The MagPi

The official Raspberry Pi magazine

RASPBERRY PI
SMART GADGETS

Hack home devices with super small computers!

Dog Translator

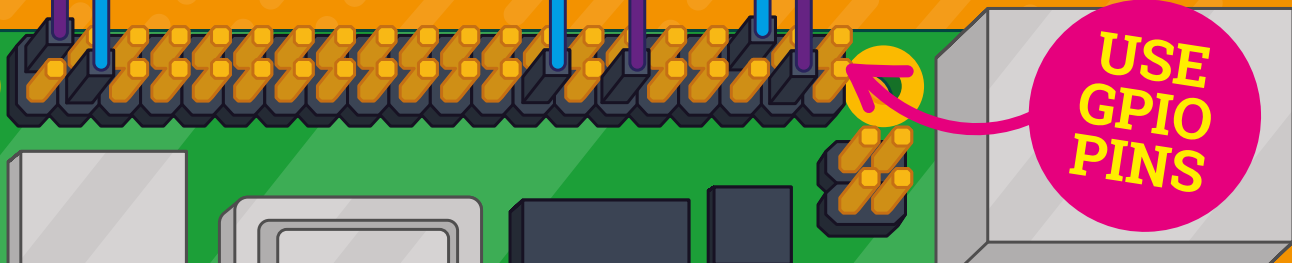HeaterMeter Barbecue

Snore Detector

Smart Bartender

Guitar Pedal

Cat Speedometer

USE GPIO PINS

Raspberry Pi PRESS

GLOBAL DELIVERY
magpi.cc/store

WIN! AIR QUALITY KITS UP FOR GRABS!
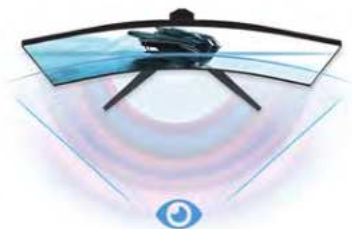
WELCOME
to *The MagPi* 110

As time goes on, I'm increasingly becoming a militant 'right to repair' advocate. I think it's right to fix and improve things, rather than chucking them in the bin and buying afresh.

It makes financial sense to build your gadgets or repair them, and you can improve classic kit with Raspberry Pi technology. With Raspberry Pi, you can make an outrageous idea work. You can't buy a dog translator, cat speedometer, or snore detector. With Raspberry Pi, your ideas – or the wild ideas of other makers – become reality.

And, it makes sense to know how something works. Build something with your own hands and it's truly yours.

Your Smart Gadgets feature (page 30) has all the information you need to transform classic devices into super-smart widgets, and build your crazy gadgets for the home. And while you're in a creative mood, don't forget Halloween is coming. Your spooky upgrades feature (page 68) is packed with ideas for entertaining local kids with electronics.

I hope you enjoy this issue. It's packed with great ideas and fun makes. Let us know what you repair, improve, or create for yourself.

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

We love our subscribers and shower them with free gifts. I suggest you take out a subscription right away so the next issue lands on your doorstep.

@LucyHattersley

GET A
RASPBERRY PI ZERO W KIT
WITH A SUBSCRIPTION!
PAGE 28

Contents

> Issue 110 > October 2021

30

20

Package Thief Deterrent

16

Solar System Display

Tutorials

The Big Feature

38

Make retro noises

62

Build an Android tablet

68

Spooky project upgrades

76

Keybow 2040

Reviews

Community

86

Matt Gray interview

WIN 1 OF 10 AIR QUALITY KITS FOR **RASPBERRY PI**

95

The next
generation of
documentation

Raspberry Pi introduces a new documentation hub. By **Lucy Hattersley** and **Alasdair Allan**

Raspberry Pi has launched the next generation of its documentation that can be viewed today (**magpi.cc/documentation**).

Writing on the Raspberry Pi blog (**magpi.cc/bringondocs**), Alasdair Allan, Head of Documentation at Raspberry Pi, says: "The documentation repository itself had been left to grow organically. When I arrived, it needed to be

▼ The page is built automatically following a push to the master branch on the GitHub page, where anybody can issue a pull request

restructured, and a great deal of non-Raspberry Pi specific documentation needed to be removed, while other areas were underserved and needed to be expanded."

Replacing Markdown

The new documentation design "comes from the same team of folks that built the excellent PDF-based documentation for Raspberry Pi Pico and RP2040."

The PDF documentation for RP2040 and Raspberry Pi Pico was built off an Asciidoc-based toolchain, and "we knew from the outset that we'd want to migrate the current Markdown-based documentation to Asciidoc. It'd offer us more powerful tools going forwards, and a lot more flexibility," says Alasdair.

Get involved

The Asciidoc source for the new documentation site is hosted on GitHub (**magpi.cc/documentationgit**), and the Documentation site is built and updated automatically using GitHub Actions when someone pushes to the master branch.

"However, we'll mostly be working on the develop branch in the repository, which is the default branch you'll now get when you take a fresh checkout, and also the branch you should target for your pull requests," says Alasdair (see **magpi.cc/documentationpullrequest**).

"We've always taken pull requests against the Markdown-based source behind our documentation site. Over the years, as the documentation set has grown, there have been hundreds of community contributors, who have made over 1200 individual pull requests, ranging from fixing small typos to contributing whole new sections.

"With the introduction of the new site, Raspberry Pi will continue to take pull requests against the new Asciidoc-based documentation.

"However, we're going to be a bit more targeted around what we'll accept into the documentation, and will be looking to keep the repository focused on Raspberry Pi-specific things, rather than having generic Linux tutorial content," writes Alasdair.

> **" If you're already a contributor, the easiest thing to do is to take a fresh checkout of the repository "**

"I'm hoping that everyone that has contributed to the documentation over the years will see the new site as a big step towards making our documentation more accessible – and, as ever, we accept pull requests. However, if you're already a contributor, the easiest thing to do is to take a fresh checkout of the repository, because things have changed a lot." M

▲ The new documentation page is built with Asciidoc and is more focused on Raspberry Pi

▲ The new Astro Pi looks similar to the original, although the camera gives it away

New Astro Pi in space
for a new mission

The next series of Astro Pi missions have begun, and will have a hardware upgrade as well

Mission Space Lab and Mission Zero are back once again: the challenge where teams of students send code to the ISS (International Space Station) to be run on a couple of Raspberry Pi systems that are installed on the station. These Astro Pi were launched in 2015, though, and are long due an upgrade.

Not only will the new Astro Pi computers use a Raspberry Pi 4, they'll also receive a powered-up Sense HAT and more sensors and features for students to use. This includes a High Quality Camera, Coral machine learning accelerator, colour and luminosity sensor, and a passive infrared sensor. This joins the current sensors of a gyroscope; an accelerometer; a magnetometer; and humidity, temperature, and pressure sensors.

"The new hardware makes it possible for teams to design new types of experiments," the Astro Pi team says. "With the Raspberry Pi High Quality Camera they can take sharper, more detailed images and, for the first time, teams will be able to get full colour images of the beauty of Earth from space. This will also enable

teams to investigate plant health with the higher quality optical filter used in conjunction with the IR-sensitive camera. Using the Coral machine learning accelerator, teams will also be able to develop machine learning models that enable high-speed, real-time processing."

Taking part in Astro Pi

There are two Astro Pi missions to choose from: Mission Zero and Mission Space Lab. Young people can participate in one or both of the missions, and entry is completely free.

Mission Zero

In Mission Zero, young people write a simple Python program that takes a sensor reading and displays a message on the LED screen. This year, participation in Mission Zero also gives young people the opportunity to vote for the names of the two new Astro Pi units being sent to the ISS. Mission Zero can be completed in around an hour, and is open to anyone aged 7 to 19 years old. Every eligible entry is guaranteed to run on board the ISS

▲ Team Jupiter took this photo of Lake Balkhash in Kazakhstan

▲ The Coral hardware is a little add-on that can be used with Astro Pi.

> **There are two Astro Pi missions to choose from: Mission Zero and Mission Space Lab**

and participants will receive an official certificate with the exact time and location of the ISS when their program ran.

Mission Zero runs until 18 March 2022, and you can read more here: **magpi.cc/missionzero**.

Mission Space Lab

Mission Space Lab is for teams of young people who want to run their own scientific experiments on the Astro Pi units aboard the ISS. It runs over eight months in four phases, from idea registration to data analysis.

Have a look at the winning teams from last year (**magpi.cc/astropi21**) for amazing examples of what teams have investigated in the past. But remember – the new Astro Pi units offer exciting new ways of investigating life in space and on Earth. Mission Space Lab team mentors just need to send their team's experiment idea by 29 October 2021: **magpi.cc/missionspacelab**. Ⓜ

▲ A beautiful ocean photo by Team Mechabot - with the HQ camera, future photos will be awesome

Upgrade in Numbers

COMPUTER: Raspberry Pi B+ - 700MHz processor with 512 MB RAM	**COMPUTER:** Raspberry Pi 4B - Quadcore 1.5 GHz processor with 8GB RAM
MACHINE LEARNING ACCELERATOR: None	**MACHINE LEARNING ACCELERATOR:** Coral
CAMERA: Raspberry Pi Camera Module - 5MP sensor, 2592 x 1944 resolution	**CAMERA:** High Quality Camera - 12.3 MP, 4056 x 3040 resolution
SENSORS: Gyroscope, accelerometer, magnetometer, temperature, humidity, pressure	**SENSORS:** Gyroscope, accelerometer, magnetometer, temperature, humidity, pressure, colour, luminosity, passive infrared
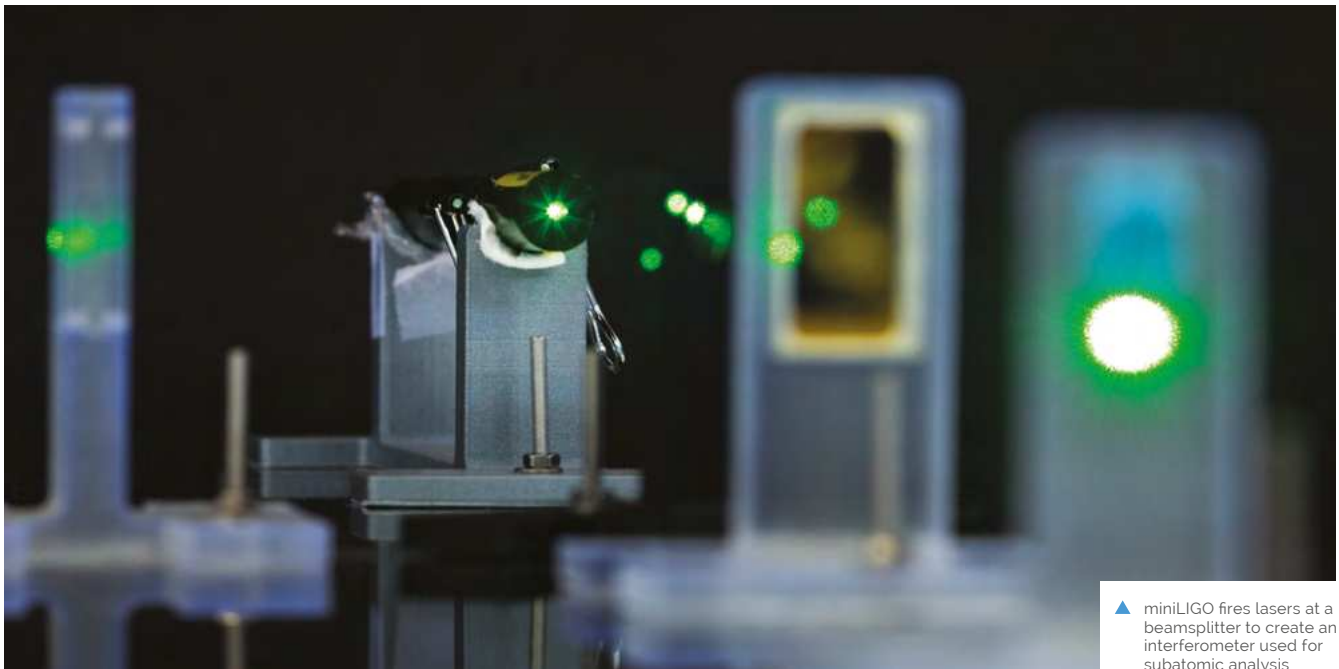
▲ miniLIGO fires lasers at a beamsplitter to create an interferometer used for subatomic analysis

miniLIGO
gravitational
wave detector

A Nobel-prize winning concept inspired PhD students to make their own gravitational wave tracker using Raspberry Pi. **Rosie Hattersley** explains

MAKER

Husni Almoubayyed

Syrian-born PhD student Husni uses code and data to study the universe and often combines them with his Raspberry Pi maker projects

magpi.cc/miniligo

After hearing about the science prize-winning LIGO (Laser Interferometer Gravitational-wave Observatory, magpi.cc/ligo) six PhD students at Carnegie Mellon University in Pittsburgh decided to see whether they could emulate the idea.

LIGO detects the gravitational waves in space caused by celestial objects colliding and predicted by Albert Einstein in his General Theory of Relativity.

Understanding space has always interested Husni Almoubayyed, so he was attracted to the idea of something that can monitor gravitational effects "stretching and contracting space around us on a very tiny scale". While there was no expectation of building their own LIGO device, not least because of the expense involved, Husni and fellow students wanted to help communicate to a wide audience

how LIGO works. miniLIGO is designed to be a "simplistic and affordable" prototype mimicking the laser-interferometer gravitational wave observatory experiment that won the Nobel Prize in physics in 2017.

Sensitive subject

The team had an advantage because Husni had worked on the original LIGO project when he was studying at Glasgow University as part of a scientific collaboration. "While I mostly worked on data analysis at that time, I wanted to experience some of LIGO's engineering challenges," he says, of which one is creating an interferometer (**magpi.cc/ligoifm**) for highly detailed analysis work. For example, CalTech's LIGO interferometers can measure a distance 1/10,000th the width of a proton!
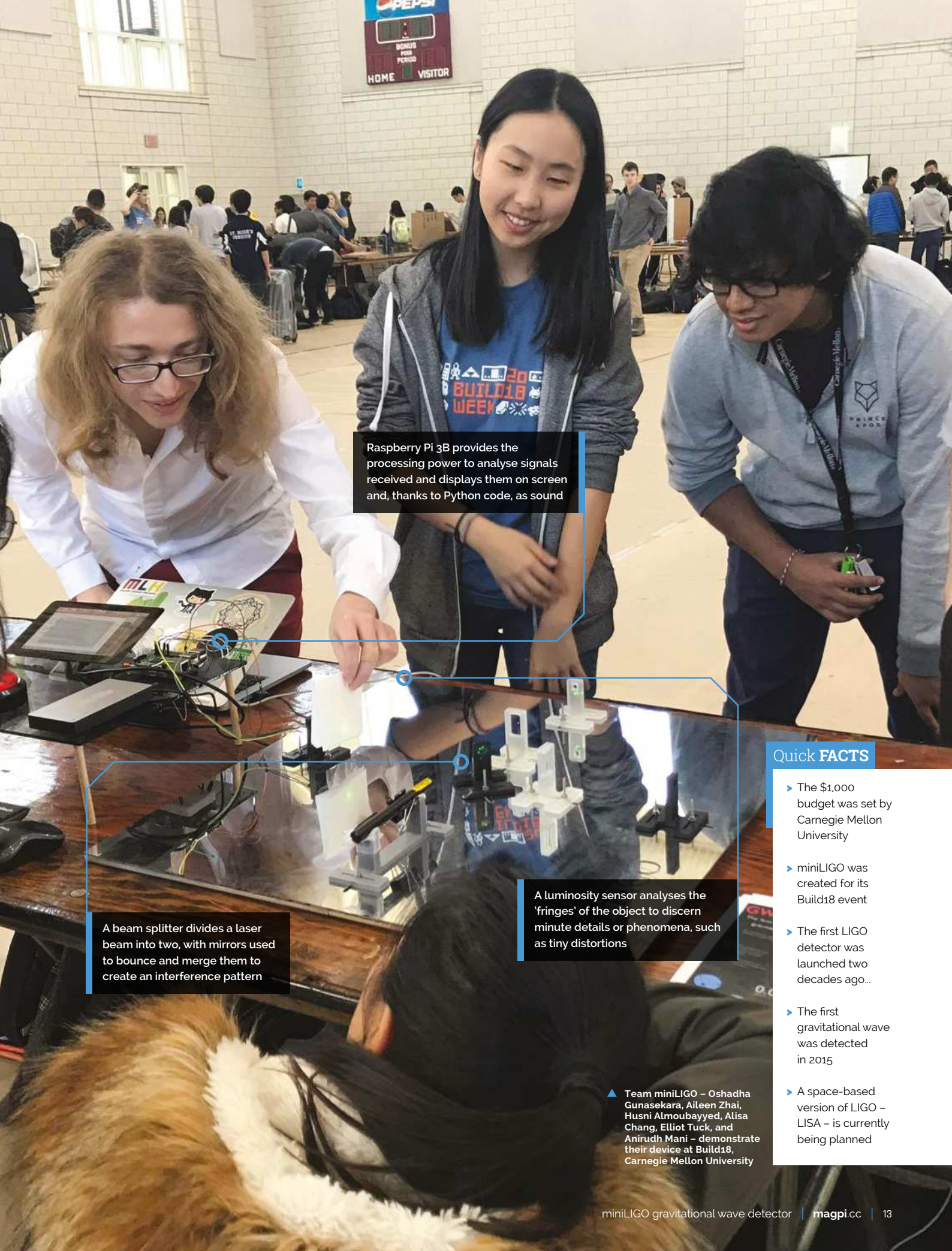
Raspberry Pi 3B provides the processing power to analyse signals received and displays them on screen and, thanks to Python code, as sound

A beam splitter divides a laser beam into two, with mirrors used to bounce and merge them to create an interference pattern

A luminosity sensor analyses the 'fringes' of the object to discern minute details or phenomena, such as tiny distortions

Quick FACTS

> The $1,000 budget was set by Carnegie Mellon University

> miniLIGO was created for its Build18 event

> The first LIGO detector was launched two decades ago...

> The first gravitational wave was detected in 2015

> A space-based version of LIGO – LISA – is currently being planned

▲ Team miniLIGO – Oshadha Gunasekara, Aileen Zhai, Husni Almoubayyed, Alisa Chang, Elliot Tuck, and Anirudh Mani – demonstrate their device at Build18, Carnegie Mellon University

▶ Schematic of the interferometer used for miniLIGO
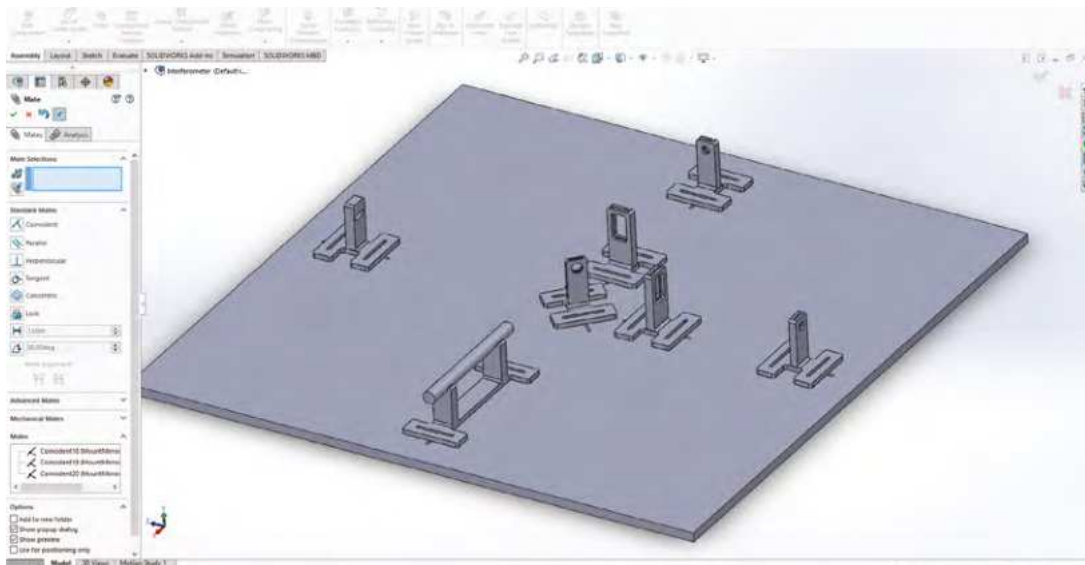
▼ The base is made of 2-way mirrored acrylic and laser-cut with annotations to indicate where the 3D-printed parts should fit
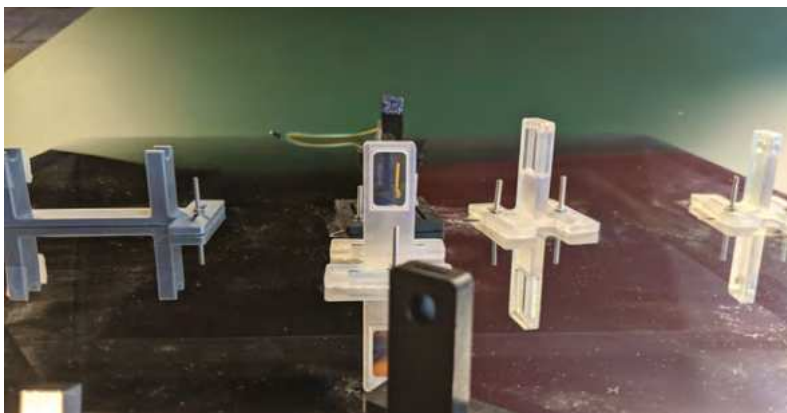
> ❝ LIGO has an extremely accurate damping system that works to shield it from vibrations and seismic changes ❞

▼ Putting together miniLIGO ahead of its public demonstration
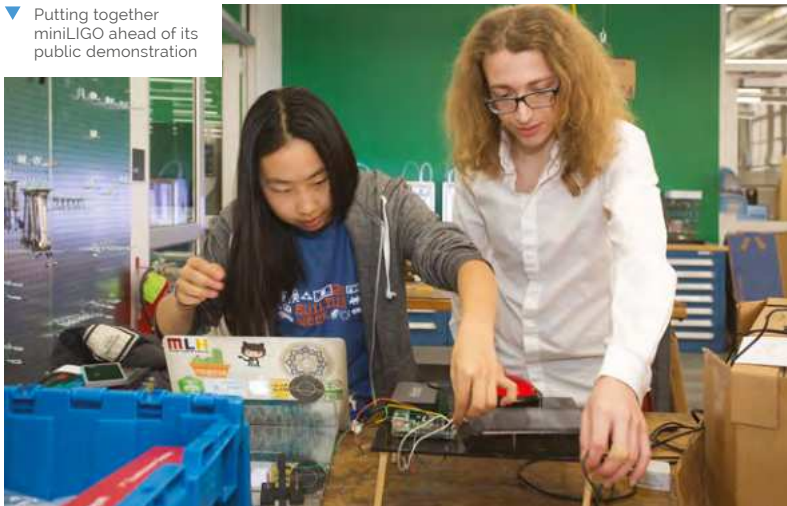
"LIGO is an extremely expensive and large-scale experiment," says Husni, and some major decisions were needed about how to create a prototype cost-effectively. With the aim of building the miniLIGO project for less than $1000, it was clear that some compromises would be needed, while mimicking the original LIGO concept as closely as possible.

"LIGO has an extremely accurate damping system that works to shield it from vibrations and seismic changes," Husni explains. "We considered using a six-axes motion sensor as a low-cost alternative, and to correct the signal computationally using the sensor, but ultimately decided to leave it out as we deemed that it would not be a good use of time and resources as stronger sources of noise would still dominate the signal."

Having used microcontrollers and Raspberry Pi for wearable computing and music-based projects, the possibilities of single-board computers were clear. (One of the other projects you'll find on the **husni.space** website is a graphical tablet that makes music with a corresponding artistic visualisation.) "Raspberry Pi allows us to easily connect all the pieces of the project, and has enough processing power to do signal processing in real time," he says. Raspberry Pi provides easy interaction with the light sensor, while also having plenty of processing power to analyse and process wave signals in real-time. It also meant the project could be portable – the team successfully ran miniLIGO from a battery pack.

Getting set up
A Raspberry Pi 3B, connected to a portable battery and a Bluetooth speaker, and a GPIO library are at the heart of miniLIGO. The team used an Adafruit High Dynamic Range Light Sensor and

The two perfectly reflective mirrors are used to build the Michelson interferometer
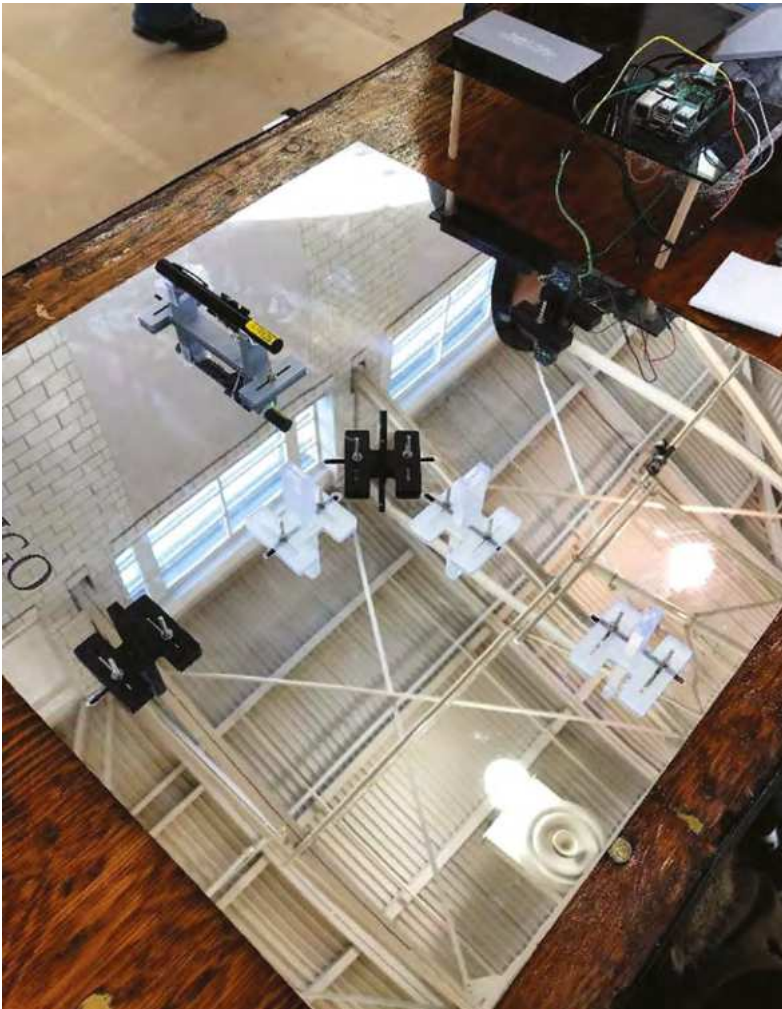
optical equipment, including the lenses and laser from Edmund Optics, then added a monitor and a speaker. Access to the university 3D printing lab meant they were able to 3D-print their own case.

When it came to writing software for miniLIGO, they decided to use Python code, largely because of the "mature and easy-to-use" Python library, from which they chose some of the common scientific computing libraries such as numpy, scipy, and matplotlib.

When miniLIGo's sensors detect a signal, its software analyses it, visualises it in real-time and plays a sound representation of it in Python. The team also wrote the software to simulate a gravitational wave signal whenever the red button on the device is pressed.

