspbian desktop, the board draws 2.89 W, hitting a peak of 7.28 W under a worst-case synthetic CPU and GPU workload – a hefty increase over Raspberry Pi 3 B+.



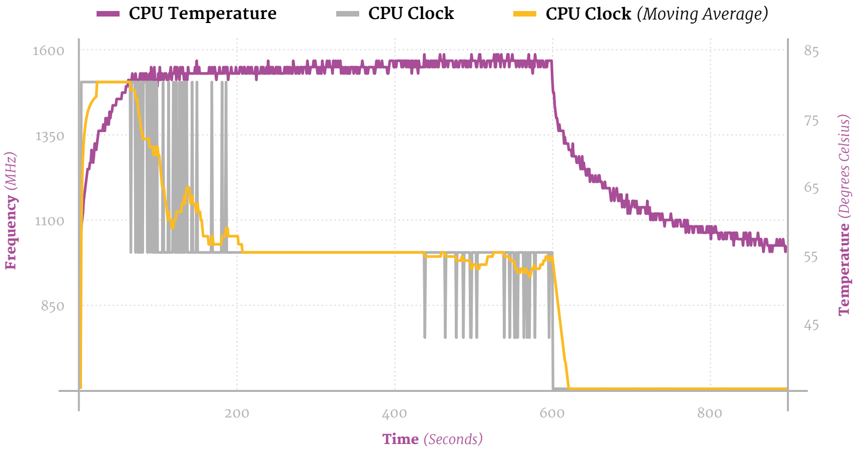
Thermal Imaging

Thermal imaging shows that Raspberry Pi 4, using the launch-day firmware, runs hot even at idle, with hot spots at the USB controller to the middle-right and power-management circuitry to the bottom-left. Under a heavy synthetic load, the SoC hits 72.1°C by the 60-second mark.



Thermal Throttling

Raspberry Pi 4 manages to go longer than Raspberry Pi 3 B+ before the synthetic workload causes it to throttle; but throttle it does after just 65 seconds. As the workload runs, the CPU drops from 1.5GHz to a stable 1GHz, then dips as low as 750MHz towards the end.



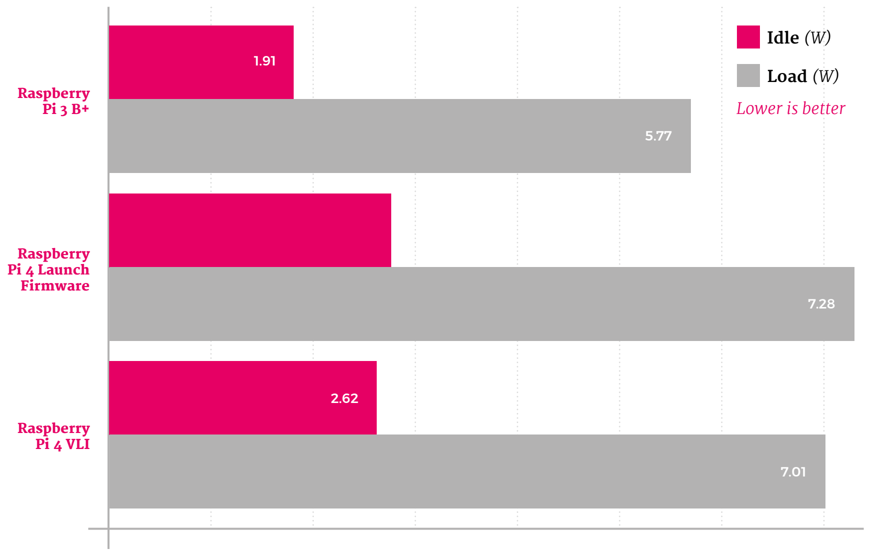
RASPBERRY PI 4 VLI FIRMWARE

USB power management brings some relief for Raspberry Pi heat

The first major firmware update developed for Raspberry Pi 4 brought power management features to the Via Labs Inc. (VLI) USB controller. The VLI controller is responsible for handling the two USB 3.0 ports, and the firmware update allowed it to run cooler.

Power Draw

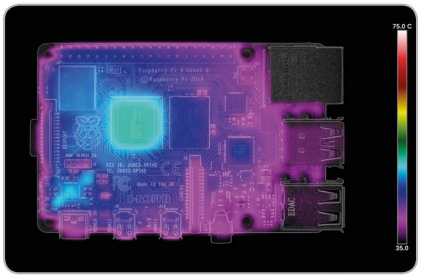
Even without anything connected to Raspberry Pi 4's USB 3.0 ports, the VLI firmware upgrade has a noticeable impact: idle power draw has dropped to 2.62 W, while the worst-case draw under a heavy synthetic workload sits at 7.01 W.



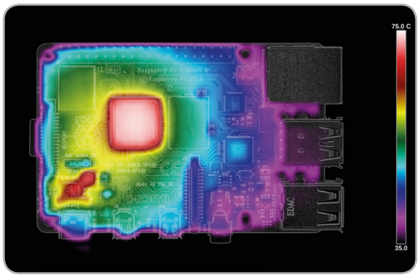
Thermal Imaging

The biggest impact on heat is seen, unsurprisingly, on the VLI chip to the middle-right, the VLI firmware helps keep the SoC in the centre and the power-management circuitry at the bottom-left cooler than the launch firmware. The SoC reached 71.4°C under load – a small, but measurable, improvement.

Raspberry Pi 4 VLI firmware *(Idle)*

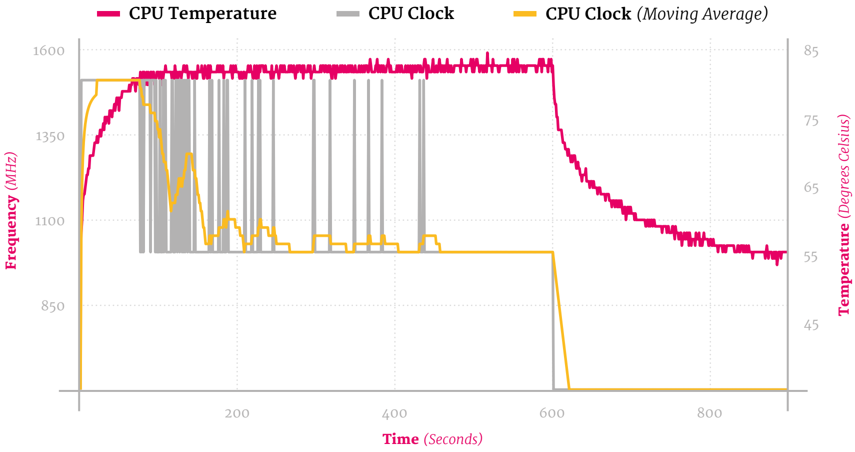


Raspberry Pi 4 VLI firmware *(Load)*



Thermal Throttling

Enabling power management on the VLI chip has a dramatic impact on performance in the worst-case synthetic workload: the throttle point is pushed back to 77 seconds, the CPU spends more time at its full 1.5GHz speed, and it doesn't drop to 750MHz at all. The SoC also cools marginally more rapidly at the end of the test.



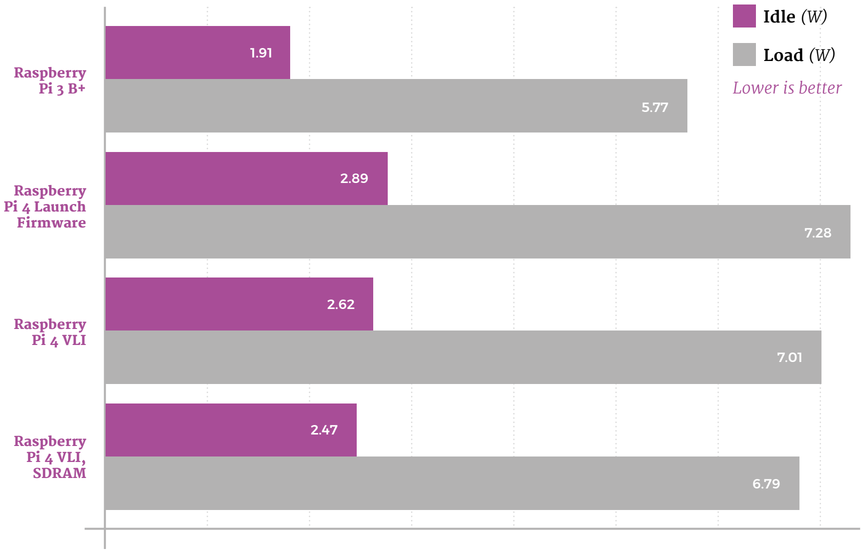
RASPBERRY PI 4 VLI, SDRAM FIRMWARE

With VLI tamed, it's the memory's turn now

The next firmware update, designed to be used alongside the VLI power management features, changes how Raspberry Pi 4's memory – LPDDR4 SDRAM – operates. While having no impact on performance, it helps to push the power draw down still further at both idle and load.

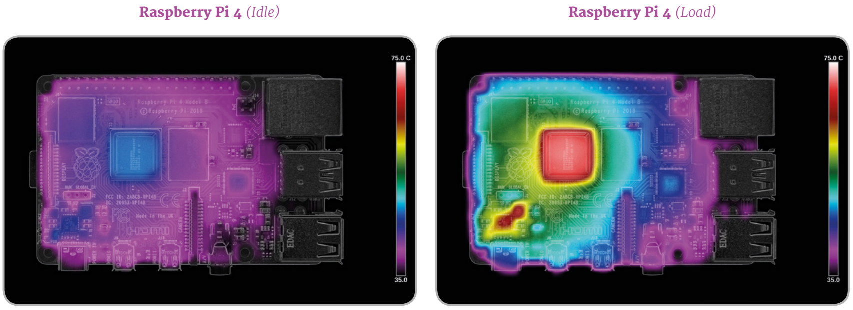
Power Draw

As with the VLI update, the SDRAM update brings a welcome drop in power draw at both idle and load. Raspberry Pi 4 now draws 2.47 W at idle and 6.79 W running a worst-case synthetic load – a real improvement from the 7.28 W at launch.



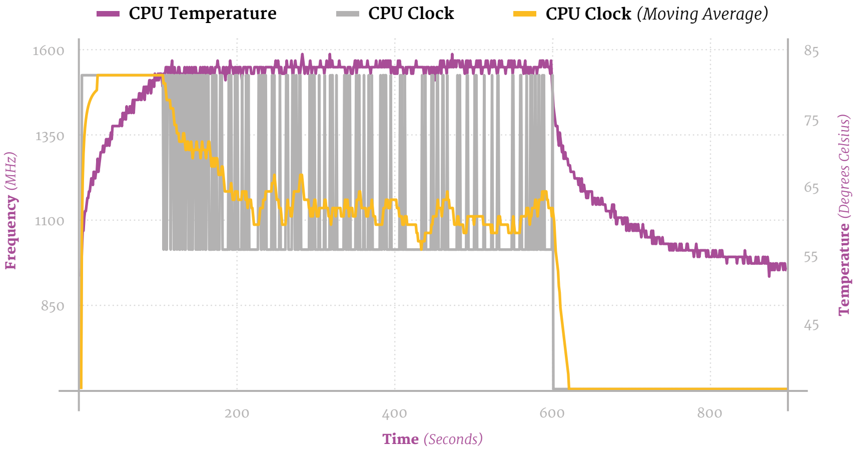
Thermal Imaging

Thermal imaging shows the biggest improvement yet, with both the SoC and the power-management circuitry running considerably cooler at idle after the installation of this update. After 60 seconds of load, the SoC is noticeably cooler at 68.8 °C – a drop of nearly 3 °C over the VLI firmware alone.



Thermal Throttling

A cooler SoC means better performance: the throttle point under the worst-case synthetic workload is pushed back to 109 seconds, after which Raspberry Pi 4 continues to bounce between full 1.5GHz and throttled 1GHz speeds for the entire ten-minute benchmark run – bringing the average speed up considerably.



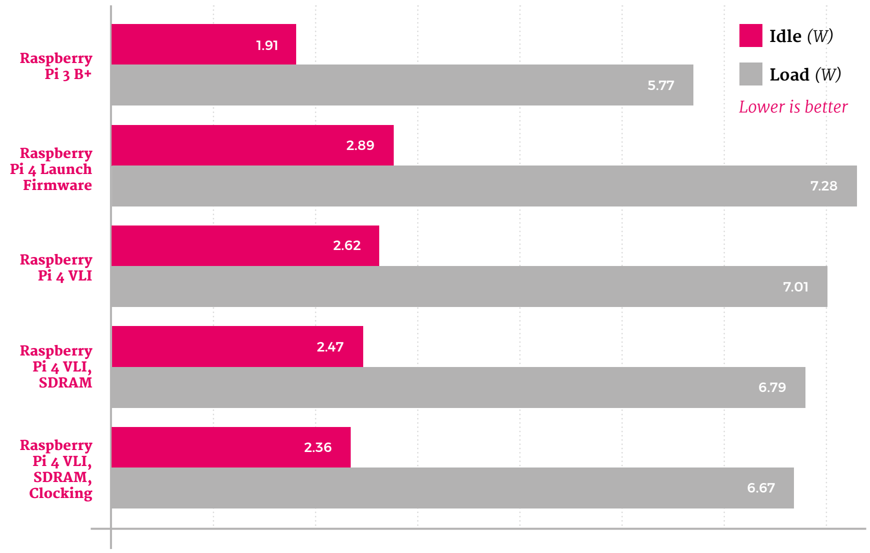
RASPBERRY PI 4 VLI, SDRAM, CLOCKING, AND LOAD-STEP FIRMWARE

September brought still more firmware improvements

September 2019's firmware update includes several changes, while bringing with it the VLI power management and SDRAM firmware updates. The biggest change is how the BCM2711B0 SoC on Raspberry Pi 4 increases and decreases its clock-speed in response to demand and temperature.

