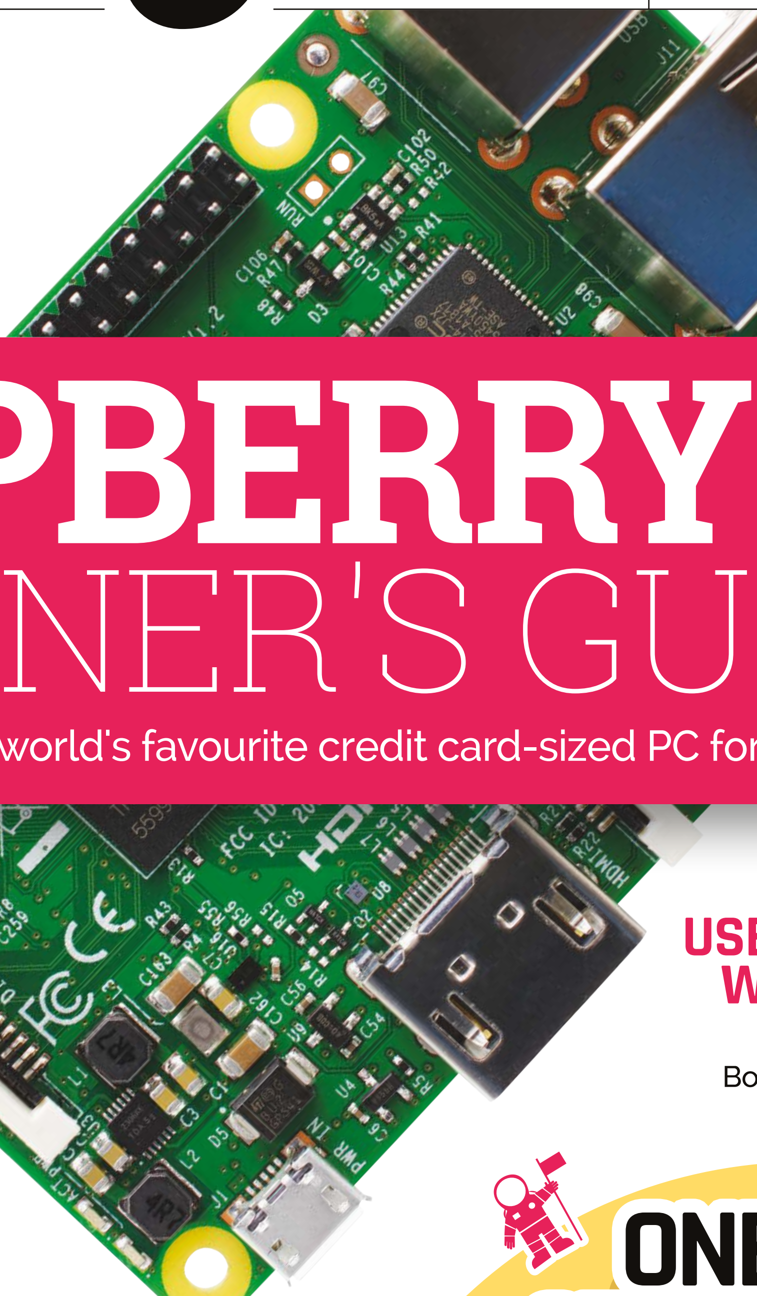
ALL NEW RASPBERRY PI GETTING STARTED GUIDE

The MagPi

The official Raspberry Pi magazine

Issue 49 September 2016

raspberrypi.org/magpi



BUILD A FOOTBALL GAME IN SCRATCH

Forget FIFA. Code your very own footy game today!

RASPBERRY PI BEGINNER'S GUIDE

How to set up and use the world's favourite credit card-sized PC for the first time

VOICE-CONTROL YOUR PI

Take command with Amazon Alexa

USE YOUR PI 3 WITHOUT AN SD CARD

Boot via USB with our expert guide

ONE SMALL STEP FOR PI

Emulate the Apollo computer on your Raspberry Pi

Also inside:

- A BRAILLE MUSICAL INSTRUMENT ANYONE CAN PLAY
- **SMALL BUT MIGHTY ZEROBORG MOTORBOARD**
- MORE OF YOUR RASPBERRY PI PROJECTS
- **THE PIPER PI LAPTOP TESTED & RATED**

Issue 49 • Sept 2016 • £5.99

09

9 772051 998001

THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMUNITY

strato

Strato Pi Touch Display

Raspberry Pi 3
Raspberry Pi Touch Display
Strato Pi board

Power Supply 12-28VDC
Real Time Clock
Buzzer
Optional RS232/RS485 & UPS

Aluminium and steel rear chassis
Brushed aluminium front bezel
Optional back box for wall mount

SFERALABS
www.sferalabs.cc

WELCOME TO THE OFFICIAL PI MAGAZINE!

Despite its small size, the Raspberry Pi is positively overflowing with possibilities. For hackers and makers it's a blank canvas, easel, and palette rolled into one. While many are very happy to use it as the affordable computer it is, for most hobbyists it's the key to unlocking a whole world of possibilities. Of course, you can't paint a landscape without learning the basics, so – starting on page 16 – the newest member of the magazine team, Lucy Hattersley, shows you how to get to grips with world's most famous credit card-sized computer. If you're a new owner, or want to help a friend or relative, it's a fantastic way to get started.

When it comes to hacking and making, the sky is definitely not the limit, because in our feature starting on page 68 we show you how to emulate the very same computer systems that took us to the moon in the 1960s, with the out-of-this-world Apollo Pi project.

Enjoy the issue!

Russell Barnes
Managing Editor

THIS MONTH:

16 GET STARTED WITH RASPBERRY PI
Just got your first Pi? Nail the basics with our beginner's guide

46 LEARN TO CODE IN C
Resident programming expert Simon Long continues his series

58 HAVE FUN WITH SCRATCH
Get lost in space and build your own football game

68 GO TO THE MOON AND BACK
Emulate the Apollo computer that sent humankind to the moon

FIND US ONLINE raspberrypi.org/magpi **GET IN TOUCH** magpi@raspberrypi.org

The MagPi

EDITORIAL
Managing Editor: **Russell Barnes**
russell@raspberrypi.org
Features Editor: **Rob Zwetsloot**
News Editor: **Lucy Hattersley**
Sub Editors: **Laura Clay, Phil King, Lorna Lynch**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Mike Kay**

PUBLISHING
For advertising & licensing:
russell@raspberrypi.org +44 (0)7904 766523
Publisher: **Liz Upton**
CEO: **Eben Upton**

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

SUBSCRIPTIONS
Select Publisher Services Ltd
PO Box 6337
Bournemouth
BH1 9EH | **+44 (0)1202 586 848**

CONTRIBUTORS
Sam Aaron, Wes Archer, Alex Bate, Luke Castle, Mike Cook, Ioana Culic, Gareth Halfacree, Richard Hayler, Spencer Organ, Phil King, Simon Long, Matt Richardson & Richard Smedley

Contents

Issue 49 September 2016

raspberrypi.org/magpi

TUTORIALS

> ALEXA ON RASPBERRY PI **44**
Get voice commands working using Amazon's Alexa

> INTRO TO C PART 3 **46**
Simon Long's continuing beginner's guide to C

> RHYTHMIC GYMNASTICS **48**
Mike's Pi Bakery continues its Olympics-themed projects

> WYLIODRIN CAR MONITOR **52**
Count the number of passing cars with a Pi

> SONIC PI: PUT ON A SHOW **54**
Tips to put on your own live Sonic Pi set

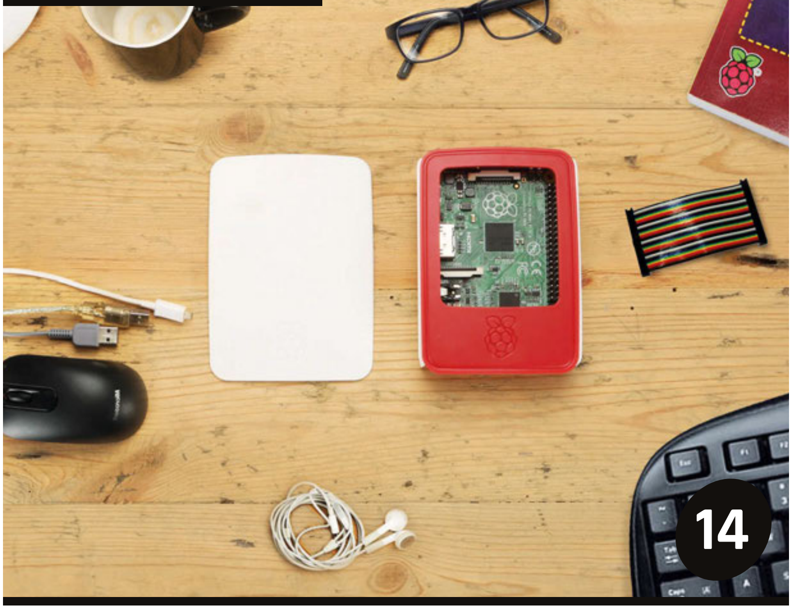
> ARCADE MACHINE BUILD PT 3 **56**
Get the display working on your DIY RaspCade

> SCRATCH FOOTBALL **58**
Forget FIFA, it's time to make your own football game

> SCRATCH ANIMATION **60**
Get lost in space and learn how to animate in Scratch

> MOTION TIME-LAPSE VIDEO **62**
Create a portable, moving, time-lapse device

COVER FEATURE

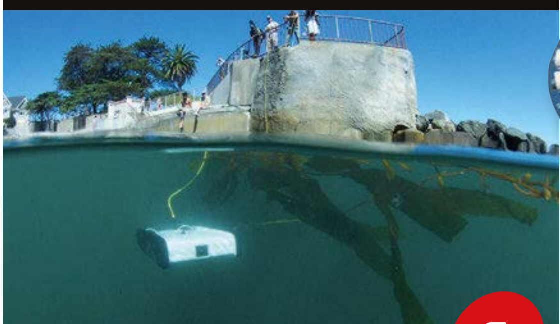


14

GETTING STARTED WITH RASPBERRY PI

IN THE NEWS

Trident ROV submarine



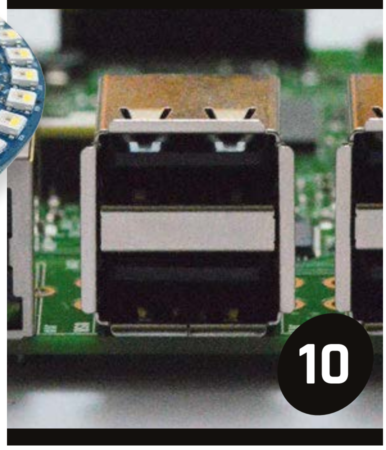
How the crowdfunded ROV sub is being powered by a Raspberry Pi **6**

MATRIX CREATOR **8**

The MATRIX Creator developers plan to open up IoT technology to everyone

Ethernernet & USB Booting

The Raspberry Pi 3 now supports USB and Ethernet booting



10

THE BIG FEATURE

68

APOLLO PI

Boldly go where no Pi has gone before by emulating the computers used on the Apollo moon missions

WIN A PIBORG YETIBORG

piborg.org

95

ZACH IGIELMAN

The creator of PiPiano is the subject of our first community spotlight

88

YOUR PROJECTS

32

ZERO 360

James Mitchell creates amazing panoramas with this very special project

Earthquake Pi 34
An earthquake alarm built using a Raspberry Pi and a toothbrush motor

Tablet Ocarina 36
Creating music that both the blind and sighted can read

Torus Visual Music 38
How Amsterdam club nights have been made better with the power of Pi

NESPi 40
An accurate recreation of a NES, scaled down in size and with working mini-cartridges

REGULARS

> NEWS 06
The biggest stories from the world of Raspberry Pi

> TECHNICAL FAQ 64
Got a problem? Maybe we can help!