Husni explains: "miniLIGO is based on a Michelson interferometer setup that has several applications in physics and astronomy. Although it is nowhere near precise enough for gravitational wave detection, miniLIGO can be used to detect sources of noise that LIGO encounters, such as when lightning strikes nearby." M

Warning!
Laser pointer

Be careful when using laser pointers in your projects and don't aim a laser pointer at a person's head.

magpi.cc/lasersafety

Make it mini

To make your own miniLIGO, you'll need a Raspberry Pi 3B powered by a portable battery pack. The code can be found at: **magpi.cc/miniligogit.**

01 A high distance laser and a 50:50 beam splitter splits the laser into the two arms of an interferometer. You'll need two mirrors and two 75:25 beamsplitters to create two Fabry-Perot cavities.

02 Use an acrylic board as the base for the interferometer, and 3D-print or attach holders to it at exactly equal heights. Use an Adafruit High Dynamic Range Sensor or a Raspberry Pi HQ Camera to capture movements.

03 Attach a screen to show visualisations and a Bluetooth speaker to alert you to any waves detected. You could also add a trigger button to initiate demo visualisations.

Package
Thief Deterrent

Here's a decidedly tongue-in-cheek make that is an extremely humorous take on a home security system. **Nicola King** stands well clear

MAKER

Ryder, aka 'Ryder Calm Down'

Ryder is a software engineer who likes to build ridiculous inventions in his spare time and document them on his YouTube channel Ryder Calm Down.

magpi.cc/ryderyt

Many of us are increasingly ordering goods online, with the consequence that more and more parcels and boxes are landing on doorsteps everywhere. From a security aspect, it can be a little worrying when our precious new purchases are left in plain sight waiting for us to return home. A fun and light-hearted project by Canadian maker Ryder shows one unique way of handling it.

Flour power

Following the theft of a package from his porch earlier in the year, Ryder decided to take action. "It had never happened to me before and I was a little frustrated," he recalls. "I'd been doing some image classification work at the time, and figured this would be a great opportunity to build something to deter people from stealing packages from my porch."

> **"** It's great at detecting whether there is or is not a package, but it also sometimes recognises my cats as packages! **"**

▲ A Raspberry Pi 4 runs the software, with a relay board used to control 12 V components such as the flashing light, siren, and sprinkler

Armed with a Raspberry Pi 4, a security camera, and various other ingredients, he set about making something that certainly caused a stir in the neighbourhood, as his must-watch YouTube video demonstrates (**magpi.cc/packagetheft**). "I built the project more for entertainment value than actual use (since I've only ever had one package stolen), but I consider it a success either way," he tells us.

So, how does the alarm system work? Using a security camera pointed at Ryder's door, images from the camera are pulled by his Raspberry Pi and processed by a custom machine learning model to detect if there is or isn't a package.

"If a package has been taken unexpectedly, Raspberry Pi sends signals to a relay (via its GPIO pins) to activate a variety of alarms to entice the 'thief' to drop the package," Ryder explains. Those surprises for thieves include a sprinkler, a loud truck horn, and a flour shower.

"This is definitely a great project if you're looking to get your feet wet into machine learning – though I don't recommend you implement the sprinkler, flour, or air horn components," he says. "All the participants in my video were friends and neighbours of mine, and implementing the exact same setup for the public will likely get you into some sort of trouble," he cautions. However, he does suggest that you can modify the project to send you an email when a package arrives, or has been taken unexpectedly, which may be a more prudent route to take!

Averting a cat-astrophe

Computer vision is central to Ryder's project, and he used Google Cloud AutoML to train the machine learning model. "I wanted something to include in my video that was easiest for those getting started with machine learning to use," he reveals. "Training a model yourself can be quite intimidating."

A garden sprinkler is activated via a relay to spray the offender with water

When a package theft is detected by the camera and computer vision, a red flashing light and siren are triggered

Ryder also added a flour blaster using compressed air and a solenoid

Warning!
Don't Assault

You may get into trouble if you attack innocent porch visitors with flour, or anything else for that matter. Do not deploy that aspect of the project in a real-world environment.

Quick **FACTS**

> This project took Ryder around a month to complete.

> You'll need quite a few items to make this…

> "A combination of Python, TensorFlow, and the last of my sanity!"

> Find the code on GitHub: **magpi.cc/ packagetheftgit**

> He's working on a Raspberry Pi-powered autonomous car

It wasn't plain sailing to get this system working as he wanted. Ryder originally had all alarms and sprinklers go off at the same time, but decided to move them to separate threads with separate timings in order to best soak his very obliging friends and neighbours. "The model [is] great at detecting whether there is or is not a package," he says "but it also sometimes recognises my cats as packages!" He also sensibly trained the system to disarm when he arrived to pick up his parcels.

Many co-operative friends and neighbours have embraced his undertaking, "though I did buy my neighbours a few boxes of chocolate afterwards to say sorry for the noise", Ryder admits. For the time being, he has dismantled this project, "since I don't really have the need for it, but I'm brainstorming some version two ideas – one of which involves a net." Any future front-step felons may be in for a shock! **M**

▲ With a camera pointed down at the porch, Ryder trained a machine learning model to spot when a package was present – and recognise known faces to disarm the alarm

▲ Wiring up the capture suit components

Machine Learning
Prosthetic Arm

This intelligent arm learns how to move naturally, based on what the wearer is doing, as **Phil King** discovers

MAKER

James Bruton

Former toy designer, current YouTube maker, and general robotics, electrical and mechanical engineer, James is a fan of doing it yourself and innovation by trial and error.

magpi.cc/ jamesbruton

Known for his robotic creations, popular YouTuber James Bruton is also a keen Iron Man cosplayer, and his latest invention would surely impress Tony Stark: an intelligent prosthetic arm that can move naturally and autonomously, depending on the wearer's body posture and limb movements.

"It's a project I've been thinking about for a while, but I've never actually attempted properly," James tells us. "I thought it would be good to have a work stream of something that could be useful."

Motion capture suit

To obtain the body movement data on which to base the arm's movements, James considered using a brain computer, but this would be unreliable without embedding electrodes in his head! So, he instead opted to train it with machine learning.

For this he created a motion capture suit from 3D-printed parts to gather all the data from his body motions: arms, legs, and head. The suit measures joint movements using rotating pieces with magnetic encoders, along with limb and head positions – via a special headband – using MPU-6050 inertial measurement units and Teensy LC boards.

Collected by a Teensy 4.1, this data is then fed into a machine learning model running on the suit's Raspberry Pi Zero using AOgmaNeo (**magpi.cc/aogmaneo**), a lightweight C++ software library designed to run on low-power devices such a microcontrollers.

"AOgmaNeo is a reinforcement machine learning system which learns what all of the data is doing in relation to itself," James explains. "This means that you can remove any piece of data and, after training, the software will do its best to replace the missing piece with a learned output. In my case, I'm removing the right arm and using the learned output to drive the prosthetic arm, but it could be any limb."

While James notes that AOgmaNeo is actually meant for reinforcement learning, "in this case we know what the output should be rather than it being unknown and learning through binary reinforcement."

To train the model, James used distinctive repeated motions, such as walking, so that the prosthetic arm would later be able to predict what it should do from incoming sensor data. He also spent some time standing still so that the arm would know what to do in that situation.

New model arm

With the machine learning model trained, Raspberry Pi Zero can be put into playback mode to control the backpack-mounted arm's movements

Part of the motion capture suit, the headband is equipped with an IMU to gather movement data
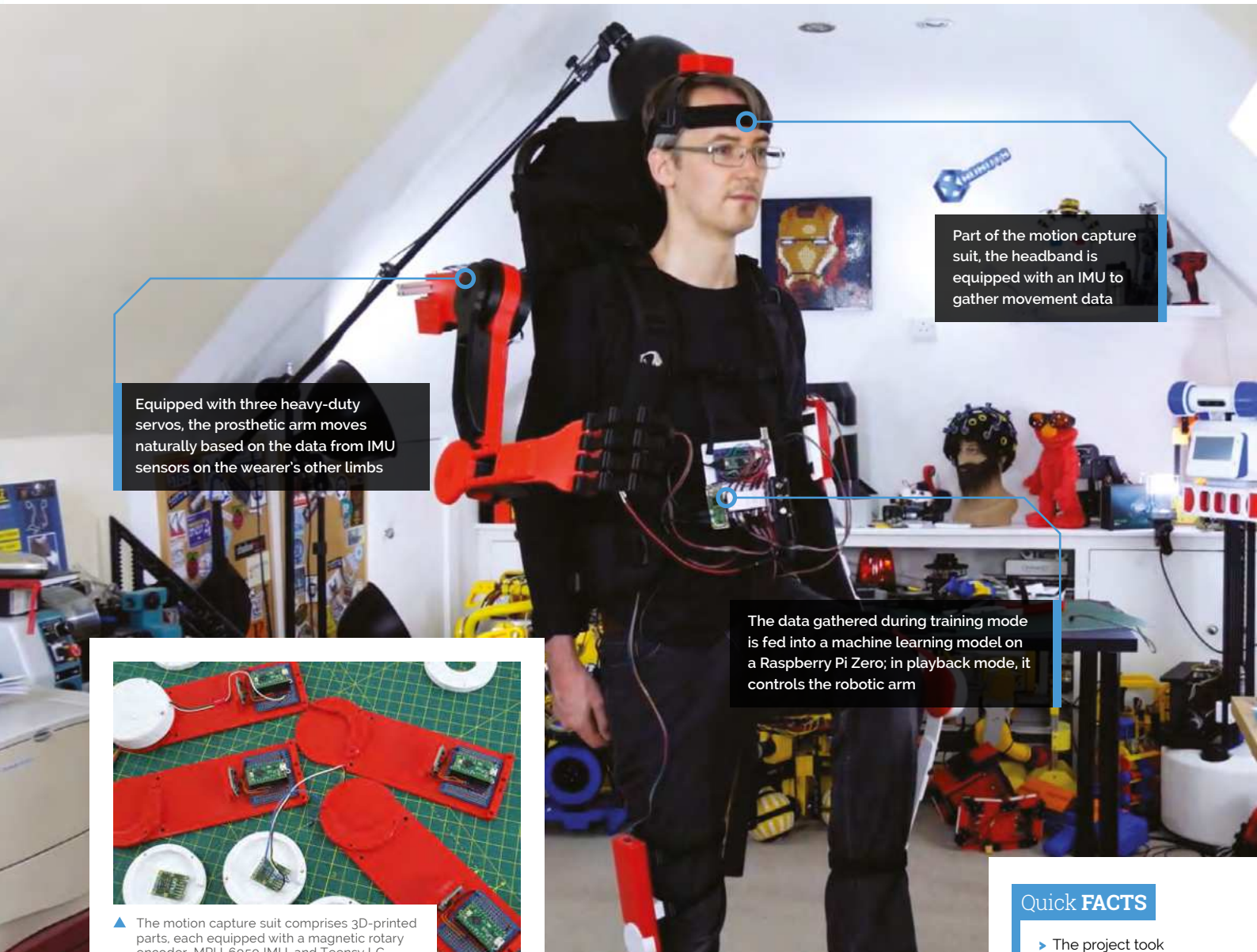
Equipped with three heavy-duty servos, the prosthetic arm moves naturally based on the data from IMU sensors on the wearer's other limbs

The data gathered during training mode is fed into a machine learning model on a Raspberry Pi Zero; in playback mode, it controls the robotic arm

▲ The motion capture suit comprises 3D-printed parts, each equipped with a magnetic rotary encoder, MPU-6050 IMU, and Teensy LC

❝ Raspberry Pi Zero can be put into playback mode to control the backpack-mounted arm's movements intelligently ❞

intelligently. It can then duplicate what the wearer's real right arm was doing during training depending on the positions and movements of other body parts.

So, as he demonstrates in his YouTube video (**magpi.cc/mlarmyt**), if James starts walking on the spot, the prosthetic arm swings the opposite way to his left arm as he strides along, and moves forward as raises his left leg. If he stands still, the arm will hang down by his side. The 3D-printed hand was added purely for aesthetic reasons and the fingers don't move.

James admits that the project is highly experimental and currently an early work in progress. "I'd like to develop this concept further," he says, "although the current setup is slightly overambitious and impractical. I think the next step will be to have a simpler set of inputs and outputs."

While he generally publishes his CAD designs and code, the arm "doesn't work all that well, so I haven't this time. AOgmaNeo is open-source, though (free for personal use), so you can make something similar if you wished." What would you do with an extra arm? Ⓜ

Quick **FACTS**

> The project took James a few months to create

> He designed and 3D-printed the robot arm parts

> It uses three high-torque XM540-W270-T servos

> The arm is mounted on 2020 aluminium extrusions

> These are fitted to a standard frame backpack

Solar System Display

Dmytro Panin has taken one small device and one giant leap towards creating a near-perfect digital orrery, as **David Crookes** explains
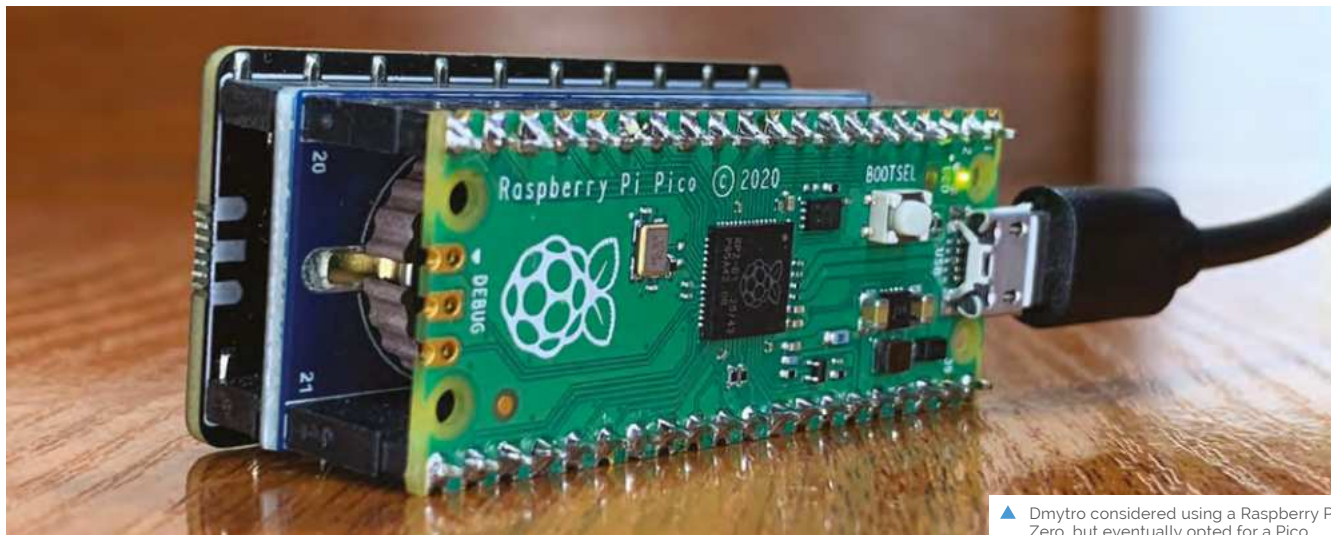
▲ Dmytro considered using a Raspberry Pi Zero, but eventually opted for a Pico

MAKER

Dmytro Panin

Dmytro is a programmer based in Kyiv, Ukraine, and he wrote his first line of code aged eight. He works for a large provider of nearshore software engineering services.

magpi.cc/ picosolar

For centuries, astronomers have used orreries to simulate the motion of the planets around the sun. These mechanical models of the Solar System originated around 150 BCE, but they gained their name when English nobleman Charles Boyle, the Fourth Earl of Orrery, commissioned his own planetarium in 1713 based on a design by clock makers George Graham and Thomas Tompion.

Dmytro Panin would love to have one. "They're a bit on the bulky side, but I may attempt to build an orrery one day," he says. In the meantime, he has produced a digital version that is equally captivating and useful, powered by a Raspberry Pi Pico microcontroller. "Astronomy has always fascinated me, and I wanted something that I could look at and see the planets in the Solar System progress over time," he continues. "So I built a visual reminder."

A star build

His project is out of this world. As well as Pico, it uses two other main components: a Waveshare Precision RTC expansion module with an on-board DS3231 chip and a Pimoroni Pico Display Pack. This allows the device to calculate the planets'
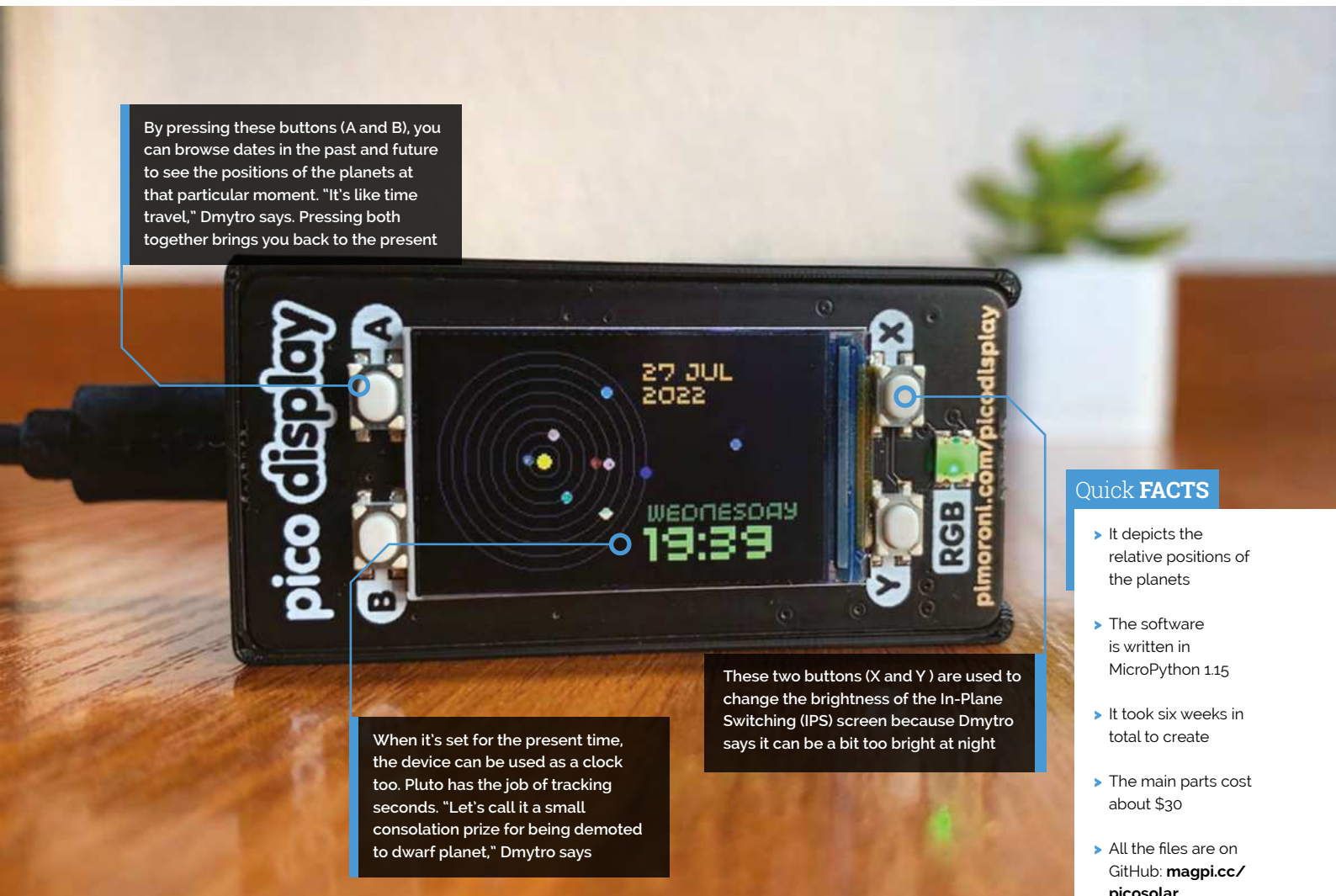
positions based on the current time and date. It then presents the planets revolving around the sun, showing their relative circular orbits.

> ❝ The cool part is that no external data is needed ❞

"The cool part is that no external data is needed," Dmytro says. "The positions of the planets are calculated on board and the only thing that the system knows is the current time, which comes from the RTC module." It makes the build a relatively straightforward one to replicate and Dmytro has been more than happy to make it open-source. He even shares his design for a 3D-printed case.

But why choose the Raspberry Pi Pico for this project? "Microcontrollers are amazing: you have a tiny object which can execute your commands and do virtually everything with extremely low power consumption. When I found out about the new RP2040 microcontroller and checked the specs, I knew it could became part of my next build," he says. "I was particularly interested

By pressing these buttons (A and B), you can browse dates in the past and future to see the positions of the planets at that particular moment. "It's like time travel," Dmytro says. Pressing both together brings you back to the present

These two buttons (X and Y) are used to change the brightness of the In-Plane Switching (IPS) screen because Dmytro says it can be a bit too bright at night

When it's set for the present time, the device can be used as a clock too. Pluto has the job of tracking seconds. "Let's call it a small consolation prize for being demoted to dwarf planet," Dmytro says

in using MicroPython instead of C to program a microcontroller."

Space to think

Dmytro has also added some neat flourishes to his creation. To start with, he wanted to figure a role for Pluto – the celestial body demoted to the status of dwarf planet by the International Astronomical Union in 2006. "I decided to represent the seconds by having Pluto bounce, starting from the full amplitude to just rolling at the bottom of the screen when a minute is about to end," he says.

He also made use of the Pico Display's tiny buttons so that past and future planetary positions can be viewed on the screen. Most of the time, however, it just sits on a shelf in his line of sight, allowing him to keep track of the current positions. "It means I can always see it," he says. "But the downside is that I can end up mindlessly staring at Pluto jumping around without realising how much time has passed."

▲ With just three main components and a 3D-printed case, the Solar System Display is a simple, yet very effective, project

Cyberdeck portable
entertainment system

Action adventurers and sci-fi tales inspired a Raspberry Pi-concealing portable box of tricks. **Rosie Hattersley** peers inside

MAKER

Justin Forsythe

Tennessee-based Justin's forays into making sparked his MA in media psychology, investigating the relationship between people and technology.

Though more of a wordsmith than a tech expert, Justin Forsythe had enough of an interest in technology to take a master's degree focusing on our relationship with it. "I've always been interested in how things work," he says simply. The maker community caught his attention and, starting with soldering, he began to investigate the uses of PCBs and Raspberry Pi. Inspiration for what would become the Cyberdeck soon arrived.

Encouraged by the success of his first electronics project in which he revived and added capacity to the HDMI ports of his Onkyo audio receiver, Justin began to cast around for what to make next. "The versatility, affordability, and size of Raspberry Pi really made it a smart buy during a time when I wasn't sure what my next project would be," he says. He knew it was to be something practical. He also found the entertainment factor of RetroPie and other retro gaming emulators intriguing, and had seen examples in portable cases.

Electric dreams

"Somewhere around this time" he began reading the novel *Neuromancer*, "which elicited images of clandestine hackers whipping out cases that contain computers with powerful capabilities. I've always been a fan of James Bond and *Mission Impossible*. This definitely helped put my head in the right place for making something fun and useful."

Not content building a 'toy without a real-world application', he decided to make use of the word processing and internet access possibilities
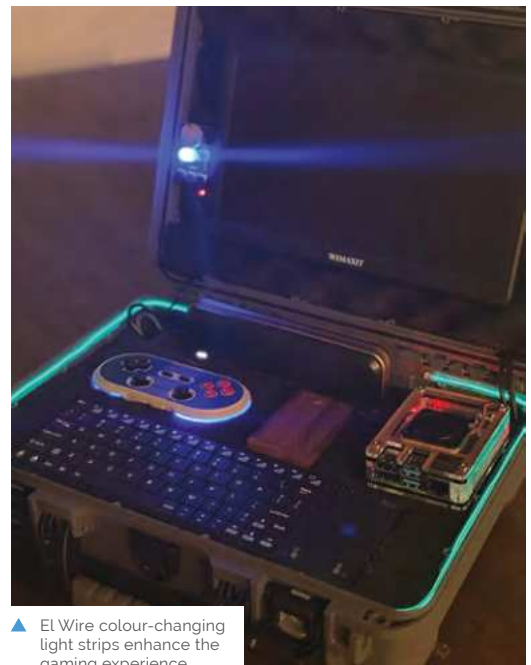
▲ El Wire colour-changing light strips enhance the gaming experience

of Raspberry Pi OS, and to combine that with a gaming pad inside a ruggedised case. "A self-contained system with wireless access also has value in an emergency situation," he reasons, suggesting it's a practical option to take on a camping, fishing, or sailing trip. What would make his idea less like reinventing the laptop was having the game station and PC parts sit alongside

Kodi TV streaming software provides on-the-go entertainment

A separate control panel is used to adjust the neon lighting scheme

Justin chose a classic wood-effect enclosure to hide the built-in fan which cools the adjacent Raspberry Pi 4 when the Cyberdeck is in use

Quick **FACTS**

> Justin was reading William Gibson's *Neuromancer* when inspiration struck…

> and the idea for a briefcase-based device was born

> In a further sci-fi nod, Cyberdeck boasts *Tron*-like LED trim

> A hidden lighting panel controls crafty colour changes

> All the better to amuse his kids!

▲ The bespoke decal hints at something intriguing inside, despite the case's deliberately subtle colour

each other, plus some serious attention to the Cyberdeck's aesthetics.

Raspberry Pi's size offers perfect portability for a project in which space management was critical: "The challenge [was] to arrange the components in such a way that they are attractive and conveniently oriented in relation to one another."

Case study

He sourced a Nanuk 910 waterproof case from C4Labs, which has a fan inside and pins to connect to Raspberry Pi, "so that when the board receives power, it's also being cooled". Furthermore, the case design meant "I could throw this entire project in the river, pull it out and open it, and the setup would be dry and ready for use."