Power Draw

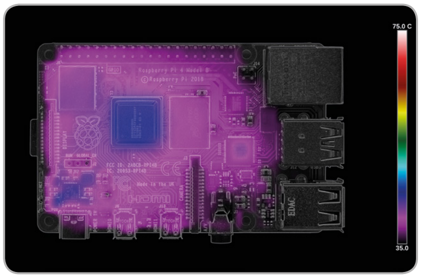
The September firmware update has incremental improvements: idle power draw is down to 2.36 W and load under the worst-case synthetic workload to a peak of 6.67 W, all without any reduction in raw performance or loss of functionality.



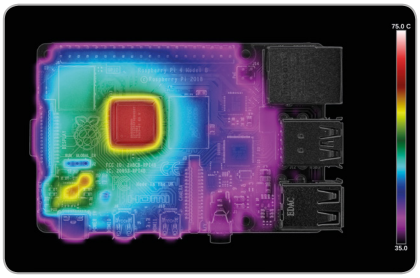
Thermal Imaging

Improved processor clocking brings a noticeable drop in idle temperature throughout the circuit board. At load, everything's improved – the SoC peaked at 65 °C after 60 seconds of the synthetic workload, while both the VLI chip and the power-management circuitry are clearly cooler than under previous firmwares.

Raspberry Pi 4 VLI, SDRAM, Clocking firmware *(Idle)*

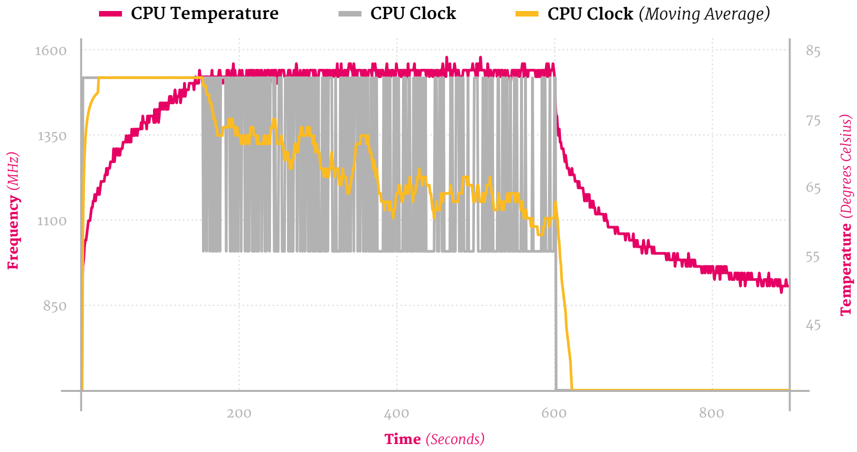


Raspberry Pi 4 VLI, SDRAM, Clocking firmware *(Load)*



Thermal Throttling

With this firmware, Raspberry Pi 4's throttle point under the worst-case synthetic workload is pushed back all the way to 155 seconds – more than double the time the launch-day firmware took to hit the same point. The overall average speed is also brought up, thanks to more aggressive clocking back up to 1.5GHz.



RASPBERRY PI 4 BETA FIRMWARE

Currently in testing, this beta release is cutting-edge

Get the latest firmware for Raspberry Pi

To upgrade your Raspberry Pi to the latest firmware, open a Terminal window and enter:

```
sudo apt update
sudo apt full-upgrade
```

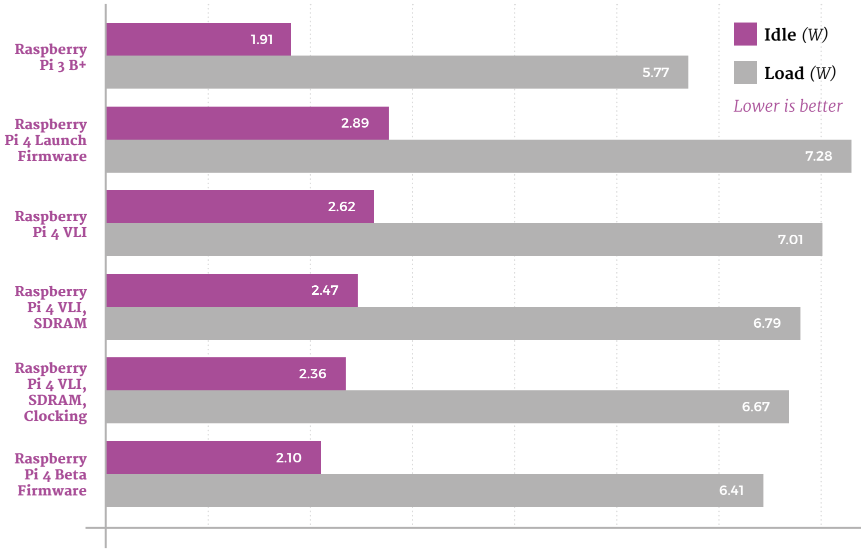
Now restart Raspberry Pi using:

```
sudo shutdown - r now
```

Nobody at Raspberry Pi is resting on their laurels. Beta firmware is in testing and due for public release soon. It brings with it many improvements, including finer-grained control over SoC operating voltages and optimised clocking for the HDMI video state machines.

Power Draw

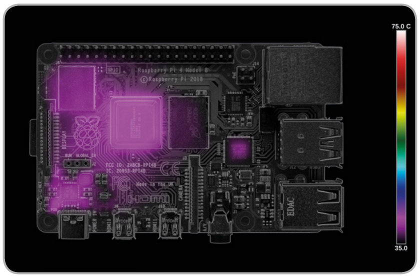
The beta firmware decreases power draw at idle to reduce overall power usage, while tweaking the voltage of the SoC to drop power draw at load without harming performance. The result: a drop to 2.10 W idle, and 6.41 W at load – the best yet.



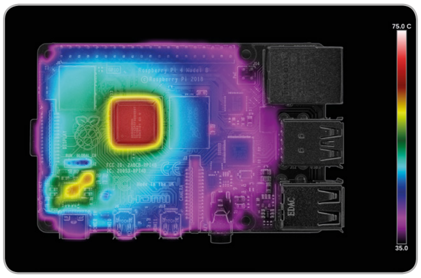
Thermal Imaging

The improvements made at idle are clear to see on thermal imaging: the majority of Raspberry Pi 4's circuit board is below the bottom 35°C measurement point for the first time. After 60 seconds of load, there's a smaller but still measurable improvement, with a peak measured temperature of 64.8°C.

Raspberry Pi 4 beta firmware *(Idle)*

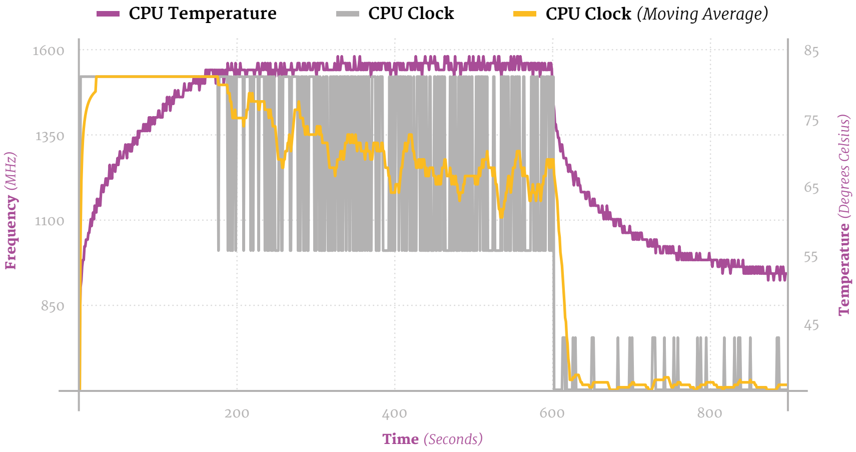


Raspberry Pi 4 beta firmware *(Load)*



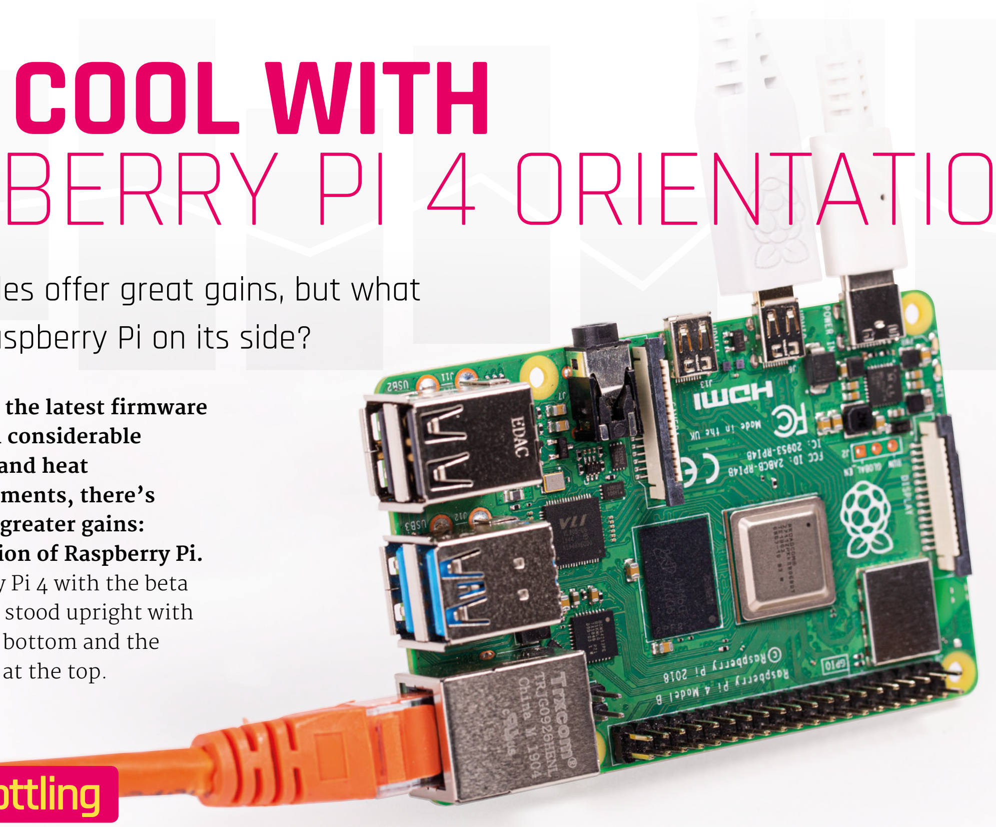
Thermal Throttling

While Raspberry Pi 4 does still throttle with the beta firmware, thanks to the heavy demands of the synthetic workload used for testing, it delivers the best results yet: throttling occurs at the 177s mark while the new clocking controls bring the average clock speed up markedly. The firmware also allows Raspberry Pi 4 to up-clock more at idle, improving the performance of background tasks.



KEEP COOL WITH RASPBERRY PI 4 ORIENTATION

Firmware upgrades offer great gains, but what about putting Raspberry Pi on its side?

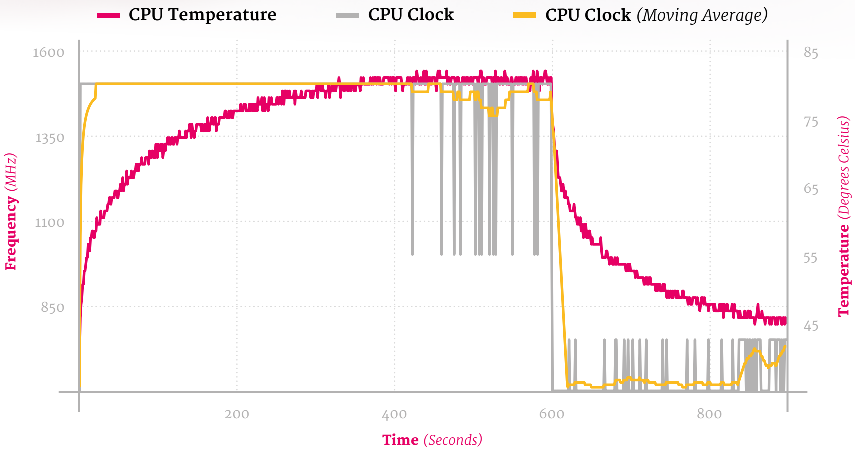


While running the latest firmware will result in considerable power draw and heat management improvements, there's a trick to unlock even greater gains: adjusting the orientation of Raspberry Pi. For this test, Raspberry Pi 4 with the beta firmware installed was stood upright with the GPIO header at the bottom and the power and HDMI ports at the top.

Thermal Throttling

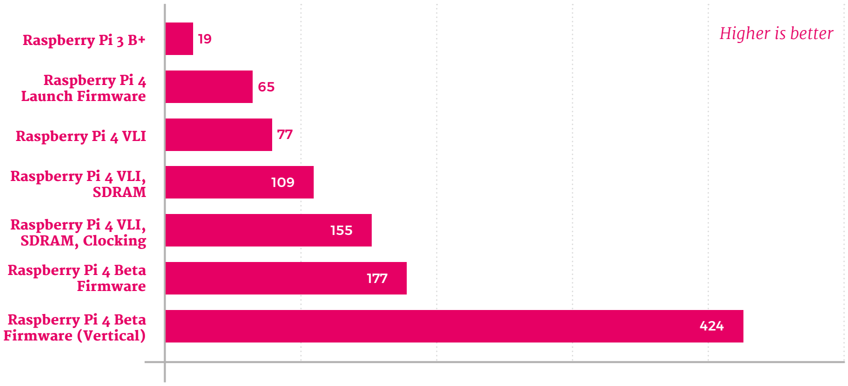
Simply moving Raspberry Pi 4 into a vertical orientation has an immediate impact: the SoC idles around 2°C lower than the previous best and heats a lot more slowly – allowing it to run the synthetic workload for longer without throttling and maintain a dramatically improved average clock speed.