> BOOK REVIEWS 84
This month's best reads for coders and hackers

> THE FINAL WORD 96
Matt Richardson tells us how libraries are using Pi

COMMUNITY

> THIS MONTH IN PI 86
We round up other goings-on in the Pi community

> EVENTS 90
Find out about Jams and Pi gatherings this month

> YOUR LETTERS 92
We answer your letters about the magazine and Pi

REVIEWS

> ZEROBORG 76

> ENVIRO PHAT 78

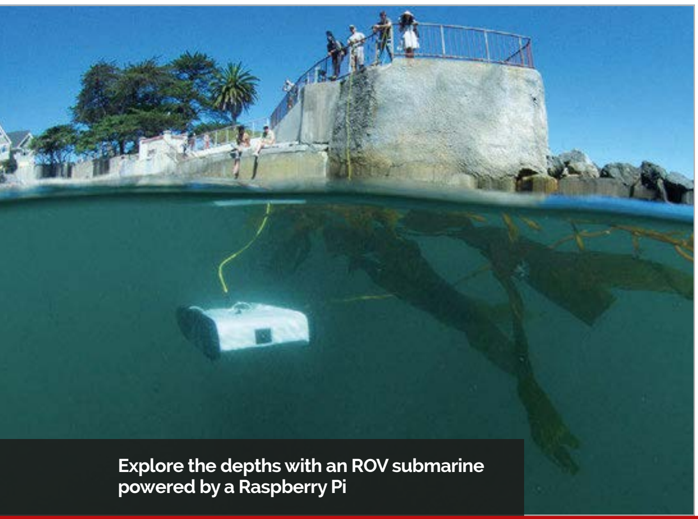
> ZERO4U 80

> PIPER 82

TRIDENT ANNOUNCES IT'S POWERED BY PI

The crowdsourced remote submarine project reveals that it's controlled by a Raspberry Pi 3

The Raspberry Pi has been used on land for robots, in the air for quadcopters, on the sea in racing yachts and drone boats, and there's even a couple in space right now as well. The Raspberry Pi is conquering everywhere but under the waves, in the murky depths. Now, this is all set to change.



Explore the depths with an ROV submarine powered by a Raspberry Pi

"Trident is a high-performance micro ROV (remotely operated vehicle) submarine that sends live video to a pilot at the surface who can control it," explains Eric Stackpole, co-founder of OpenROV. The Berkeley-based startup makes open-source ROV submarines, and Trident is its latest project. "Its hydrodynamic design makes it optimal for penetrating through current as well as searching large unknown areas, and it's built to be extremely portable and rugged so it can be used in places that have never been explored before. Trident will be the first ROV submarine that is affordable for the majority of consumers, and the aim of the project is to give people the ability to participate in a field they may not otherwise have access to."

The Trident project was Kickstarted in 2015, with an aim to be released at the end of 2016. After asking for $50,000, the Kickstarter went up and past $800,000 to ensure it was fully funded. While this may have been in November last year, it was only recently that the team revealed that a Raspberry Pi 3 was the computer brain living inside the Trident. The Raspberry Pi 3 came out after the Kickstarter was funded, though, so why has it been used?

"One of the main things we like about Raspberry Pi is its community," Eric tells us. "We've been amazed by how many incredible devices have been built using Raspberry Pi, and we certainly hope Raspberry Pi developers will come up with creative projects on the OpenROV platform. The Raspberry Pi 3 also offered the best computational performance, features, and form factor in our price range. We're using the built-in WiFi on the Pi to communicate with external

EXPLORING SUNKEN SHIPS

The SS Tahoe sank in Lake Tahoe in 1940. What can a mini submarine discover about it?

The ghostly wreck of an ancient steamship lies 120 metres down, on the bed of Lake Tahoe

An intact porthole reflects the lights of the Trident as it creeps around this watery grave

The prototype of the Trident is not much different from the finished design, with sleek curves that allow it to fly through the water

payloads that can be mounted on Trident, and as time goes on we hope to develop software that uses advanced control systems, computer vision, and perhaps even machine learning to give Trident additional capabilities. We've even been talking about ways to use software for autonomous operations."

The machine certainly looks impressive. Videos show it cuts neatly and quickly through the water, making sharp turns on a whim (see more here: **magpi.cc/2baXRAf**). It connects to the surface via a tether for better communication, using a wireless buoy to then connect to the controls. The tether can have a range of up to 300 metres, although the Trident shouldn't be going any deeper than 100 metres anyway.

"Trident is only one step in our journey to popularise telerobotics as a tool for exploration," says Eric. "As we develop technology which will allow people to control vehicles through the internet and share data with thousands of others, we hope that we can not only change how many discoveries about our planet are being made each day, but also who the people making those discoveries are."

Find out more at **openrov.com**.

> "One of the main things we like about Raspberry Pi is its community"

The bare-bones motherboard connects to the Raspberry Pi 3 at the front, which in turn controls much of the craft

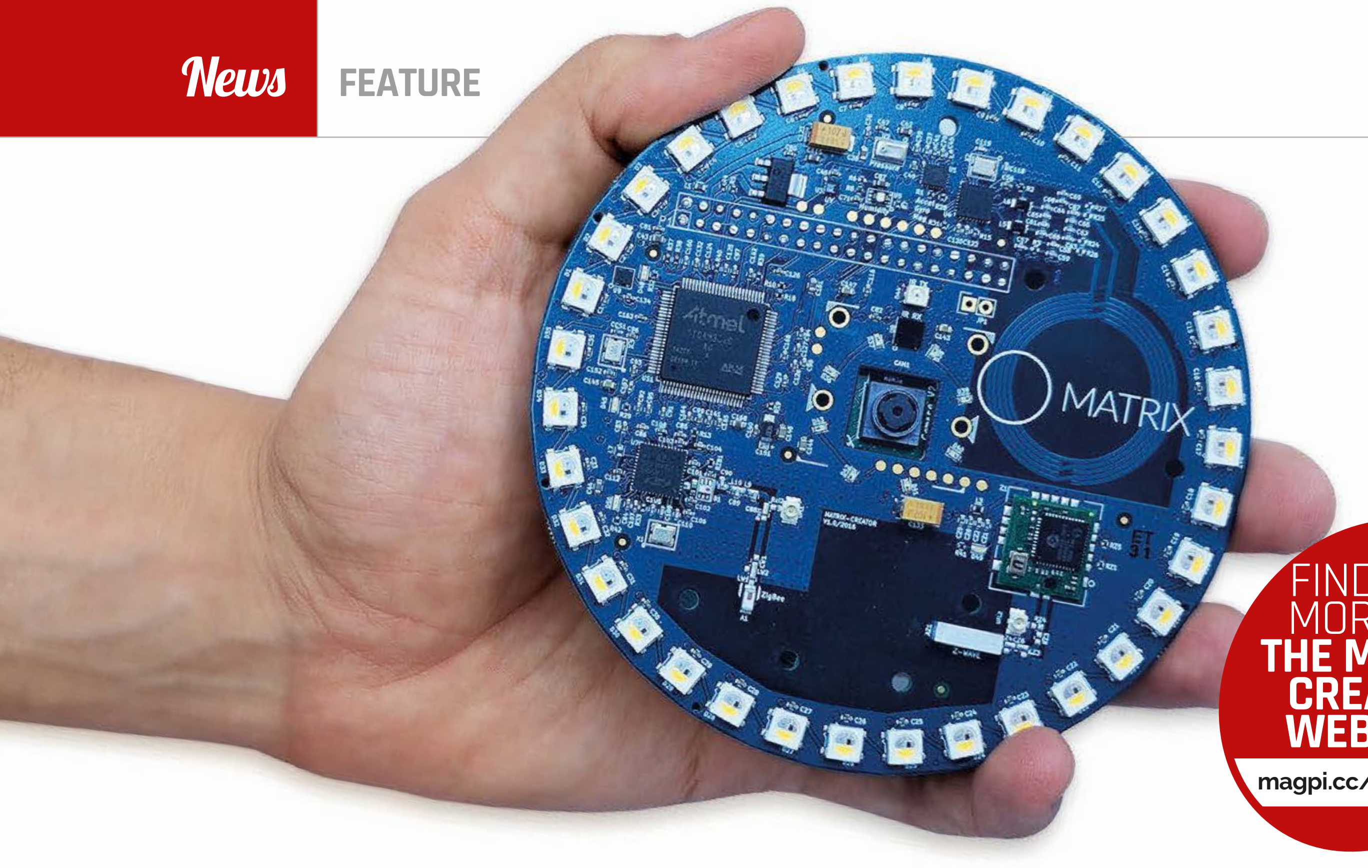
MAKING AN OPEN-SOURCE ROV

"Since the beginning, a major cornerstone of our company has been building community," says Eric Stackpole. "Although Trident and most of our other hardware designs have chiefly been developed internally, we've found that sharing our work tends to create a stronger, more capable user base, and we hope it will also fuel more innovation in the field of low-cost underwater exploration. On the software side of things, there are a ton of open-source libraries and programs which we've used or modified, and we hope that our software will be useful to the community as well. We've put literally thousands of hours into developing our software and we want that to be a useful tool for others." You can see the team's open-source work on OpenROV's GitHub page: **magpi.cc/2blGl9f**.