❝ Justin began reading the novel Neuromancer, "which elicited images of clandestine hackers whipping out cases that contain computers" ❞

▶ The plethora of downloadable gaming ROMs inspired the idea of a dual entertainment and practical project

▲ The Cyberdeck provides plenty of fun for the family
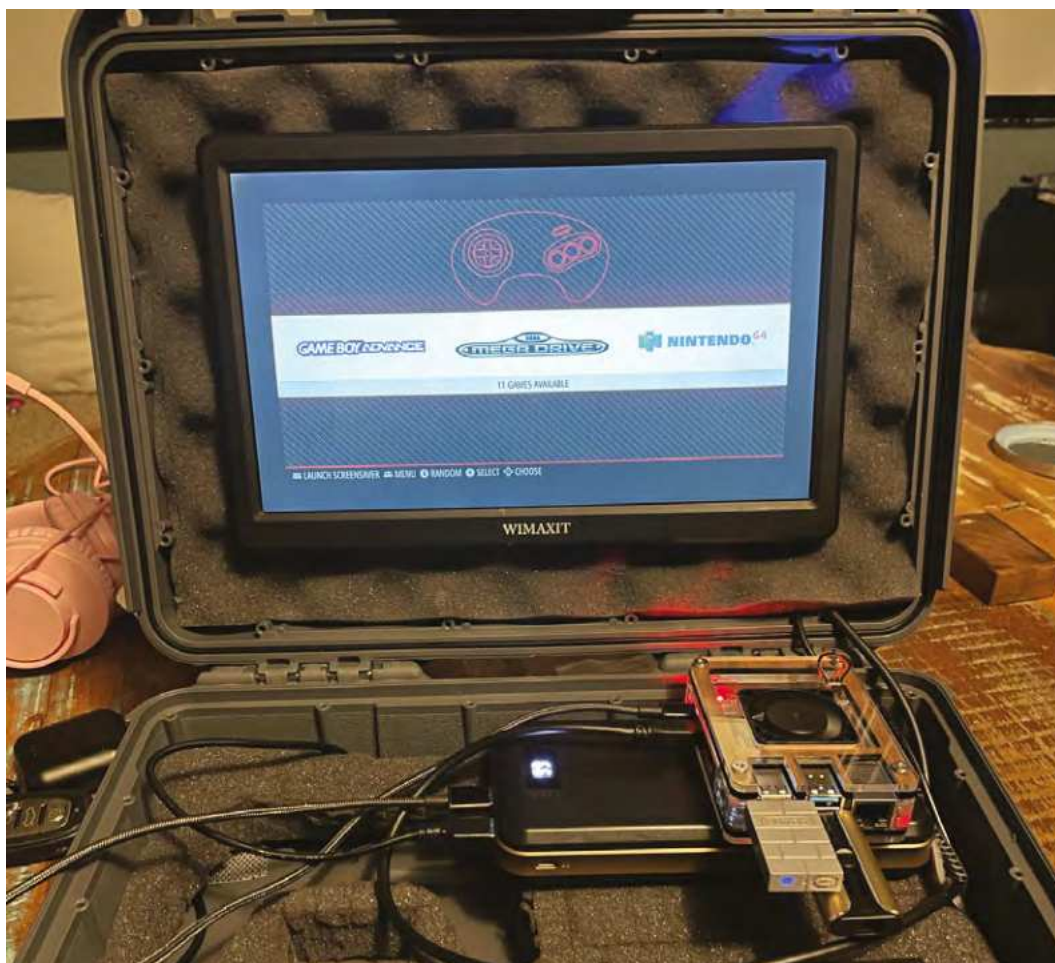
Next, Justin decided his Cyberdeck would look "so much cooler" with a keyboard. The Rii K12 model he chose, along with a 10.1-inch 1080p screen and 30,000mAh portable battery left room for a "small framing" of protective foam and a little wriggle room, if needed, to adjust the position of the cables. To his delight, there was even room for a control board for the funky light strip he'd got his eye on.

This adds to the atmosphere when the Cyberdeck is being used in retro game emulation mode to play the legal ROMs he acquired online. Since he's largely satisfied with its look and its practical, portable qualities, Justin says any future refinements are likely to be along the lines of small solar panels to boost the battery power even further, and a possible new version with a Faraday cage inside. "As to Raspberry Pi, itself, I do have a couple of ideas in the works, but I'll keep quiet for now!" he tantalises. M

On the case

A sturdy, ideally waterproof case, a handheld gaming pad, a power pack, and a small screen to connect to Raspberry Pi 3 or 4 form the basics of the Cyberdeck setup.

01 Prepare SD cards with RetroPie (and any additional entertainment apps you wish) and Raspberry Pi OS , then place it in the case along with the battery, screen, mini keyboard, and the gamepad.

02 Measure and cut foam to fit inside the case and cushion the components, strapping the screen and Raspberry Pi in place with Velcro. Add optional LED strips to enhance the Cyberdeck's visual appeal.

03 Check everything works by booting up Raspberry Pi. Reassemble in the case, connecting Raspberry Pi to the case's internal fan, then hooking up all the other components to it.

Zippy

Tomasz Burzy aimed for the stars with his latest project but reality brought his ambition back down to earth. The result is no less impressive, as **David Crookes** discovers

MAKER

Tomasz Burzy

Tomasz says he has ADHD: "I don't have the attention span for everyday tasks, yet somehow I have the focus to design, engineer, program, and produce solutions to those problems."

magpi.cc/zippy

Since February this year, NASA's Perseverance rover has been trundling across the surface of the Red Planet as part of the space agency's Mars 2020 mission. Its cameras have taken many amazing photographs of a barren landscape and eerie sky, attracting millions of viewers and inspiring a new generation of scientists and engineers.

Tomasz Burzy is among those enthralled by NASA's endeavour. "I became interested in robotics a while ago and love seeing the work of NASA's Jet Propulsion Laboratory and Boston Dynamics," he says. "The Mars rovers quickly became my favourite pieces of engineering and when I saw Perseverance, I wanted to build a replica."

❝ I only learned Python at the start of 2020 ❞

Unfortunately, Tomasz had no experience of making a remote-control vehicle, so his partner suggested he should work on something a little smaller first. This led to the creation of a tinier tank called Zippy which he based around a Raspberry Pi Zero W computer. "It's helped me become acquainted," he says – although his dream of replicating the rover is still very much in mind.

Game on

Since starting the project, Zippy has evolved and the latest version – 3.0 – is an impressive build. It makes use of a printable modular tank model designed by Nahuel Taibo (**magpi.cc/roverplatform**). It also uses an L298N H-bridge motor driver to operate 12 V high-torque metal gear motors, powered by an 11.4 V lithium polymer battery.

One version of his code uses a web server to access Zippy remotely; another allows the tank to be controlled using an Xbox One gamepad. The decision to use Raspberry Pi Zero W was primarily driven by the small amount of available room inside the plastic model, but Tomasz found it works perfectly well.

"Zippy doesn't have a lot of space inside of him after you put in the motors – once the motor driver and the battery are in there, it gets pretty cramped," he explains. "But Raspberry Pi Zero W fits in perfectly and it's a powerful enough computer to allow me to add features later on."

▼ The components, including the L298N motor driver, fit snugly inside Zippy's casing
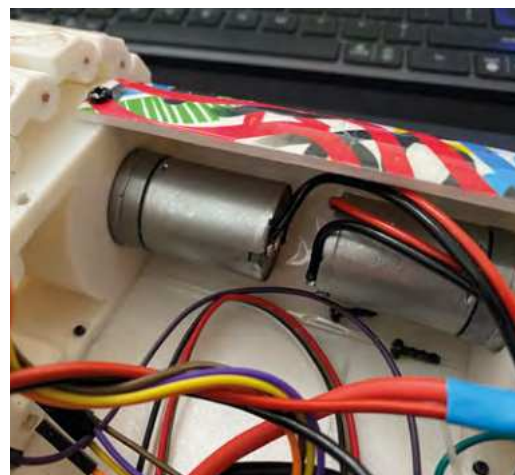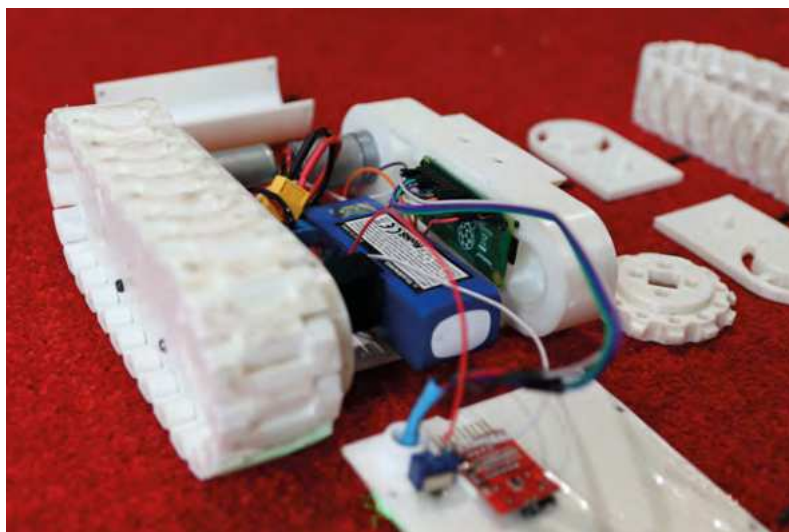
▲ The original motor gears were grinding on grass, so they were replaced with an improved set

The waterproof Osmo Action camera is capable of filming 4K HDR video and taking Ultra HD images

Tomasz's brightly coloured, 3D-printed case contains a Raspberry Pi Zero W running a script that listens for inputs before turning its GPIO pins on and off

Zippy is an agile tank that's easy to control thanks to the integration of an Xbox One gamepad

Quick **FACTS**

> The components fit into the centre of the tank

> Tomasz designed parts to hold them in place

> It took hours to get the controller working

> An RGB LED indicates what's happening in the script

> All the files are on GitHub: **magpi.cc/ zippyos**

To get things moving, Tomasz had to spend time writing the code. "I only learned Python at the start of 2020, but I've been working with electronics on and off for a while," he tells *The MagPi*. He used a Python library called approxeng.input to get the Xbox One controller working. "It lets me connect almost any control to Raspberry Pi and use its inputs," he says.

Room for a view

To top things off nicely, a camera has been placed onto Zippy. "I mounted an action camera because Zippy has an interesting low-to-the-ground point-of-view," Tomasz explains. "During one of the test drives, a fox became interested in Zippy and we got some great close-ups. Sadly, that footage was lost when my NAS failed."

Despite such a setback, Tomasz hasn't lost heart and he intends to continue to persevere with his Perseverance-inspired build by making future improvements. As it stands, Zippy struggles to ride over sand, rocks, and long grass, but it makes light work of carpets and flat ground and can zip around at a fair old speed. "I'd like to add headlights or a first-person-view camera," Tomasz reveals. The sky really is the limit for a project such as this. M

◄ Zippy struggles with some terrain, but Tomasz says it trundles across pebbles "with determination"

JOIN FOR 12 MONTHS AND GET A
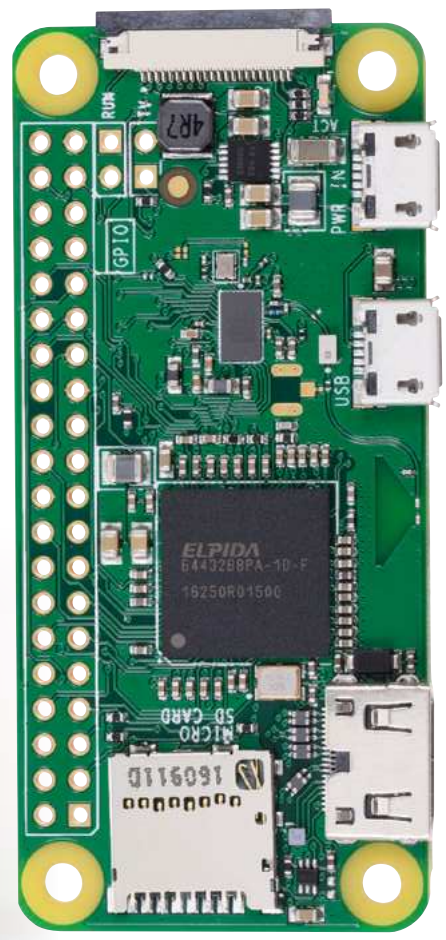
FREE Raspberry Pi Zero W Starter Kit

WITH YOUR FIRST 12-MONTH SUBSCRIPTION

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

WORTH
£20

▷ Raspberry Pi Zero W

▷ Raspberry Pi Zero W case with three covers

▷ USB and HDMI converter cables

▷ Camera Module connector

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

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

RASPBERRY PI
SMART GADGETS

Transform an ordinary object into a smart thing by putting a Raspberry Pi in it - **by Phil King**

Smart gadgets are everywhere these days, but you don't need to buy them ready-made. Using Raspberry Pi as a brain makes it easy to transform existing appliances and devices into smart ones. Or, you can even use it to build your own custom products from scratch, to suit your own precise requirements.

Here we present a range of amazing projects to create smart devices for every aspect of your life. We hope they inspire you to make your own – let us know if you do!

p33
Hello, friend
Hiya, Doggy!
Dog Translator

p33
Cat Speedometer

p35
Guitar Pedal

p34
Snore Detector

p32
HeaterMeter Barbecue

p32
Smart Bartender

Home appliances

Smart home appliances are really taking off, but you don't need to trash that old washing machine or fridge. Make it smart with a Raspberry Pi.

Washbeary Pi

Washing machines contain a microcontroller to manage their various actuators (such as the motor, heater, and drain pump) and sensors. So why not control one with Raspberry Pi instead? This way, you can create your own custom washing programs. YouTuber askDrAlex demonstrates his work in progress in a couple of videos, with Raspberry Pi controlling an old machine's motor. A similar project by tom-2015, Washberry Pi, is detailed on GitHub: **magpi.cc/washberrypi**.

> **Maker:** askDrAlex
> **url:** magpi.cc/washbearypi

ThermOS Smart Thermostat

While smart thermostats are useful for controlling your central heating system remotely, they have a few limitations and can be expensive. Needing a new smart thermostat to replace the six old thermostats controlling his complex home heating system, Joe opted to build his own powered by a Raspberry Pi 4. Using the HAP-Python HomeKit framework, he can control his system's six heat zones for different rooms from his iPhone. How cool, or hot, is that?

> **Maker:** Joe Truncale
> **url:** magpi.cc/thermos

Refrigerator/ Freezer Monitor

After his wife's carefully preserved breast milk was ruined by a freezer failure, along with three racks of ribs, Rick was determined to create a smart alert system for his fridge. Costing around $30, it uses a BME280 sensor to monitor the fridge or freezer's temperature, linked via a flat cable to a Raspberry Pi Zero W outside the appliance. Data is reported to a web dashboard, and a high-temperature alert text can be sent to a phone.

> **Maker:** Rick Kuhlman
> **url:** magpi.cc/fridgemonitor

Water Dispenser

After calculating that he spent 15 hours a year filling water bottles from his fridge's painfully slow dispenser, Chris set about building an intelligent water dispenser. With a Raspberry Pi 4 controlling a solenoid switch for a water pump, connected to the kitchen sink supply via a long tube, it can automatically fill three sizes of water bottle. Housed in a sleek black 3D-printed casing, dubbed the 'Hydrobot 5000', it looks really cool too.

> **Maker:** Chris Courses
> **url:** magpi.cc/waterdispenser

> "Rick was determined to create a smart alert system for his fridge"

RELATED PROJECTS

- Raspberry Pi baby monitor **magpi.cc/babycam**

- Uber Home Automation **magpi.cc/uberhome**

- Smart Waste Bin **magpi.cc/smartbin**

- KDE Plasma Bigscreen TV **plasma-bigscreen.org**

⚠️

Warning!
Electrical Safety

Please be careful when working with electrical projects around the home. Especially if they involve mains electricity.

magpi.cc/ electricalsafety

Food and drink

I f you want your Raspberry Pi to help you cook a raspberry pie, or other dishes and beverages, there's a smart project for that.

HeaterMeter Barbecue

If you want to be a true 'pitmaster', you'll want to know all about this open-source controller for your barbecue. Thermistor, or thermocouple probes, are used to monitor the grill, food, and ambient temperatures and adjust the blower to maintain the perfect cooking heat. Raspberry Pi adds a web interface and alarm notifications, so you can sit back and relax. You can buy a ready-made HeaterMeter board, or build your own using the comprehensive online guide.

Maker: Bryan Mayland
url: magpi.cc/heatermeter

BrewPi

The key to successful beer brewing is to carefully control the fermentation temperature. The documentation on the BrewPi site shows how to hack a standard fridge to convert it into a smart fermentation chamber by adding a heater and sensors. A ready-made BrewPi Spark unit monitors and controls the temperature, connected to a Raspberry Pi which logs data and provides a web interface, but it's possible to build your own DIY version. You could even use BrewPi for sous vide-style cooking.

Maker: Elco Jacobs
url: brewpi.com

> "Hack a standard fridge to convert it into a smart fermentation chamber"

RELATED PROJECTS

- Picrowave **magpi.cc/picrowave**

- Tea-Pi **magpi.cc/teapi**

- Voice-Controlled Coffee Machine **magpi.cc/voicecoffee**

- Chamber Sourdough Incubator **magpi.cc/chamber**

OnionBot Sous Chef

Keen cook Ben found himself pondering how to automate simpler kitchen tasks, such as softening onions, to save time and enable him to focus on culinary creativity. His solution is a robot sous chef with a Raspberry Pi brain. With a suspended thermal sensor array to detect the temperature of the pan below, and a Camera Module to help monitor cooking progress using machine learning, a servo motor controls the dial on the induction stove.

Maker: Ben Cobley
url: magpi.cc/onionbot

Smart Bartender

For the ultimate in luxury, you could hire your own personal bartender. But it's far cheaper to just make a robotic one using Raspberry Pi. With the ability to mix up to six ingredients together at the same time, this one can serve you anything from a rum and coke to a Long Island iced tea. You could customise the device to change ingredients and add more drinks, or even link it up to a voice assistant.
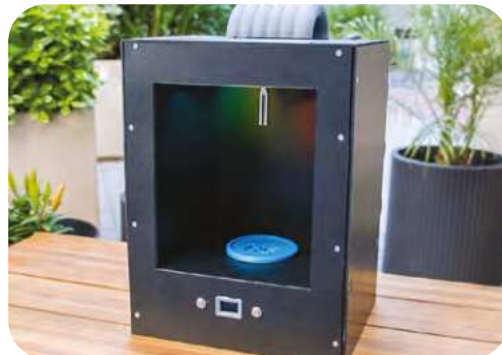
Maker: Hacker Shack
url: magpi.cc/bartender

Pets and critters

What could be better for your next pet project than making a project for your pet? Or maybe one to catch unwanted critters.

Daphne's Catflap says WASH YOUR PAWS @DaphneFlap · 4h
Most splendid Daphne, your nose is so pink and moist. My life could not be any better.

Tweeting Cat Flap

Cat flaps are ideal for letting your furry feline friend wander in and out of the house as it pleases – which, let's face it, is what cats do. But what if a flap could do more? Since Daphne was already active on social media, mainly to complain about her 'staff' Kate Bevan. Bernie Sumption thought she deserved a more tech-savvy cat flap. The result is a portal that tweets a photo and message every time Daphne passes through it (**@DaphneFlap**).

Maker: Bernie Sumption & Kate Bevan
url: magpi.cc/tweetcatflap

Dog Translator

Ever wondered what your dog is trying to tell you by barking? YouTuber NerdStoke has come up with the solution (allegedly): a Dog Translator. His pet pooch Nugget wears a harness with a microphone to pick up his barks. A Raspberry Pi Zero then converts them into human speech to be played back over speakers. OK, so the speech comprises prerecorded phrases played based on the time of day, but it's a fun project and easily customisable.

Maker: NerdStoke
url: magpi.cc/dogtranslator

Cat Speedometer

These giant hamster wheels for cats are brilliant for indoor exercise, but have you ever wondered how fast your cat can run on one, and how far? Shawn decided to find out by fitting his wheel out with a Raspberry Pi Zero and a reed switch base to sense the six magnets around the wheel passing it. From this, distance, speed, and top speed can be calculated and logged. Shawn discovered his cats were running 4–6 miles a night, at up to 14 mph.

Maker: Shawn Nunley
url: magpi.cc/catspeedo

IoT Kitteh

Cats may wander miles around their neighbourhood. To find out the location of his puss Lily, Jeremy invented a vest to track her position using GPS. For communication purposes, he fitted it with a cellular connection. After realising that a cat wouldn't be able to answer the phone when called, he opted to send a message to the device via AWS IoT and have it call him back automatically, as well as (perhaps more useful) sending GPS co-ordinates.

Maker: Jeremy Wall
url: magpi.cc/iotkitteh

RELATED PROJECTS

- Hamster Fitness Tracker **magpi.cc /hamsterfitness**

- RaspiTrap humane mousetrap **magpi.cc/raspitrap**

- Dog Treat Dispenser **magpi.cc/dogtreat**

- Cat Prey Detector **magpi.cc/catprey**

- BLE Cat Door **magpi.cc/blecatdoor**

"The cat flap tweets a photo and message every time Daphne passes through it"

Health and fitness

From Fitbits to Peloton bikes, the world is awash with smart gadgets to help you stay fit and well, but why not exercise your mind by making one?

"It uses computer vision and machine learning to help track your HIIT workout progress"

Shareable Smartbike

Keanu wanted to track his cycling routes using GPS and log his speed and ride length data. Rather than buying a standard bike computer, he decided to create his own which would also turn the lights on and off automatically, as he kept forgetting. It features a Raspberry Pi, Arduino, and lots of wiring. Since he wanted to be able to share the bike without interfering with his own ride data, he also added an RFID scanner to identify the rider.

Maker: Keanu Dav
url: magpi.cc/smartbike

HIIT Personal Trainer

High-intensity interval training is a popular way to keep fit, and James has created a clever system and web app to enable you to train at home. It uses computer vision and machine learning on Raspberry Pi to help track your HIIT workout progress in real-time (at around 30 fps). It acts as an electronic referee, tracking the user's detected poses and movements, and then scoring them. There's even a leaderboard so multiple users can compete.

Maker: James Wong
url: magpi.cc/hiitpi

RELATED PROJECTS

- YogAI Smart Personal Trainer **magpi.cc/yogai**

- Smart Bottle **magpi.cc/smartbottle**

- SmartEdge Gym Buddy **magpi.cc/gymbuddy**

- Gymnasticon Smart Bike **magpi.cc/gymnasticon**

RIoT Brick Run Tracker

Wanting to log his progress in the Brecon Beacons 10 Peaks ultramarathon, Alan created his own GPS run tracker in a 3D-printed case. As showcased in *The MagPi* magazine issue #93 (**magpi.cc/93**), the RIoT Brick enabled family and friends to access a web dashboard to track how he was doing in the gruelling endurance race. Powered by a 20,000 mAh battery pack, the Brick was also equipped with numerous sensors to log temperature, humidity, pressure, altitude, and light level readings along the route.
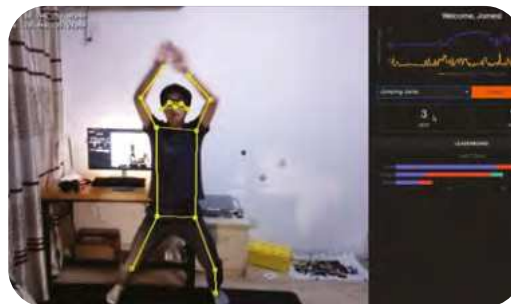
Maker: Alan Peaty
url: magpi.cc/riotbrick
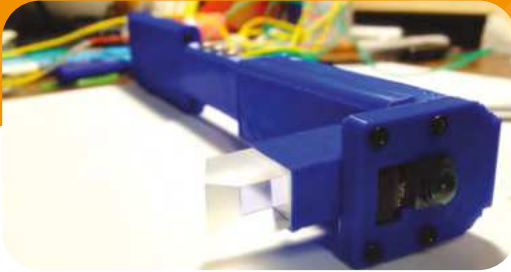

Smart Snoring Correction

An estimated 25% of adults snore regularly, which can be a problem for them or their partner. Bryan and Brayden have come up with a solution: a DIY device based around a Neosensory Buzz bracelet, normally used to alert people with hearing difficulties to doorbells and alarms. A sound inference base running on Raspberry Pi recognises snoring sounds from a mic and sends a signal to the Buzz bracelet to wake the user temporarily so they stop snoring.

Maker: Bryan and Brayden Staley
url: magpi.cc/snoring

Clothing and accessories

I f you want to show off your new project, what better way than to wear it? Or you may want a better way to organise your clothes.

Raspberry Pi Zero Prism

While the original Google Glass never really took off, smart glasses are still a thing. Instead of forking out big money for ready-made ones, however, it's possible to make your own. One inspiring example is Ricardo's Prism, which fits a Raspberry Pi Zero, Camera Module, and micro LCD into a 3D-printed wearable case, with a beam splitter and convex lens to focus the HUD image. It can do facial, image, and optical character recognition, as well as internet browsing.

Maker: Ricardo Ferro
url: magpi.cc/pizeroprism

RELATED PROJECTS

- Pi Camera Cap
  magpi.cc/picameracap

- Sock Matcher
  magpi.cc/sockmatcher

- CNC Embroidery Machine
  magpi.cc/cncembroidery

- Social Media without the Internet (smart suit) **magpi.cc /socialmediasuit**

Smart Wardrobe

Some people are predicting that smart wardrobes (aka closets) will be the next big thing for the home, but Tan See Youu explored the concept back in 2015 using a Raspberry Pi. The project includes an RFID reader to detect tags the user has placed into every item of clothing in the wardrobe. Once an item is registered, it can be recognised and the system will tell you, via an Android smartphone app, when you last wore it.

Maker: Tan See Youu
url: magpi.cc/smartwardrobe

Smart Watch

Smartwatches are incredibly popular and the swankier models are often worn as a status symbol. For the ultimate in one-upmanship, however, you could be wearing your own custom-created, one-of-a-kind watch. That's what Tyler did, creating his Smart Watch to impress a friend at work. It comprises a Raspberry Pi Zero W and LCD screen, housed in a 3D-printed case that he designed. With the addition of a tiny joystick and buttons, it can even play Minecraft!

Maker: Tyler Suard
url: magpi.cc/smartwatch

> "For the ultimate in one-upmanship, you could be wearing your own custom-created, one-of-a-kind watch"

Multi-effects Guitar Pedal

For those who like to wear an electric guitar, a DIY effects pedal will make it sound even better. We've seen a few based on Raspberry Pi, but Raphaël's is incredible, featuring a touchscreen and offering serious sonic capabilities with an innovative method for combining effects. It is able to digitally create a complex parallel signal chain: "Something like the modular synth version of guitar pedals," he says. No off-the-shelf pedal can do anything like it.

Maker: Raphaël Isvelin
url: magpi.cc/guitarpedalyt

arm DevSummit

Learn to Write Fast and Run Fast on Arm

Join Arm DevSummit
Virtual Event Live on October 19-21, 2021

OVER 140 AGENDA ITEMS ACROSS THESE SEVEN TRACKS:

- Cloud Native for the Endpoint, Edge and the Cloud
- The Future of Mobility is Software Defined
- Developing Interactive Consumer Experiences
- Practical Software Development for a Productive IoT
- AI in the Real World: From Development to Deployment
- Hardware Design Methodology
- Optimizing Arm Technology for PHC, Cloud and 5G Infrastructure

DON'T MISS THESE HOT WORKSHOPS FOCUSING ON PRACTICAL SOFTWARE DEVELOPMENT FOR A PRODUCTIVE IOT:

- Introducing Keil® Studio and Cloud-based Development for IoT and Embedded Applications
- Containers Over-The-Air: Building Your First Edge AI Device the Easy Way
- Ultra-Low Power Wi-Fi Made Easy – Develop a Battery Powered Wi-Fi Sensor in Hours

Visit devsummit.arm.com to Register

Make retro noises
with Raspberry Pi

Recreate the sounds of the 16-bit computer
era with authentic vintage audio chips

MAKER

K.G. Orphanides

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

@KGOrphanides

You'll Need

> OPL2 Audio Board
> **magpi.cc/opl2**

> OPL2 board libraries
> **magpi.cc/opl2lib**

> 3.5 mm to 3.5 mm stereo audio cable

> Speaker with 3.5 mm input

> 7 × female-female Dupont jumper cables

The Yamaha OPL2 FM synthesis chip – also known as the YM3812 – was a powerhouse of 1980s and early 1990s digital audio. Released in 1985, it appeared on the classic AdLib, Game Blaster, and early Sound Blaster cards, as well as Yamaha's own keyboards. These relatively inexpensive OPL2 sound cards soon found favour among PC game developers keen to make up for the limitations of the timer-controlled internal PC speaker. In this project, we're going to relive the heady days of 1980s computer music by connecting Cheerful Electronic's OPL2 Audio Board to Raspberry Pi's GPIO, playing classic game soundtrack files and programming a simple controller to play your own tunes.