There are several factors at work: having the components oriented vertically improves convection, allowing the surrounding air to draw the heat away more quickly, while lifting the rear of the board from a heat-insulating desk surface dramatically increases the available surface area for cooling.



Throttle Point Timing

This chart shows how long it took to reach the throttle point under the synthetic workload. Raspberry Pi 3B+ sits at the bottom, soft-throttling after just 19 seconds. Each successive firmware update for Raspberry Pi 4, meanwhile, pushes the throttle point further and further – though the biggest impact can be achieved simply by adjusting Raspberry Pi's orientation.



REAL WORLD TESTING

Synthetic benchmarks aside, how do the boards perform with real workloads?

Looking at the previous pages, it's hard to get a real idea of the difference in performance between Raspberry Pi 3B+ and Raspberry Pi 4. The synthetic benchmark chosen for the thermal throttle tests performs power-hungry operations which are rarely seen in real world workloads, and repeats them over and over again with no end.

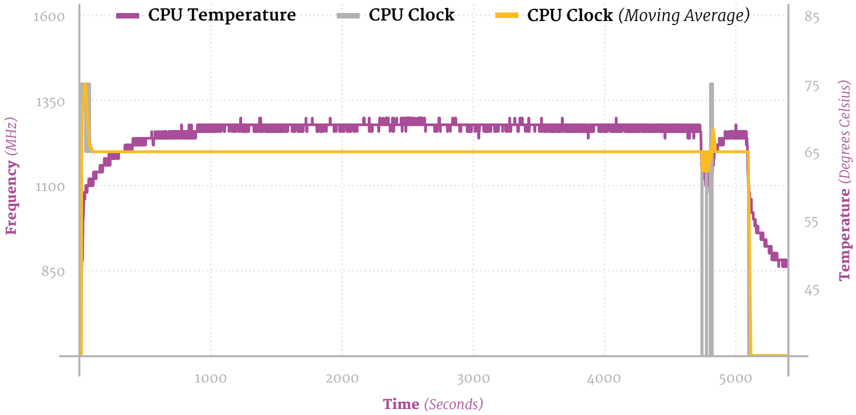
Compiling Linux

In this test, both Raspberry Pi 3B+ and Raspberry Pi 4 are given the task of compiling the Linux kernel from its source code. It's a good example of a CPU-heavy workload which occurs in the real world, and is much more realistic than the deliberately taxing synthetic workload of earlier tests.

With this workload, Raspberry Pi 4 easily emerges the victor. Despite its CPU running only 100MHz faster than Raspberry Pi 3B+ at its full speed, it's considerably more efficient – and, combined with the ability to run without hitting its thermal throttle point, completes the task in nearly half the time.

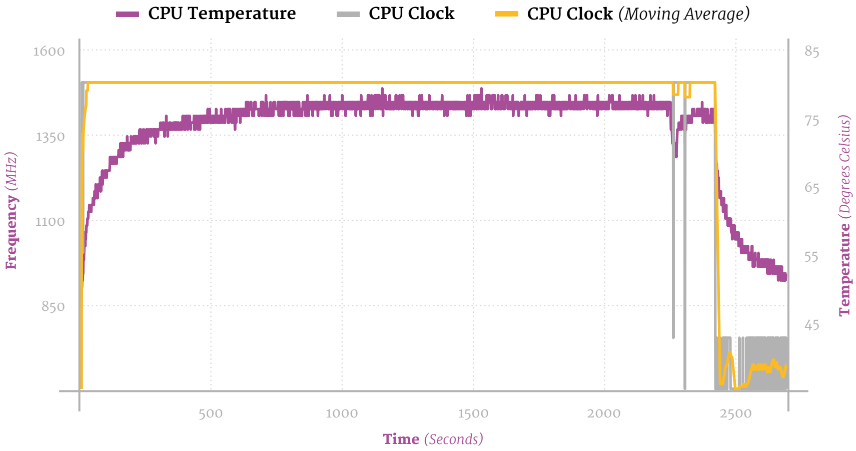
Kernel compile: Raspberry Pi 3B+

Raspberry Pi 3B+ throttles very early on in the benchmark compilation test and remains at a steady 1.2GHz until a brief period of cooling, as the compiler switches from a CPU-heavy workload to a storage-heavy workload, allows it to briefly spike back to its 1.4GHz default again. Compilation finished in 5097 seconds – one hour, 24 minutes, and 57 seconds.



Kernel compile: Raspberry Pi 4

The difference between the synthetic and real world workloads is clear to see: at no point during the compilation did Raspberry Pi 4 reach a high enough temperature to throttle, remaining at its full 1.5GHz throughout – bar, as with Raspberry Pi 3 B+, a brief period when a change in compiler workload allowed it to drop to its idle speeds. Compilation finished in 2660 seconds - 44 minutes and 20 seconds.



TIM GOVER ON FIRMWARE DEVELOPMENT

Bridging the worlds of hardware and software, Tim Gover focuses on the firmware

PROFILE

Tim Gover

Tim Gover is a software engineer at Raspberry Pi, where he works on firmware for booting, power management, and displays.

raspberrypi.org

“I’m a software engineer and work on the firmware,” Tim Gover explains, after having sent across the very latest beta firmware build for testing. “I spend most of my time working on the bootloader, including support for board bring-up and manufacturing. I also get involved with display stuff like the HDMI and HVS firmware drivers.”

Just what is firmware?

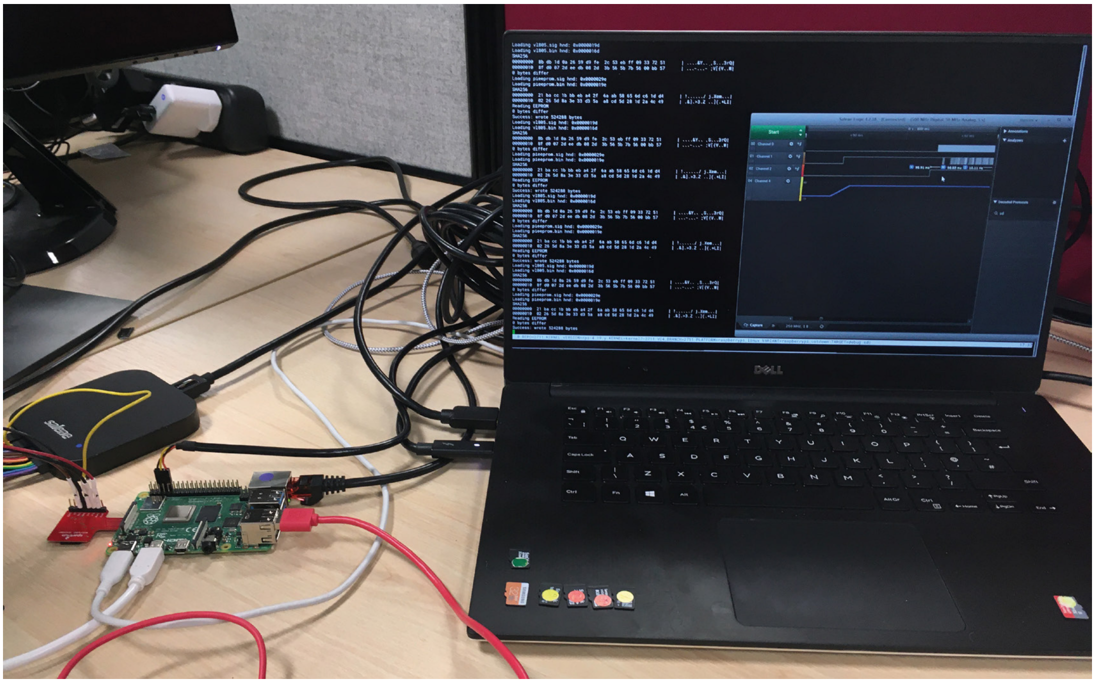
“On Raspberry Pi, firmware normally refers to software running a dedicated processor – instead of the main CPU – to support complex hardware. There are multiple firmwares, including the BCM2711 ROM, bootloader, USB 3.0 controller, WiFi, and the VideoCore **start.elf**.”

Raspberry Pi 4 handles things a little differently to previous models, however – in particular, when it comes to the part known as the second-stage bootloader.

“The second stage bootloader is responsible for initialising the SDRAM and loading the VideoCore firmware into memory from external storage – microSD/Ethernet/USB. It’s loaded by the ROM directly into the VideoCore L2 cache, so must be small, and is equivalent to **bootcode.bin** which was loaded from the microSD on Raspberry Pi 3 and earlier.”

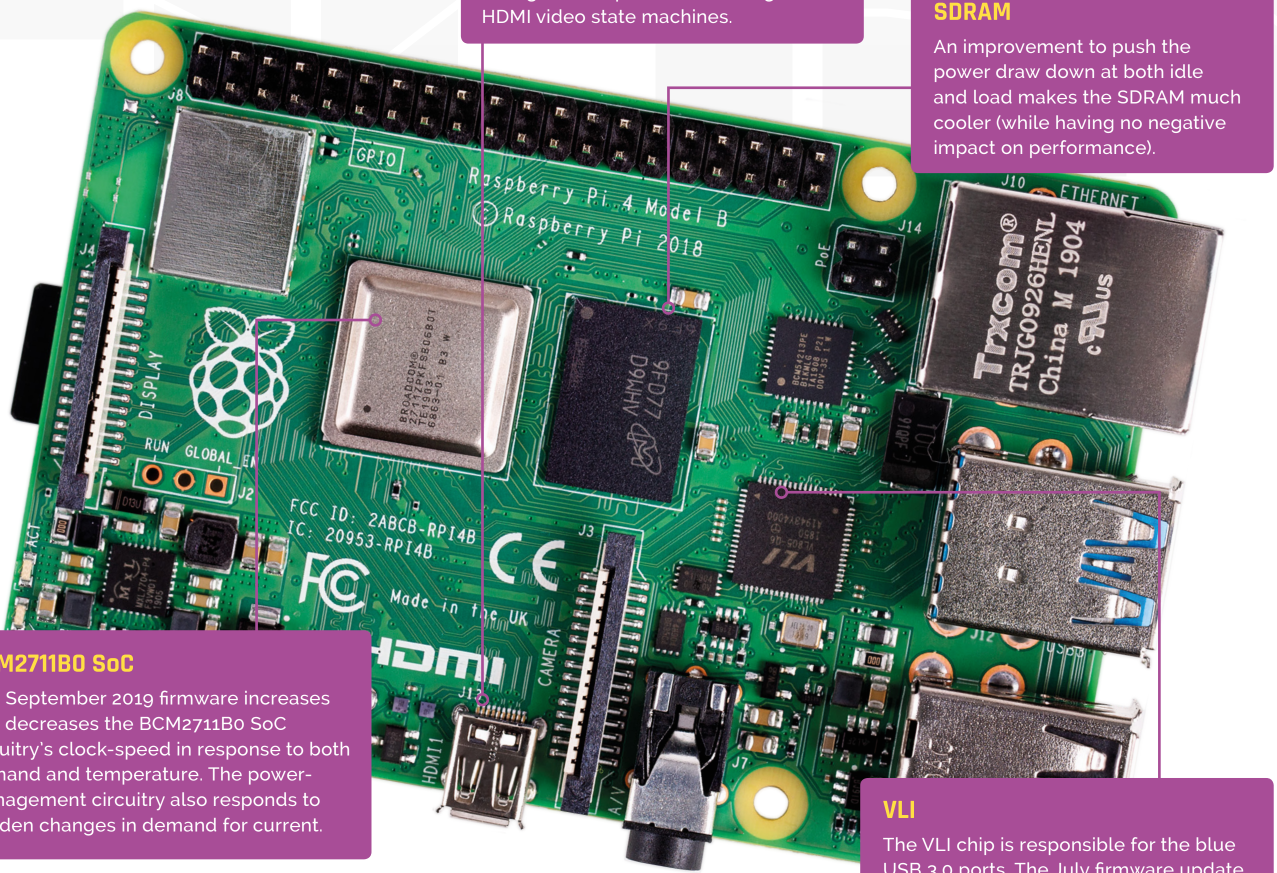
To allow for the bootloader to be easily upgraded, Raspberry Pi 4 comes with something earlier models do not: a small amount of on-board storage known as an electrically erasable programmable read-only memory (EEPROM).

“An EEPROM for the second stage on BCM2711 seems to be a nice middle ground, compared to requiring a microSD (or more expensive on-board eMMC) for a **bootcode.bin**,” Tim explains. “It was never going to be practical to bake more functionality into the chip ROM. Relatively simple USB 2.0 support for mass storage has been



▸ Tim Gover’s toolbox: Raspberry Pi 4 with microSD breakout board, logic analyser, and UART serial communications on the laptop

Raspberry Pi components improved by firmware



HDMI

The new beta firmware introduces finer-grained control over SoC operating voltages and optimised clocking for the HDMI video state machines.

SDRAM

An improvement to push the power draw down at both idle and load makes the SDRAM much cooler (while having no negative impact on performance).