An entrance into the SS Tahoe itself – watch you don't get the tether stuck, though

What's beyond this long-unopened door in the dark depths?

Someone needs to clean their bathroom; no wonder the ship is abandoned



FIND OUT MORE ON **THE MATRIX CREATOR WEBSITE:**
magpi.cc/2bisuQV

CREATE ALMOST ANYTHING WITH MATRIX CREATOR

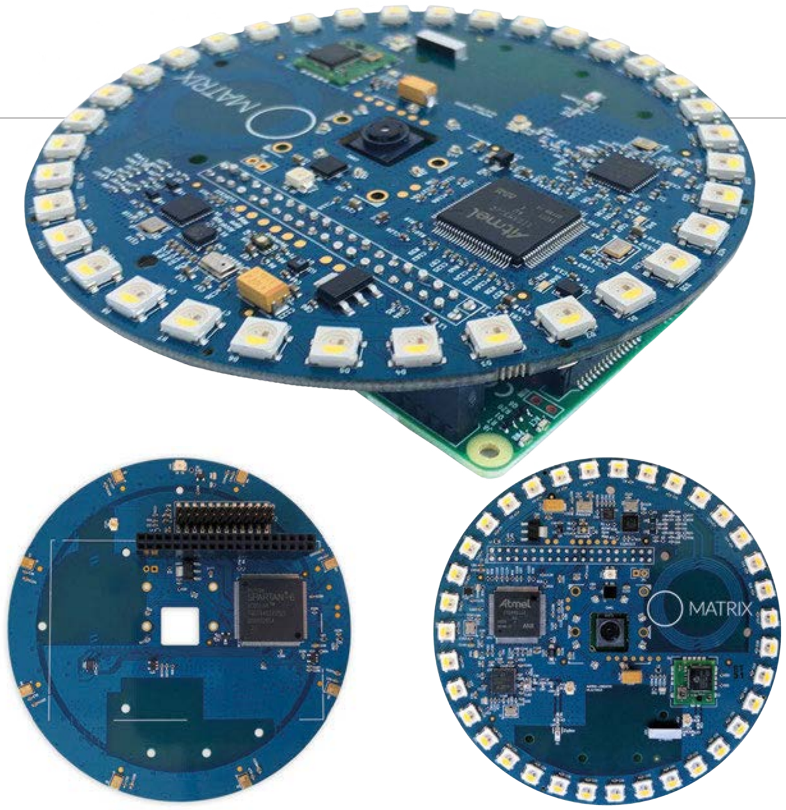
The all-inclusive hardware add-on for the Raspberry Pi hopes to open up the world to the tiny computer

COMPONENTS

- 8 MEMS microphone array (DIY Amazon Echo)
- FPGA (Xilinx Spartan 6)
- Microcontroller (ARM Cortex M3)
- Temperature sensor
- Ultraviolet sensor
- Pressure sensor
- 3D accelerometer
- 3D gyroscope
- 3D magnetometer
- Humidity sensor
- LEDs

There are plenty of add-on boards and HATs for the Raspberry Pi that add functionality to the computer, from simple things like LEDs or motor controllers, to sensor suites and enterprise security. The MATRIX Creator is something a little different, though, adding a huge number of functions in a bid to open up development with the Raspberry Pi to IoT and beyond.

"The MATRIX Creator is an all-inclusive hardware device that connects to the GPIO pins on the Raspberry Pi and provides a means for any developer around the world to start making machine learning, computer vision, and Internet of Things applications within minutes," explains Rodolfo Saccoman, CEO of AdMobilize, the company behind the MATRIX. "We see it as being a building block for the democratisation of IoT and AI. Just as how the iPhone created an all-in-one hardware device that allowed developers to unleash their creativity and build amazing apps, which in turn revolutionised the world, we see the MATRIX Creator doing the same thing for a multitude of industries such as smart homes, intelligent buildings, robotics, security, industrial control, smart retail, drones, custom maker projects, and many more."



Above The MATRIX has many functions, including a multitude of sensors and microphones, as well as a ring of LEDs around the edge

IoT for all

Rodolfo believes the MATRIX is something anyone can use. With this in mind, the team have created an operating system based on Raspbian, MATRIX OS, that allows people to start programming the MATRIX straight away in JavaScript. "MATRIX Creator is great as both a teaching and a development tool, aimed at anything from a middle school computer class to hobbyists and universities," Rodolfo tells us.

It's taken a year to get this far in terms of development, and Rodolfo is delighted with the response to the board so far:

"The reactions have been incredible, truly surpassing our greatest expectations! Since we launched it at the National Maker Faire in June, we have received positive feedback and excitement from the community. While at the Faire, we won the National Maker Faire Editor's Choice blue ribbon and have since been featured in magazines and websites. In addition, we completely sold out of our first batch of MATRIX Creators and since shipping them, we've had countless people on social media express their excitement about it and what they are going to use it for."

> "The reactions have been incredible, truly surpassing our greatest expectations"

Even though the MATRIX sold out in June, it's available to purchase from the website; due to a lot of work from the team, they've managed to get the board priced pretty low at $99 (£76). You can find out more about it here: **magpi.cc/2bisuQV**.

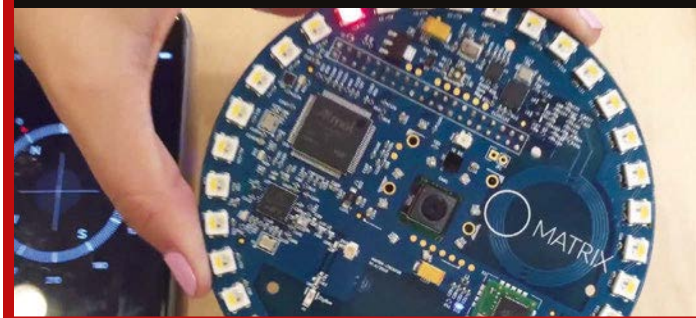
"Our small team is so humbled by the love we have received from the Raspberry Pi and maker community," Rodolfo concludes. "Our goal is to develop a powerful and user-friendly creation platform for people to create amazing, intelligent things... we think machine intelligence and Internet of Things can be for everyone!"

PROJECTS WITH MATRIX

Here's what people have already been making

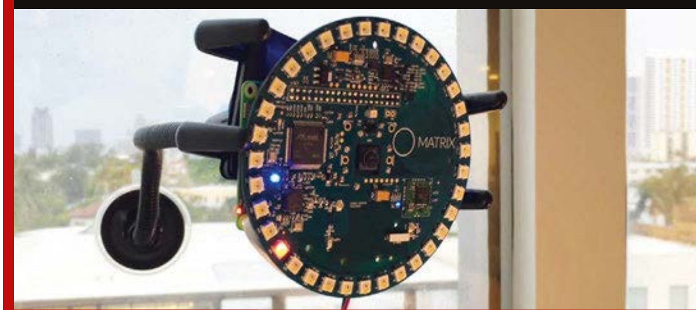
SELF-BALANCING ROBOT magpi.cc/2biUwMa

Balancing robots are those that work on just two wheels. This one runs about just fine, even with a higher centre of gravity. The accelerometers and gyroscopes in the MATRIX help make sure it stays balanced, thanks to a bit of extra code.



LED COMPASS magpi.cc/2biWfBa

This one is quite simple but very effective: using the magnetometer on the MATRIX board, the Raspberry Pi determines a compass direction. This is then displayed on the board via the LEDs around the edge. It updates in real time.



FACE TRACKING magpi.cc/2biXWP4

Similar to the compass, this project uses a connected camera (attached to the middle of the MATRIX) to track the user's face. It uses this data to locate where the face is in relation to the MATRIX, and lights an LED that's in that direction.

ETHERNET & USB BOOT

New update makes it possible to boot from USB drives and networks

It is now possible to boot a Raspberry Pi 3 from a USB storage device or directly from a network connection.

These new boot modes enable Pi owners to start up Raspberry Pi 3 devices with alternatives to the traditional SD card. The Pi can now be booted from an attached USB storage device, such as a hard drive, SSD drive or thumb drive. You can even boot a Raspberry Pi without any storage device attached, by loading the operating system from another computer on the same network.

The boot process

"There's a small boot ROM, which is an unchanging bit of code used to boot the device," explains Gordon Hollingworth, Raspberry Pi's director of engineering, in his blog on the Raspberry Pi site (**magpi.cc/2bdmnhY**). "It's the boot ROM that can read files from SD cards and execute them.

"When the Pi is powered up, or rebooted, it tries to talk to an attached SD card," he continues. "[It] looks for a file called bootcode.bin; if it finds it, then it loads it into memory and jumps to it. This piece of code then continues to load up the rest of the Pi system, such as the firmware and ARM kernel."

The potential to boot to the Raspberry Pi was included at the hardware level with the Raspberry Pi 3. "While squeezing in the quad A53 processors, I spent a fair amount of time writing some new boot modes," reveals Gordon. "Needless to say, it's not easy squeezing SD boot, eMMC boot, SPI boot, NAND flash, FAT file system, GUID and MBR partitions, USB device, USB host, Ethernet device, and mass storage device support into a mere 32kB."

He notes that this boot mode hasn't been enabled by default, as they first wanted to check that it worked as expected. The boot modes are enabled in one-time programmable (OTP) memory, so you need to enable the boot mode on your Raspberry Pi 3 first. This is done using a config.txt parameter.

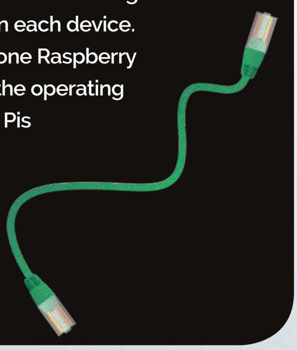
Unfortunately, the new boot options are only available in the Raspberry Pi 3 – you can't USB- or Ethernet-boot a Pi Zero or older models. "The boot code is stored in the BCM2837 device only," says Gordon, "so the Pi 1, Pi 2, and Pi Zero will all require SD cards."