01 Connect your OPL2 Audio Board

Use a jumper cable to connect VCC on the OPL2 board to board pin 2 (5 V) of Raspberry Pi's GPIO. Connect GND to pin 6, A0 to pin 16 (GPIO 23), Data to 19 (GPIO 10), Latch to 15 (GPIO 22), Shift to 23 and Reset to pin 13 (GPIO 27). The pins have changed through various iterations of the OPL2 board's software, so check the documentation. Finally, use the 3.5 mm audio output on the board to connect speakers or headphones. Powered speakers are the best choice, as they're less susceptible to audible interference, but we've had no trouble using headphones. For the sake of your ears, make sure the board's physical volume control is at minimum before you start testing.

02 Set up Raspberry Pi

We strongly recommend using a fresh install of Raspberry Pi OS for this hardware project. The OPL2 board controller software requires

Raspberry Pi's Serial Peripheral Interface (SPI) to be active. Open a Terminal and type `sudo raspi-config`. Select option 3 'Interface options', then 'P4 SPI'. Select Yes to enable it, then Finish.

You'll also need the WiringPi library, but this comes installed by default on current versions of Raspberry Pi OS and will be pulled down automatically during installation if it's not found.
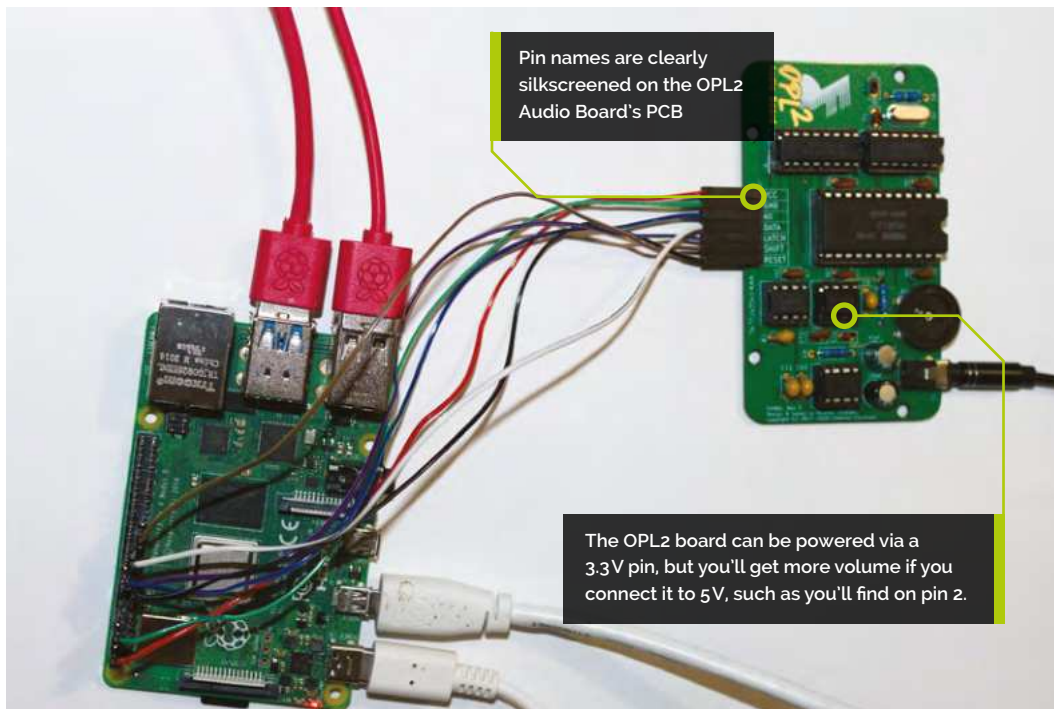
03 Install and test the software

Open a Terminal window and type:

```
git clone https://github.com/DhrBaksteen/ArduinoOPL2.git
cd ArduinoOPL2
sudo ./build
```

If all dependencies are met, you'll be asked to choose which of Cheerful's boards you're using. Select option 2 for the OPL2. Once built, the installer will show you a connection diagram. We've

> You can buy the OPL2 board as a kit to solder together yourself or, like us, as a fully assembled item

Pin names are clearly silkscreened on the OPL2 Audio Board's PCB

The OPL2 board can be powered via a 3.3 V pin, but you'll get more volume if you connect it to 5 V, such as you'll find on pin 2.

already wired up our board, but double-check that against the pin numbers shown to make sure you've used the ones it expects.

Make sure your speaker is connected and switched on; enter **Y** to play the test tune. If you don't hear anything, check your volume control and GPIO connections, then run the demotune executable in **ArduinoOPL2/examples_pi/demotune/** to re-test.

04 Play some tunes

The OPL2 board's accompanying tools include a command-line player for id Software's IMF AdLib audio format, the VGZ and VGM formats used to encode audio for AdLib cards, as well as the OPL2 relatives in Sega's consoles, and the DOSBox Raw OPL (DRO) audio format created to record OPL2 and OPL3 output from DOS games. Some example files are included; others can be found online.

To try the player out with the Jill of the Jungle theme, open a Terminal and enter:

```
cd ArduinoOPL2/examples_pi/opl2play
./opl2play tunes/jill.vgz
```

If you want to be able to run the player from anywhere, add it to your path by typing the following in the Terminal:

```
export PATH="$PATH:$HOME/Software/
ArduinoOPL2/examples_pi/opl2play/"
```

To make that permanent, run:

```
mousepad ~/.bashrc
```

Then paste that `export PATH` line in at the end of the file and save.

05 Get ready to code

The **examples_pi** directory contains handy tools, libraries, and code examples that are worth looking at and playing with. They're written in C++ and use the WiringPi library.

Make a directory and copy the **/examples_pi/demotune/midi_instruments.h** file into it. This is a General MIDI instrument library based on one created by The Fat Man for the original AdLib card. We're going to write an extremely simple controller application so we can play a monophonic tune live using Raspberry Pi's number keys. We'll use ncurses to detect key presses, so open a Terminal and type:

```
sudo apt-get install libncurses5-dev
libncursesw5-dev
```

06 Make your own synth

You can download the extended source code for this micro-project at **magpi.cc/inputcpp**. Following along will guide you through key interface documentation for the OPL2 board and some of its core features.

Top Tip 👍

MIDI input

To receive input from USB MIDI devices, RtMidi (**magpi.cc/rtmidi**) is worth a look.

input-print_version.cpp

> Language: **C++**

```cpp
001.  #include <ncurses.h> /* remember to sudo apt-get
002.  install libncurses5-dev libncursesw5-dev */
003.  #include <OPL2.h>
004.  #include <stdio.h>
005.  #include <wiringPi.h>
006.  #include "midi_instruments.h"
007.
008.  OPL2 opl2;
009.
010.  int kbhit()
011.  {
012.      int ch = getch();
013.      if (ch != ERR) {
014.          ungetch(ch);
015.          return 1;
016.      } else {
017.          return 0;
018.      }
019.  }
020.
021.  int main(int argc, char **argv) {
022.      opl2.begin();
023.
024.      // Load a harpsichord instrument and assign
          it to channel 0.
025.      Instrument harpsichord =
          opl2.loadInstrument(INSTRUMENT_HARPSIC);
026.      opl2.setInstrument(0, harpsichord);
027.
028.      initscr();
029.      cbreak();
030.      scrollok(stdscr, TRUE);
031.      while (1) {
032.          if (kbhit()) {
033.              int key = getch();
034.              if (key==49){
035.                  opl2.playNote(0, 5, NOTE_C);
036.                  // Play note C-5 on channel 0
037.              } else if(key==50){
038.                  opl2.playNote(0, 5, NOTE_D);
039.              } else if(key==51){
040.                  opl2.playNote(0, 5, NOTE_E);
041.              } else if(key==52){
042.                  opl2.playNote(0, 5, NOTE_F);
043.              } else if(key==53){
044.                  opl2.playNote(0, 5, NOTE_G);
045.              } else if(key==54){
046.                  opl2.playNote(0, 5, NOTE_A);
047.              } else if(key==55){
048.                  opl2.playNote(0, 5, NOTE_B);
049.              } else if(key==56){
050.                  opl2.playNote(0, 6, NOTE_C);
051.              } else if(key==57){
052.                  opl2.playNote(0, 6, NOTE_D);
053.              } else if (key==48){
054.                  opl2.playNote(0, 6, NOTE_E);
055.              }
056.              delay(500); // Hold the note for 500ms
057.              opl2.setKeyOn(0, false); // Release
          the note
058.          }
059.      }
061.  }
```

We're going to include the opl2 library and instruments header file, capture keyboard input, set an instrument, and associate specific keys on the keyboard with notes to be emitted by the sound card.

Open Geany (or your preferred syntax-highlighting text editor) on Raspberry Pi's desktop and create a file called **input.cpp** in the same directory that you copied **midi_instruments.h** to.

07 Find your voice

To see which instruments are available for you to use, open **midi_instruments.h** and take a look through it. In our example code, we load a harpsichord voice from **midi_instruments.h** and set it to a channel. We then use the OPL2 library's

`playNote` function to specify the channel we wish to play a note on (0 in this case), its ISO octave number (5), and the note we wish to play. Useful examples and explanations can be found in the Working with Instruments (**magpi.cc/workingwithinstruments**) entry in the OPL2 board's documentation and the **indepth.md** file that comes with its libraries. These include instructions for creating your own instruments and setting sustain – download **input_sustain.cpp** (at **magpi.cc/inputsustaincpp**) to see how this could work with our sample code.

08 Cutting keys

We're going to handle key press detection with a popular, if limited, approach to keyboard

Raspberry Pi GPIO BCM numbering

OPL2 Audio Board to Raspberry Pi wiring guide

OPL2 board pin	VCC (5V/3.3V)	GND	A0	Data	Latch	Shift	Reset
Raspberry Pi GPIO pin (Board Numbering)	2 (5V)	6	16	19	15	23	13
Raspberry Pi GPIO pin (BCM Numbering)	5V	GND	23	10	22	11	27

character capture using `getch()` from the ncurses library. It works here because we only want to play one note at a time and will be fixing the duration of each note in code. A more sophisticated control application would benefit from the SDL library (**magpi.cc/sdllib**), particularly its handling of KEYUP events.

Once a keystroke has been captured, it's written to a variable called `key`. We've then used a short series of if/else statements to have each keycode invoke a note from the OPL2.

09 Cracking the (key)code

We've used the number keys to provide a control surface of just over an octave, not including semitones, on a standard Raspberry Pi keyboard. These tend to be the same across all layouts, but other keys do not.

That's because keycodes are based on character, not physical position, so an AZERTY keyboard user will have trouble if software is hard-coded to use the WASD keys for movement.

To check the key numbering for your keyboard, see the list at **magpi.cc/javakeycodes** or download, compile, and run **keycheck.cpp** (from **magpi.cc/keycheckcpp**) to display the keycodes. This is useful if you wish to increase the range of your Raspberry Pi-powered instrument by assigning notes to more keys.

10 Ready to compile

With your code in place, it's time to compile **input.cpp**. The online version of the file includes an extra routine to display which note you've played.

Open a Terminal in the directory containing **midi_instruments.h** and **input.cpp**. Raspberry Pi OS comes with the GCC compiler already installed. We'll tell it to use the OPL2, WiringPi, and ncurses libraries. Type:

```
gcc -std=c++11 -Wall -o input input.cpp
-lOPL2 -lwiringPi -lncurses
```

Then type `./input` to try it out and make some (slightly beepy, monophonic) music with your number keys. Press **CTRL+C** to exit when you're done.

11 Further projects

If you have an Arduino microcontroller, you can connect the OPL2 Audio Board to it to transform the board into a DOSBox-X compatible Sound Blaster 1.0 clone, which you can then connect to Raspberry Pi via USB. Connection and configuration instructions are provided at **magpi.cc/opl2dosbox** and you can find our comprehensive guide to running retro DOS games using DOSBox-X at **magpi.cc/dosemulation**. **M**

Top Tip 👍

Take it to the bank

To create and import OPL2 instruments, try Wohlstand's OPL3BankEditor **magpi.cc/opl3bank**
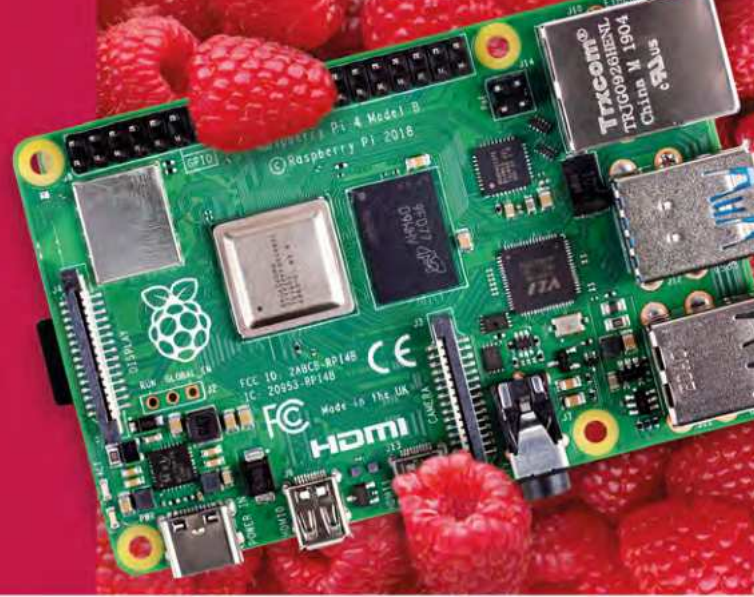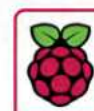
Ultimate home server: add sound and vision

Turn your server into a media powerhouse with Plex and Mopidy

Create as many libraries as you like by dividing media into directories

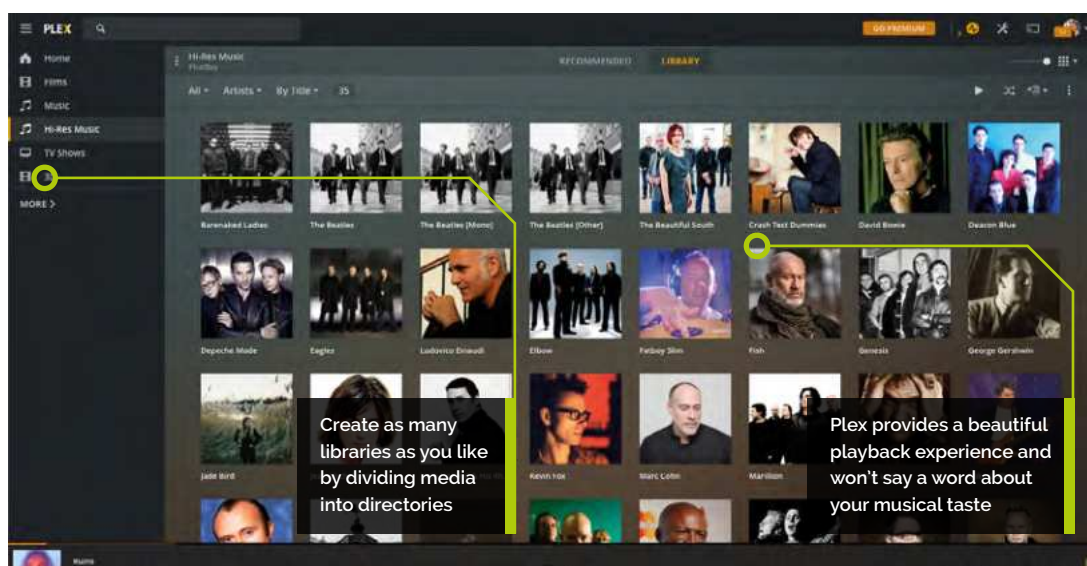Plex provides a beautiful playback experience and won't say a word about your musical taste

MAKER

PJ Evans

PJ is a writer, software engineer, and tinkerer. His ambient music collection has been described as "inspiring" (PJ Evans) and "just a load of random noise" (Mrs Evans, music teacher).

twitter.com/ mrpjevans

You'll Need

> Raspberry Pi ultimate server **magpi.cc/108**

> Plex **magpi.cc/plex**

> Mopidy **magpi.cc/mopidy**

Sound and vision is where we're at for our ultimate server project this month. Sure, serving files is all very well (see *The MagPi* magazine issue #109, **magpi.cc/109**) but what about a decent media interface? Luckily we're spoilt for choice. We're going to add two media serving solutions, Plex and Mopidy, to our server. Plex is a great choice for playing back your media anywhere in the world and Mopidy can turn your server into a jukebox as well as a multi-room audio streaming system. We'll also look at how to create your file structure and how to add content from different sources. Let's hit the play button.

(**magpi.cc/109**). We want to be sure our external drive is mounting on startup. Reboot and then issue this one simple command in the terminal:

```
df -h
```

The 'df' command is short for 'disk free' and shows you a summary of disk usage. If all is well, you'll see the external drive (partition **sdb1**) mounted to **/mnt/huge** (or whatever you decided to call the mount point). If /mnt/huge does not appear in the list, refer back to the previous tutorial and make sure the entry in **/etc/fstab** is correct.

01 Check, check, and check again

Before we plunge into the world of media streaming, let's make sure your ultimate server is correctly configured as per last month's tutorial

02 Set up your library

Media streaming tends to involve a lot of files. Before you find yourself in a mess, take some time to consider how you would like to structure

► The command `df -h` shows that the external drive has correctly mounted

all your media. How you do this is up to you, but when it comes to ripping CDs, many encoders such as abcde (A Better CD Encoder, **magpi.cc/abcde**) will automatically rip to a given structure, so it can be helpful to plan this out now. See **Figure 1** for an example of a well-laid-out file structure. Don't feel that you have to follow this – if you would prefer more complex (or simple!) divisions, that's completely up to you.

Top Tip 👍

Quickly copy your media

Moving a lot of media files over the network can be very time-consuming. Luckily, our USB drive can be mounted onto the source machine for super-fast copying. Remember, the host operating system will need a driver that can understand ext4 partitions.

03 Add some sounds

Now we have lots of fast USB 3.0 space to play with, we can add some media that we'd like to stream. This is a massive topic in itself, so we'll just give you some pointers here. You can now legally buy MP3s from many sources such as Amazon and Apple. These are free from DRM (Digital Rights Management), meaning you can play them back on
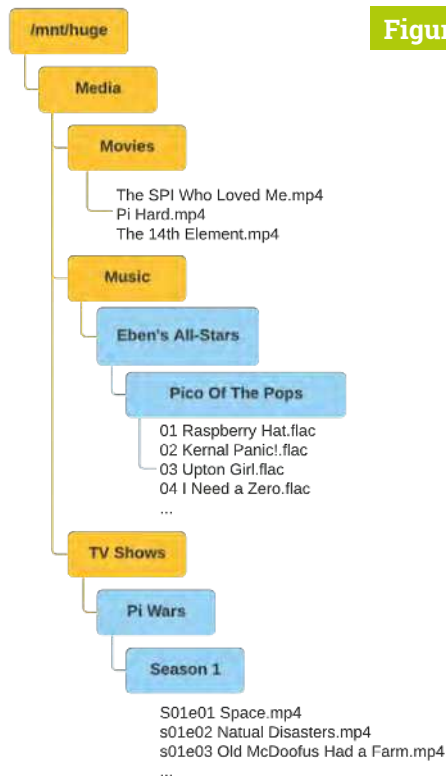

Figure 1

```
/mnt/huge
    Media
        Movies
            The SPI Who Loved Me.mp4
            Pi Hard.mp4
            The 14th Element.mp4
        Music
            Eben's All-Stars
                Pico Of The Pops
                    01 Raspberry Hat.flac
                    02 Kernal Panic!.flac
                    03 Upton Girl.flac
                    04 I Need a Zero.flac
                    ...
        TV Shows
            Pi Wars
                Season 1
                    S01e01 Space.mp4
                    s01e02 Natual Disasters.mp4
                    s01e03 Old McDoofus Had a Farm.mp4
                    ...
```

► **Figure 1** Here's an example of a sensible directory structure for your media files. Remember to create it on your external mounted drive

your Raspberry Pi media system without issues. If you want to get the best quality possible, consider buying in FLAC format (no lossy compression) from services such as HDTracks (**hdtracks.com**) or Qobuz (**qobuz.com**).

04 Going to the movies

Adding mainstream movies can be a bit challenging as there are no legal vendors of DRM-free movies. However, you can use Plex's own free streaming service to access content with a wide range of (sometimes cheesy) movies to watch, as well as video podcasts. You can also use plug-ins to access all of YouTube. Of course, if you have a collection of home movies, it's a great way to get them up on the big screen.

05 Let's Plex!

We're going to install Plex, one of the most popular media streaming services available. It supports a dizzying array of file formats for both audio and video. Plus, clever built-in transcoding systems mean it can compress on-the-fly to ensure the smoothest of playbacks. All with a very friendly, good-looking user interface. Installation takes a few steps, so follow the commands in the **install_plex.sh** listing to add Plex's repository, update, and then install and configure Plex. Once complete, we can check Plex is running using this command:

```
sudo systemctl status plexmediaserver
```

You should see that Plex is 'loaded' and 'active (running)'. Press **Q** to exit and you're ready to configure!

06 First-time Plexing

When we installed Plex, it started a web server on port 32400. So all you need to do is open a web browser on another machine (or the server itself if you've installed a desktop) and go to:

```
http://ultimate.local:32400/web
```

(Replace the server name if you didn't use 'ultimate' when setting it up in part 01 of this series.)

What you should see now is the Plex welcome screen. Be patient: it can take a few seconds on

the first run. To get the best out of Plex, create a free account as prompted using an authentication provider such as Google or just your own email address.

07 Initial setup

Plex will take you through a short wizard to get things set up. Start by giving your Plex server a name; we've chosen 'ultimate' to be consistent. You can also opt for access to your Plex library from anywhere on the internet. Plex will attempt to open a port on our home router so you can access your media from the **plex.tv** website. If you don't want to do this, ensure the checkbox is clear. Click 'Next' to start adding your libraries. One-by-one, select a library 'type' (music, movies, etc.) when navigating to the corresponding folder on your external drive; e.g. for 'Music', use **/mnt/huge/Media/Music**.

08 Library management

Plex will now start scanning your folders for content. It will also access online databases to add metadata to its library to make your collection as complete as possible. Depending on the size of your collection, this process will take anywhere between seconds and hours. It's a background process though, so you can start using Plex right away as your media appears. You'll also notice that Plex will also offer you online content – you can disable this if you wish or sign up for a 'Plex Pass' for a lot more features. If you don't like what Plex has 'found' for your media types, you can always click the pen icon and edit the metadata; Plex doesn't always get it right.

09 Access it anywhere

One of the great things about Plex is the availability of its client applications. Not only is there a rock-solid web-based interface (which works great on Raspberry Pi OS's Chromium) but support for a wide range of devices including PlayStation, Xbox, Apple TV, Amazon Fire devices – and for mobile, iOS and Android. If you created an account earlier, all you have to do is install the app and log in. Your ultimate server will be detected and added. Now you can access and stream your media collection from any location on the internet.
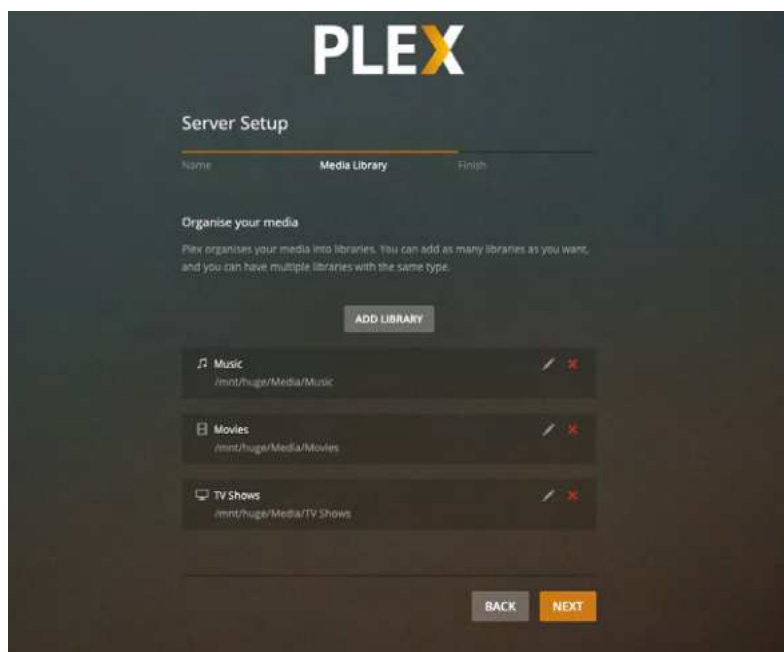
10 Streaming to headless devices

One restriction of Plex is that it requires a device with some kind of interface to work and plays audio to that interface. If we wanted to play audio on the ultimate server itself, or stream to another Raspberry Pi computer that is connected to speakers, we need to use Mopidy, a purpose-built audio streaming solution. Luckily, Plex and Mopidy can live side-by-side and even share libraries. To install Mopidy and its user interface, Iris, follow the commands in the **install_mopidy.sh** listing. Once complete, you can access the user interface here:

```
http://ultimate.local:6680/iris/
```

▲ Once Plex is installed, you need to add each directory and specify what type of media it contains

install_plex.sh

DOWNLOAD THE FULL CODE:
magpi.cc/installplexsh

> Language: **Bash**

```bash
001.  #!/bin/sh
002.
003.  # Plex's repository uses HTTPS as the transport, so we need to
      install support
004.  sudo apt update
005.  sudo apt install apt-transport-https ca-certificates curl
006.
007.  # Get Plex's public key and add the Plex repo
008.  curl https://downloads.plex.tv/plex-keys/PlexSign.key | sudo apt
      key add -
009.  echo deb https://downloads.plex.tv/repo/deb public main | sudo
      tee /etc/apt/sources.list.d/plexmediaserver.list
010.  # Install the server
011.  sudo apt update
012.  sudo apt install plexmediaserver
```
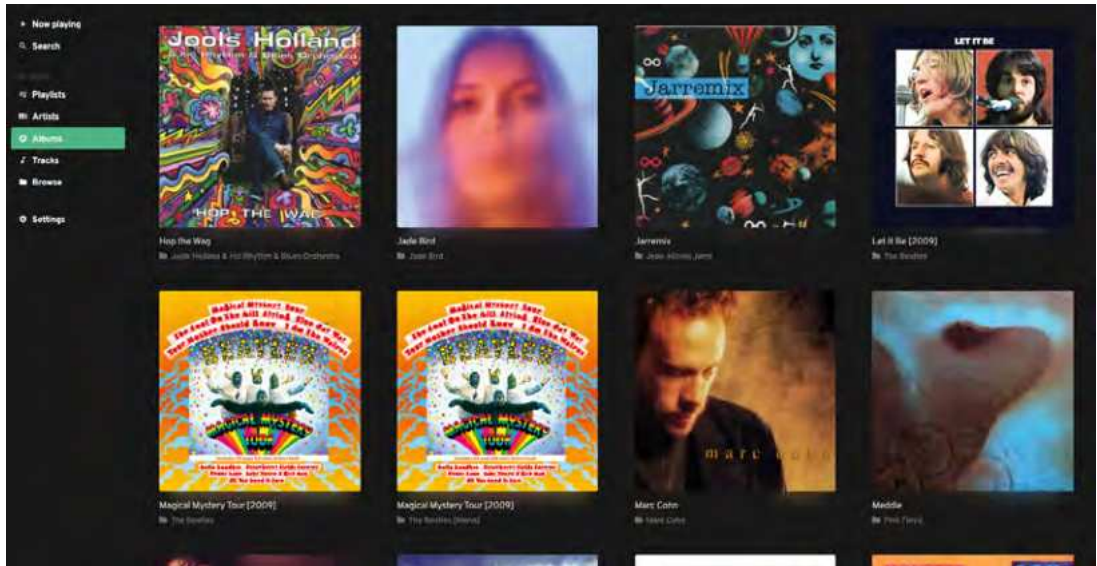
► Mopidy is an alternative to Plex, designed for local playback and streaming

⚠️

Warning!