BCM2711B0 SoC

The September 2019 firmware increases and decreases the BCM2711B0 SoC circuitry's clock-speed in response to both demand and temperature. The power-management circuitry also responds to sudden changes in demand for current.

VLI

The VLI chip is responsible for the blue USB 3.0 ports. The July firmware update enabled it to run cooler. Deployment of this firmware took longer than expected after early testing revealed a bug, since resolved, with selected USB 3.0 devices.

> “We've already committed to supporting USB mass storage boot and IPv6 network boot”

replaced with Gigabit Ethernet plus USB 3.0 over PCI Express. These driver interfaces are much more complicated, and also tend to require DMA access to SDRAM.”

What’s the beta’s secret?

“The new DVFS code adds more operating points for Arm [CPU] frequencies,” says Tim, “and picks the best voltage according to the operating point. The voltage also has to be suitable for all the other blocks on the chip – e.g. V3D – so this is a fairly complicated change.”

There are other, more minor changes, too, including a more sensible default for Raspberry Pi 4’s USB On-The-Go (OTG) functionality. “By default the USB OTG controller is not enabled unless `dtoverlay=dwc-otg` is added to **config.txt**,” explains Tim. “This saves about 35 mW for most users who only use the USB-C connection for power.”

What else should we know?

“Dynamic HDMI clocking – the HDMI state machine clock is now dynamically changed to match the resolution requirements using the updated firmware clock infrastructure. This makes it better at reducing the idle power to the minimum required for the current display configuration – e.g. a single high-definition display should have lower idle power consumption than a 4Kp30 or dual-display configuration.”

What’s next on the to-do list?

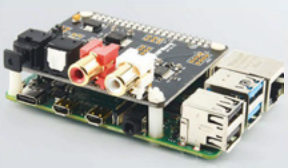
“We’ve already committed to supporting USB mass storage boot and IPv6 network boot, so that will keep me busy for a while!”

Universal & Unique

UniPiCase

for Raspberry Pi 4

Perfect option for your DIY or high-volume commercial projects



Fits all kinds of HATs
Ideal PoE HAT case

Use with HAT or Pi alone
Simple and professional

Rapid, tool-free assembly
GPIO cable pass-through

Two wall mount options
Well ventilated

www.UniPiCase.com

Designed and manufactured in North America

GPIO **Xmas Tree**

▶ pocketmoneytronics ▶ **magpi.cc/Q5eraD** ▶ £4 / $5

A mini, programmable Christmas tree accessory that's easy to code in Scratch and Python. **Rob Zwetsloot** decks his Raspberry Pi with one

SPECS

LIGHTS:
5 × LEDs

PROGRAMMING LANGUAGES:
Scratch, Python

TIME TO FINISH:
Roughly five minutes

We love all the GPIO Christmas decorations that you can get for Raspberry Pi, and while we also really enjoy putting stuff together like those Christmas kits, we sometimes just like to plug something in and see it go.

The GPIO Xmas Tree is something like that – and probably the easiest one to code yet! It's also nice and small, sitting over six GPIO pins rather than taking up all 40. This way, you can add a little festive flair to your Raspberry Pi, while still working on other projects that require the use of some GPIO pins.

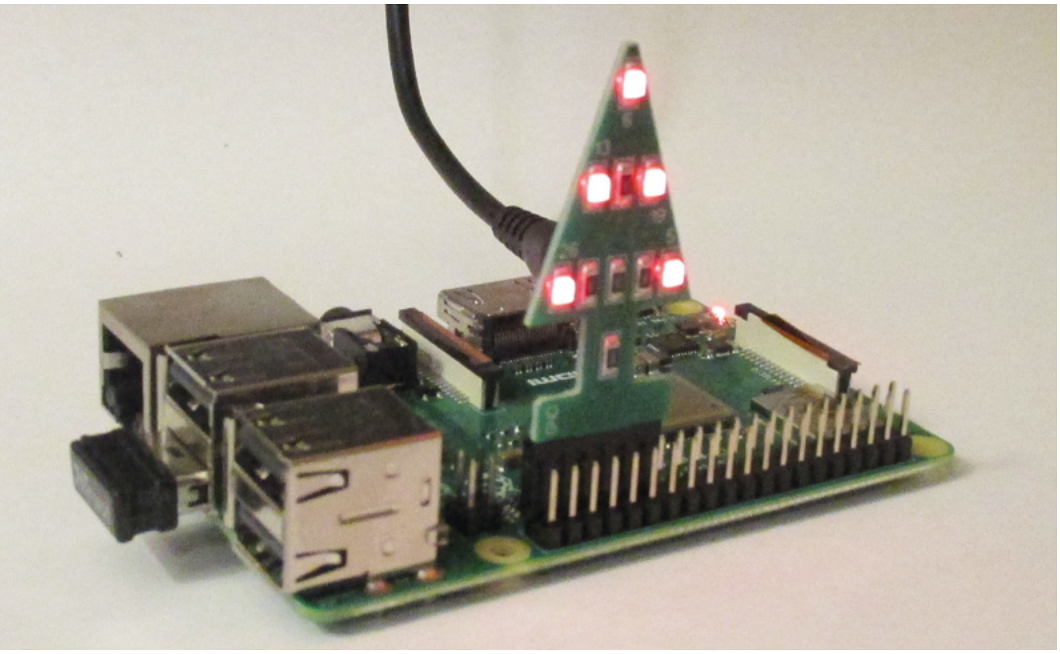
> " One of the unique things about this tree is that you can program it with Scratch! "

Easy coding

One of the unique things about this tree is that you can program it with Scratch! Scratch on Raspberry Pi has a built-in GPIO library, allowing you to code physical objects, including this tree's five LEDs.

This is one of the few cases, though, where something so simple is much easier to do via Python, especially thanks to the GPIO Zero library, requiring less than ten lines of code to create a wonderful twinkling effect!

We quite adore this little tree – it's cheap and cheerful and could be someone's first Raspberry Pi project on Christmas morning, with a quick and very cool result.



▶ While there's nowhere to hang your baubles, this tiny tree is fun to program

Verdict

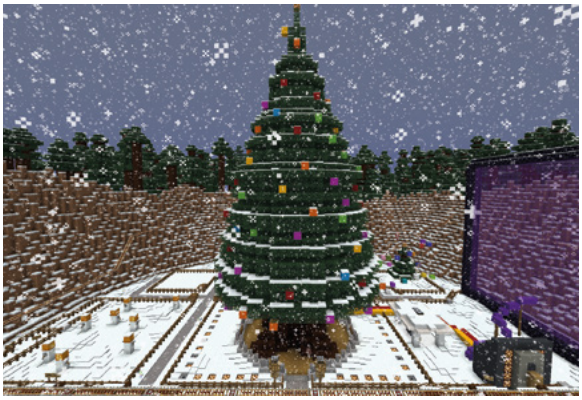
This fun little project will make your Raspberry Pi work desk festive, or make a young maker's first steps at Christmas wonderful.

9/10

10 Best: Christmas projects

Liven up your living room with these fantastic holiday builds

We love seeing people make their own festive projects at this time of year, and we always try to make sure we include one of our own every December issue. Check out page 40 for this year's smart lights. If you're still feeling a bit humbug about it all, here are ten other incredible Christmas projects to get you in the spirit.



▲ Minecraft-controlled Christmas tree

VR to RL

Minecraft can be hacked with a bit of code so that you can make it do as you wish. But this also means that, via more code, it can interact with reality. So, David Stevens has made it so changes to the Christmas tree in the game alter the lights outside. Clever!

magpi.cc/DdipTY

▲ Smart Gingerbread House

Edible electronics, sorta

Gingerbread is delicious, and houses made from it are a staple decoration in some homes. Estefannie takes it a step further and gives hers lights and motors. Honestly, ours wouldn't last long enough for us to do that.

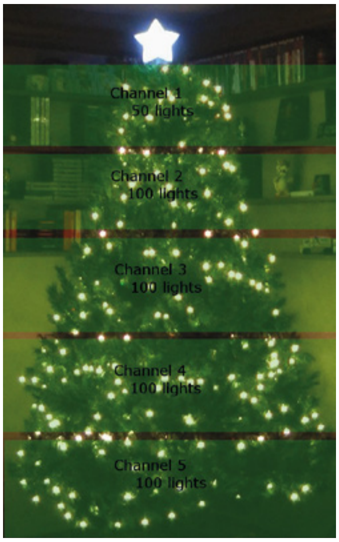
magpi.cc/AG4pbj

▼ Raspberry Pi Christmas Tree Light Show

Advanced tree lights

We quite like the lights we made earlier in the mag, but if you want to do some serious tree hacking, we suggest taking a look at this amazing project on Instructables.

magpi.cc/irP5Nh



▲ Secret Santa Babbage

Upgrade your office

If you find fishing names out of a hat a bit old-fashioned, you can always have Babbage Bear choose for you. Squeeze his hand and he'll print out a piece of paper showing who you need to buy for. £5 limit, though.

magpi.cc/Hbaw2t

▼ Raspberry Pi Christmas Light Display

Home light automation

This is a serious amount of lights all over this house to be controlled by a Raspberry Pi. David does so using a phone to connect to Raspbian via VNC.

magpi.cc/jAtBw3

▲ Pireplace

Raspberry Pi fireplace

Some people like to turn on the fireplace video on Netflix; others prefer to create digital fires of their own. This one also does the impossible and can cycle through different colours of flame.

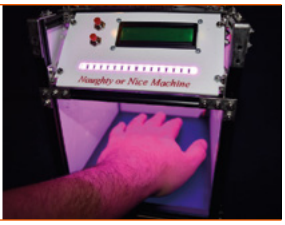
magpi.cc/GpvLJe

▶ Naughty or Nice machine

Make a list

A modern take on the palm readers you used to find on seaside piers, although this machine takes a more binary approach: are you naughty… or nice? Think happy thoughts.

magpi.cc/cyyW58



▲ Lightshow Pi

Drop the house bass

Another project where a house has been kitted out with full Raspberry Pi-powered lights – only this one plays music and syncs to it as well. Especially phat dubstep tracks.

lightshowpi.org

▲ Santa detector

Catch Saint Nick

This is a fun Scratch project that allows you to go to sleep happy in the knowledge that you won't miss Santa. Maybe you'll find out exactly how big his sack of toys is.

magpi.cc/xGNNQw

The MagPi Christmas Card Cover

Hack the magazine

A couple years ago, we made a little project that allowed you to turn a copy of *The MagPi* into a light-up Christmas card. Grab the PDF, print the cover on card, and give it a go!

magpi.cc/52

SEND US YOUR HOLIDAY PROJECTS!

Made something with Raspberry Pi for this holiday season? Send us photos on Twitter (**@TheMagPi**) or via email (**magpi@raspberrypi.org**) so we can hopefully feature you in the next issue!

Learn computing systems with Raspberry Pi

It's one thing to learn about computers, and quite another to learn about computing

The Elements of Computing Systems

CREATOR
Noam Nisan and Shimon Shocken

Price:
£25/$35

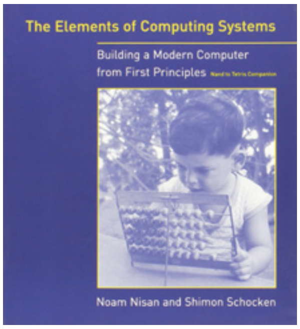
magpi.cc/XRyTzM

Modern computing systems are built on a stack of technologies. Right at the top, you have the operating system and high-level languages like Python. These sit above a virtual machine that communicates via assembly language to the hardware, which itself is built on a system of chips and logic gates. Head down and you'll hit physics.

Professional programmers often don't know what's going on deep down in the computer system. They rely on the fact that what they do works.

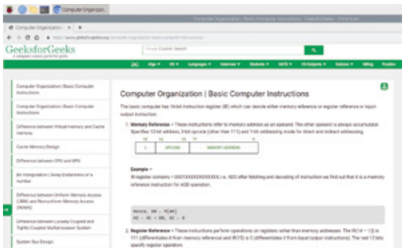
The Elements of Computing Systems is the book behind the more popular Nand to Tetris course (aka 'nand2tetris'). Using a hardware simulator to build a NAND (NOT-AND) gate, the latter is then used to build all the other chips and gates that form a computer. You then use that to build a general-purpose computer system, called Hack; and a compiler, called Jack. You use these to build an operating system.

The corresponding website (**nand2tetris.org**) has projects, software installation, and sample programs made with Jack (including the aforementioned Tetris).



The Elements of Computing Systems
Building a Modern Computer from First Principles
Nand to Tetris Companion
Noam Nisan and Shimon Schocken

Websites



GEEKSFORGEEKS

The GeeksforGeeks site should be bookmarked anyway, but take a look at the Computer Organization and Architecture tutorials section.

magpi.cc/R622SJ

STACK EXCHANGE

Stack Exchange has two useful boards for you: Computer Science (**magpi.cc/T4uM4h**) and Electrical Engineering (**magpi.cc/infrdB**). Sign up with both.

stackexchange.com

CPU SIMULATOR

There are countless CPU simulation programs and sites to explore, many based on the Little Man Computer model (**magpi.cc/NAphAm**). Why not start with this offering?

tools.withcode.uk/cpu

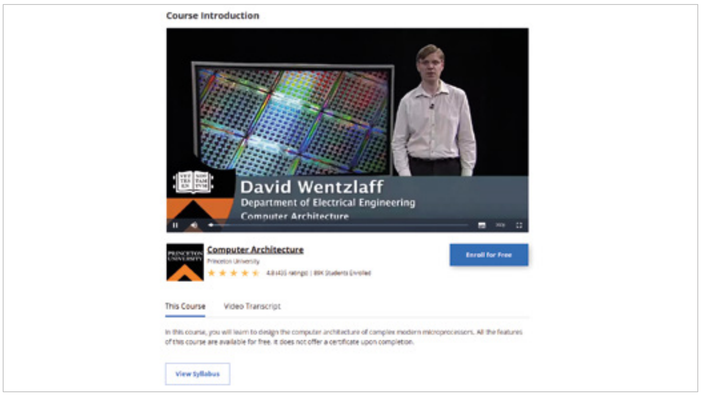
Computer Architecture

CREATOR

David Wentzlaff

Price:
Free

magpi.cc/ikxjQd



Course Introduction

David Wentzlaff
Department of Electrical Engineering
Computer Architecture

Computer Architecture
Princeton University
★★★★★ 4.8 (435 ratings) | 89K Students Enrolled

Enroll for Free

This Course | Video Transcript

In this course, you will learn to design the computer architecture of complex modern microprocessors. All the features of this course are available for free. It does not offer a certificate upon completion.