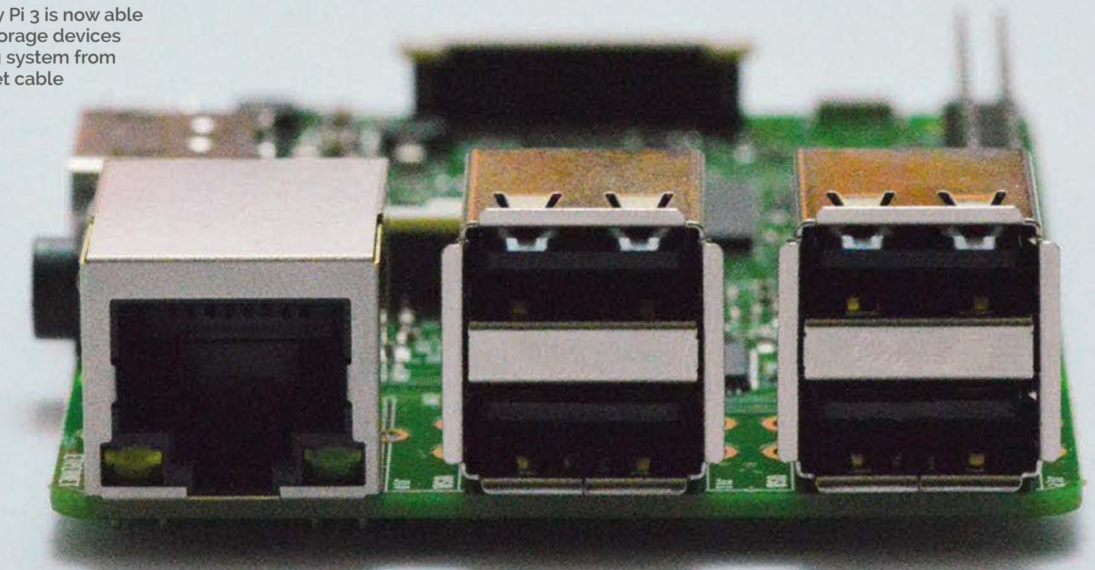
NETWORK BOOT

Network booting enables a computer to load all of its software over a network. This is useful in a number of cases, such as remotely operated systems or computers used in data centres. Network boot enables devices to be updated, upgraded, and completely re-imaged, without IT managers having to work manually on each device.

It's now possible to use one Raspberry Pi with an SD card to load the operating system to other Raspberry Pis on the network.

More information for network boot can be found in the Raspberry Pi documentation: **magpi.cc/2aYArsw**





Right The Raspberry Pi 3 is now able to boot from USB storage devices or load an operating system from an attached Ethernet cable

Mass effect

There are lots of advantages to using a mass storage device over an SD card. Hard drives, whether traditional spinning platter or flash storage, tend to be much larger than SD cards and storage space is much cheaper at high levels.

The new boot mode makes it far easier for users to build projects that require large data storage without requiring both an SD card and hard drive. Flash thumb drives can be easily recycled into effective hard drives. Many users will find it easy to locate spare flash drives.

There are some issues with using a USB mass storage device. "Some flash drives power up too slowly," says Gordon, and "some flash drives have a very specific protocol requirement that we don't handle; as a result of this, we can't talk to these drives correctly. An example of such a drive would be the Kingston DataTraveler 100 G3 32GB."

But thanks to the sterling efforts of Raspberry Pi's work-experience student Henry Budden, a list of working SSDs has been made: SanDisk Cruzer Fit 16GB, SanDisk Cruzer Blade 16GB, Samsung 32GB USB 3.0 drive, MeCo 16GB USB 3.0.

Going Ethernet

The benefits of network boot are less immediately obvious, but it's a more intriguing technical process.

"SD cards are difficult to make reliable unless they are treated well. They must be powered down correctly," explains Gordon. "A Network File System (NFS) is much better in this respect, and is easy to fix remotely." NFS file systems can be shared between multiple Pis, meaning that you only have to update and upgrade a single Pi.

"I would like to thank my Slack beta testing team who provided a great testing resource for this work," concludes Gordon. "It's been a fun few weeks."

MASS STORAGE BOOT

The USB specification allows for a mass storage class which many devices implement, from the humble flash drive to USB attached hard drives. This includes microSD readers, but generally it refers to anything you can plug into a computer's USB port and use for file storage.

You can find more information about the boot process for USB mass storage in the Raspberry Pi documentation: **magpi.cc/2bmLY9n**.

FORMULA PI

The new robot racing series that teaches you about robotics as you live out your F1 fantasies

“When I was young, watching motor racing with my father, I realised that most people were interested in who the best driver was, but I was interested in which car was the fastest.” So says Timothy Freeburn, who you may know as the director of PiBorg, the company that makes amazing Raspberry Pi robots and kits. “For me the competition was about the best engineering, not the best drivers.”

With this in mind, Timothy and the PiBorg team have launched a Kickstarter for a brand new autonomous racing league, one where engineering and software programming skills are the key to victory: Formula Pi.

With a track built above PiBorg’s office, teams from around the world can enter by simply submitting their code for the race season, so they can be installed onto the special YetiBorg racers that have been built for the competition.

“We set about designing a slower vehicle, one that could self-right if it got flipped over, and the YetiBorg was born,” explains Tim. “In order to keep the costs down, we use a Raspberry Pi Zero and camera on the YetiBorg, with our ZeroBorg motor controller. A 9V rechargeable battery powers both the motors and the Pi, and we’ve had running times on the track of 30 mins, so more than sufficient run time for a good race. The YetiBorg uses large wheels to give ground clearance on both sides; this makes for very entertaining ‘interactions’ between robots, as they very easily flip over when locking wheels with an opponent.”

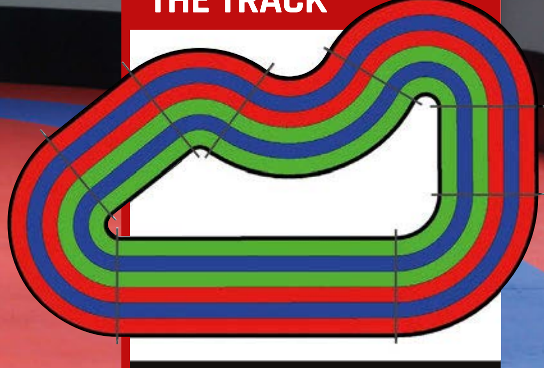
The Kickstarter will be over by the time you read this article, but fear not: if you want to enter and there are slots left, you’ll be able to do so on the Formula Pi website.

FIND OUT MORE ABOUT FORMULA PI

formulapi.com

Above The vehicle of choice for the contestants in Formula Pi is the YetiBorg

THE TRACK



A purpose-built, brightly coloured track atop the PiBorg offices will be used for the racing series. The colours and slower speed of the robots allow for the camera on the YetiBorg to do some image processing while navigating the course.

Like all good race tracks, it has a tricky chicane along with a nice long straight and other gentler corners. The handling of the course will truly test the code’s worth.

THE PI-POWERED LASER CUTTER

Theo Lasers is looking to make cutting and engraving with light more accessible, and has turned to the Raspberry Pi to drive its open-source hardware

Left Despite its low power requirement, the Theo Lasers cutter can easily create intricate cuts in a range of materials

"I gave my job up 18 months ago after volunteering in a makerspace," Grant Macaulay explained from his stand at the Maker Faire UK event, where he was showcasing prototypes of education-centric laser cutters created by his startup company Theo Lasers. "Artists were coming in and cutting A4 sheets of paper on big industrial laser cutters, and I thought, 'Right, I've been a maker, I've been making stuff under Creative Commons, I'm going to make a laser cutter with a laser cutter and give it all away with the plans, the grbl CNC tool, and an Android mobile phone app.'"

The result: affordable, low-power laser cutters housed in an eye-catching laser-cut wooden chassis. Since their unveiling in April, Grant's been hard at work with his team to improve upon his initial designs. "I wanted more from this laser cutter," he says, four months on from his Maker Faire UK interview. "I dreamt of the full potential possible and what I as a maker was crying out for! I spent weeks banging my head against a brick wall, trying to get grbl to do what I wanted it to. It wasn't working.

"I phoned up my friend Gavin, who is an absolute genius at programming, and asked him to come and have a look and give me some guidance. He was so enthusiastic; he turned to me and said, 'What do you want from me? That list will take me about eight hours!' Fantastic! Far more than I could have possibly ever hoped for; what a morale booster!"

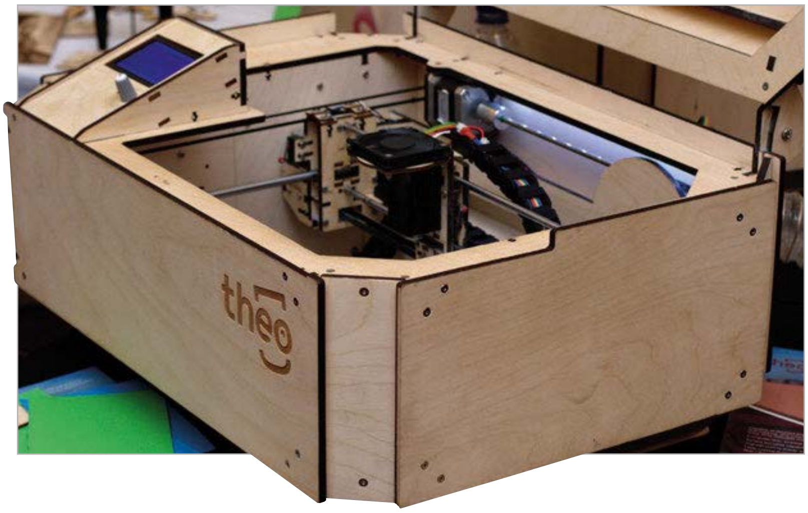
Outgrowing its home

Adding new features – including the ability to load files for cutting on an SD card, support for the RAMPS 1.4 CNC board from the RepRap project, a built-in display, Bluetooth connectivity, and even a temperature sensor for safety – soon meant the project was hitting the upper limits of the Arduino Mega on which it was based. Fortunately, there was a solution readily available: the Raspberry Pi.

WHY OPEN SOURCE?

"When I gave up my job, the first thing I did was build the Microslice from end to end to see if it was pivot or persevere. I then made the Ultimaker Original, because that's all the same sort of size and mechanics; they released that under a Creative Commons licence and they're still here," explains Grant of his company's beginnings. "I've learned from these guys, and I want to give something back to the community and makers. And that's where Theo is. I want to make a Kickstarter on this; I want to start the business, employ people, and get a premises in Glasgow where I live, but this is me giving back as well as launching my own business."

The attractive wooden housing - laser-cut, of course - hides sophisticated yet maker-friendly internals



"We turned to the Raspberry Pi, and this opened up so many exciting possibilities and options," Grant explains. "We immediately offloaded the peripheral devices – the SD card and display – and changed from an Android app and Bluetooth connectivity to browser-based software and WiFi."