The laws covering the ripping of CDs, DVDs and Blu-rays vary from country to country. The private copying of copyrighted material is illegal in the United Kingdom. Check what you want to do is legal before proceeding!

magpi.cc/ripping

11 Adding media to Mopidy

In the previous step we installed the 'local' plugin, which allows Mopidy to access local file systems for media. One restriction is that it can only support a single directory. Start by editing the config file:

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

install_mopidy.sh

> Language: **Bash**

DOWNLOAD THE FULL CODE:

⬇ **magpi.cc/installmopidysh**

```
001.   #!/bin/sh
002.
003.   # Add Mopidy repo and install
004.   wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key
       add -
005.   sudo wget -q -O /etc/apt/sources.list.d/mopidy.list https://
       apt.mopidy.com/buster.list
006.   sudo apt update
007.   sudo apt install mopidy
008.   sudo adduser mopidy video
009.
010.   # Configure (be careful with this line!)
011.   printf "[http]\nhostname = 0.0.0.0" | sudo tee -a /etc/mopidy/
       mopidy.conf
012.
013.   # Install user interface
014.   sudo apt install python3-pip
015.   sudo pip3 install Mopidy-Iris
016.   sudo pip3 install Mopidy-Local
017.   sudo sh -c 'echo "mopidy ALL=NOPASSWD: /usr/local/lib/
       python3.7/dist-packages/mopidy_iris/system.sh" >> /etc/sudoers'
018.
019.   # Enable and start service
020.   sudo systemctl enable mopidy
021.   sudo systemctl start mopidy
```

Add the following at the end of the file:

```
[local]
media_dir = /mnt/huge/Media/Music
```

Remember to change this path if you're using a different naming system or structure. Save the file (**CTRL+X**), and instruct Mopidy to scan the contents, then restart:

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

Reload Mopidy in your web browser and the media should be available. Now you can play music using your server as the audio source should you wish.

12 Advanced Mopidy

Now you have Mopidy installed, you can upgrade to a few special features. Mopidy supports a wealth of plug-ins and one of the most popular is mopidy-spotify, which, you guessed it, turns Mopidy into a Spotify client (although you need a premium account for this). Also, you can install Snapcast to create a synced multi-room audio system for a fraction of the price of solutions such as Sonos. For more detail and tutorials on all of this, check out *The MagPi* issues #96, #97, and #98.

Next time

In the fourth part of this series, we're going to look at how Raspberry Pi OS can be put to work to make your network a better place. Access your files securely from anywhere in the world, protect yourself from tracking ads, and more. 🅜

Stream viewer-controlled projects
with Surrogate.TV

Want a new level of interaction in your livestreams? Add Surrogate.TV to your Raspberry Pi projects for super low-latency audience control

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

I f you cast your mind back many years, you'll remember the craze of Twitch Plays Pokémon. Using inputs from the user chat, a computer would play through the video game Pokémon Red, often with chaotic results.

The popularity of these streams has resulted in many similar ideas, whether directly or indirectly, and Surrogate.TV is one of these. Not only does it allow viewers to control what they see, what they see can be a full-on Raspberry Pi contraption like a painting robot or little fighting robots in real life, with negligible latency. The features are much more rich than in the past as well, so let's explore how to set up your own fun interactive projects on Surrogate.TV.

▼ Your game, as seen from the dashboard, when a Raspberry Pi is set up and switched on

01 Assemble your surrogate

A camera is essential for this project; however, it doesn't matter what kind of camera you use. If you're using a Raspberry Pi Camera Module or HQ Camera, plug it in the usual way via the CSI port, with the silver connectors on the connector ribbon facing in the direction of the SD card slot.

For a USB webcam, Surrogate.TV recommends using a USB 3.0 port if possible, which are the blue ones in the middle of a Raspberry Pi 4.

Before connecting it all up to your Raspberry Pi projects, it's a good idea to run through the basics in the rest of this tutorial first.

02 Flashing Surrogate image

We're going to use Raspberry Pi Imager to create our SD card, which you can grab from **magpi.cc/imager**. Once that's installed on your computer, download the Surrogate image (a .gz file) from **magpi.cc/surrogateimg**, and then finally open the Imager application.
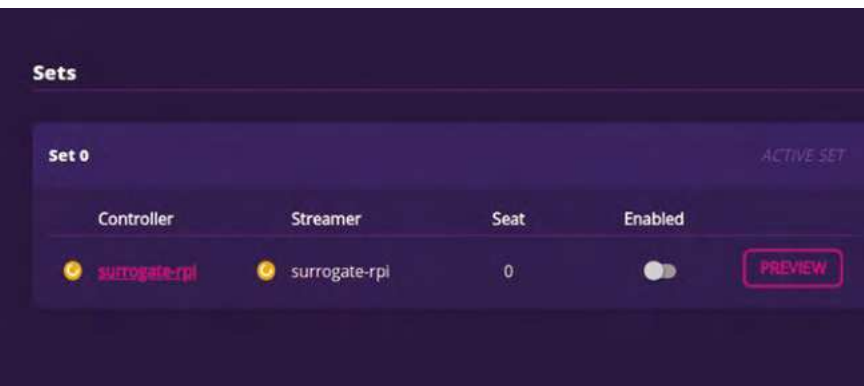
Click on Choose OS and select 'Use custom' from the bottom of the list. From here, select the image file you just downloaded. Now click on Choose Storage and select the SD card you'd like to use. Click on Write, then just wait for it to complete.
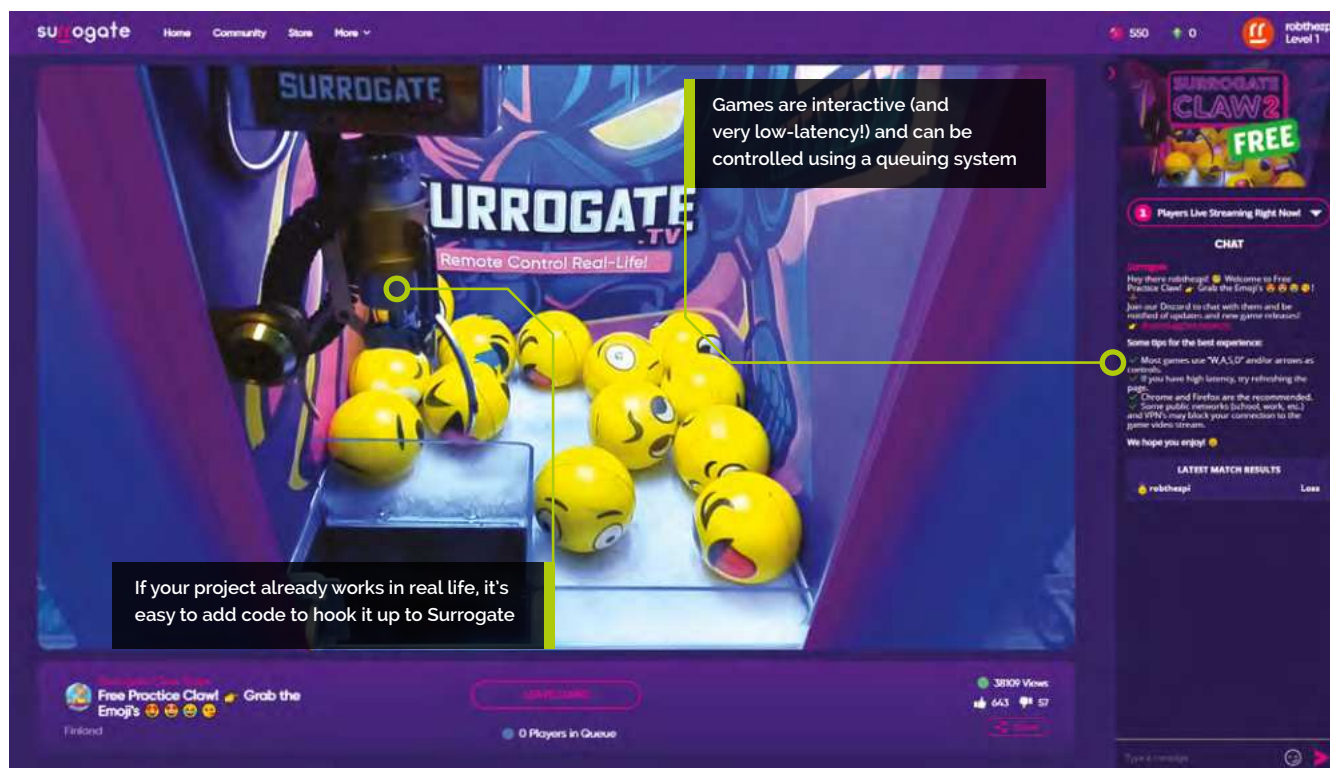
Don't worry about starting it up yet, as we'll need to create a game for it to connect to first.

03 Starting with Surrogate.TV

You need to create an account at **surrogate.tv** on another computer – we'll trust you understand how to set up an online account. Once that's done

Games are interactive (and very low-latency!) and can be controlled using a queuing system

If your project already works in real life, it's easy to add code to hook it up to Surrogate

and you're logged in, it's time to create a 'game' – the kind of thing you'll have viewers interact with via Raspberry Pi. Click on 'Create a game' and it will give you a checklist of stuff you'll need – make sure you have your wireless details to hand, like it says, and then press Start.

Give your game a temporary name and URL – these can be changed later, but for now it will help us get all set up. Click Next.

04 Connecting to Raspberry Pi

Now you can plug in your Raspberry Pi – you won't need it connected to a monitor in this part. From your other computer, copy the token it displays, and then open your wireless settings and look for the hotspot that Raspberry Pi has created – the SSID should be **surrogate-rpi**, and the password is **surrogatetv**.

When you're connected to the hotspot from Raspberry Pi, click Continue on your other computer. You will now be taken to a web page on Raspberry Pi.

05 Set up Raspberry Pi

You should see a field to enter the token you copied in the previous step – paste it in and click Continue, and it will ask you to add the details

of your wireless network. Enter the SSID and password of your home network and leave the rest of the options. Click Connect; you will be kicked off Raspberry Pi as it turns the hotspot off and attempts to connect to your wireless network.

Reconnect to your own home wireless network from your computer. Go back to **surrogate.tv** and click 'Skip to dashboard', where you should be able to find and control your 'game'/Raspberry Pi.

> **"** If all is working, a yellow light should be showing next to the controller name **"**

06 Game testing

If your Raspberry Pi doesn't show up as a controller in the Sets category, reboot Raspberry Pi and wait a few minutes. If that still doesn't work, go back, reflash, and try again, double-checking your WiFi details.

If all is working, a yellow light should be showing next to the controller name. If you click Preview, the camera you connected to Raspberry Pi should start up and the feed will be displayed on screen.

So far so good; now we need to check if the full system works and streams.
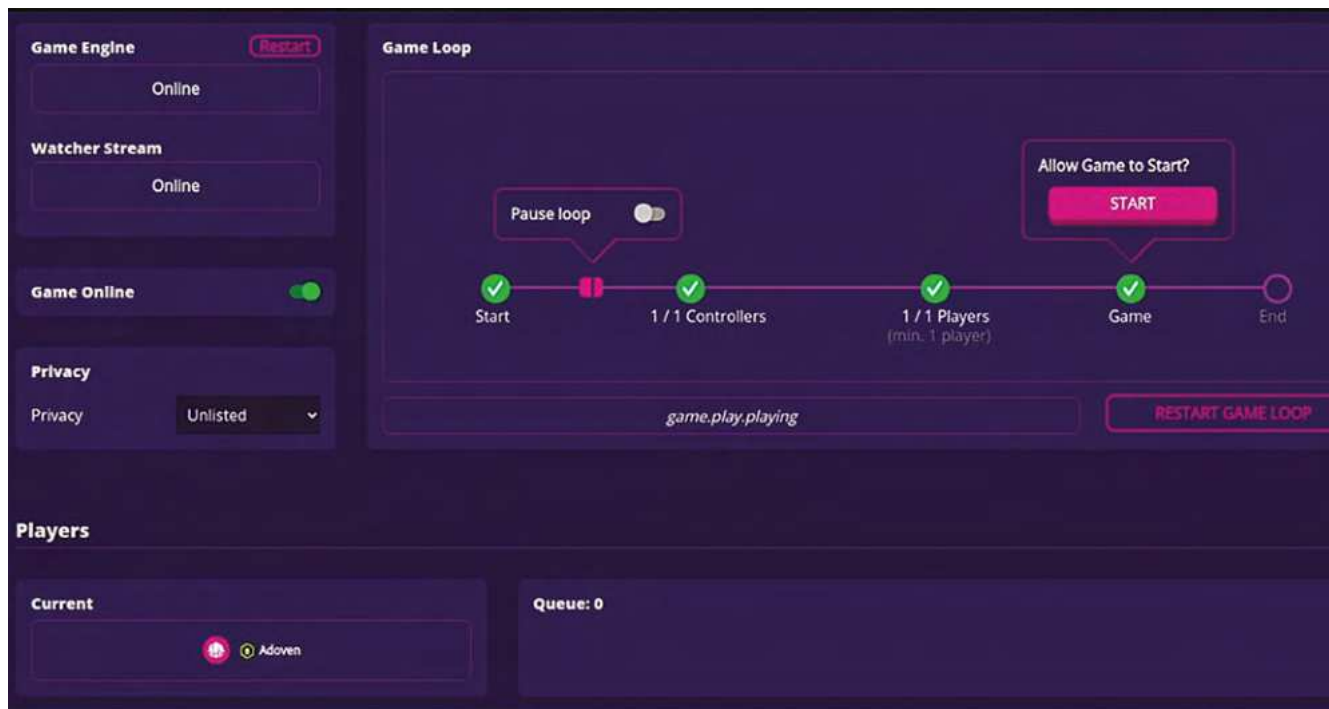
You'll Need

> Raspberry Pi 4

> Surrogate image file: **magpi.cc/ surrogateimg**

> Raspberry Pi Camera Module: **magpi.cc/camera**

> Or a USB webcam

> Another computer with WiFi

Top Tip 👍

Compatibility

The Surrogate image is based on Raspberry Pi OS, so it should be no problem to port your projects to it.

▲ The game loop allows you to see how the game is working in real time

07 Stream testing

On the game we set up, there's a toggle you can click labelled Enabled. Do that and the yellow icon should turn green. Toggle the Game Online setting above this as well, and then click on 'Go to game' on the left and it will give you the stream interface, hopefully with your camera feed.

This feed is the watcher stream, which has some delay; however, if you click 'Play now' you'll be momentarily allowed into the game. This in-game stream is very low-latency – saving that bandwidth for when you need it.

08 Going for a loop

You may have heard the term 'game loop' in regards to game design – breaking the interactive part of the game down to its core. In this instance it's fairly similar, but with audience participation it's showing what a viewer will do when controlling a game. As we've set up a simple game to just test

> ❝ You can build the robot and wire it up to the GPIO on Raspberry Pi ❞

everything, there's nothing really to do right now; however, you can see how it will work by going back to the dashboard in a separate tab and see how it shows the current state of the stream.

09 Game templates

Included in Surrogate.TV are some templates for games to check out, including a simple robot using a fairly cheap robot kit. Change your Raspberry Pi settings by going to the dashboard and clicking on its name under Controller. Change it from Simple Game to 'bot' from the drop-down list.

Following the instructions here, you can build the robot and wire it up to the GPIO on Raspberry Pi in the correct way. Mount your Raspberry Pi to the robot, turn it on with a mobile power supply, and you'll be immediately be able to play with the robot from your computer.

10 Custom game setup

You can also create custom code for custom games that can control your own projects (or different robots). It's a good idea to get SSH started up on your Surrogate Raspberry Pi, so

hook up a monitor and keyboard to it. Login with username **pi** and password **creator1337**, and then type `sudo raspi-config` to open the command line configuration tool. Go to Interfaces, SSH, and enable it.

Now is also a good time to change your password by using `sudo passwd pi`. Before you disconnect the monitor and keyboard, type in `hostname -I` to get the IP address so you can SSH in from your other computer.

11 Custom game coding

In the case of the robot code from the template, you'll find it in **games/bot/game.py** in the file structure of Raspberry Pi. Here you can play around with the example code to see how it works. Of note is the Python library surrotg which

you can link to other Python code to pass on controls via the stream interface.

Otherwise, check out the handy tutorial from Surrogate.TV (**magpi.cc/customgame**) for the basics you need on getting a fresh script running. There's also documentation for diving deeper in how to use Surrogate here: **magpi.cc/surrogatedocs**.

12 Examples to try

If you don't have a project in mind to hook up, you can find some examples that might spark your imagination at **hackster.io/surrogatetv**. There's some neat stuff involving robots and servos to get you started. You can also check out some of the games the Surrogate.TV folks stream on their website, as well as other users of the platform. M
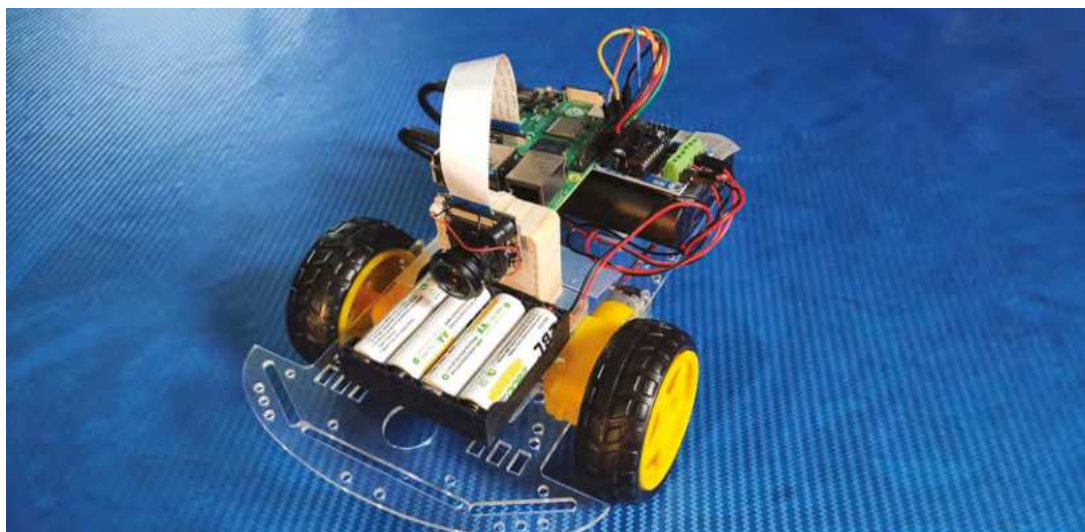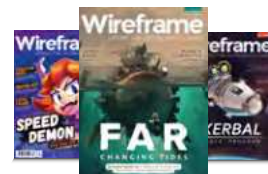

Top Tip 👍

Playability

Make your game engaging to viewers by giving them a high score to go for!

◄ This simple robot can be entirely controlled via Surrogate using a template code
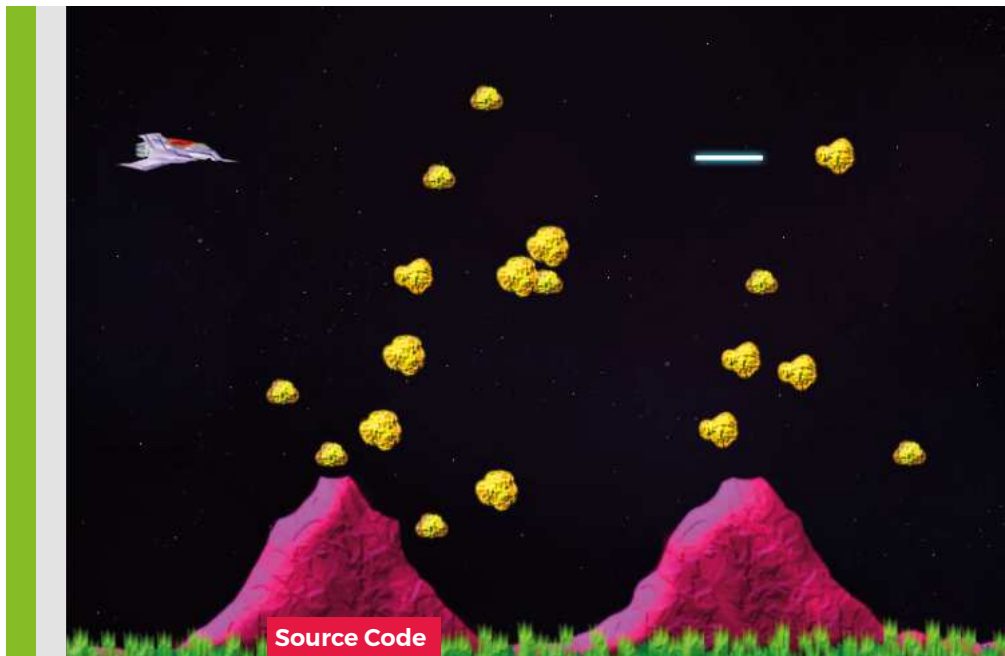
Source Code

Recreate Gradius's rock-spewing volcanoes

Code an homage to Konami's classic shoot-'em-up

AUTHOR
MARK VANSTONE

R eleased by Konami in 1985, *Gradius* – also known as *Nemesis* outside Japan – brought a new breed of power-up system to arcades. One of the keys to its success was the way the player could customise their Vic Viper fighter craft by gathering capsules, which could then be 'spent' on weapons, speed-ups, and shields from a bar at the bottom of the screen.

A seminal side-scrolling shooter, *Gradius* was particularly striking thanks to the variety of its levels: a wide range of hazards were thrown at the player, including waves of aliens, natural phenomena, and boss ships with engine cores that had to be destroyed in order to progress. One of the first stage's biggest obstacles was a pair of volcanoes that spewed deadly rocks into the air: the rocks could be shot for extra points or just avoided to get through to the next section. In this month's Source Code, we're going to have a look at how to recreate the volcano-style flying rock obstacle from the game.

Our sample uses Pygame Zero and the `randint` function from the random module

to provide the variations of trajectory that we need our rocks to have. We'll need an actor created for our spaceship and a list to hold our rock Actors. We can also make a bullet Actor so we can make the ship fire lasers and shoot the rocks. We build up the scene in layers in our `draw()` function with a star-speckled background, then our rocks, followed by the foreground of volcanoes, and finally the spaceship and bullets.

In the `update()` function, we need to handle moving the ship around with the cursor keys. We can use a `limit()` function to make sure it doesn't go off the screen, and the **SPACE** bar to trigger the bullet to be fired. After that, we need to update our rocks. At the start of the game our list of rocks will be empty, so we'll get a random number generated, and if the number is 1, we make a new rock and add it to the list. If we have more than 100 rocks in our list, some of them will have moved off the screen, so we may as well reuse them instead of making more new rocks. During each update cycle, we'll need to run through our list of rocks and update their position. When we make a rock, we give it a speed and direction, then when it's

updated, we move the rock upwards by its speed and then reduce the speed by 0.2. This will make it fly into the air, slow down, and then fall to the ground.

From this code, we can make rocks appear just behind both of the volcanoes, and they'll fly in a random direction upwards at a random speed. We can increase or decrease the number of rocks flying about by changing the random numbers that spawn them. We should be able to fly in and out of the rocks, but we could add some collision detection to check whether the rocks hit the ship – we may also want to destroy the ship if it's hit by a rock. In our sample, we have an alternative, 'shielded' state to indicate that a collision has occurred. We can also check for collisions with the bullets: if a collision's detected, we can make the rock and the bullet disappear by moving them off-screen, at which point they're ready to be reused.

That's about it for this month's sample, but there are many more elements from the original game that you could add yourself: extra weapons, more enemies, or even an area boss. **w**

Vulcan Venture

Here's Mark's volcanic code. To get it working on your system, you'll need to install Pygame Zero – full instructions are available at **wfmag.cc/pgzero.**

```python
# Gradius
import pgzrun
from random import randint

jet = Actor('jet',(400,300))
bullet = Actor('bullet', center=(850, 0))
rocks = []

def draw():
    screen.blit("background", (0, 0))
    drawRocks()
    screen.blit("foreground", (0, 0))
    bullet.draw()
    jet.draw()

def update():
    if keyboard.up: jet.y = limit(jet.y-5,50,550)
    if keyboard.down: jet.y = limit(jet.y+5,50,550)
    if keyboard.left: jet.x = limit(jet.x-5,10,790)
    if keyboard.right: jet.x = limit(jet.x+5,10,790)
    if keyboard.space :
        if bullet.x >= 850 : bullet.pos = (jet.x,jet.y+5)
    if bullet.x < 850: bullet.x += 20
    updateRocks()

def limit(n, minn, maxn):
    return max(min(maxn, n), minn)

def drawRocks():
    for r in range(0, len(rocks)):
        rocks[r].draw()

def makeRock(pos):
    r = len(rocks)
    if r < 100:
        rocks.append(Actor('rock'+str(randint(1,3)), center=pos))
    else:
        r = getOldRock()
        rocks[r].pos = pos
    rocks[r].speed = randint(6,12)
    rocks[r].dir = (randint(0,60)-30)/10

def updateRocks():
    if randint(0,10) == 1: makeRock((215,480))
    if randint(0,10) == 1: makeRock((540,480))
    shieldsUp = False
    for r in range(0, len(rocks)):
        if rocks[r].y < 800:
            rocks[r].y -= rocks[r].speed
            rocks[r].x += rocks[r].dir
            rocks[r].speed -= 0.2
            if jet.colliderect(rocks[r]):
                shieldsUp = True
            if bullet.colliderect(rocks[r]):
                rocks[r].y = 800
                bullet.x = 850
    if shieldsUp == True:
        jet.image = "jet2"
    else:
        jet.image = "jet"

def getOldRock():
    for r in range(0, len(rocks)):
        if rocks[r].y >= 800:
            return r

pgzrun.go()
```


▲ The *Gradius* volcanoes spew rocks at the player just before the end-of-level boss ship arrives.

Laura Sach

MAKER

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

@CodeBoom

Martin O'Hanlon

MAKER

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

@martinohanlon

Part 10

Create GUIs with Python:
Stop-frame animation

Build your own stop-frame animated GIF creator

This project uses a Raspberry Pi Camera Module and guizero to make a stop-frame animation application (Figure 1).

To complete this project, you'll need a Raspberry Pi with an official Camera Module (or High Quality Camera). If you need help connecting up the Camera Module, take a look at the 'Getting started with the Camera Module' guide at **rpf.io/picamera**. You will need guizero installed with the optional 'images' functionality, which you can install by running this command in the terminal:

```
pip3 install guizero[images]
```

This project is broken down into stages:

1. Taking a picture with the camera and displaying it on a GUI
2. Taking multiple pictures and saving them to a GIF
3. Allowing the user to change the GIF
4. Tidying up the GUI

Take a picture
Start by creating this program.

```
# Imports ---------------
from guizero import App, Picture, PushButton
from picamera import PiCamera

# Functions -------------
def capture_image():
    camera.capture("frame.jpg")
    viewer.image = "frame.jpg"

# Variables -------------
camera = PiCamera(resolution="400x400")

# App -------------------
```

```
app = App(title="Stop frame animation")

take_next_picture = PushButton(app,
text="Take picture", command=capture_image)
viewer = Picture(app)

app.display()
```

Note that the higher the resolution, the greater the processing time. 400×400 is small but really quick to process.