View Syllabus

Princeton University's Computer Architecture course is on Coursera and is widely regarded as one of the best 'all-rounders'. The course covers architecture, pipelining, cache, superscalar, and works its way up to multiprocessors.

The course is led by David Wentzlaff, Associate Professor, Electrical Engineering at Princeton, and he leads you through the concepts. By the end of the course, you will have a good understanding of the different types of processor architectures.

You can enrol on the course for free, although there is no certificate for completion.

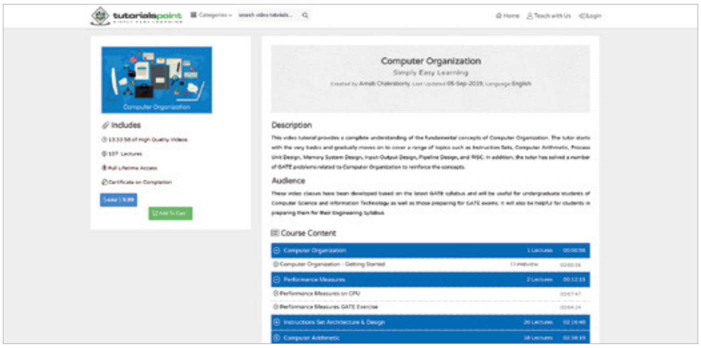
Computer Organization

CREATOR

Parrot

Price:
£100 / $130

magpi.cc/kx2Cr7



Computer Organization
Simply Easy Learning
Created by Arnab Chakraborty, Last Updated 05-Sep-2019, Language English

Includes
13:33:58 of High Quality Videos
107 Lectures
Full Lifetime Access
Certificate on Completion

Description
This video tutorial provides a complete understanding of the fundamental concepts of Computer Organization. The tutor starts with the very basics and gradually moves on to cover a range of topics such as Instruction Sets, Computer Arithmetic, Process Unit Design, Memory System Design, Input Output Design, Pipeline Design, and RISC. In addition, the tutor has solved a number of GATE problems related to Computer Organization to reinforce the concepts.

Audience
These video classes have been developed based on the latest GATE syllabus and will be useful for undergraduate students of Computer Science and Information Technology as well as those preparing for GATE exams. It will also be helpful for students in preparing them for their Engineering Syllabus.

Course Content

Computer Organization	1 Lectures	00:00:56
Computer Organization - Getting Started	Preview	00:00:56
Performance Measures	2 Lectures	00:12:11
Performance Measures on CPU		00:07:47
Performance Measures GATE Exercise		00:04:24
Instructions Set Architecture & Design	20 Lectures	02:16:46
Computer Arithmetic	16 Lectures	02:36:19

Computer Organization starts with computer and performance, before moving on to processor unit design and memory system design, then input-output design and pipeline design techniques.

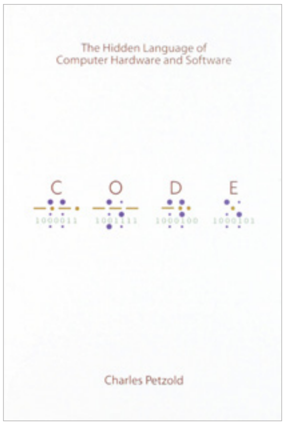
Of particular interest for Raspberry Pi fans is its coverage of RISC – reduced instruction set computers.

It's developed using the GATE (Graduate Aptitude Test in Engineering) syllabus, making it useful for undergraduates. But the course content of use to anybody interested in learning computing architecture.

The price is listed at $132, but it's frequently on sale (it was $10 at the time we went to press, so look out for that).

Books

Add these titles to your computing bookshelf



The Hidden Language of Computer Hardware and Software

C O D E

Charles Petzold

CODE: THE HIDDEN LANGUAGE OF COMPUTER HARDWARE AND SOFTWARE

This book is famous for explaining complex concepts to non-technical people. It doesn't teach anything about programming, but is a great place for absolute beginners.
magpi.cc/BVUnby

BUT HOW DO IT KNOW?

This book is designed to bridge the gap between knowing the major parts of a computer and starting a complete course. The first three chapters are available on Google Books (magpi.cc/tQrtY5).
buthowdoitknow.com

ALGORITHMS TO LIVE BY

Learn how computer science can be applied in the real world (and gain a further appreciation for the computing concepts in the meantime).
magpi.cc/ynWm9S

Liz Clark
aka Blitz City DIY

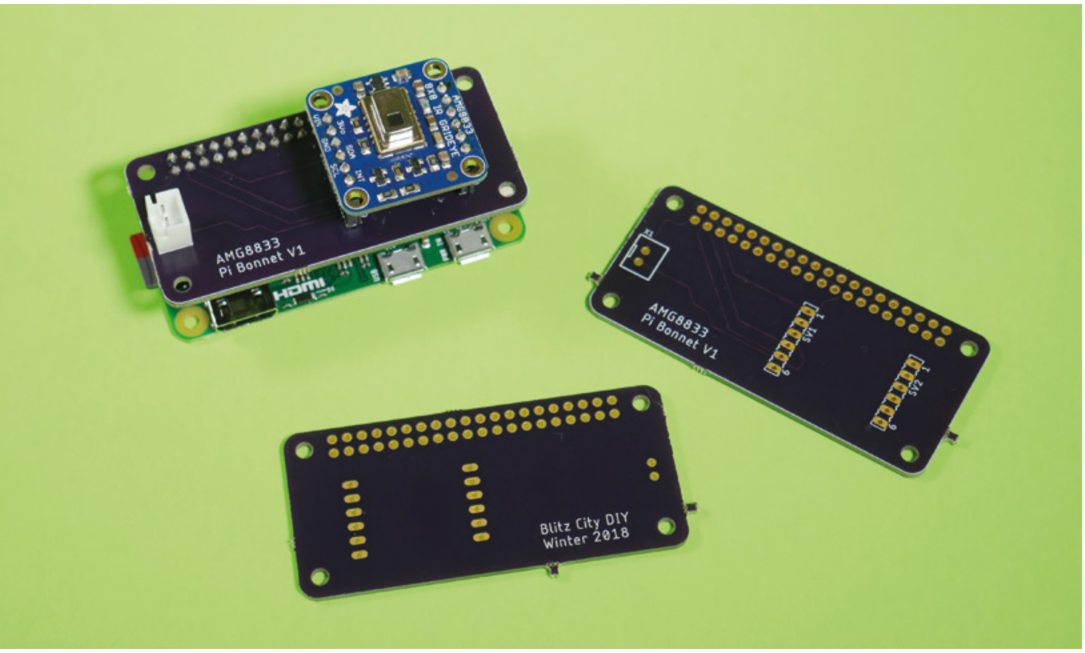
The creator of Blitz City DIY talks to us about making amazing YouTube videos of cool builds

> Day job **Vlogger** | > Community role **YouTuber** | > URL **magpi.cc/FDpqSi**

Not everyone has been tinkering with electronics for decades. Some are new to the hobby, and it's easy to figure out why: information on the internet is easier to obtain than ever, and the low cost of Raspberry Pi has helped to further make it accessible. Liz is one of those newcomers.

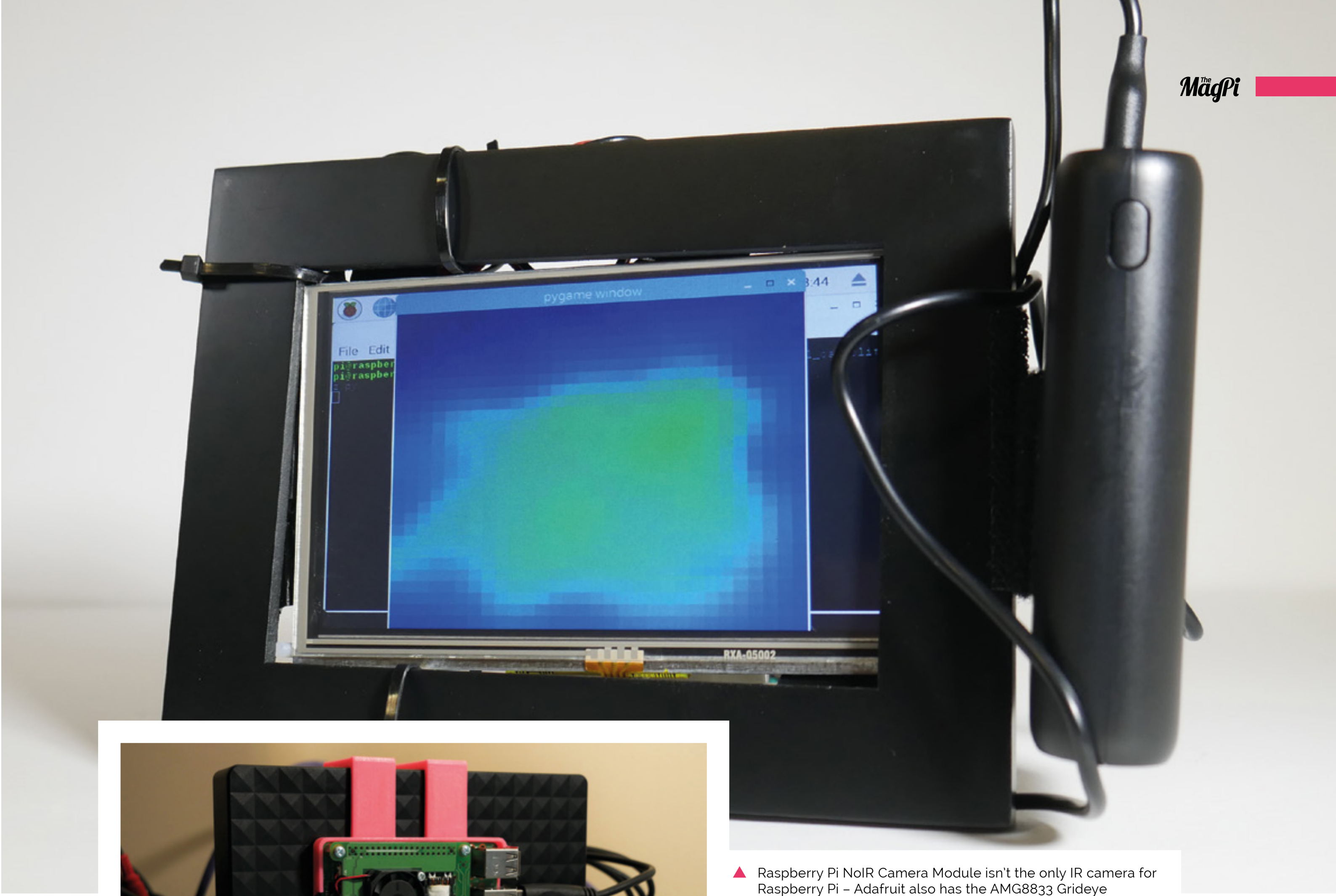
"I actually got a bit of a 'late' start to making (although I'm a firm believer that it's never too late to learn something new)," Liz tells us – and we completely agree. "I was in college and decided that I should try and learn some coding without really knowing what that meant. I was majoring in music technology, so I began my search in the music arena and learned about Arduino and all of the MIDI projects that people were beginning to make with them. I dabbled a bit off and on with it, but it wasn't until about four years ago that I really got serious about it and now I'm doing a bit of everything: CircuitPython, 3D printing and design, PCB design, and of course a hearty portion of Raspberry Pi. Outside of hardware, though, I am a long-time avid knitter, cross-stitcher, and sewer."

We'd also argue that textile arts are a form of making as well, just a bit more low-tech.