The shift to a new control platform didn't harm the freedom of the underlying hardware, which Theo Lasers is making available under a permissive open-source licence. "All the way through, we have kept backwards compatibility for Theo connectivity. Theo is fully grbl, OctoPrint, and Theo Controller compliant, accessible via USB, SD card, or WiFi connectivity. Theo can also be run from batteries and topped up by solar power, making Theo fully computer- and grid-free!"

Theo Controller

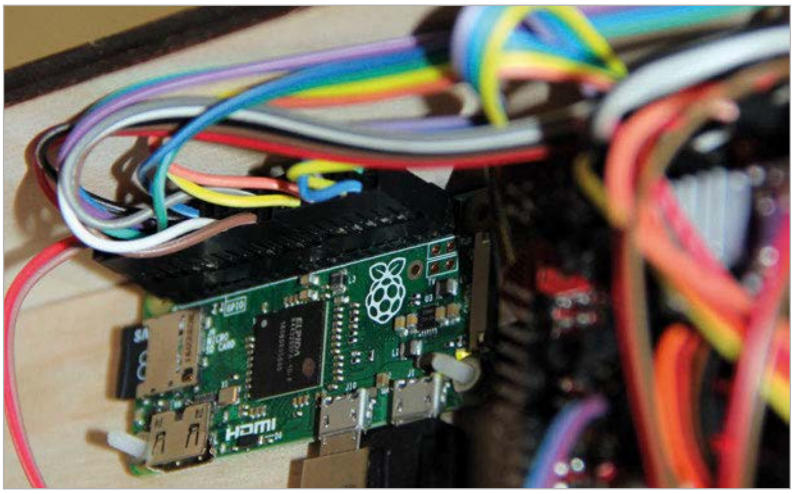
The shift from a microcontroller to a fully featured microcomputer opened up even more possibilities than Grant had first imagined. "We've now gone on to develop our own web service called Theo Controller," he says, "which also allows for full photo engraving using several picture formats, and the progress can be monitored and recorded using the Raspberry Pi Camera."

It's this which helps to truly set the Theo Lasers cutter apart from its competition. As well as the ability to be controlled from any device with a web browser, the integration of the Raspberry Pi Camera Module allows for live yet safe viewing of the cutting or engraving process. For devices which, even in their commercial incarnations, have been known to start small fires when the laser gets stuck, that's something that adds real value, not to mention allowing for safer use in education, where students can see the cutting process without risking their eyesight.

> "Theo Lasers is making the hardware available under an open-source licence"

At the time of writing, Grant and his team at Theo Lasers are planning to launch a Kickstarter campaign in early September for a trio of main models: a 1W entry-level engraver, plus 3W and 5W cutters. Regardless of the crowdfunding campaign's success, all software and hardware is to be released under a permissive open-source licence.

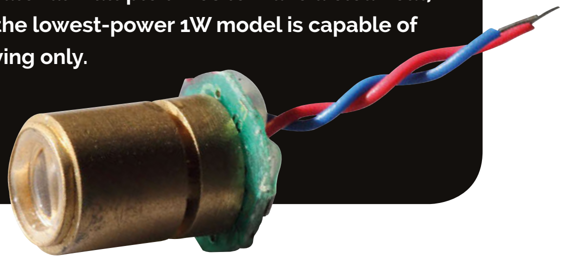
More information is available from **theolasers.com**.



Above A Raspberry Pi Zero provides the brains for the clever Theo Controller platform

DIODES VERSUS TUBES

To keep costs down, the Theo Laser cutters use laser diodes rather than the more common carbon dioxide (CO_2) filled tubes. Essentially the same technology as found in CD-ROM and Blu-ray drives, the laser diode runs at significantly lower power – 5W compared with 35W up to several hundred watts – and takes up less room than a tube-based laser. While this allows a Theo Laser to run entirely from battery power, it has an impact on efficiency: a Theo Laser may need to pass over material multiple times to make a clean cut, while the lowest-power 1W model is capable of engraving only.



GETTING STARTED WITH RASPBERRY PI

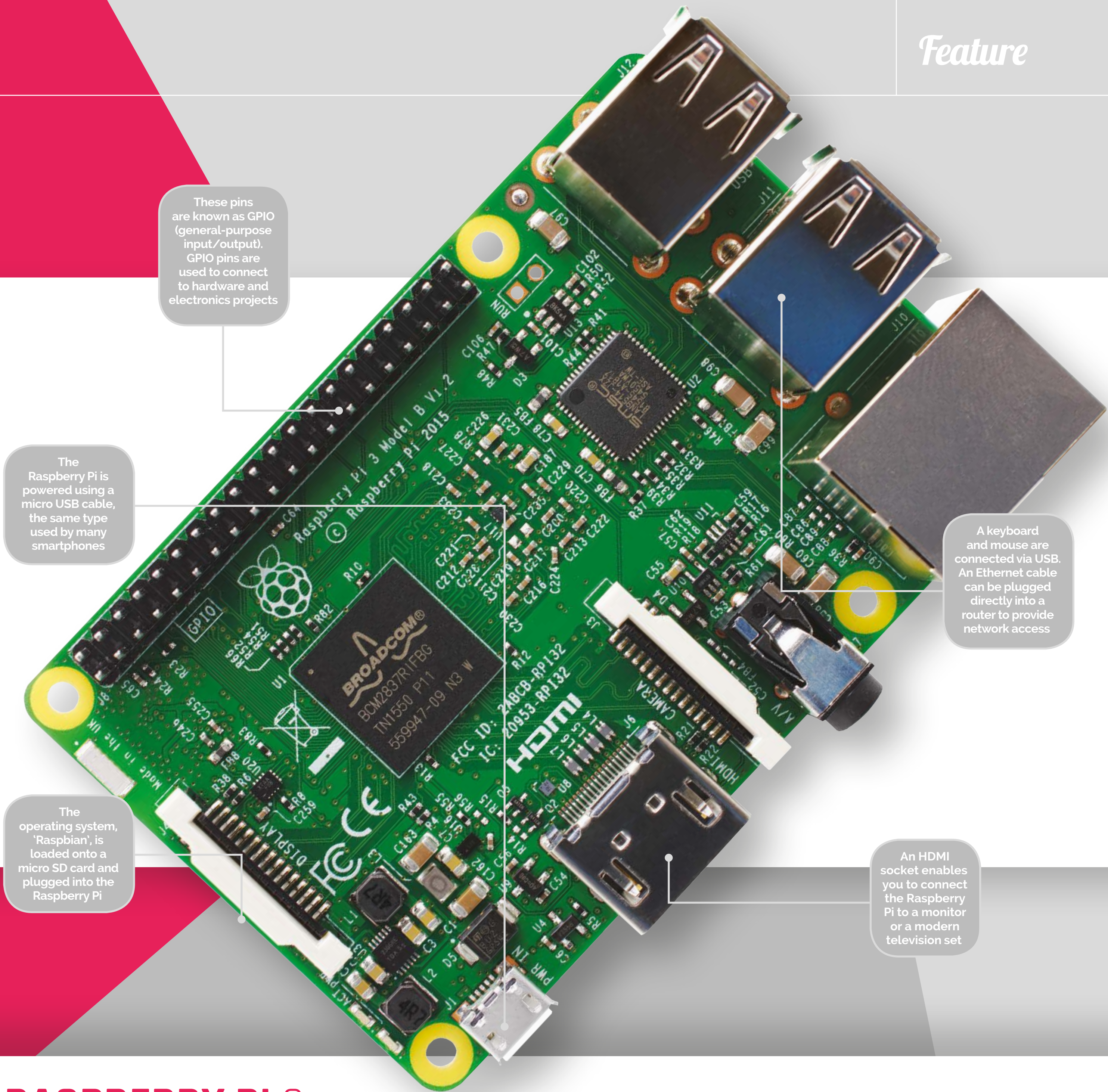
Creating amazing projects is easy with a Raspberry Pi, but first you need to plug it in and set up Raspbian, the default operating system. This guide will get you up and running in no time

The Raspberry Pi is a wonderful microcomputer that brims with potential. With a Raspberry Pi you can build robots, learn to code, and create all kinds of weird and wonderful projects.

Hackers and enthusiasts have turned Raspberry Pi boards into fully automated weather stations, internet-connected beehives, motorised skateboards, and much more. The only limit is your imagination.

But first, you need to start at the beginning. Upon picking up your Raspberry Pi for the first time, you're faced with a small green board of chips and sockets and may have no idea what to do with it. Before you can start building the project of your dreams, you'll need to get the basics sorted: keyboard, mouse, display, and operating system.

Creating projects with a Raspberry Pi is fun once you've mastered the basics. So in this guide, we're going to take you from newbie zero to Raspberry Pi hero. Grab your Raspberry Pi and let's get going.



RASPBERRY PI 3

The Raspberry Pi 3 is the latest model, and the version recommended for most newcomers

SD card

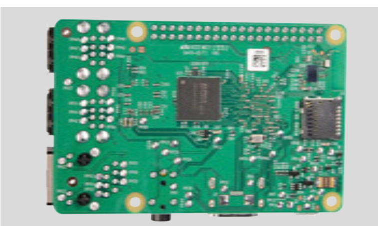
On the underside of the Raspberry Pi 3 board is the SD card slot. You preload the operating system onto a micro SD card and use it to boot up the Raspberry Pi.

Wireless network

The Pi 3 is the first Raspberry Pi to feature built-in wireless LAN and Bluetooth. This enables you to connect to a wireless router and get online without using a WiFi dongle.

1.2GHz ARM CPU

Featuring the latest 1.2GHz quad-core ARM CPU (central processing unit), the Raspberry Pi 3 is faster than many smartphones, and powerful enough to be used as a desktop computer.



RASPBERRY PI ZERO

Ultra-low-cost, super-tiny, and incredibly powerful, the Pi Zero is the tiniest Raspberry Pi computer

The Pi Zero is an ultra-low-cost and incredibly small microcomputer packed onto a single board. It's roughly a third the size of the Raspberry Pi 3, and has a teenie price tag ($5, or around £4).

For all that, the Pi Zero is packed with enough power to handle demanding computer projects.

Despite its diminutive stature, the Pi Zero is no toy. The Pi Zero is a fully fledged microcomputer with a 1GHz ARM CPU and 512MB RAM. It packs enough technology to run the full version of Raspbian, just the same as the Raspberry Pi 3.

The smaller board is more minimalist than other Raspberry Pi units, which makes it more challenging to set up. But it's a rewarding device that's ideal for creating Internet of Things, wearable, and embedded projects.