The GUI contains a PushButton and Picture. When the button is pressed, the `capture_image` function is called. The function uses the camera to capture an image and save it as **frame.jpg**. The picture is then displayed in the Picture widget. Test the program (**stopframe1.py**, overleaf). When you click the 'Take picture' button, the image should be displayed on the GUI (**Figure 2**).

Figure 1 A simple stop-frame animation

stopframe1.py

> Language: **Python 3**

```
001.   # Imports ---------------
002.
003.   from guizero import App, Picture, PushButton
004.   from picamera import PiCamera
005.
006.   # Functions ------------
007.
008.   def capture_image():
009.       camera.capture("frame.jpg")
010.       viewer.image = "frame.jpg"
011.
012.   # App ------------------
013.
014.   app = App(title="Stop frame animation")
015.
016.   camera = PiCamera(resolution="400x400")
017.   take_next_picture = PushButton(app, text="Take picture",
018.   command=capture_image)
019.   viewer = Picture(app)
020.
021.   app.display()
```

Take multiple images and save to a GIF

An animation is made of multiple pictures, known as frames. Each frame in the animation will be slightly different to the last and when played together at speed, the animation will appear to move.

In this step, you will change your GUI to keep a list of all the frames taken and use PIL (Python Imaging Library) to save the frames as an animated GIF which will be displayed in the viewer.

At the top of your program, import the Image module from PIL:

```
from PIL import Image
```

Create a list to store the frames of your animation:

```
frames = []
```

To keep track of how many frames have been taken, import a Text widget, add it to your app, and set it to 0.

```
from guizero import App, Picture,
PushButton, Text

total_frames = Text(app, text="0")
```

> ❝ An animation is made of multiple pictures, known as frames ❞

Each time a new image is captured, you will need to open it and append it to your list of frames:

```
def capture_image():
    camera.capture("frame.jpg")
    viewer.image = "frame.jpg"

    frame = Image.open("frame.jpg")
    frames.append(frame)
    total_frames.value = len(frames)
```

The len (length) of the frames list is then used to update the text in total_frames.

Your program should now look similar to **stopmotion2.py**. Test it and make sure the number of frames increases each time you take a picture.

Save as a GIF

You can PIL to save all the frames as one animated GIF. Create a new save_animation function to save the frames as **animation.gif**.

```
def save_animation():
    if len(frames) > 0:
        viewer.show()
        frames[0].save(
            "animation.gif",
            save_all=True,
            append_images=frames[1:])
        viewer.image = "animation.gif"
    else:
        viewer.hide()
```

There is a lot happening here, but by breaking down the code you see how this works. If the number of frames in the list is greater than 0, then the viewer is shown, otherwise it is hidden.

```
if len(frames) > 0:
    viewer.show()
    ...
else:
    viewer.hide()
```

The frames are then saved to a file called **animation.gif**. The first frame (frames[0]) is saved, the remaining frames (frames[1:]) are appended, and all are saved to the animated GIF.

▲ **Figure 2** Take a picture

stopframe2.py

➤ Language: **Python 3**

```
001.    # Imports ---------------
002.
003.    from guizero import App, Picture, PushButton, Text
004.    from picamera import PiCamera
005.    from PIL import Image
006.
007.    # Functions -------------
008.
009.    def capture_image():
010.        camera.capture("frame.jpg")
011.        viewer.image = "frame.jpg"
012.
013.        frame = Image.open("frame.jpg")
014.        frames.append(frame)
015.        total_frames.value = len(frames)
016.
017.    # Variables -------------
018.
019.    frames = []
020.
021.    camera = PiCamera(resolution="400x400")
022.
023.    # App -------------------
024.
025.    app = App(title="Stop frame animation")
026.
027.    total_frames = Text(app, text="0")
028.    take_next_picture = PushButton(app, text="Take picture",
029.    command=capture_image)
030.    viewer = Picture(app)
031.
032.    app.display()
```

```
frames[0].save(
        "animation.gif",
        save_all=True,
        append_images=frames[1:])
```

The **animation.gif** is then shown on the viewer.

```
viewer.image = "animation.gif"
```

Call the save_animation function at the *end* of the capture_image function to create and display the animation.

```
def capture_image():
    camera.capture("frame.jpg")
    viewer.image = "frame.jpg"

    save_animation()
```

Your code should now be similar to **stopframe3.py** (overleaf). Test it out.

Delete the last frame

At the moment, if you make a mistake while creating your animated GIF, you have to start again from the beginning.

You should modify your GUI to allow the last frame taken to be deleted, so if a mistake is made you can undo the change.

Create a new function which will remove or pop the last frame from the list, save the changed animation and then display it.

```
def delete_frame():
    if len(frames) > 0:
        frames.pop()
        total_frames.value = len(frames)

    save_animation()
```

The length of the frames list is checked before attempting to pop the last item. An error would be raised if you tried to pop an item from an empty list.

Add a PushButton to the GUI to call the delete_frame function, by inserting this code:

```
delete_last_picture = PushButton(controls,
align="left", text="Delete last",
command=delete_frame)
```

Note: You could also modify the GUI to allow you to delete any frame, not just the last one.

stopframe3.py

> Language: **Python 3**

```
001.    # Imports ---------------
002.
003.    from guizero import App, Picture, PushButton, Text
004.    from picamera import PiCamera
005.    from PIL import Image
006.
007.    # Functions -------------
008.
009.    def capture_image():
010.        camera.capture("frame.jpg")
011.        viewer.image = "frame.jpg"
012.
013.        frame = Image.open("frame.jpg")
014.        frames.append(frame)
015.        total_frames.value = len(frames)
016.
017.        save_animation()
018.
019.    def save_animation():
020.        if len(frames) > 0:
021.            viewer.show()
022.            frames[0].save(
023.                "animation.gif",
024.                save_all=True,
025.                append_images=frames[1:])
026.            viewer.image = "animation.gif"
027.        else:
028.            viewer.hide()
029.
030.    # Variables -------------
031.
032.    frames = []
033.
034.    camera = PiCamera(resolution="400x400")
035.
036.    # App -------------------
037.
038.    app = App(title="Stop frame animation")
039.
040.    total_frames = Text(app, text="0")
041.    take_next_picture = PushButton(app, text="Take
042.    picture", command=capture_image)
043.
044.    viewer = Picture(app)
045.
046.    app.display()
```

Changing the timing

Each frame is displayed for the default duration time of 100 milliseconds. Include a Slider widget in your GUI to allow the duration to be changed.

Add it to the list of imports.

```
  from guizero import App, Picture,
PushButton, Text, Slider
```

Then create the widget in the app.

```
    Text(app, text="Duration")
  duration = Slider(app, start=100, end=1000,
command=save_animation)
```

The start and end parameters will be the minimum and maximum times you can set for the frame duration.

Each time the slider is changed, the save_animation function will be run.

Update the save_animation function to use the duration value when saving the GIF.

```
    frames[0].save(
            "animation.gif",
            save_all=True,
            append_images=frames[1:],
            duration=duration.value)
```

> ❝ Each frame is displayed for the default duration time of 100 milliseconds ❞
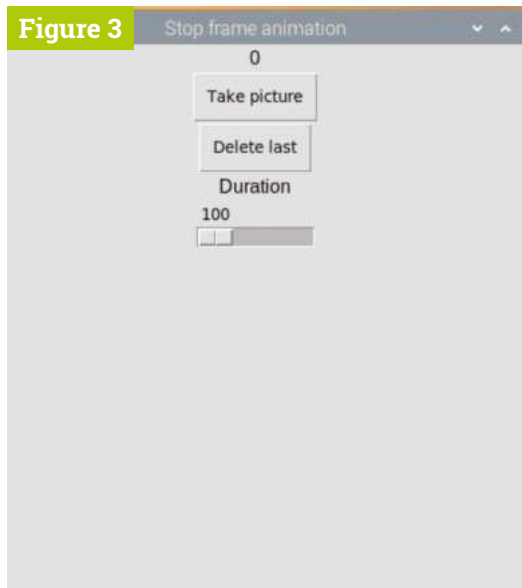
▲ **Figure 3** Controls stacked at the top

stopframe4.py
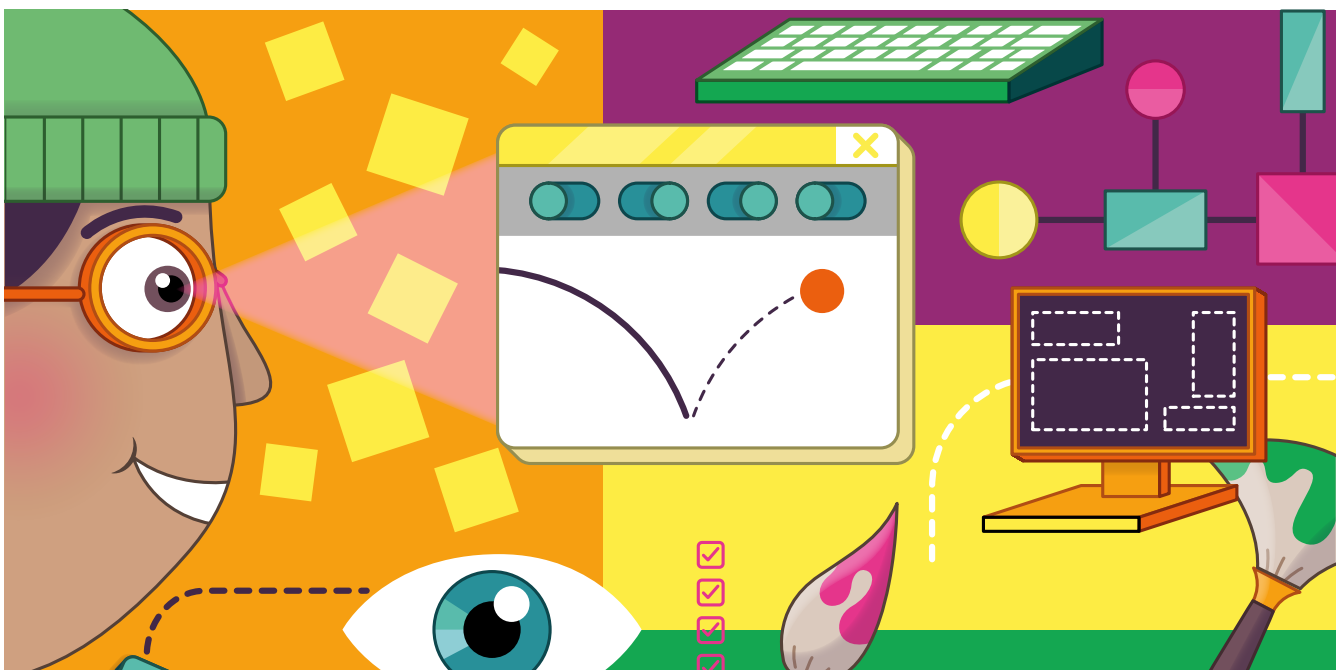
> Language: **Python 3**

```
001.    # Imports ---------------
002.
003.    from guizero import App, Picture, PushButton, Text,
004.    Slider
005.    from picamera import PiCamera
006.    from PIL import Image
007.
008.    # Functions -------------
009.
010.    def capture_image():
011.        camera.capture("frame.jpg")
012.        viewer.image = "frame.jpg"
013.
014.        frame = Image.open("frame.jpg")
015.        frames.append(frame)
016.        total_frames.value = len(frames)
017.
018.        save_animation()
019.
020.    def save_animation():
021.        if len(frames) > 0:
022.            viewer.show()
023.            frames[0].save(
024.                "animation.gif",
025.                save_all=True,
026.                append_images=frames[1:],
027.                duration=duration.value)
028.            viewer.image = "animation.gif"
029.        else:
030.            viewer.hide()
031.
032.    def delete_frame():
033.        if len(frames) > 0:
034.            frames.pop()
035.            total_frames.value = len(frames)
036.
037.        save_animation()
038.
039.    # Variables -------------
040.
041.    frames = []
042.
043.    camera = PiCamera(resolution="400x400")
044.
045.    # App ------------------
046.
047.    app = App(title="Stop frame animation")
048.
049.    total_frames = Text(app, text="0")
050.    take_next_picture = PushButton(app,
            text="Take picture", command=capture_image)
051.    delete_last_picture = PushButton(app,
            text="Delete last", command=delete_frame)
052.    Text(app, text="Duration")
053.    duration = Slider(app, start=100, end=1000,
            command=save_animation)
054.
055.    viewer = Picture(app)
056.
057.    app.display()
```

Your code should now resemble **stopmotion4.py**. Try it out.

Align the controls

At the moment, the controls are taking up a lot of room stacked at the top of the GUI (**Figure 3**).

Create a Box and align it to the top of the GUI to hold the controls, first adding it to the imports.

```
   from guizero import App, Picture,
PushButton, Text, Slider, Box

  controls = Box(app, align="top")
```

Modify the widgets so that they are in the controls box and set the **align** parameter to **"left"**. For example:

```
  total_frames = Text(controls, text="0",
align="left")
```

Aligning widgets to the left inside the box will make them stack up next to each other. Repeat this for rest of the controls so they are all put into the top box and lined up next to each other.

Your complete program should look similar to **11-stop-motion.py** (overleaf). **M**

Create Graphical User Interfaces with Python

For further tutorials on how to make your own GUIs with guizero, take a look at our book, *Create Graphical User Interfaces with Python*. Its 156 pages are packed with essential info and a range of exciting projects.
magpi.cc/pythongui
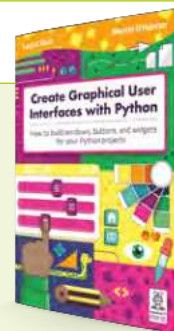
11-stopmotion.py

> Language: **Python 3**

```python
001.  # Imports --------------
002.
003.  from guizero import App, Picture, PushButton, Text,
004.  Slider, Box
005.  from picamera import PiCamera
006.  from PIL import Image
007.
008.  # Functions -------------
009.
010.  def capture_image():
011.      camera.capture("frame.jpg")
012.      viewer.image = "frame.jpg"
013.
014.      frame = Image.open("frame.jpg")
015.      frames.append(frame)
016.      total_frames.value = len(frames)
017.
018.      save_animation()
019.
020.  def save_animation():
021.      if len(frames) > 0:
022.          viewer.show()
023.          frames[0].save(
024.              "animation.gif",
025.              save_all=True,
026.              append_images=frames[1:],
027.              duration=duration.value)
028.          viewer.image = "animation.gif"
029.      else:
030.          viewer.hide()
031.
032.  def delete_frame():
033.      if len(frames) > 0:
034.          frames.pop()
035.          total_frames.value = len(frames)
036.
037.      save_animation()
038.
039.  # Variables -------------
040.
041.  frames = []
042.
043.  camera = PiCamera(resolution="400x400")
044.
045.  # App -------------------
046.
047.  app = App(title="Stop frame animation")
048.
049.  controls = Box(app, align="top")
050.  total_frames = Text(controls, text="0", align="left")
051.  take_next_picture = PushButton(controls, align="left",
052.      text="Take picture", command=capture_image)
052.  delete_last_picture = PushButton(controls,
053.      align="left", text="Delete last", command=delete_
053.      frame)
053.  Text(controls, align="left", text="Duration")
054.  duration = Slider(controls, align="left", start=100,
055.  end=1000, command=save_animation)
056.
057.  viewer = Picture(app)
058.
059.  app.display()
```

Build an Android
tablet with Raspberry Pi

Emteria.OS is a version of Android for Raspberry Pi. Let's use it in its natural, portable environment, and use it for development

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

Top Tip 👍

Installer issues

Run the installer as Administrator if you're having issues, and check if the SD card runs, even if the final checks fail.

Android on Raspberry Pi has been a topic of discussion for years in specific spaces, and while there have been attempts in the past to port Android, emteria.OS has been the best way to use Android on Raspberry Pi for a few years now. With some recent great updates, we're revisiting this great OS to get it working as an Android tablet, and see how it can be used for Android dev.

01 Raspberry Pi hardware

You'll need a Raspberry Pi 4 for this project so that you can make the most out of your custom tablet, although you can use a Raspberry Pi 3B+, Raspberry Pi 400, and Compute Module 3 with emteria. Emteria also supports HDMI on Raspberry Pi so you can connect it to a monitor, TV, or other kind of small display depending on if you want to use it as a tablet for development.

With an official touchscreen, you can cut down on some wires and have it work in a traditional Android way.

02 Build your tablet – part 1

To make sure everything is working before fully committing to our build, we suggest just connecting Raspberry Pi 4 to the display at first. You may need to construct your screen, however that is quite simple and there are full instructions here: **magpi.cc/screenbuild**.

You do not need to add the jumper wires between the control board and Raspberry Pi though, as you will need to power Raspberry Pi and the display separately. You can use the special USB splitter cable that came with the SmartiPi Touch 2 case if you wish to make it easier down the road, however you will at least need a micro USB power supply and a USB C power supply.

03 emteria.OS Installer

You'll need to create an account for emteria. OS, which you can do at **emteria.com**. Once you've done that, you can head to the download page at **magpi.cc/emteriainstaller** and download the installer. Run the program once it's downloaded to

Emteria is a version of Android that can run on Raspberry Pi and other hardware

Using a touchscreen and case, you can create your own Android tablet using emteria

install – you may need to tell Windows to 'run the software anyway' if a warning window pops up.

Once this is done, open the Emteria installer app, and you will be asked to login using the account details you created earlier. From there, select your hardware (Raspberry Pi 4), choose the right SD card if you have multiple plugged in, and then click flash for the installer to do its job.

04 Initial emteria.OS boot

Once the install has finished, plug your SD card into your Raspberry Pi connected to the touchscreen. Put the power in for the screen (the micro USB on the controller board), and then plug the power into Raspberry Pi 4.

It will cycle through the boot a few times before finally showing the emteria logo pulsing – after a while this will transition to a grey screen and do some set up before allowing you to click on next, where you'll perform the classic Android setup.

05 Initial emteria.OS setup

The touchscreen will be working at this point, however you can always connect a mouse and keyboard if you want to run through the process a little quicker. After setting your time zone time, and date, you then need to select your WiFi network to connect to. Once again, if you've used Android this should be familiar.

Now you need to activate your install (although you can test it out for eight hours before doing this). We logged in via our account, but you can enter a key if that's easier for you. Skip the next two steps, agree to the EULA, and hit reboot for everything to be applied.

06 Build the tablet – part 2

If installation and setup went fine, you can install your Raspberry Pi and screen into the SmartiPi case. Once again there is a handy build guide on how to do this: **magpi.cc/smartipibuild**.

At this point you can add a Raspberry Pi Camera Module to the whole case build, however at the

> ❞ It will cycle through the boot a few times before finally showing the emteria logo ❞

▲ Affixing Raspberry Pi to the back of the official touchscreen requires only screws and a DSI ribbon cable

time of writing it is not supported in the OS. The emteria team are working on it though, and some USB webcams are supported right now, so if you're desperate to take photos there are ways.

07 Install extra software

As emteria is not an official version of Android, you'll not have access to the standard Play Store. You can still sideload APK files if you download files to the device though, and there is F-Droid which is a free and open-source app store for Android. Opening up this app will prompt you to do an update before installing any apps, just like you would on the Play Store. You will have to allow it to install third-party apps from the system settings, which it will help you do.

08 Advanced settings

You've probably noticed that the tablet is currently set to landscape. With no gyro inside, you cannot automatically change the orientation. Bringing up the emteria settings allows you change the orientation to portrait (90 degrees), or even flip the entire screen over (180 degrees). You'll also notice there are a load of the developer settings sitting in here, such as keeping the screen awake (useful if you have it plugged into a power supply

for this stage), along with other settings for remote work, Ethernet connections, launching apps at boot, etc.

> ❝ One of the great features of emteria is its ability to act as a kiosk ❞

09 Other Android differences

While many parts of emteria work like stock Android, there are a few differences that are worth noting. First of all, as there is no dedicated off button on a Raspberry Pi, you'll find there is a shutdown app in the library that allows you to turn off, reboot, and head to recovery. The browser is not Chrome, and other default apps are swapped with non-Google based ones. You can still login to Google services using the browser though, and there are other browsers you can try as well.

10 Kiosk mode

One of the great features of emteria is its ability to act as a kiosk – a terminal with limited functions that you set yourself. It's meant for more
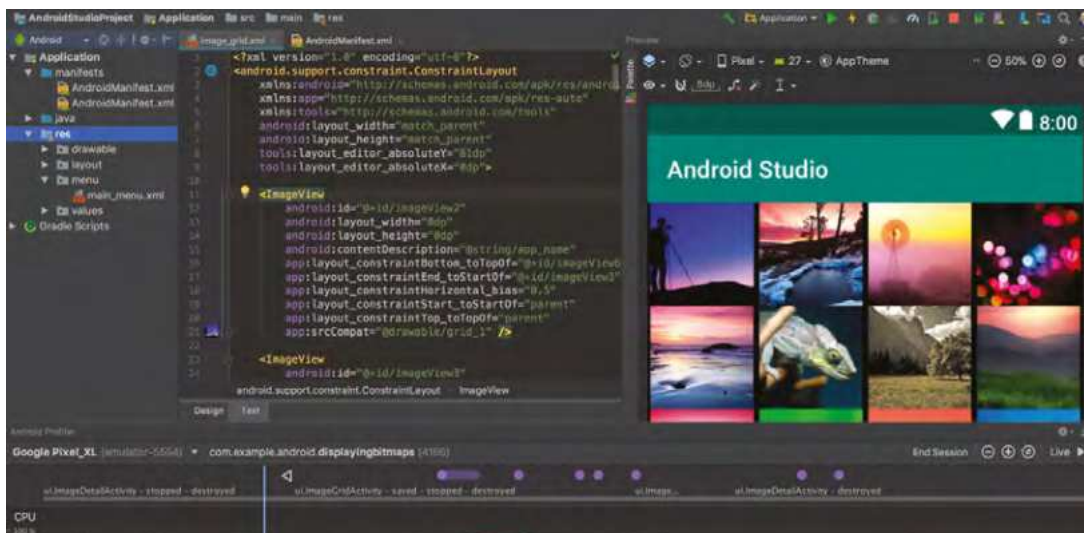
Android Studio is the development suite for Android apps

Choosing specific apps that can only be used in Kiosk mode can be great for kids or cafés

public places that might require people to sign up to a service, or get directions in a large area.

To turn it on, go to the emteria settings and toggle Kiosk state. Set a password so you can turn it back off, and then select Kiosk or Web launcher. In Kiosk launcher you can choose 'approved apps', which will be the only apps the users can use. Similarly, web launcher sets a selection of bookmarked websites that are the only ones that can be visited. Tapping ten times on the screen will bring back the Kiosk settings menu.

11 Portability

As well as a portable battery, taking a power supply with you can also work in case you stop in a place with power. SmartiPi Touch 2 also has a stand, making it easier to use on the go as well, however it does mean it's not quite as slim when folded up – the stand part can actually be removed with a screw to cut down on this size, making the whole thing a little easier to take with you.

12 Development tips

Emteria is based on the latest version of Android – Android 11 – making it a great development environment for Android apps. Even just connecting it to a monitor, or using the VNC functions, will allow you to test out apps you're creating.

Heading to **developer.android.com** will give you some great beginner tutorials and sample code on how to start, as well as the tools needed to build apps. **M**
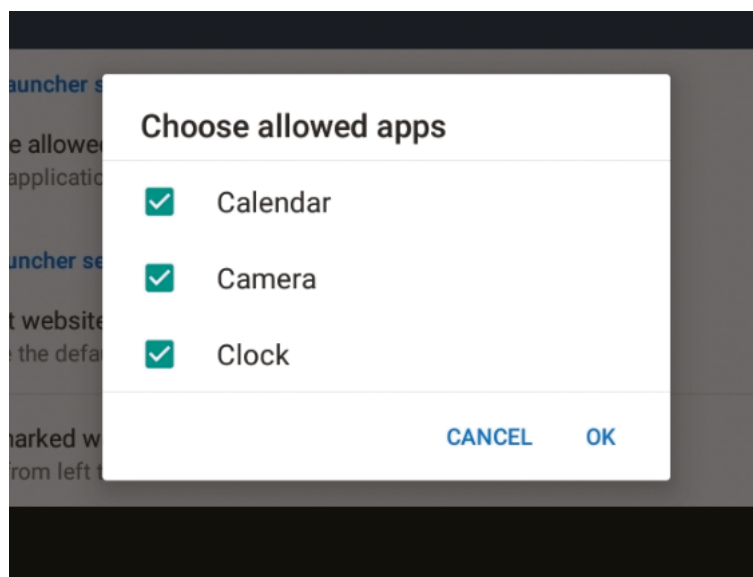
Emteria uses a specific program, much like Imager, to burn the latest version of the OS to an SD card
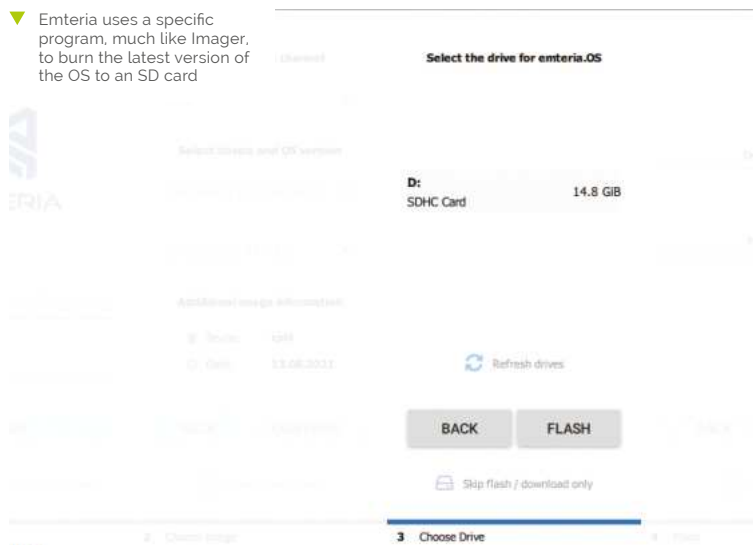
THE Official
RASPBERRY PI
HANDBOOK
2022

50 Raspberry Pi
Hacks & Hints

iPod Classic
Spotify Player

BeccaCam
Digital Camera

Raspberry Pi
Pico projects

Build a
Home Assistant

Raspberry Pi
Amiga 600

Walking
Bipedal Robot

Updated!
QUICKSTART GUIDE

Start incredible projects
with Raspberry Pi

SPOOK UP YOUR RASPBERRY PI

Unnerving upgrades for your pedestrian projects

As dusk draws nearer and the days get colder, we approach the most frightful night of the year, where ghouls and ghosts come out to play and witches and werewolves can be heard in the distance. While we can't help you repel these repugnant ruffians, we can enable you join them with some blood-curdling builds.

This year we're going to help you take your regular projects and make them scary, allowing you to save some time and put on the perfect Halloween party. Read on, if you dare…

Malicious Magic Mirror

Add a serious scare to your smart mirror – visitors beware!