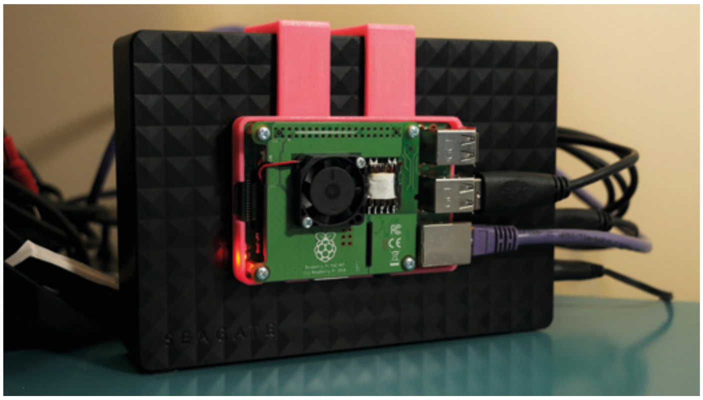


Designing a PCB is no simple feat, and this one really makes the thermal camera project a bit easier for Liz

Why did you start your channel?
Related to my 'late' start in making, I had been out of college for about two years and was starting to feel a little stuck. I really wasn't working on any creative projects and I was worried that I was going to permanently fall into that rut working full-time. I also have a video background and I hadn't filmed and edited anything for fun at that point since school, so my channel was basically born from a place of worry and/or quarter-life crisis. It was a very surreal and odd thing for me to do because I definitely lean more toward the introverted side, so the idea of me talking on camera about things was completely outside of my comfort zone. However, I've become a lot better



▲ Raspberry Pi NoIR Camera Module isn't the only IR camera for Raspberry Pi – Adafruit also has the AMG8833 Grideye



▲ A familiar sight to many Raspberry Pi owners – a Raspberry Pi-powered media server. We like the 3D-printed mounting to cut down on the footprint

> I actually got a bit of a 'late' start to making (although I'm a firm believer that it's never too late to learn something new)

at speaking as a result, and my goal for working on creative projects and keeping up my video skills wouldn't have been possible without my channel. It's also led to some really amazing opportunities that otherwise I wouldn't have had, so I'm so glad I took the initial risk.

When did you first learn about Raspberry Pi?

I first learned about Raspberry Pi a couple of years ago, around the time that I was starting my channel. It honestly seemed a little mysterious to me at first because I wasn't quite sure what it was or what it did, but I quickly fixed that and fell in love with Linux and all things single-board computers.

Any future Raspberry Pi plans?

I'm actually working on a big Raspberry Pi project right now. It's going to be a MIDI-powered robot xylophone. I'm using tiny solenoid motors to strike the keys on a glockenspiel and I'm using MIDI-in over UART on Raspberry Pi. I also made a custom HAT PCB to connect up the MIDI-in circuit and multiplexers to Raspberry Pi's GPIO. It's definitely a concept that's been done before, but having a music background and being a former mallet player, I'm really excited to create my own version.

Blitz City DIY Raspberry Pi projects

"I rigged up a thermal camera using the AMG8833 thermal camera module from Adafruit. It was before I had a 3D printer, so the housing was in a modified picture frame. That project also ended up becoming my first PCB – I designed up a quick Bonnet-type board to just easily route the thermal camera module to Raspberry Pi's GPIO.

"I also have an OpenMediaVault instance running on a Raspberry Pi that I use as a home media server and network backup for my computers, and I do run Steam Link on a Raspberry Pi as well. I think that's one really cool thing about Raspberry Pi in general: you can run a lot of different types of projects on them and there's always more flavours of Linux becoming available for them too."

This Month in Raspberry Pi

Coolest Projects 2020!

Next year's Coolest Projects events are coming together

We love the Coolest Projects – it's amazing to see the imagination and ingenuity of young makers in multiple fields. More details about the upcoming 2020 events in the USA and UK were revealed – as well as the opening of registration – as we were going to print, so we've squeezed them in here!

The event is both a celebration and an exhibition to inspire and enable innovation, creativity, entrepreneurship, and technology skills in young makers. Join us to support hundreds of young innovators and celebrate their accomplishments. Participants and visitors will also get the chance to take part in exciting hands-on tech activities, see inspirational speakers and leading technology experts, and join a global community of digital makers.

All projects are welcome in any programming language or using whatever hardware you like, whether you're a beginner or a seasoned creator. You can work as an individual or as part of a team of up to five. Coolest Projects events are open to all levels of skill; the focus is on creativity, participation and, most of all, having fun!

COOLEST PROJECTS USA
THE WORLD'S LEADING TECHNOLOGY EXHIBITION FOR YOUNG PEOPLE
MARCH 7, 2020
DISCOVERY CUBE ORANGE COUNTY CALIFORNIA
COOLESTPROJECTS.ORG/USA
#COOLESTPROJECTS
Raspberry Pi

Coolest Projects USA

- 7 March 2020
- Discovery Cube Orange County, CA, USA
- coolestprojects.org/usa
- magpi.cc/fKpgFA

COOLEST PROJECTS UK
THE WORLD'S LEADING TECHNOLOGY EXHIBITION FOR YOUNG PEOPLE
4 APRIL 2020
THE SHARP PROJECT, MANCHESTER
COOLESTPROJECTS.ORG/UK
#COOLESTPROJECTS
Raspberry Pi

Coolest Projects UK

- 4 April 2020
- The Sharp Project, Manchester UK
- coolestprojects.org/uk

Christmas comes early!

Some makers have been getting their Christmas projects sorted well in advance

We often get messages and emails from readers who would love us to feature their project – and usually we do in our project showcase feature! Unfortunately, when it comes to seasonal projects, we don't always have time or space – so here's a couple we were sent this year that we'd love to show off a bit of!

Interactive Nativity Scene

MAKER: Tomasz Siroń

Tomasz was so keen to let us know about his plans, he messaged us in March about his project. It's a nativity scene in Poland that you can control via the internet!

"I wanted to show off our original project, which I built together with friends," Tomasz said. "There is a crib controlled by the internet. Everyone can control the crib. The site was created using Bootstrap and jQuery. Hosting is on our own basement server. We use a virtual machine (Debian) on Proxmox.

"We used a Raspberry Pi 3, two IP cameras, and two Arduino Nanos… In the crib, we installed the computer speaker – carols were heard around the city centre. I built it with my colleagues over about a month."

He's brought it back for another year. Find out more here: **magpi.cc/nXqYzp**.

Raspberry Pi Christmas Lights

MAKER: Greg Macaree

We got this through via email from Greg in early November, and it's a project that he's been working on and upgrading for a while now!

"At the end of last year, I added a string of 500 pixels (LEDs) to a Raspberry Pi and created an animated Christmas tree for the front garden," Greg told us in the email. "This year, we've taken it up a notch: we'll be running over 6000 pixels for our Xmas Show – still controlled by a Raspberry Pi, although connected via dedicated hardware."

It's a spectacular sight, and they even used it for Halloween! We love a good project you can recycle and repurpose – take a look here: **magpi.cc/igXcwi**.

MagPi Monday

Amazing projects direct from our Twitter

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made. Here is a small fraction of them. Follow along at the hashtag #MagPiMonday.

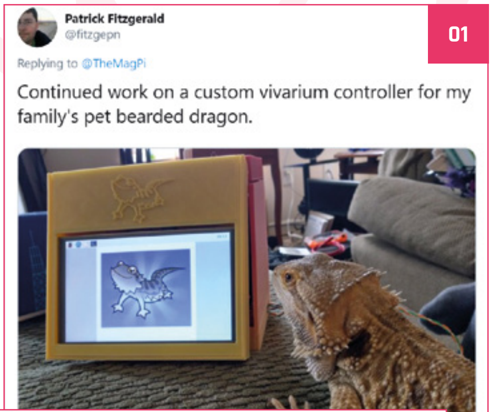
01. We're not sure what's cooler: the applications of this project, or the bearded dragon logo

02. Halloween was just happening as we were released last issue, meaning there were still some great seasonal spooks to see

03. This lovely-looking arcade cabinet that folds down to be stored is wonderful

04. We love seeing updates on the automated mowing robot

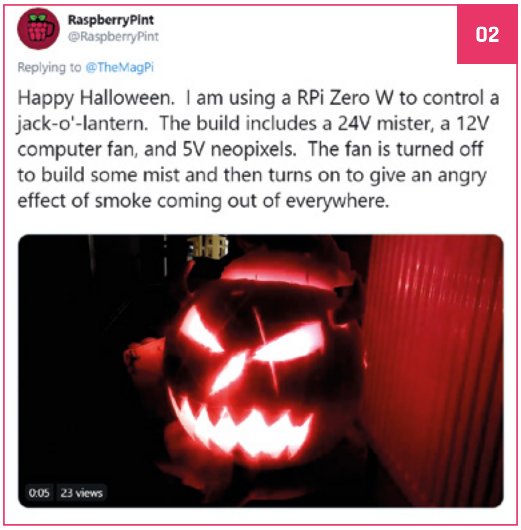
05. Mike of CamJam and Pi Wars fame showed off this amazing-looking wooden robot


01

Patrick Fitzgerald
@fitzgepn

Replying to @TheMagPi

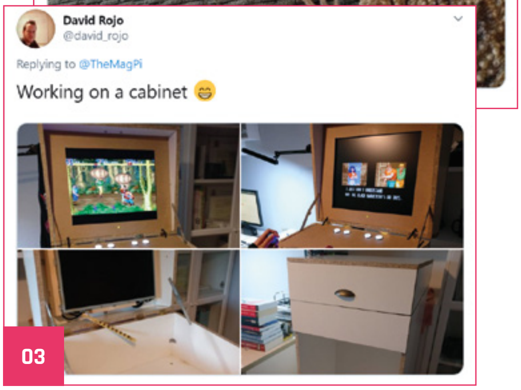
Continued work on a custom vivarium controller for my family's pet bearded dragon.


02

RaspberryPint
@RaspberryPint

Replying to @TheMagPi

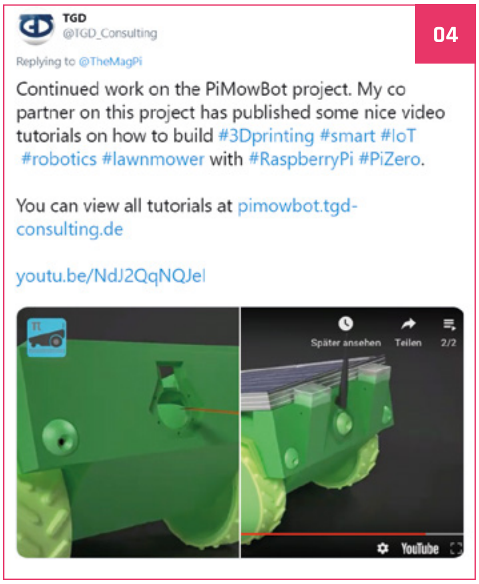
Happy Halloween. I am using a RPi Zero W to control a jack-o'-lantern. The build includes a 24V mister, a 12V computer fan, and 5V neopixels. The fan is turned off to build some mist and then turns on to give an angry effect of smoke coming out of everywhere.


03

David Rojo
@david_rojo

Replying to @TheMagPi

Working on a cabinet 😄


04

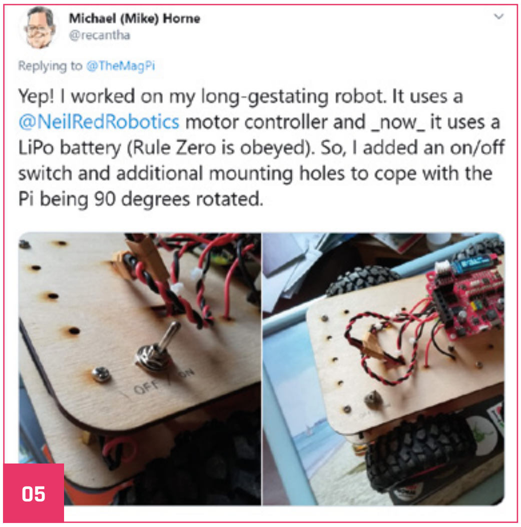
TGD
@TGD_Consulting

Replying to @TheMagPi

Continued work on the PiMowBot project. My co partner on this project has published some nice video tutorials on how to build #3Dprinting #smart #IoT #robotics #lawnmower with #RaspberryPi #PiZero.

You can view all tutorials at pimowbot.tgd-consulting.de

youtu.be/NdJ2QqNQJeI


05

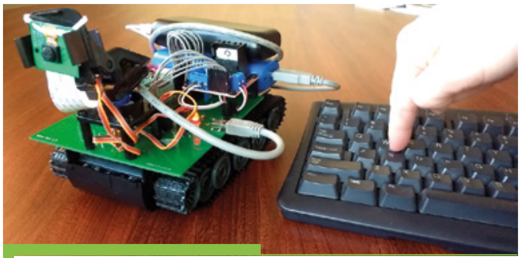
Michael (Mike) Horne
@recantha

Replying to @TheMagPi

Yep! I worked on my long-gestating robot. It uses a @NeilRedRobotics motor controller and _now_ it uses a LiPo battery (Rule Zero is obeyed). So, I added an on/off switch and additional mounting holes to cope with the Pi being 90 degrees rotated.

Crowdfund this!

Raspberry Pi projects you can crowdfund this month



Low-Latency Real-time Robot Control Software Guide

Less of a product and more documentation, this will give you some direction on how to remotely control a robot in real time over WiFi. We've done tutorials on stuff like this, but this plans to go a bit more in-depth.