To keep the size down, the Pi Zero features a smaller-than-normal mini HDMI socket. You'll almost certainly need a mini HDMI-to-HDMI adapter or cable to connect the Raspberry Pi to a television or monitor.

Alternatively, hackers can hook up an RCA cable directly to the video headers on the board. RCA cables are the red, white, and yellow plugs that you find on older televisions. This feature makes the Pi Zero a great choice for retro gaming enthusiasts.

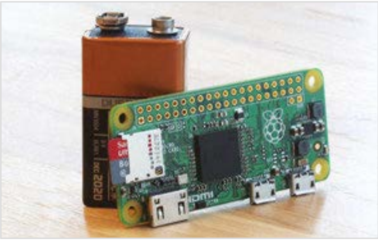
PI ZERO

Powerful processor

The Pi Zero packs a sizzling 1GHz single-core ARM 11 CPU with 512MB RAM. Despite its diminutive size, it's 40 percent faster than the original Raspberry Pi model.

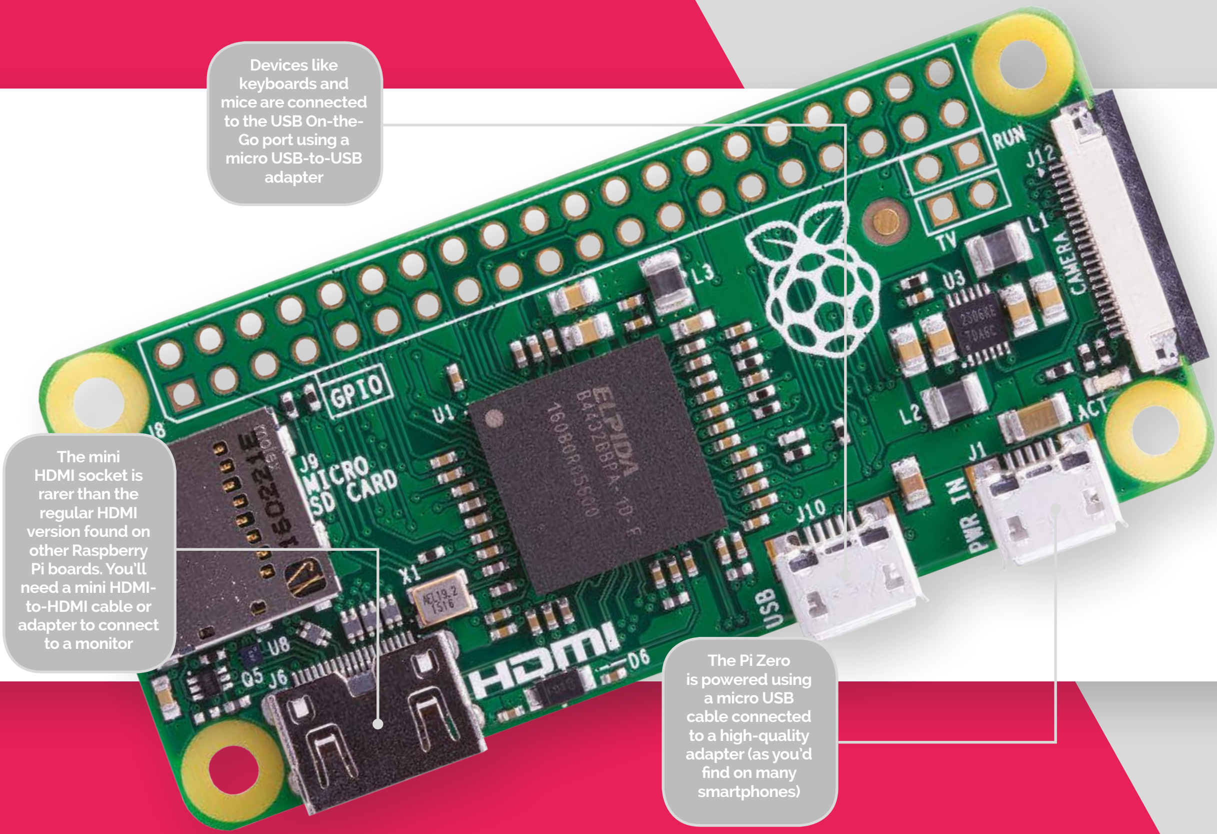
Tiny form factor

The Pi Zero offers a full computer experience, complete with the Raspbian operating system, and is only a third the size of the original Raspberry Pi.



GPIO to go

The full GPIO header sits along the side of the Pi Zero board. These holes enable makers to attach hardware to the Pi Zero, and you can experiment with electronics projects.



The Pi Zero board uses the same micro USB power input as other Raspberry Pi devices, and you can use an official adapter or salvage a high-quality power supply from a mobile phone (2A output is recommended).

Ports are minimal on the Pi Zero, and it sports a single USB port that's smaller than a regular one. You'll need a micro USB-to-USB adapter to connect your keyboard. You may also want a USB hub to connect a mouse and other devices like a USB camera.

A recent version update, Pi Zero v1.3, has a built-in camera connector. Like the other Raspberry Pi devices, you can connect a Raspberry Pi Camera Module or NoIR Camera Module directly to the Pi Zero. This enables you to turn the Pi Zero into a super low-cost camera for taking photos and recording videos.

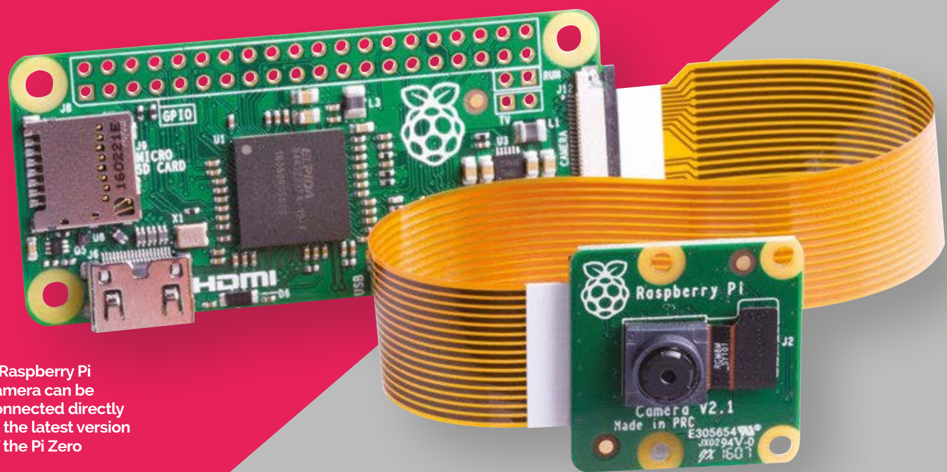
Thanks to the low power draw of the Pi Zero, this is ideal for time-lapse photography. You just set it up and let it get on with it.

Hooking a Pi Zero up to the internet requires either a USB-to-Ethernet adapter or, more commonly, a WiFi dongle.

Amazingly, the Pi Zero even has the full 40-pin GPIO header of the other Raspberry Pi models, but you don't get the pins pre-built onto the board. Instead, you need to solder two 20-pin male headers to the GPIO holes.

Setting up a Pi Zero is slightly more tricky than a Raspberry Pi 3, but it's also a lot of fun. The end result is a super-cheap, super-powerful computer that runs a full operating system.

> “Ports are minimal on the Pi Zero, and it sports a single USB port that's smaller than a regular one”



A Raspberry Pi camera can be connected directly to the latest version of the Pi Zero

EQUIPMENT YOU'LL NEED

All the kit you need to get a Raspberry Pi up and running for the first time

You don't require much to get your Raspberry Pi started: a micro SD card from an old camera, a smartphone charger, a recycled HDMI cable, and a keyboard and mouse are all you need.

Most items can be sourced from computer hardware around the house, or begged and borrowed from friends and family. If you're looking for the ultimate in low-cost computing; the Raspberry Pi is it.

You should be able to source, salvage, and scavenge most equipment you need to get a Raspberry Pi up and running. To get the most out of your Raspberry Pi in the long term, though, you should use high-quality components.

A good micro SD card from a named brand will be faster and more reliable. Not all USB power adapters are born equal, either. A reliable branded adapter will provide a steady stream of power, even when you attach multiple devices.

The Raspberry Pi board isn't shy, and it'll work just fine naked, but a good case keeps the board safer and makes it easier to store. There's a huge range of cases available, and many offer unique features such as waterproofing, stackability, or wall mounting.

The official Raspberry Pi case is a slick piece of kit that's perfect for any Pi user. Made of five parts that click together, it enables you to quickly open the case and access the board and GPIO pins.

Any equipment you can't recycle can be picked up from the Raspberry Pi Shop (**magpi.cc/2bnamFF**) or from distributors like Element14 (**element14.com**), Allied Electronics (**alliedelec.com**), and RS Components (**magpi.cc/2bnapBl**).

MICRO SD CARD

The micro SD card acts as the hard drive for your Raspberry Pi. You install the Raspbian operating system onto the card, then all your documents, files, and projects are saved to it as you work.

Raspberry PI fan Jeff Geerling did a community favour by purchasing over a dozen different micro SD cards and benchmarking each one. The results were pretty dramatic, with some cards running up to four times as fast as others. Samsung Evo+ and SanDisk Extreme are two popular brands worth looking out for, and both are fairly cheap. You can read more at **magpi.cc/2bncFs3**

HDMI cable

An HDMI cable is the easiest way to connect your Raspberry Pi to a computer monitor or television. You don't need an expensive one, and most people recycle one from an old games console or DVD player.

USB power

A good 2A or 2.5A power supply provides you with enough power to run a Raspberry Pi with all kinds of peripherals connected. You can buy an official Universal Power Supply (**magpi.cc/2a14pye**).

Keyboard

Any standard USB keyboard can be used to enter commands to your Raspberry Pi. You can use a Bluetooth keyboard with the Raspberry Pi 3, or any other Pi with a Bluetooth dongle attached. A wired keyboard is easier to use when setting up your Raspberry Pi.

Mouse

Any standard mouse will work with the Raspberry Pi, although ones with two buttons (non-Apple mice) work better. Like keyboards, a Bluetooth mouse will work once it's paired, but a wired mouse works as soon as you plug it in.

INSTALLING RASPBIAN

Discover how to use NOOBS to quickly set up the Raspbian operating system on your Raspberry Pi

Before you start using your Raspberry Pi, it needs to have an operating system (OS). This is the software used to start the hardware, and open and close programs.

Many computers use a specific operating system tied to the hardware. You'll probably be used to Windows on a PC and OS X on a Mac computer.