Magic mirror projects are very cool and very practical, and also completely terrifying if you program them to give jump scares. When we first covered them in the December 2016 issue, creator Bradley Melton was already adding scary pictures to replace the inspirational quotes.

Sadly, this module was never released for the Magic Mirror system (**magicmirror.builders**), but you can get a nice little module that adds seasonal lights here: **magpi.cc/mmlights**.

If you do want to add something truly scary to your existing magic mirror, though, look no further than Charles Gantt's 'spooky surprise' tutorial here: **magpi.cc/spookymirror**. It has details on how to add your own PIR sensor, and instructions on how to program your mirror to do the scares.

Mirror in motion

How the motion sensing works in practice

Shocking Santa Detector

Go from ho ho ho to no no no with this motion-sensing trap – give it a go!

The Santa Detector is a classic project from the Raspberry Pi Foundation, making use of a PIR motion sensor hooked up to some Scratch code so you can find out when Big Red has arrived. Instead of Santa, though, what if it just detected any passer-by to give them a fright?

Upgrading this project for our uses is fairly simple. Instead of flashing a picture of Santa, we can replace it with a scary picture. We can also add a scream sound effect to Scratch to go with the picture. Lastly, we'll have to camouflage it so it can reach the maximum number of people for a scare.

magpi.cc/santadetector

A NEW LOOK

Instead of jolly old Saint Nick, we want a scary photo. We found this picture of a skull on black that is sure to give anyone the heeby jeebys. You can use any other picture you like, or draw one yourself. To add a picture to Scratch, go to the costumes tab, hover over the circle 'Choose a Costume' button at the bottom, and select upload costume. Delete the other costumes.

1

WE ALL SCREAM

From our repository of spooky upgrades, you will find a file called **Wilhelm_scream. mp3**. In the Sounds tab, hover over the round button at the bottom, and click Upload File to upload this new scary scream to your project. In the code, replace 'whoop' in the **play sound** block with 'Wilhelm_scream'.

2
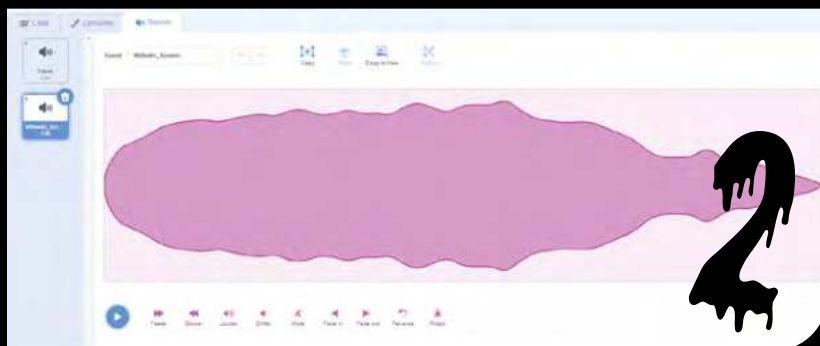
Similarly spooky

POSSESSED PORTRAIT

This seemingly normal painting is actually a monitor in a frame with a little PIR motion sensor, activating an animation of the portrait screaming at passers-by. It's the same concept as the Santa detector on a grander scale.

magpi.cc/possessedpic

HAUNTED JACK IN THE BOX

Instead of a PIR motion sensor, this project uses a Raspberry Pi Camera Module to detect when someone is specifically looking at it, setting off an automated Jack-in-the-box with variable speeds and sounds for maximum frights.

magpi.cc/hauntedjack

Download new images and code from magpi.cc/spookyupgrades

Hidden FROM SIGHT

3

When hiding your PIR sensor, you can go simple or advanced. We like the simple method of secreting ours on a shelf nestled among figures or DVDs. Make sure the monitor is visible to anyone who will be passing by your hiding spot. If you want to go advanced, there are some nice PIR cases you can print from Thingiverse.

Frightening Photo Booth

Create eerie images with these cunning code tweaks – fun for all ages!

A familiar sight at official Raspberry Pi events, the All-Seeing Pi is a fun photo booth that lets you add stickers to photos of yourself and tweet them out into the world. It's a fun thing to replicate at parties and other kinds of events, including perhaps an upcoming Halloween party of your own.

To make this appropriately alarming, a first step will be to change the overlays from fun sunglasses and wigs to fangs and witches' hats. We can add an extra, basic overlay filter with a little trick as well.

magpi.cc/allseeingpi

Awful accessories

It's quite simple to switch out the overlays for spooky props – simply add any images you have (or ones from our repository) to the **overlays** folder of your All-Seeing Pi code. They should be PNG files with transparency. Give them simple file names and then head to **overlay_functions.py** and change the contents of the overlays list to the names of the new images; e.g.,

```
overlays = ['witchhat', 'fangs', 'catears'].
```

1

Similarly spooky

CREEPY FACE TRACKING PORTRAIT

Using a camera and a screen in a much different way – to change a creepy face so that it's always looking at you. Always.

magpi.cc/trackingportrait

STOMACH SHOT

Freak out your friends by making your camera and screen portable so that you can see through yourself for the perfect zombie costume.

magpi.cc/stomachshot

Fearful filters

We're going to cheat a bit here to make some filters. Draw or find some photo borders with a clear centre and open them in a photo editing program. Add a layer below the border and fill it in with a light green or a light brown colour. Reduce the opacity of the coloured layer to 30% and voilà, instant filter. Test it with a photo on the layer below the border and colour to make sure it's what you want, and then save it to the overlays folder as before, updating the list in **overlay_functions.py**. Alternatively, we've created some you can use.

> " IT'S A FUN THING TO REPLICATE AT PARTIES AND OTHER KINDS OF EVENTS "

Double trouble

If you add an extra button to your All-Seeing Pi, you can have both the filters and props show up at once. Create a separate folder for the filters called **filters** and then copy the **overlay_functions.py** file to this folder, renaming it filter_functions.py. Update the code inside with the new folder location name, and change the overlays list to filters. Update **allseeingpi.py** by adding:

```
from filter_functions import *
```

Follow the steps in the All-Seeing Pi tutorial to add the extra filters to the images you take. There's also a version of that code in our repository.

DASTARDLY DISPLAY LIGHTING

Go from everyday LEDs to haunted Halloween colours – hang them from trees!

The amazing temperature-controlled stair lights by Lorraine Underwood (@LmcUnderwood) are a great way to set up decorative lights anywhere – they don't just need to be on stairs. They also don't have to be a rainbow. We have a couple of options here to make it a little more Halloween-themed, and that's mainly down to the colour palettes of the lights. Think oranges, greens, and purples, classic colours for this spooky time of the year.

magpi.cc/tempstairs

	blueviolet	#8A2BE2	rgb(138,43,226)
	darkviolet	#9400D3	rgb(148,0,211)
	darkorchid	#9932CC	rgb(153,50,204)
	darkmagenta	#8B008B	rgb(139,0,139)
	purple	#800080	rgb(128,0,128)
	indigo	#4B0082	rgb(75,0,130)

EERIE RGB

NeoPixel lights use an RGB colour scale, assigning the red, green, and blue elements an intensity value from 0 to 255. For example, a ghastly green would be (0, 255, 0). For purple and orange, we need a specific mix: purple is (128, 0, 128) and orange is (255, 140, 0). You can also cheat for black by setting some LEDs to (0, 0, 0)

1

TERRIFYING TEMPERATURES

The original aim of the project was to help you figure out the temperature. You can still do that with this version if you wish, switching out the original colours for our spooky ones. We suggest orange for a colder colour, then green, and finally purple for a warmer temperature.

2

> **GO ALL OUT AND DECORATE YOUR HOME WITH LIGHTS AND MUSIC POWERED BY RASPBERRY PI**

Mysterious
MUSIC

You can have music dictate how your lights move using LightShow Pi from Adafruit (**magpi.cc/lightshowpi**). Even if you're not going to play the music, you can still use some tracks to help randomise and animate your display – quicker than programming it yourself.

3

Similarly spooky

POPLAWSKI'S HOLIDAY LIGHTS

With an evergrowing selection of decorations, the Poplawskis not only make their front porch spooky, they let you control it over the internet.

magpi.cc/poplawski

RASPBERRY PI HALLOWEEN LIGHTS AND MUSIC SHOW

Go all out and decorate your home with lights and music powered by Raspberry Pi. We've also seen people hack cheap projectors in recent years for this as well…

magpi.cc/hallolightsmusic

Keybow **2040**

▶ Pimoroni ▶ **magpi.cc/keybow** ▶ £50 / $57

An RP2040-based auxiliary keyboard for macros, samples, and streams. **Rob Zwetsloot** gives them a tickle

Ever since the release of Raspberry Pi Pico, we've been interested in its ability to power keyboards, both standard and custom. We've seen a couple of DIY projects doing just this, and Pimoroni even released an add-on with a soft keys (RGB Keypad: **magpi.cc/rgbkeypad**) for Pico at its launch. It was only a matter of time before the RP2040 – the chip that powers Pico – would be used on its own dedicated keypad with real keys. This inevitable product ended up being another Pimoroni product: Keybow 2040.

It comes with one of three sets of key switch types depending on your preference: mechanical (clicky-clacky), linear (silent), and tactile (silent with some resistance). All versions have an LED in the switch which is fully programmable, and the keys are standard sized so you can easily swap them out for others if you'd like. There's also a cheaper base kit for which you'll need to provide your own switches and keycaps.

Lock on

Some construction is required, but it's very minimal – add the switches, sandwich the layers together, and then put the keys on. We were done in a couple minutes, at which point you can actually start using the Keybow as a number pad by plugging it in.

Unlike a standard Pico, the Keybow is connected via a USB-C cable to your computer of choice. There doesn't seem to be any technical

> ❝ Much cheaper and more hackable than some of the commercial alternatives ❞

▶ Each key is individually programmable, with customisable RGB LEDs as well

▲ Everything you get in the box – it seems a lot, but it's a quick build

advantage to having USB-C; however, it does mean the basic design has some forward compatibility for Pimoroni. Also, it's nice to have more stuff making use of USB-C. A couple of hardware buttons are also included: one to reset the Keybow, and one to hold down to do the boot selection if you want to tinker with the code.

Custom combo

The code itself is written in CircuitPython, which is derived from the MicroPython – both can be used on a regular Pico and they're largely similar as well. Whatever changes you need can be gleaned from the abundant example programs that are installed on Keybow 2040.

With little effort we were able to change the way the keys work – with constantly lit buttons that change when pressed, and passing along a specific combos of keys for when they're pressed rather than a single numpad key. Due to the way it loads scripts, you can immediately test out your edits, making for very quick prototyping and bug fixing.

We've been using a linear version for weeks now during streams as a custom stream deck, and it works perfectly with our custom setup (the code for which we'll drop on **magpi.cc/github**). It's sturdy, turns on quickly, the LEDs are bright, and we've had no noticeable lag on a button press. With the removable caps, you can add custom ones with specific labels like a mute icon or specific scene modes, etc., making it perfect for this, and much cheaper and more hackable than some of the commercial alternatives. 🄼

Verdict

A premium-feeling product that is easily customisable and priced much more fairly than less-open alternatives.

10 /10

PoE USB Hub HAT
for **Raspberry Pi Zero**

▶ The Pi Hut ▶ **magpi.cc/poeusbzero** ▶ £22 / $25

Add Power over Ethernet and three USB ports to Raspberry Pi Zero. By **Phil King**

▶ Four pogo pins on the upper USB board connect to test pads on Raspberry Pi Zero

Verdict

Adds both PoE and extra USB ports to Raspberry Pi Zero without using the GPIO header.

8/10

The PoE USB HAT for Raspberry Pi Zero not only adds an Ethernet port, but enables you to power everything via the same standard RJ45 cable. A nice bonus is the inclusion of three full-size USB Type A ports to connect peripherals.

This double-decker HAT features the Ethernet port and fully isolated SMPS (Switch Mode Power Supply) on the bottom board, with a second board stacked on top holding the three USB ports. Coming already assembled, the two boards are held together by hex pillars.

Extra pillars and screws are supplied to mount your Raspberry Pi Zero on the top – maybe this should be called a HAB (Hardware Attached on Bottom) rather than a HAT!

Free header

The USB board has four pogo pins that connect to test pads on the underside of Raspberry Pi Zero, so you need to ensure it's mounted tightly and accurately to make solid connections.

An obvious advantage of mounting the HAT
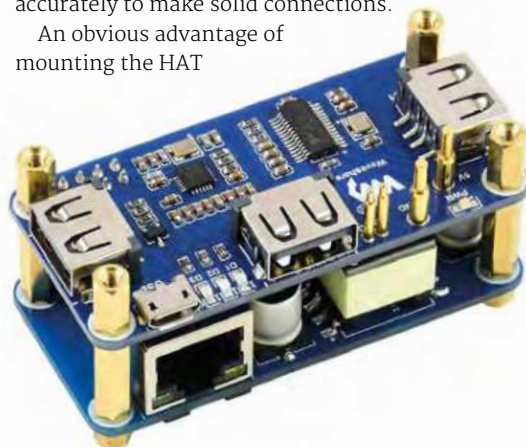
▲ With Raspberry Pi Zero mounted on top, the double-decker HAT adds three USB ports on the top board and PoE on the bottom

> ❝ Leaves Raspberry Pi Zero's GPIO header free to connect other boards ❞

underneath Raspberry Pi Zero is that it leaves the latter's GPIO header free to connect other boards or electronic components.

One caveat is that you'll need an 802.3af compatible router for Power over Ethernet to work. If not, you could always add a PoE switch or injector between the router and HAT. With an input voltage of 37 to 57 V DC and output of 5 V 2 A, up to around 13 W of power can be supplied, which should be plenty for most uses. Alternatively, if you don't need PoE, you could simply power Raspberry Pi Zero normally via micro USB, which will in turn power the HAT.

While we've seen DIY solutions for adding PoE or standard Ethernet connectivity to Raspberry Pi Zero, and USB ports via a hub, this HAT offers a neat all-in-one solution. ⧫

SunFounder Basic starter kit for Raspberry Pi Pico

- Based on Raspberry Pi Pico
- Quickly learn embedded programming and electronic circuits
- Comes with 283 items, 18 projects
- Detailed online tutorials
- Quick-response technical support

Available at: sunfounder.com/products/pico-starter-kit

SunFounder Pico Car Kit
Completely Open-Source

Pico Car Kit

Raspberry Pi Pico is included

- Obstacle Avoidance
- Object Tracking
- Speed Measurement
- Mileage Calculation
- Line Following
- Cliff Detection

Available at: sunfounder.com/products/raspberrypi-pico-car

Electronic Clock for
Raspberry Pi Pico

▶ The Pi Hut ▶ **magpi.cc/picoclock** ▶ £24 / $27

A large LED clock powered by Pico and
with customisable functions. By **Phil King**

SPECS

DISPLAY:
22×7 pixel
green LED,
plus status
sidebar and
top weekday
display

FEATURES:
DS3231
RTC with
temperature
sensor, backup
battery,
photosensor,
buzzer, 3 ×
push-buttons,

DIMENSIONS:
216 (W) x 79 (H)
x 25 mm (D)

Verdict

Solidly built with a
large display, it's
a cool clock, and
the demo code
is customisable
– although you'll
need to know
your C++.

8/10

The size of this Electronic Clock by
Waveshare is impressive. At 21.9 by 7.9 cm,
it's a chunky device and its 22×7 green LED
display with large pixels is readable from a
fair distance.

Lifting off the rear panel reveals the electronics
and two female headers to mount your Raspberry
Pi Pico (not included). First, however, you'll need
to flash Pico with the software. The online manual
(**magpi.cc/picoclockguide**) is confusing, as
there's no need to compile anything with terminal
commands if using a Raspberry Pi: just drag the
downloadable UF2 across to the mounted Pico.

You can then mount Pico onto the rear headers,
close the rear panel (securing it with the supplied
screws), plug it in, and the clock will start working.

Customisable clock
With settings adjusted using three push-buttons
on the side, standard clock functions include two
alarms, a timer, and an hourly chime option, all
using the on-board piezo buzzer. There's also the
option of scrolling the display every three minutes
to show the date and temperature (in °C or °F, read
from the on-board RTC chip's built-in sensor).

That RTC (real-time clock), along with a CR2032
coin cell battery, means the clock will remember

▲ A Raspberry Pi Pico flashed with the software can
be slotted into the clock's rear female headers

> ❝ The USP of this clock is
> the ability to customise
> the example code to
> add extra functions ❞

the correct time even if disconnected from 5 V
power (via the supplied USB to micro USB cable),
although you'll lose any custom settings. There's
also an option to automatically adjust the LED
brightness based on the built-in photoresistor.

The unique selling point of this clock is the
ability to customise the example code and add
extra functions, which is done solely in the
C/C++ programming languages – there's no
official MicroPython support. **M**

▲ The chunky LED display is impressive and has rear panel holes for
wall mounting
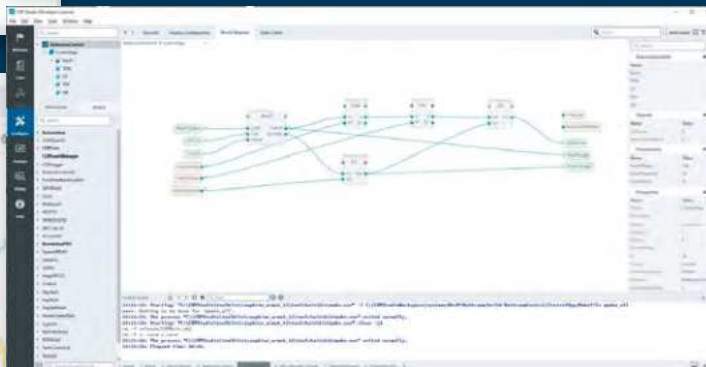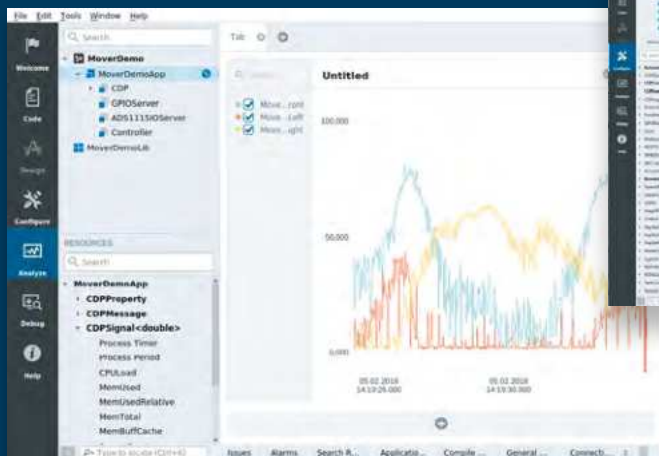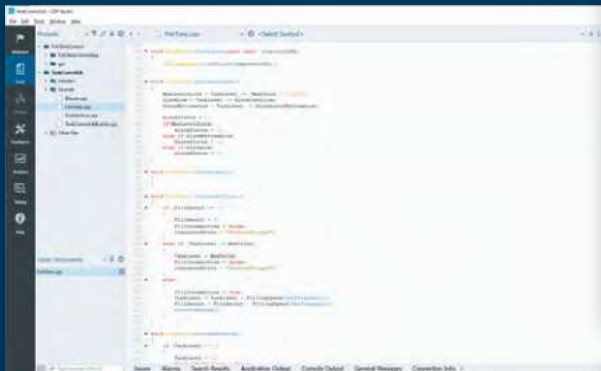
10 Amazing:

Raspberry Pi instruments

Get musical with these Raspberry Pi-powered musical creations

Raspberry Pi is an incredible piece of hardware for audio, and we know many people that use it for amazing sound systems. Did you know you can also create music on it? Not just with code either, as these projects show! M

▲ Hex Keys

Isomorphic keyboard

One of the great instruments designed by Mike Cook, this laser-cut keyboard is quite different to a traditional piano keyboard, but can still make amazing tunes.

magpi.cc/pibakery

▲ Soundfighter

A new challenger

What do you do once you've mapped piano keys to a PS2 controller? You build a second, identical piano and duke it out in Street Fighter. Go home and be a piano man.

magpi.cc/soundfighter

▶ Piano stairs

Up the scale

You don't just have to make music with your hands – with a little inspiration from *Big*, you too can turn a little fun stair workout into a musical experience.

magpi.cc/pianostairs

▲ Trill MIDI Guitar

New electric guitar

Another invention from Mike Cook, this excellent MIDI guitar uses special Trill touch sensors and a little display to create a cool, customisable guitar.

magpi.cc/trillguitar

▲ LED piano visualiser

Sound to light

These kind of visualisers are really popular on YouTube for pianos, giving them a rhythm game vibe. Create one for your own piano.

magpi.cc/pianovisualiser

▶ Pedal Pi

Effects via Raspberry Pi

This amazing project isn't an instrument per se, but it is part of one – a guitar effects pedal with a huge amount of customisation.

magpi.cc/pedalpi

▲ Self-playing xylophone

Robotic glockenspiel

With as few wires as possible, Stéphane was able to create this automated xylophone that uses many solenoids to play out a MIDI file on the classic instrument.

magpi.cc/xylophone
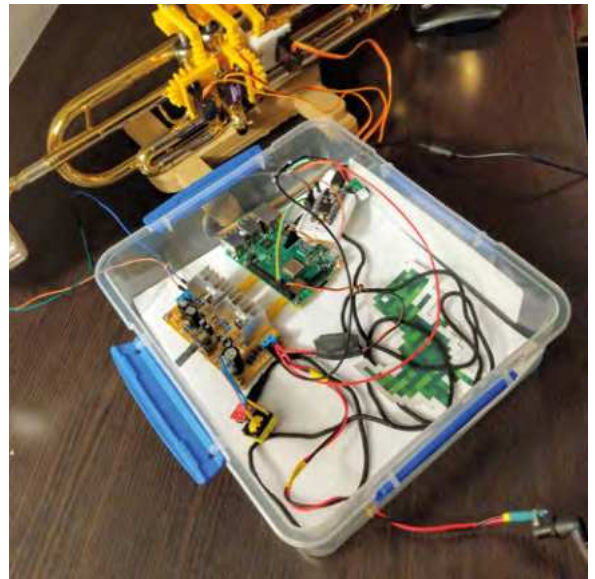
▲ MIDI Trumpet Robot

Automated jazz

This trumpet is played by robotic fingers depressing the valves, with MIDI sound actually playing from a speaker at the mouthpiece. It's controlled over the web.

magpi.cc/miditrumpet

▶ Ultrasonic theremin

Spooky tunes

With an ultrasonic distance sensor, you can control notes on Sonic Pi to create eerie tunes, perfect for a 1950s alien invasion B-movie vibe.

magpi.cc/theremin

▶ Raspberry Pi Guitar Rig

One-man band

Building an amp into an acoustic guitar was only the first step. Adding a Raspberry Pi and touchscreen to control it on the fly to change synthesizer settings makes this extra-special.

magpi.cc/guitarrig

Learn networking
with Raspberry Pi

Discover more about computer networking using these resources. By **Phil King**

An Introduction to Computer Networking for Teachers
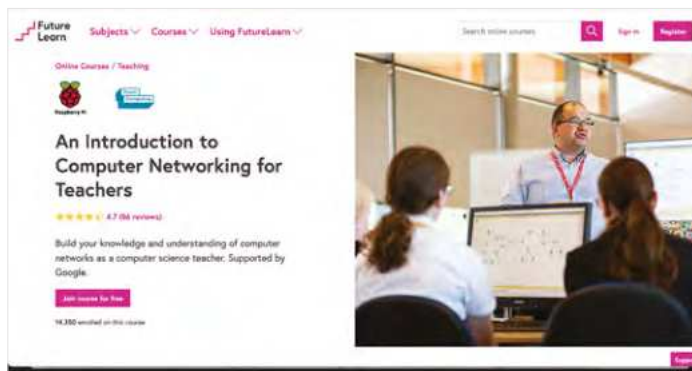
AUTHOR

Raspberry Pi Foundation

Price:
Free

magpi.cc/introtonetworking

Computer networking gives us access to the internet and allows us to communicate with people all over the world, but how many of us know how it really works?

Aimed at computer science GCSE teachers, this foundation course is a useful starting point for anyone curious about computer networks. It takes around six hours to complete, over three weeks, although you can work at your own pace. Signing up for a free account gives you five weeks' access to the course, but there's the option to upgrade for continued access and to receive a certificate of achievement.

The self-guided course teaches you the fundamentals of networking. Week one, 'Creating Connections', covers the basics of computer network connections; week two, 'Packets & Protocols', explores IP address and packets, along with protocols such as TCP/IP, DHCP, and NAT. The final week, 'The Net as a Network', covers the World Wide Web, cloud computing, and cybersecurity. **M**

Online courses

Study networking with these interactive courses

LINUX NETWORKING AND ADMINISTRATION (LFS211)
Aimed at IT professionals, this self-paced course from the Linux Foundation is pricey ($299) but offers 50 hours of course material including videos and assignments.
▶ **magpi.cc/lfs211**

NETWORKING WITH PYTHON
Another free course from the Raspberry Pi Foundation, it covers the use of sockets in network programming, including methods for testing applications.
▶ **magpi.cc/flpythonnetwork**

A TO Z LINUX NETWORKING CONCEPTS
This Udemy course (£59.99) aims to improve your Linux network administration skills by learning to build, maintain, and secure a computer network.
▶ **magpi.cc/azlinuxnetwork**

Raspberry Pi Networking Cookbook

AUTHOR
Rick Golden

Price:
£29 / $45

magpi.cc/ rpinetworking

For a Raspberry Pi-specific take on networking, this 352-page book features a collection of server-side recipes to get you started and develop your knowledge and skills. From basics such as remote access using SSH, it goes on to cover file sharing with Samba, configuring a static IP address, generating a firewall with ufw, setting up an Apache web server, creating a wireless access point with hostapd, and monitoring network traffic with Wireshark. Bonus chapters cover Internet of Things projects (using a GrovePi board) and setting up a Raspberry Pi cluster.
While this Second Edition, released in 2016 when we were running Raspbian Jessie on Raspberry Pi, is a little dated in some aspects, it does contain a fair amount of useful knowledge. **M**
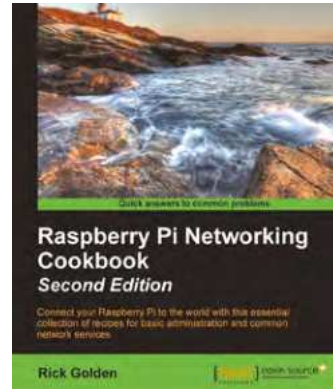
Mastering Python Networking

AUTHOR
Eric Chou

Price:
£38 / $50

magpi.cc/ pythonnetworking

Fully updated for Python 3, the 576-page Third Edition explores using the power of Python libraries to master network engineering tasks, covering topics such as network automation, programmability, and data analysis. It's aimed at IT professionals and operations engineers, and at least a basic knowledge of Python programming and networking is required to make the most of it. After an overview of the TCP/IP suite and Python, followed by low-level device interactions, it moves on to cover APIs, the Python-based Ansible automation framework, network security and testing, and network monitoring.
Later chapters tackle building network web services with Flask, AWS and Azure cloud networking, network data analysis with Elastic Stack, working with Git, and using Jenkins to automatically create operations pipelines. **M**
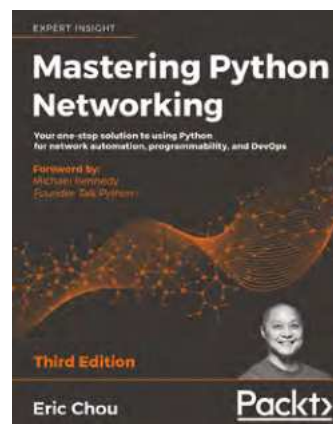
Free web resources

Check out these online networking resources

DEBIAN REFERENCE
Since Raspberry Pi OS is based on Debian, the networking section of the online Debian Reference manual is ideal for learning a host of useful commands.
▶ **magpi.cc/ debianrefnetwork**

RASPBERRY PI FORUMS
For troubleshooting Raspberry Pi-specific networking issues, the 'Networking and servers' section of the official forums is a good place to go for help
▶ **magpi.cc/ networkingforum**

LINUX KERNEL ARCHIVES
The networking section of the Linux Kernel Organization archives is packed with information on just about every aspect of Linux networking.
▶ **magpi.cc/ kernelnetworking**

Matt Gray

A YouTuber with a penchant for making, with an amazing Raspberry Pi-powered Game Boy Camera

> ▶ Name **Matt Gray** | ▶ Occupation **Broadcast Engineer**
> ▶ Community role **YouTuber** | ▶ Website **magpi.cc/mattgray**

▶ The Fast WiFi Adapter is a modern way to interact with classic tech

Matt Gray has been making for years – both in terms of cool projects and also cool YouTube videos. He's recently been in the spotlight for a great hack of a Game Boy Camera; however, he deserves a bit more fanfare than that for all he does.