▶ kck.st/2WrbPlH

Black Raspberry

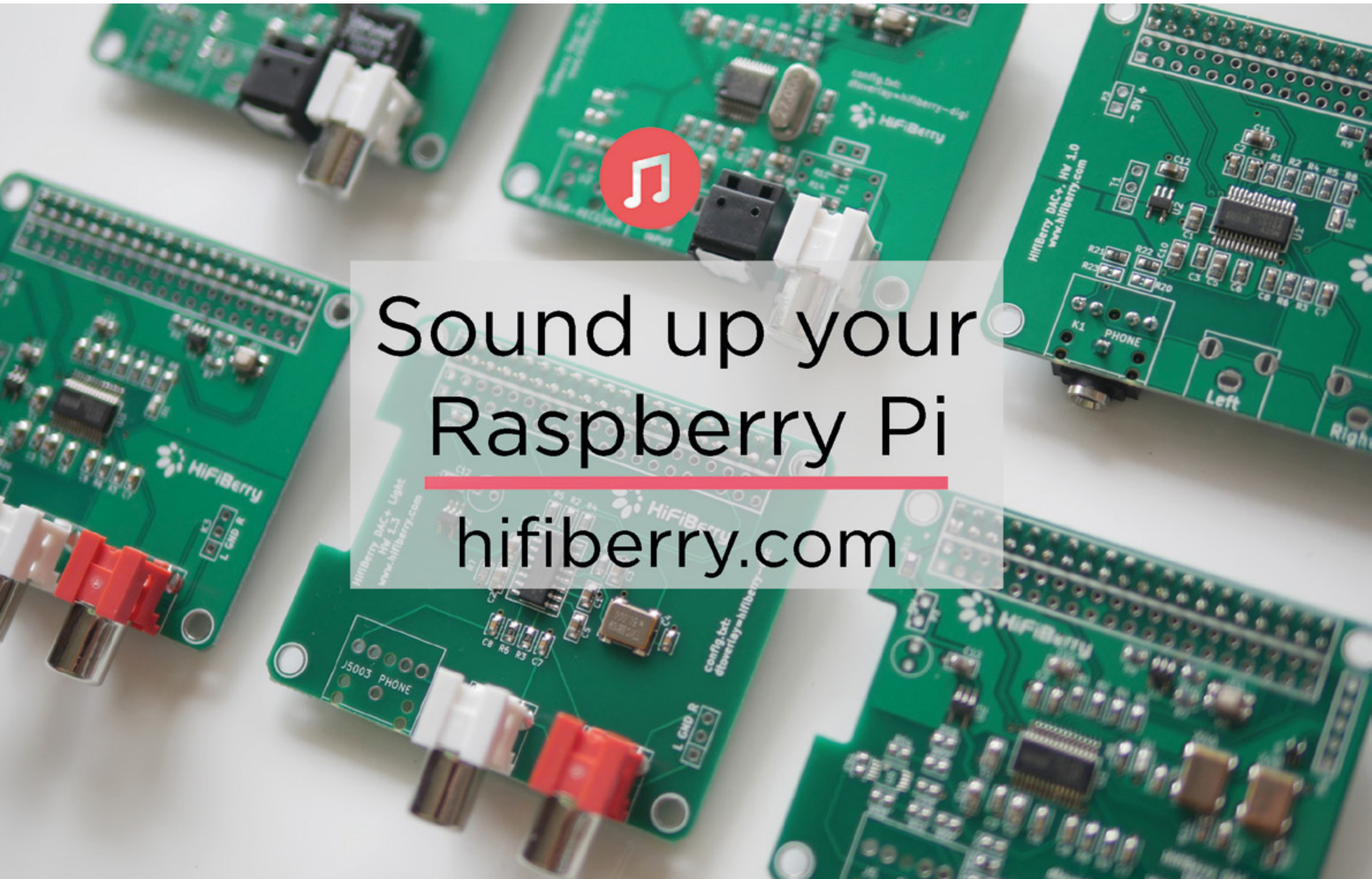
This case designed for a Raspberry Pi 4 does a bit more than just keep a Raspberry Pi protected – it also extends out the I/O (USB, Ethernet, etc.) to a more classic configuration along the 'rear' of the case.

▶ kck.st/2peSisq

CROWDFUNDING A PROJECT?

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

magpi@raspberrypi.org



Raspberry Jam Event Calendar

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

01. Raspberry Pi Jam Mexico

- Saturday 30 November
- Tenayuca 25, Mexico City, Mexico
- magpi.cc/3wtTp5

A community event focused on gathering people who want to learn about maker culture.

02. Leeds Raspberry Jam

- Wednesday 4 December
- Dixons Unity Academy, Leeds, UK
- magpi.cc/Nnhpgn

There'll be chances to get hands-on with more digital making activities through the workshop, as well as a hackspace area.

03. Exeter Raspberry Jam

- Saturday 7 December
- Exeter Library, Exeter, UK
- magpi.cc/rpep8e

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

04. Stafford Raspberry Jam

- Tuesday 10 December
- Stafford College, Stafford, UK
- magpi.cc/3T5aT2

A meet-up for folks who have a Raspberry Pi computer and want to learn more about it..

05. Barnstaple Library Raspberry Jam

- Saturday 14 December
- Barnstaple Library, Barnstaple, UK
- magpi.cc/LXPaPG

Get together and share ideas with other Raspberry Pi enthusiasts. Everyone welcome.

06. Cornwall Tech Jam

- Saturday 14 December
- Bodmin Library, Bodmin, UK
- cornwalltechjam.uk

For anyone interested in technology, of all ages and abilities. Ask questions and learn about programming.

07. South Devon Tech Jam

- Saturday 14 December
- Paignton Library & Information Centre, Paignton, UK
- magpi.cc/9vhGQ5

A monthly informal and friendly session for anyone interested in technology, regardless of age or ability.

08. Raspberry Jam Zelzate

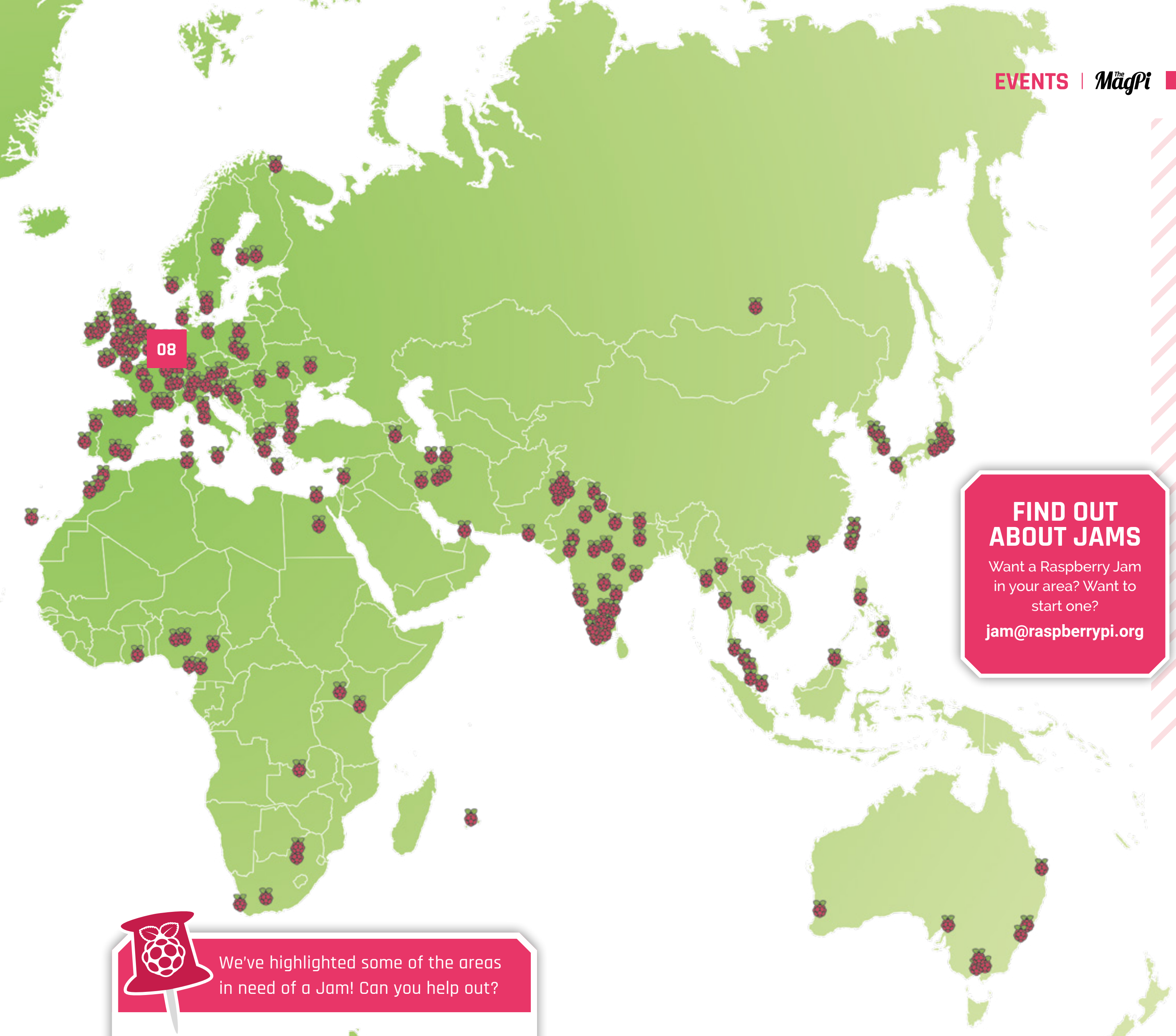
- Saturday 21 December
- Openbare Bibliotheek Zelzate, Zelzate, Belgium
- magpi.cc/agqrW8

Everyone is welcome to start, share, and work on their own projects in a fun and relaxed atmosphere.

FULL CALENDAR

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

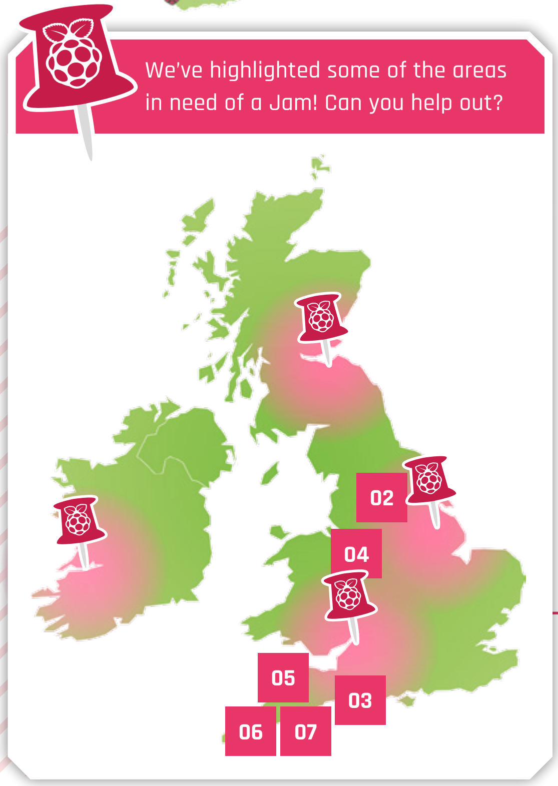
rpf.io/jam



FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one?

jam@raspberrypi.org



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

Raspberry Jam advice:
Checking in

“**We use the Eventbrite app to check in attendees where possible.** However, people tend to arrive all at once and we don’t like a queue. If it’s not possible to scan the tickets, we just check them as they come in. It means our numbers are a little off, but it keeps people happier.”

Michael Horne – Cambridge 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: **rpf.io/guidebook**

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

Your Letters

Project submission

If I've created a project that I want to submit to you, what is the best way to go about it?

Nessa via Twitter

If you drop us an email at **magpi@raspberrypi.org**, or send it to us via Facebook or Twitter, we'd be happy to look at putting it in the magazine in some form. Good pictures are necessary, so if you have some taken in decent light, that will help.

Usually, we'll ask you some questions and your project ends up in a Project Showcase – otherwise we'll try to squeeze it in elsewhere. Websites and info about what you have done are also very helpful.

Contact us!

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

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
RETRO GA
WITH
RASPBER
164 PAGES
VIDEO GAME PROJE
THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 5
INCLUDING RASPBERRY PI 4!
200 PAGES OF IDEAS
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

▲ We have plenty of new books for all your gift needs!

New books

I've been keeping an eye on your store to see if you have any new books coming out this year – you always seem to release a few good ones before Christmas! Any chance you could let me know if there will be another *Official Projects Book* or anything else new? They're always good presents.

Allister via email

Good news: as the magazine goes to print, we've had a few new books released. From us folks on *The MagPi*, there's the *Official Raspberry Pi Projects Volume 5*, which is 200 pages of Raspberry Pi 4 (and other models) goodness. It's perfect for newbies and people looking for what to actually do with their Raspberry Pi.

In addition, we have a book on *Retro Gaming with Raspberry Pi*, which is another 164 pages of amazing retro-gaming-themed projects and tutorials so that you can get your classic gaming fix at home or in your pocket.

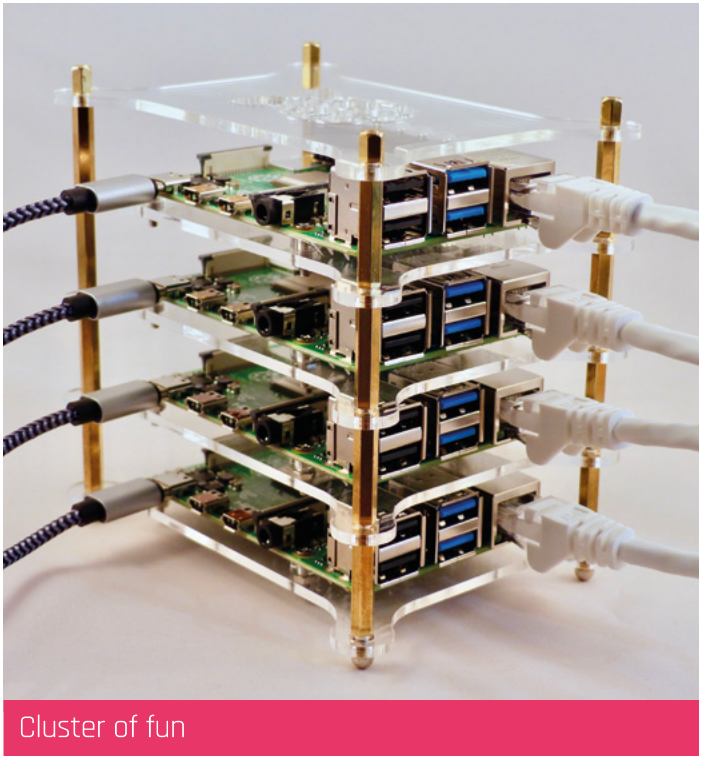
We've also updated the *Raspberry Pi Beginner's Guide* and released a brand new *Get Started with Raspberry Pi* book that comes with a Raspberry Pi 3A+! Plenty to get stuck into.