Most Raspberry Pi owners use an open-source operating system called Raspbian, which is based on Linux. The current version is based on a version of Linux called Debian Jessie, hence the name Raspbian (sometimes you'll hear it called 'Raspbian Jessie').

Linux is like Windows and Mac OS X, but more fun because it's open-source, so anybody can view the source code and improve it.

You can install a range of different OSes on a Raspberry Pi, some based on other versions of Linux, others based on Windows, and even completely unique environments like RISC OS.

Raspbian is the official OS and the one most beginners should start with. It's the simplest to install, easiest to use, and most projects and tutorials use Raspbian as their base.

Start with NOOBS

There are two approaches to installing Raspbian and other operating systems. Beginners should start with NOOBS (New Out Of Box Software). More advanced users may copy an image file containing a whole operating system directly to the SD card.

First, you must format your micro SD card to use the Windows FAT 32 format. The easiest way to do this on a Mac or Windows PC is to use a program called SD Card Formatter (**magpi.cc/2bncvkm**).

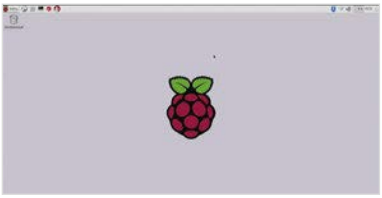
Connect your micro SD card to a Mac or Windows PC, typically using a micro SD-to-SD card adapter or a USB card reader, and use SD Card Formatter to erase the card.

Next, download the NOOBS ZIP file from **magpi.cc/2bnf5XF**. Extract the contents of the file and open the NOOBS folder. Copy the contents across to the root of the SD card. See the 'Setting up NOOBS' steps for more information.

AVAILABLE OSES

Raspbian

The official operating system is the easiest to use, and the one beginners should start with. It works a lot like other popular operating systems.

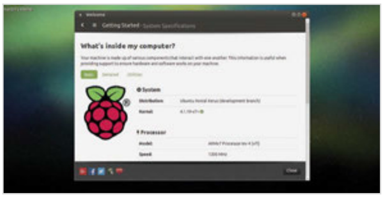


Windows 10 IoT Core

Not the full version of Windows, sadly, but Windows 10 IoT Core enables programmers to run Internet of Things and embedded projects.

Ubuntu MATE

Ubuntu is one of the world's most popular Linux operating systems, and Ubuntu MATE is a lightweight version that runs just fine on the Raspberry Pi.



NOOBS automates the process of installing Raspbian. Select the Raspbian option and click on Install to run it

NOOBS automatically copies all the files needed to run Raspbian onto your SD card

With the NOOBS files copied across, remove the micro SD card from your computer and slot it into your Raspberry Pi. Now connect the keyboard, mouse, and HDMI cable. Finally, attach the USB power to boot up the Raspberry Pi.

The Raspberry Pi will boot, displaying the NOOBS installer. By default it only has one option, 'Raspbian [RECOMMENDED]'. Place a tick next to Raspbian and click Install. Click Yes in the Confirm alert to begin installing Raspbian.

Now you just need to wait while the Raspbian file system is extracted. When it's finished, you'll see the Raspbian desktop and the message 'OS(es) Installed Successfully'. Click OK to start using your Raspberry Pi.

Installing image files

Installing an operating system from an image file is a slightly more complex procedure, but one that more advanced (and Pi Zero) users should learn. Image files are copied differently in Windows, compared to Linux and Mac computers.

In both systems, you format the micro SD card to FAT 32 as usual, then you download the operating system as an image file, a large file ending in '.img'. This file is then copied bit by bit as an exact replica to the micro SD card.

On a Windows PC, you will copy the image file using an app called Win32DiskImager (**magpi.cc/2bndEsr**). On Mac and Linux machines, most users copy the file using a command called 'dd' in the terminal.

Full instructions for copying image files for Windows, Mac, and Linux can be found on the Raspberry Pi website (**magpi.cc/1V5Oj8E**).

A good alternative for Mac owners is a program called Apple Pi Baker (**magpi.cc/2bcD53z**). This program enables you to pick the image file and the SD card, and then handles the copying automatically.

Learning how to copy image files is essential if you want to use operating systems other than Raspbian. Beginners should stick with NOOBS to install Raspbian to start with. It's much easier and is the best operating system for beginners.

OSMC

OSMC (Open Source Media Centre) is an easy way to transform your Raspberry Pi into a video and audio player.

RISC OS

RISC OS is an operating system originally designed by Acorn Computers for ARM-based systems. It's very light and completely different.

SETTING UP NOOBS

Download NOOBS

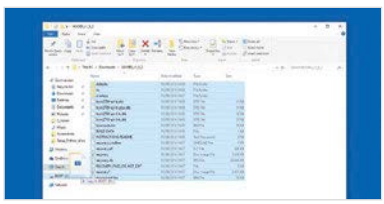
In a browser, visit **magpi.cc/2bnf5XF**. Click Download ZIP to get all the files. Open your downloads folder and locate the NOOBS file: currently it's 'NOOBS_v1_9_2'. Right-click on a Windows PC and choose Extract All, then Extract. Just double-click the file on a Mac to extract it.

Format SD card

Open SD Card Formatter and you'll see the card in the Drive letter. Change the Volume Label to BOOT so you can identify it later. Now click Option and change Format Type to Full (Erase). Ensure Format Size Adjustment is set to Off and click OK. Click Format, then OK. Click Exit to close SD Card Formatter when it's finished.

Copy NOOBS files

Open the freshly extracted folder so you can view all the files. It should have folders called **defaults**, **os**, and **overlays**, and files including **bootcode.bin** and **recovery**. Select all of the files and drag them onto the BOOT icon in the sidebar. This copies all of the files inside the NOOBS folder to the root of the SD card. It's important to copy the files inside NOOBS, and not the NOOBS folder itself.



USING RASPBIAN

Getting to grips with the Raspberry Pi's official operating system

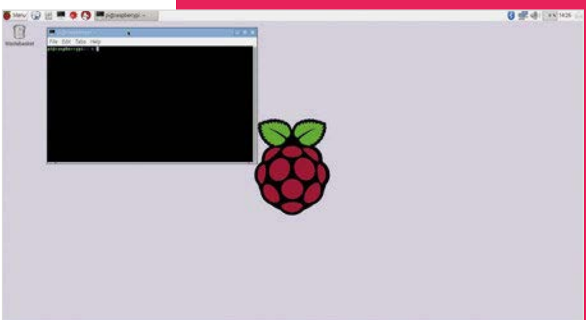
A Raspberry Pi can run many operating systems (OSes), but Raspbian is the official OS and the one most newcomers will start with.

Raspbian is a Linux operating system based on the popular Debian distribution. Fully customised for the Raspberry Pi hardware, it's usually a trouble-free experience using a Raspberry Pi with Raspbian.

One aspect of Linux that will be new to Windows and Mac users is being able to choose from different graphical interfaces. Raspbian includes one called LXDE, which stands for 'Lightweight X11 Desktop Environment'.

This heavily modified version of LXDE enables you to use a Raspberry Pi as you would another computer. You have a Menu button, which offers access to most of the programs and apps installed. Programs open in windows, which you can switch between, minimise, maximise, and close using buttons.

Many users might be wondering why this is anything special. Well, computers didn't always have windows; instead, most users used a command-line interface and entered text commands to start programs.



You'll learn how to use the terminal and control your Raspberry Pi computer using text commands

Terminal velocity

In Raspbian, you'll probably spend some time working under the hood of the desktop in a command-line environment. Next to the Menu button is the terminal, a program that enables you to enter Linux text commands. Learning how Linux works, and how to create programs that run from the command line, is part of the joy of owning a Raspberry Pi. It's a return to classic computing where you need to learn how things actually work.

Raspbian is a great environment for learning to code. Along with easy access to the command line, you get all kinds of programming environments built in: everything from MIT's Scratch to Python and Java. You even get a full working version of Mathematica, a cool maths environment that normally costs £190 to buy, with access to real-world data.

It's possible to buy SD cards pre-formatted with the Raspbian software. This saves you from having to install the operating system

Office worker

It isn't just about programming, though. You can use your Raspberry Pi as a desktop computer, and the operating system comes with LibreOffice built in. This is a full office suite of programs, similar to Microsoft Office. Its programs include Writer (word processing), Calc (spreadsheets),

Impress (presentations), Draw (vector graphics and flowcharts), Base (databases), and Math (formula editing).

Raspbian connects to the internet, and has a built-in web browser called 'Epiphany'. You also get an email client called 'Claws Mail'. Both can be accessed under Menu > Internet.

The Raspberry Pi connects to the internet using Ethernet (a cable that runs from your Raspberry Pi to a modem/router) or WiFi. It's easy to connect to a WiFi network, and we'll look at setting up both WiFi and Bluetooth next.

Settings and software

You can adjust the settings for your Raspberry Pi in two ways: using the desktop interface or a terminal program called Raspi Config.

Choose Menu > Preferences to find a collection of different system settings. Add / Remove Software can be used to find and remove packages from the Raspbian system.

Appearance Settings, Audio Device Settings, Main Menu Editor and Mouse & Keyboard Settings all adjust appearance and interaction with Raspbian. Most of the options are self-explanatory.

The Raspberry Pi Configuration choice provides more in-depth options. Here you can change your password (**raspberry** by default) and the hostname of the Pi on the network (**raspberrypi** by default). You can choose to boot to the desktop or the command-line interface (CLI), and enable and disable various hardware interface options.

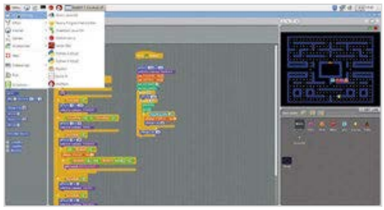
Raspi Config offers even more detailed options. Open a terminal window and then enter `sudo raspi-config`. A blue screen with options in a grey box appears. Use the up and down arrow keys to move between options; press the right and left arrow keys to move into an option (and back to the main menu). More information on these options can be found at **magpi.cc/2bnfuJF**.

The important thing about Raspbian is not to worry about experimenting with different options and settings. Feel free to explore the menus, command line, and configuration settings. You can always reset your micro SD card with NOOBS and start again.

USING THE RASPBIAN INTERFACE

Programming tools

Raspbian comes with a selection of coding tools, found under Menu > Programming. Scratch makes it easy to learn programming concepts, and popular languages like Python and Java are ready to use right out of the box.