"I'm a Broadcast Engineer by day, but I've been making videos on YouTube for over a decade now!" Matt explains to us. "I've also helped organise Electromagnetic Field Festival, a non-profit UK camping festival for those with an inquisitive mind or an interest in making things: hackers, artists, geeks, crafters, scientists, and engineers.

"I've been making bits and bobs for years, the bigger projects generally being ones that make me laugh: The Webcycle, an exercise bike that only lets you use the internet when you're pedalling.

"The Freddie Mercury Thermometer, because it seemed like no-one had made that joke yet.

"[And] the Hoverpub, where we converted a small hovercraft into a pub!"

When did you learn about Raspberry Pi?
I found out about Raspberry Pi when it very first came out, and I used the original Raspberry Pi as part of the Freddie Mercury Thermometer.

Where did the Game Boy Camera hack idea come from?
I've had my Game Boy Camera since the 1990s: it was my first digital camera, and I love it! It's a cartridge that goes into the Game Boy like a game, but converts it into a camera. The photos it takes are really low resolution (128 × 112 px) and greyscale but, considering that, I'm always surprised by how clearly the photos can come out

It was only ever designed to take 30 photos, print them using the Game Boy Printer, then delete them to make space for more. But after years in a box, I wanted to use it again, and after some searching online, I found
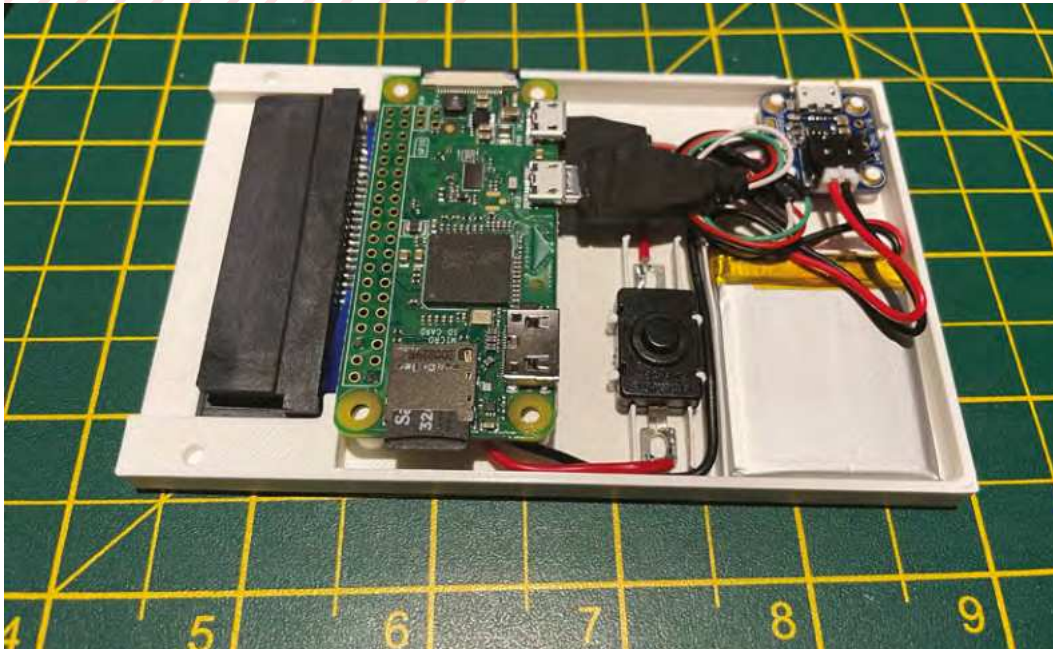
◀ We're fans of the very clean insides of the adapter

a long-winded way to get photos off the camera around 2013, and was able to start using it again to not only take photos, but also share online.

Its 30-photo limit was getting annoying after a while, and I didn't want to also have to lug a laptop and bunch of cables and adapters around with me to be able to clear it off. Then it hit

charger and a switch. I've programmed [this] Raspberry Pi so that when you turn it on with the camera plugged in, it automatically runs the cartridge reader's software to extract photos from it, and save them onto Raspberry Pi. Then it hosts a WiFi network, so you can connect to it and download the photos from its web page.

> ❝ I didn't want to also have to lug a laptop and bunch of cables and adapters around with me ❞

me. A Raspberry Pi Zero is a tiny computer I could take with me… With my newly arrived 3D printer, I could combine [Raspberry] Pi and all the other bits needed into one compact unit to take with me!

How does the Game Boy Camera hack work?
Inside the custom 3D-printed case, there's a USB Game Boy cartridge reader (**gbxcart.com**) plugged into a Raspberry Pi Zero W, which are powered by a LiPo battery via an Adafruit MicroLipo

For more details and to see it in action, please watch my YouTube video on it: **magpi.cc/gbcam**.

What else do you vlog about?
This isn't my first escapade with the Game Boy Camera. A few years ago I worked out how to take colour photos with it (**magpi.cc/gbcamcolour**), even though it's only greyscale! And you might know me from my previous project, 'Will it soft serve?' (**magpi.cc/wiss**). M

▲ Taking photos of beautiful things with a beautiful classic

MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!! M

01. This is a very neat project, and we do appreciate it being able to monitor multiple tanks at a time

02. Hopefully it still retains some of that Raspberry Pi 400 power though

03. Sensors like this are slightly harder to make than you'd think, so we appreciate this build

04. A companion rabbit robot – a rabbot? Robbit?

05. More excellent use of weight sensors with Hexabitz

06. This cute case has a secret surprise

07. A very fun project to do with people of all ages!

08. 'Robot chaos' is what we call our drawer with robots we've reviewed

09. The Model D looks very swish

10. We hope the fish appreciate their high-tech living space

11. The full *Planets Suite* by our new favourite orchestra

Jitesh Saini
@jiteshsaini10

Replying to @TheMagPi

Created this online water tank widget to work with #RaspberryPi and ultrasonic distance sensor. Anybody can access and configure their own widget without writing any server side code.
Video: youtu.be/ETrYPMRdL-E
Website: helloworld.co.in/iot/

Customisable Online Widgets for Water Tank Project

Nick Bild
@NickBild79

Replying to @TheMagPi

The Pi64 is a Raspberry Pi 400 that thinks it's a Commodore 64:

Pi64
The Pi64 is a Raspberry Pi 400 that thinks it's a Commodore 64.
hackster.io

Hexabitz
@HexabitzInc

03

Replying to @TheMagPi

Good morning 🌞
Raspberry Pi Force Gauge and Weight Scale Using Hexabitz load cell sensor 🥊 😀
#MagPiTuesday #MagPiMonday #raspberrypi #loadcell

Raspberry Pi Force Gauge and Weight Scale Using **Hexabitz** Modules

0:37 1K views

Tom Faux (Vulpestruments)
@Vulpestruments

04

Replying to @TheMagPi

My Pizero companion bot is slowly but surely coming together. It will sit on my shoulder and do things.

Aula-J
@Aula_J2019

05

Replying to @TheMagPi

😄 🐟 🍎 🌱 @Raspberry_Pi Smart Food Scale with @HexabitzInc Modules hackster.io/aula-jazmati/r...
@Hacksterio @KassabNajib @dfrobotcn

Raspberry Pi Smart Food Scale with Hexabitz Modules
In this tutorial we are building a Raspberry Pi smart food scale with Hexabitz modules
🔗 hackster.io

Penk Chen
@penk

06

Replying to @TheMagPi

A low tech but fun build — 3D printed "Offline T-Rex" Pi4 case with a hidden camera

🦖 STL files available: thingiverse.com/thing:4950479

Frank
@Frank76139220

07

Replying to @TheMagPi

Tried my hand on stop-motion and wired up a button to take a picture. Using @TheMagPi essentials camera module guide.

Dr Footleg - Roboteer @drfootleg · 1h
I created robot chaos at the British Science Festival in Chelmsford.
#MagPiMonday

08

> **Dr Footleg - Roboteer** @drfootleg · Sep 11
> I've created robot chaos here at the British Science Festival with the #EssexSteamettes and @SouthendRPJams
> Robots have been running nonstop. 1 robot broken, 2 have batteries run flat and 1 controller has been broken already! Lots of kids loving it though.

0:08 496 views

TGD
@TGD_Consulting

09

Replying to @TheMagPi

Tim spent his weekend #3Dprinting parts of the new PiMowBot Model D to see how they fit together. After further quality checks on @Creality3dprint Ender3 the new 3D model are available @Cults3D, asap, stay tuned.
#Make #DIY #3Dprinted #IoT #smart #robotics #RPi based #lawnmower

HomeMadeGarbage
@H0meMadeGarbage

10

Replying to @TheMagPi

Last week I maintained the goldfish aquarium system using Raspberry Pi ZERO.
homemadegarbage.com/goldsystem02

10:35 AM · Sep 13, 2021 · Twitter Web App

diyelectromusic
@diyelectromusic

11

Replying to @TheMagPi

I've had my @Raspberry_Pi "singing" the offstage chorus for Neptune, the last movement of my #LoFiPlanets.

My #LoFiOrchestra has now performed all seven movements of Holst's masterpiece.

#MagPiMonday

Raspberry Pi Computing Education Research Centre

The new University of Cambridge department will make learning better

I f you've been paying attention to the work of the Raspberry Pi Foundation, you'll have seen all the studies into how different backgrounds effect the way people learn and interact with computing. With this new initiative, this research will be able to be applied to new learning.

"With computers and digital technologies increasingly shaping all of our lives, it's more important than ever that every young person, whatever their background or circumstances, has meaningful opportunities to learn about how computers work and how to create with them." Philip Colligan, CEO of the Raspberry Pi Foundation, wrote in a blog (**magpi.cc/rpcercblog**). "That's our mission at the Raspberry Pi Foundation."

New research

"The new Research Centre is a joint initiative between the University of Cambridge and the Raspberry Pi Foundation," Philip continues. "[It] builds on our long-standing partnership with the Department of Computer Science and Technology. That partnership goes all the way back to 2008, to the creation of the Raspberry Pi

Foundation and the invention of the Raspberry Pi computer. More recently, we have collaborated on Isaac Computer Science, an online platform that is already being used by more than 2500 teachers and 36,000 students of A level Computer Science in England, and that we will shortly expand to cover GCSE content.

"Through the Raspberry Pi Computing Education Research Centre, we want to increase understanding of what works in teaching and learning computing, with a particular focus on young people who come from backgrounds that are traditionally under-represented in the field of computing or who experience educational disadvantage.

"The Research Centre will combine expertise from both institutions, undertaking rigorous original research and working directly with teachers and other educators to translate that research into practice and effect positive change in young peoples' lives."

If you'd like to get involved, you can subscribe to the research newsletter here: **rpf.io/research-update**.

▲ Gender studies are among those that have been commissioned from the Raspberry Pi Foundation

Best of the rest!

Other amazing things from the community

ARGON EON

The folks at Argon make great cases, and this four-bay NAS box for Raspberry Pi 4, that they're crowdfunding, looks like an amazing piece of kit. This case will be made from aluminium with a triangular design, and will also be used for cooling. It even has a cute little OLED display.

▶ **kck.st/3BR153B**

HOME AUTOMATION HAT

"Home Automation HAT is a home monitoring and automation controller HAT for the Raspberry Pi that can control up to two appliances/devices on a load of 250 VAC/7 A, 30 VDC/10 A, photo-coupling isolation Optocoupler EL357NC, allowing users to safely control high voltages or current devices."

▶ **kck.st/3ksMnbY**

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

What-If machine

A 3D-printed *Futurama* prop that you can ask any What If question, which it will answer with a random episode of *Futurama*.

▶ **magpi.cc/whatiftv**

Stock ticker

We always love a nice e-ink display project, and this one is a cool, simple one that tracks your favourite stock.

▶ **magpi.cc/stockticker**
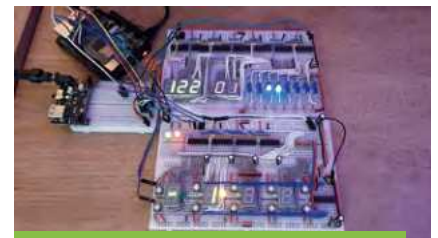
Pallet Town sequencer

Similar to our favourite digital orchestra, this sequencer plays the Pallet Town music from the original Pokémon games.

▶ **magpi.cc/pallettown**

Your
Letters
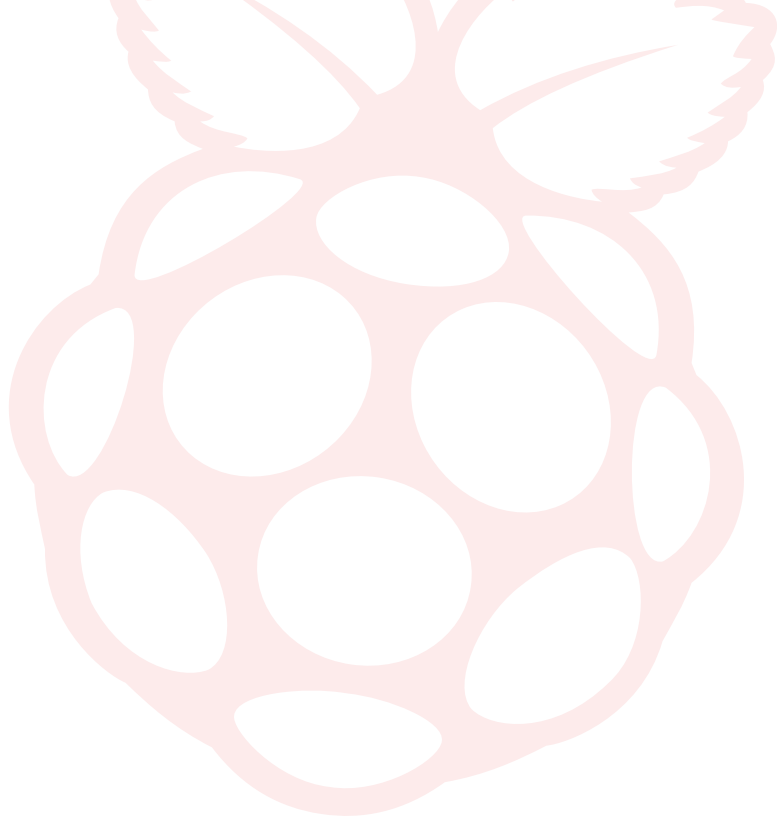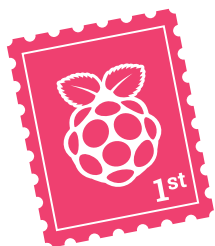
Universal projects

Now that Raspberry Pi 4 is out, what about an article about some great projects for the old Raspberry Pi 3A+? I have a Raspberry Pi 3A+, and am having a hard time finding good projects to use it for. Maybe a WiFi radio? Digital photo frame (though it might not have enough RAM)? Anyway, just thought it might be a fun article.

Wesley via email

Thanks for the suggestion! This is definitely something we've thought of doing. A lot of projects that use a Raspberry Pi 4 will work fine with a Raspberry Pi 3A+, though: even with lower power, not all projects use that CPU speed or RAM, so the only thing you might struggle with is having the right ports – and again, you can work around that.

For something like a digital photo frame, a Raspberry Pi 3A+ is also perfectly fine! As a rule of thumb, if it can run on a Raspberry Pi Zero, it can pretty much run on any Raspberry Pi.

▶ Raspberry Pi 3A+ is still a great computer that can be used for amazing projects
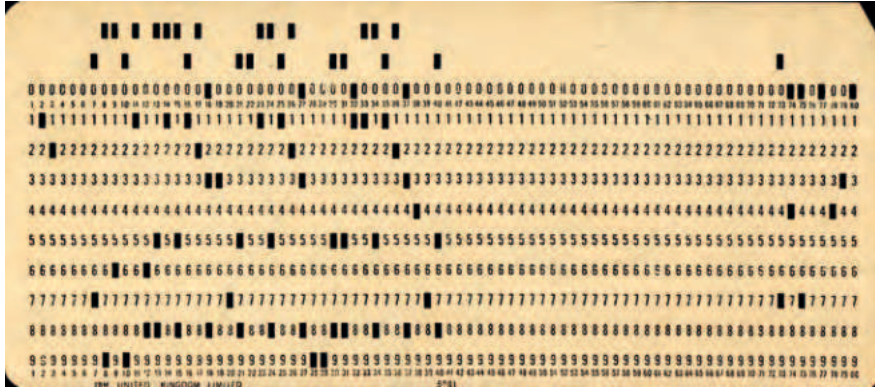
▶ Punch cards are a fascinating part of computer history, representing data simply with holes in a piece of card

Credit: Pete Birkinshaw CC BY 2.0

Astro Pi upgrades

I'm hoping to enter the next Astro Pi competition, if there is one, and I was wondering about the tech specs of the two Astro Pi up on the ISS. Are they Raspberry Pi 3? I'd like to test my idea before submitting!

Lottie via Facebook

Currently, it's a Raspberry Pi B+ inside Ed and Izzy Astro Pi. However, you'll be pleased to know that Astro Pi is not only getting a new mission, but also a huge upgrade to Raspberry Pi 4, along with some extra sensors and such. You can find out more details about the new computers on page 8.

▲ The new Astro Pi are much more powerful than the previous ones

Seeing our place in history, again

I read Lucy Hattersley's article 'Our place in history' in issue 108 of *The MagPi* magazine. As an 85-year-old retired programmer, I can relate to her in-laws' computer experience. I got into the computer field in the 1960s after a career change from teaching high school physics and math. I have used punched cards, punched paper tape, magnetic tape, floppy disks, etc. to store programs and data. I now have about a dozen Raspberry Pi [boards] for various projects. My latest project was to build a robot car with my nine-year-old great-grandson. I used a Raspberry Pi Zero W for that project. It's controlled with the Blue Dot Android App. My first computer build was based on the Motorola 6800 processor (SWTPC 6800). I built that in the late 1970s. The basic kit had 400 bytes of RAM and used an audio cassette unit for storage. I later upgraded to 32kB of RAM and an 8-inch floppy drive. It had no operating system. The firmware that it booted into was called MIKBUG. Most of my programming was done in assembly language. My favourite computer to program was DEC's PDP/11. I retired in 1998, but I have tried to keep up with technology.

William via email

Wow, that is a long history with computers. Thank you for sharing your story with us! We hope your great-grandson enjoyed his first steps with Raspberry Pi, and will have fun making projects with one for years to come.

Contact us!

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

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE FOR THE MODERN MAKER

MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | October 2021 | Issue #47

LORA

50 best 3D prints

Essential builds for your home, office, and workshop

ALSO INSIDE

TOUCH INPUT
Add capacitive touch to Pico

DIY GAMES CONTROLLER
Build your own input device

CARDBOARD PICO **SMD** PURE DATA

ISSUE #47

OUT NOW

hsmag.cc

WIN ONE OF TEN

AIR QUALITY KITS FOR RASPBERRY PI

We reviewed the Air Quality Kit for Raspberry Pi last issue, and said it was a "great kit for experimenting with monitoring air quality. Very easy to set up and start using, and with some good coding examples and excellent tutorials." We have ten up for grabs!

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

Terms & Conditions

Competition opens on **29 September 2021** and closes on **28 October 2021**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation and Raspberry Pi Trading, 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.
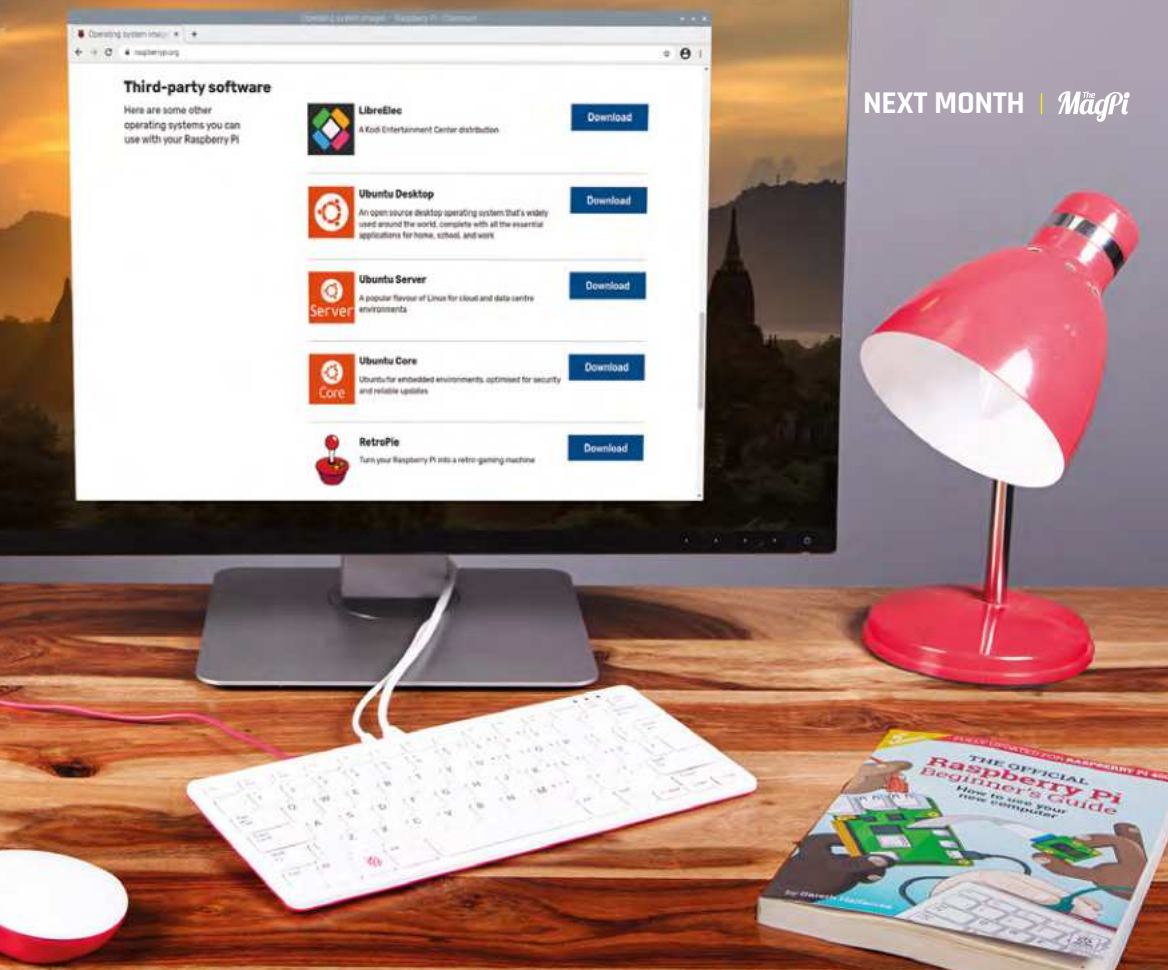
RASPBERRY PI
OPERATING SYSTEMS

THE MAGPI #111
ON SALE 28 OCTOBER

Plus!

Walk a Spider Robot

Learn circuit design

Build a Pico light game

DON'T MISS OUT!
magpi.cc/subscribe

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

A life in electronics

Who knows where the time goes? By **Mike Cook**

It was all a mistake, I was meant to be a chemist. Back when I was five-years-old I knew I wanted to be a scientist, and by ten, various science toys convinced me it was chemistry. But I was dyslexic, which was not recognised in those days. This caused my teachers to predict low-level manual work to my parents when I was five. Also dyslexia resulted in me going to a Secondary Modern school, but they had no Chemistry Lab. So when it came to picking a science subject at the age of thirteen, it was Physics.

I left school at the age of 16 and got a job as an apprentice technician at a small electronics company in Oldham, being paid the princely sum of £5.50 a week. They allowed me to do a day release where I took an ONC (Ordinary National Certificate) course in Electrical Engineering.

This, in turn, enabled me to take a combined Physics & Electronics degree at Newcastle upon Tyne Polytechnic. When I went for an interview, my first encounter with a strong Geordie accent did leave me wondering where the hell I was. The maths module had three lectures about programming in Fortran II,

these were the only lessons I had in programming, but I was hooked!

After graduation, I worked on a form of audio compression called adaptive delta, pulse code modulation or ADPCM, later to be used by Microsoft and Apple.

I was in the right place at the right time. In 1976 I built my first computer from just the information in the data sheet of the Signetics 2650 microprocessor. I got a

" I was in the right place at the right time "

lecturing position at Manchester Polytechnic's Physics department, and one day I was contacted out of the blue by Mike Bibby. He had been asked to start a magazine about home computers and an ex-student had told him about me. I agreed to do four articles, convinced the magazine would not go further. Well, it did go further: 210 issues in total with my articles under the titles of 'Body Building Course' (BBC), and 'Run the RISC'.

When Raspberry Pi took over the running of *The MagPi* magazine, I was approached to see if I could write for it. Previously I had one article published in *The MagPi* magazine issue #5 (**magpi.cc/5**), which made the front cover. However, 'Mike's Pi Bakery' began in *The MagPi* issue #33, and ran all the way to issue #109.

Having reached the grand old age of 70, I felt it was time to catch up on all those projects I had been wanting to do for years but never had the time. The first one is to convert all the Sony 8 videotapes of my children growing up into a digital format. However, I am not going for good, I expect to return for the occasional ad-hoc article.

It has been an absolute privilege to write in *The MagPi* magazine, and attending various Raspberry Pi events, and shows, meeting some of my readers. Thanks for having me.
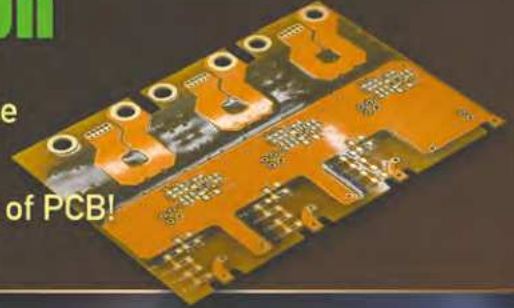
AUTHOR

Mike Cook

Despite, and maybe because of, his dyslexia Mike Cook has inspired and educated many people over the years in what is now known as physical computing.

magpi.cc/theBox