A MagPi Christmas

I want to subscribe to *The MagPi* magazine for my daughter for twelve months at the beginning of December, but wanted to know if she would receive a magazine in time for Christmas as it will be her gift. Could you let me know the date of delivery?

Kim via email

As well as this issue, we have one other issue coming out before Christmas (19 December, see more on page 97). So, depending on where you live, you will be able to subscribe for that issue in time – however, to be safe, the earlier you subscribe, the more likely you'll have a copy in time as the gift. Make sure to check out our subscription offers at **magpi.cc/subscribe**.



Cluster of fun

I like the look of the cluster computing project you did in *The MagPi*, but I do have to ask… what is the point of it? You can't really do much real computing with it.

Piers via Facebook

The cluster tutorial is more of a fun thing you can do that helps you learn computing. Sure, you're not going to be able to make a real supercomputer out of it, and even with hundreds of Raspberry Pi boards it might be a stretch. However, at least you can learn how to do it, and the kind of maths you would do with a cluster.

That being said, we are planning a more in-depth look with what you can do with it in a future issue, so watch this space.

> “We are planning a more in-depth look at what you can do with a cluster”

Strato UPS 3.0

The industrial server powered by Raspberry Pi

We have redesigned the Strato Pi UPS keeping the new Raspberry Pi 4 in mind.
The power-supply stage has been re-engineered to withstand the current demand of Raspberry Pi 4, including that of the external devices powered via USB and Strato's auxiliary power supply output.

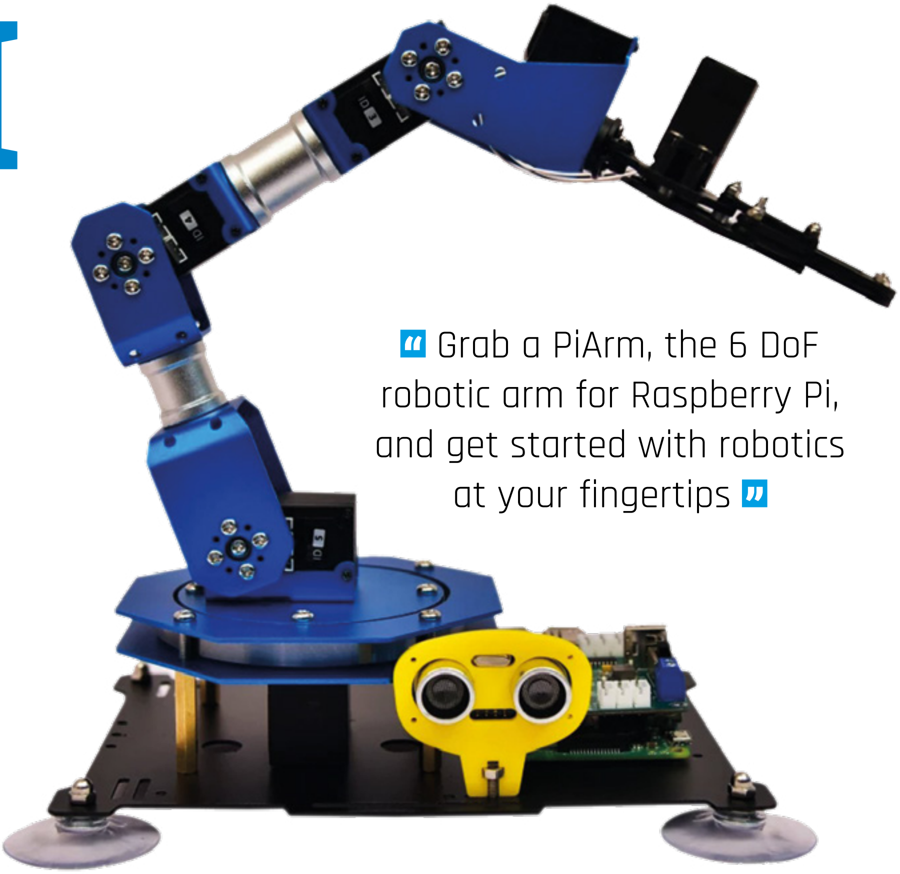
Highlights:

- 9-28Vdc power supply with surge and reverse polarity protection
- Real-time clock with replaceable back-up battery
- Optoisolated RS-232 and RS-485
- Hardware watchdog
- UPS with external lead-acid battery
- Aux UPS power supply output for external devices
- User-configurable watchdog and power management operations
- Secure element
- Buzzer
- User-upgradable firmware for the onboard micro-controller directly through Raspberry Pi

SFERALABS
www.sferalabs.cc

WIN A PIARM ROBOT ARM KIT!

We reviewed the PiArm in issue 87 of *The MagPi* and found it an impressive piece of hardware – now it's your chance to win one.



> “Grab a PiArm, the 6 DoF robotic arm for Raspberry Pi, and get started with robotics at your fingertips”

In association with
SB Components

Head here to enter: magpi.cc/win | Learn more: magpi.cc/E2BWNr

Terms & Conditions

Competition opens on **28 November** and closes on **19 December 2019**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

NEW GRAPHICAL EDITOR!

CDP STUDIO

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.

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

TOOLS FOR BETTER MAKING

MASTER DIGITAL MAKING
WITH RASPBERRY PI

THE MAGPI **#89**
ON SALE **19 DECEMBER**

Plus!

Sail a solar-powered robot boat

Make an RPI232 bulletin board

Construct a smart vision robot

DON'T MISS OUT!

magpi.cc/subscribe

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

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.org

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors
Phil King and Nicola King

ADVERTISING

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

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Sam Ribbits

Illustrator
Sam Alder

CONTRIBUTORS

Mike Cook, David Crookes, PJ Evans, Gareth Halfacree, Rosemary Hattersley, Daniel Lambton-Howard, Simon Long, Danny Staple

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

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

SUBSCRIPTIONS

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

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

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

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

Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).

Raspberry Pi PRESS

ISSN: 2051-9982.

A lifelong obsession

Lucy Hattersley has spent a long time learning to code

At the base physical level, a computer is ground-up rocks that we've tricked into thinking. It's the most remarkable achievement of the human race; one that teaches us what it means to be human.

I believe that learning to code is one of the most profound things you can do as a human on the planet today. Not just because you can get a cushy job doing something you enjoy; or even because computers run the world, so you might as well understand how they work (although that is important).

It's because teaching a machine to think helps you contemplate what you are. And learning to code (especially on a modern computer) means taking gigantic, otherworldly concepts and breaking them down into small, manageable parts. Learn to do that with a computer and you can do it all kinds of parts of life. Coding makes you a better person.

Learning the BASICs

All of this was lost to me when I first encountered a ZX81 at infant school. I remember programming a platform game where a 0 had to jump up through a moving gap in the platform above.

As with most ZX81 games, you could only write a couple of screens of code before running out of the 1kB RAM. This turns out to be a bonus when you're six years old and would prefer to go outside and play.

As I grew older, I moved on to BBC Micro at school and begged my parents for a Sinclair Spectrum at home. I loved making animated stories, simple games, and text adventures. Like most children from that era, I've paid my parents back with a lifetime of free tech support.

> “We get to work with the best computer company in the world”

School days

I remember controlling robotic turtles at school, developing my own version of Ceefax, and acid-dipping printed circuit boards.

Computing was often mayhem and even if things rarely worked, I never found the lessons dull. This is one reason why I support Raspberry Pi's low-cost endeavours.

'It's not a toy' is perhaps the worst thing a child can hear about a computer. But an incredibly well-engineered toy can get itself into the hands of young makers, and they can make anything they want with it.

Computers are more than just games consoles with keyboards. A good computer has an ecosystem of development and making that springs up naturally, both from the availability of tools and the community that grows up around it.

That community is important. We bring together the best people we can find and write about the fun stuff they're making, or the cool events they're putting on. We get to work with the best computer company in the world; on the best computer ever made. We have a lot of fun!

So thank you for putting up with me as editor of *The MagPi* magazine. I hope you've enjoyed 2019, and 2020 is going to be even better.

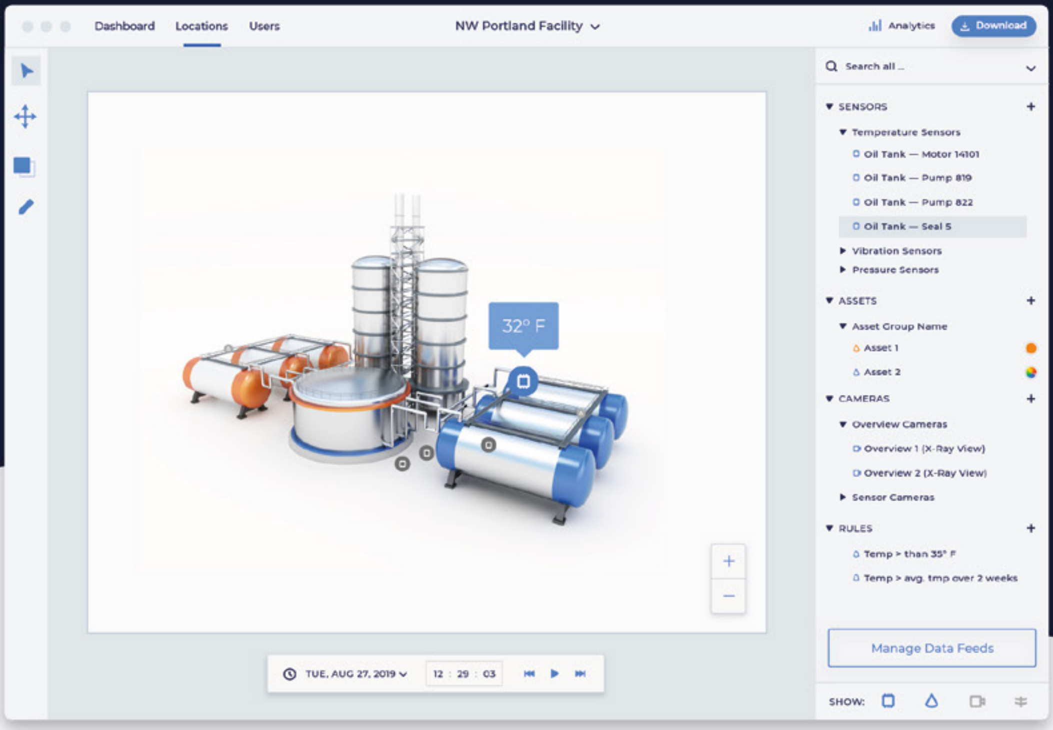
AUTHOR

Lucy Hattersley

Lucy is editor of *The MagPi*. She works with Rob, Phil, Sam, and lots of writers every month to put together this magazine and hopes you enjoy it.

@lucyhattersley

UrsaLeo



Visualize actionable data on 3D models from all your Raspberry Pi sensors.

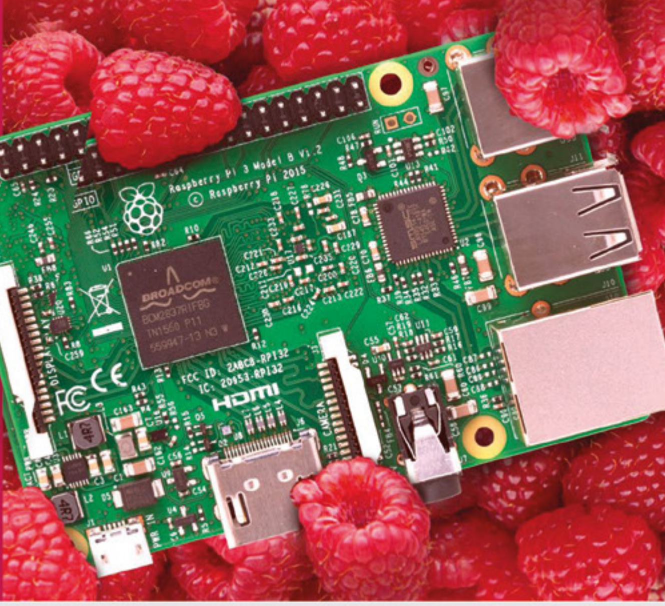
UrsaLeo technology takes Raspberry Pi sensor data and displays it on 3D models of any environment, visualizing alerts and lending context to help guide your business.

- ✓ Import your environment with a CAD drawing (Sketch Up, AutoCAD or similar) or start with photographs and measurements
- ✓ Zoom, pan and rotate through a fully rendered 3D model that combines your data with our software
- ✓ Easily place your sensor locations in the environment and define metadata for each one
- ✓ Define camera viewpoints for each sensor
- ✓ Set up alerts based on incoming data
  - Assign status colors to visualize sensor and asset information
  - Automatically jump to view any sensor generating an alert
- ✓ Multiple visualizations available, such as X-ray, rewind and replay, with more coming soon

GREAT FOR
INDUSTRY 4.0 • EDUCATION • BUILDING MANAGEMENT

Get started today
www.ursaleo.com/platform

American **Raspberry Pi** Shop



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

Partner and official reseller for top Pi brands:

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

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

USA PiShop.us

Canada BuyaPi.ca

Raspberry Pi APPROVED RESELLER