Web software

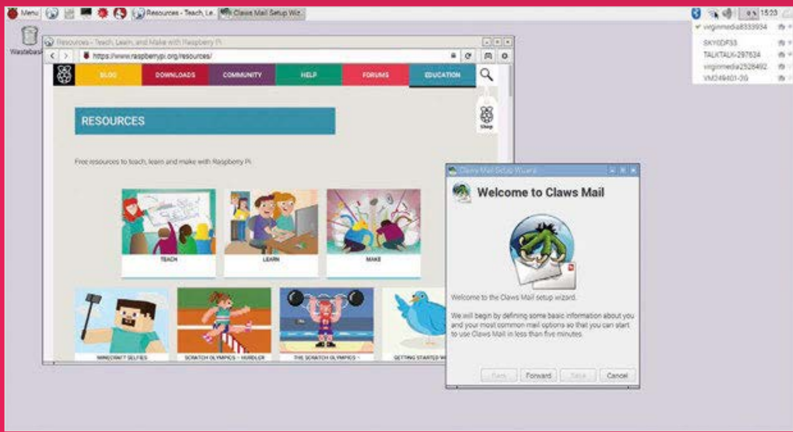
A web browser called Epiphany is built into Raspbian, along with an email program called Claws Mail. There are links to Raspberry Pi Resources and *The MagPi* under Menu > Internet.

Office suite

Raspbian features powerful LibreOffice programs like Writer and Impress. These are the equivalent of Microsoft Office apps and enable you to create documents on your Raspberry PI.

SETTING UP THE INTERNET

Get online wirelessly and quickly, with this guide to setting up wireless LAN on your Raspberry Pi


A wireless internet connection enables you to get help online and set up apps like Claws Mail

The Raspberry Pi is best when connected to the internet. You can use it to browse the web, play online videos, and send and receive emails. More importantly, you can get the latest updates and install the software packages you need for any project.

To do this, you'll need to get online. With the Raspberry Pi 3 this is easier than ever, because it now has a wireless antenna built into the board.

Other models of Raspberry Pi, including the Pi Zero, require a WiFi dongle connected to a spare USB port.

With wireless added to your Raspberry Pi, it's easy to get online. Boot into the Raspbian desktop and look for the WiFi Networks icon in the Panel (on the top-right of the display).

Click WiFi Networks and you'll see a list of all the local wireless networks. Choose your network and (if you have one) enter your password, also called the 'Pre Shared Key'. The Raspberry Pi connects to the wireless network, enabling you to get online. In this respect it's pretty much like any other computer that connects to WiFi; it will even remember the password for next time.

Once you're online, you can use the Epiphany browser to fetch webpages. Click Web Browser in the Launch Bar.

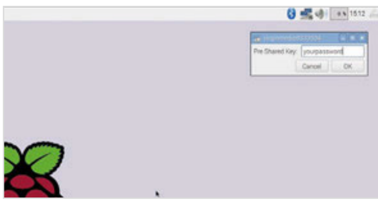
CONNECTING TO A WIRELESS NETWORK

Check for networks

Click on the Wireless Networks icon in the Panel. Raspbian will display a list of all the wireless networks available in your local area. Click on the one that's yours.

Enter your password

Enter your WiFi password in the Pre Shared Key field and click on OK. The network symbol will switch to a wireless symbol and you'll be connected.

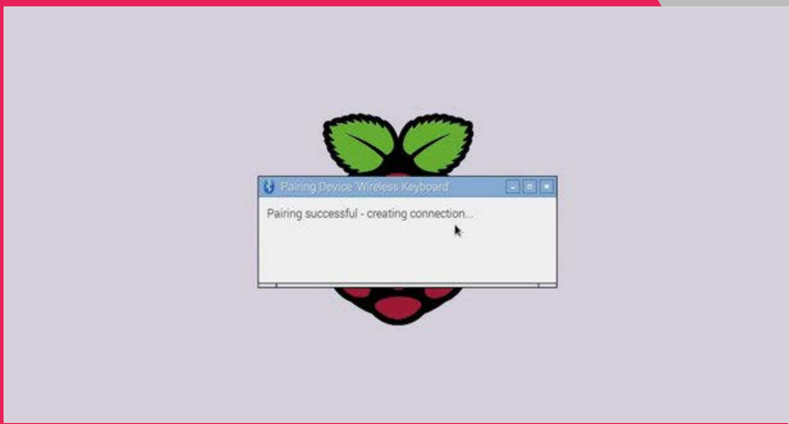


Test your connection

Test your internet connection by opening a webpage. Click on Web Browser in the Launch Bar and enter **www.raspberrypi.org** in the URL field. Press **RETURN** to load the page.

SETTING UP BLUETOOTH

Connect wirelessly to nearby devices with Bluetooth technology



Devices connected by Bluetooth work wirelessly with your Raspberry Pi

Bluetooth is another piece of technology that has been added to the Raspberry Pi 3 board. With Bluetooth you can connect wireless devices, such as mice and keyboards, directly to your Raspberry Pi.

As with wireless LAN, if you own an older Raspberry Pi model or a Pi Zero, you'll need to attach a USB dongle to use Bluetooth devices.

With Bluetooth hardware on your Raspberry Pi board, it's easy to connect to a device wirelessly, a process known as 'pairing'.

You can pair wireless gaming controllers, like a PlayStation joypad, or Android smartphones. Many Raspberry Pi projects make use of Bluetooth, enabling the Raspberry Pi to communicate with nearby electronic components and devices.

The easiest way to test out Bluetooth is to set up a wireless mouse or keyboard; both are fairly easy devices to come by.

In some ways, the process is similar to connecting to a WiFi network, but the Bluetooth device you want to connect to must be set to pairing mode first. This is also known as making the device 'discoverable'. Putting a device into pairing mode varies by device; holding down the power button until an LED flashes is fairly commonplace, but check with the instructions for your device.

You then use the Bluetooth icon in the Raspbian desktop Panel to connect to the device: choose Bluetooth > Add Device.

It's possible to put your Raspberry Pi into pairing mode by choosing **Bluetooth > Make Discoverable** from the Panel. Then you can connect to your Raspberry Pi from other Bluetooth devices like mobile phones.

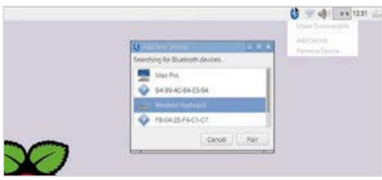
SETTING UP A BLUETOOTH DEVICE

Pairing mode

Start by putting your Bluetooth device in Pairing / Discoverable mode. We're using an Apple wireless keyboard. Hold down the power button until the LED flashes. Click Bluetooth in the Panel and choose Add Device.

Add new device

The Add New Device window opens and will scan for nearby Bluetooth devices. Some will have names, others just identifying numbers (check on the device). Choose a device from the list and click Pair.



Enter code

The Pi now attempts to pair with the Bluetooth device. You'll be asked to enter a code on the keyboard; press the buttons and **RETURN**. You can now start using the Bluetooth device with your Raspberry Pi.

GETTING TO KNOW GPIO

Discover the joy of electronics by hooking up components, wires, and hardware to the pins on a Raspberry Pi board

One of the most powerful and fun features of the Raspberry Pi is the row of pins at the top. Known as 'GPIO' (General-Purpose Input/Output), these pins enable you to hook up the Raspberry Pi to additional hardware and electronics.

There are lots of hardware attachments for the Raspberry Pi that connect directly to the GPIO pins. Many are known as HATs (Hardware Attached on Top). These connect directly to the GPIO and sit on top of the Raspberry Pi. More importantly, HATs are designed to work as soon as you connect them to the Raspberry Pi, so hardware branded as a HAT is easier to set up.

The real joy of GPIO isn't using pre-made hardware, but building your own electronics projects. You can connect the GPIO pins to all kinds of electronic circuitry and control it using the Raspberry Pi. With the right cables, you can hook the GPIO pins up to switches, buttons, sensors, buzzers, and all manner of electronic gizmos and widgets. These are used to learn all about electronics hardware and circuit building.

While it's possible to wire parts directly to the GPIO pins, most tinkerers place electronic components in a breadboard and connect this to the Raspberry Pi.

> "You can connect the GPIO pins to all kinds of electronic circuitry and control it"

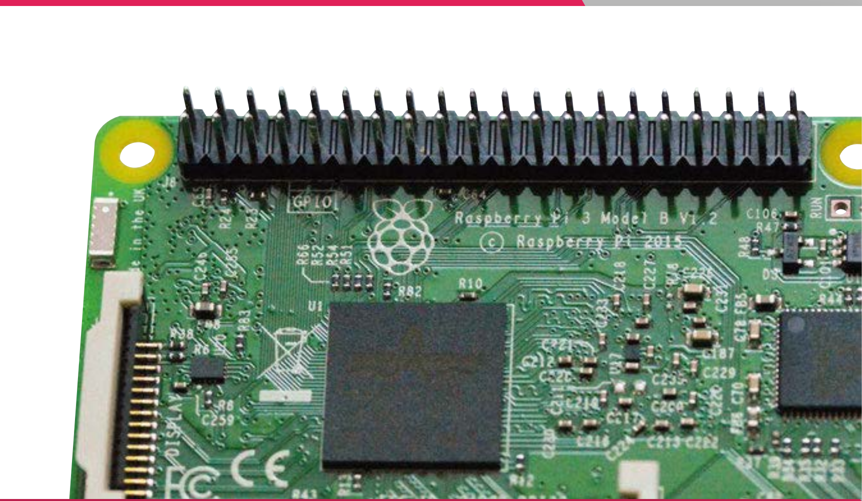
BREADBOARDS AND BREAKOUTS

Electronic components are plugged into the holes on the breadboard, and components in adjacent holes are linked. In this way, you can build up a test circuit without having to actually solder components together.

If you follow the instructions, connecting directly to the GPIO pins on a Raspberry Pi is safe, but randomly plugging in wires and power sources to the Raspberry Pi may cause bad things to happen, especially plugging in devices that use a lot of power (like motors).

Because of this, many electronics enthusiasts use a device known as a 'breakout cable' between the Raspberry Pi and breadboard. The breakout cable plugs into the GPIO pins, and into the breadboard.

There are also devices like the Explorer HAT that combine a breakout with a breadboard and enable you to create prototype circuits.



Unlike the type used for cutting bread, an electronic breadboard is a plastic slab with lots of holes in it.

Wiring a breadboard (or circuit) directly to the GPIO pins is generally safe, as long as you avoid circuits with external power sources. Most tinkerers invest in a breakout cable to go with the breadboard (see 'Breadboards and breakouts').

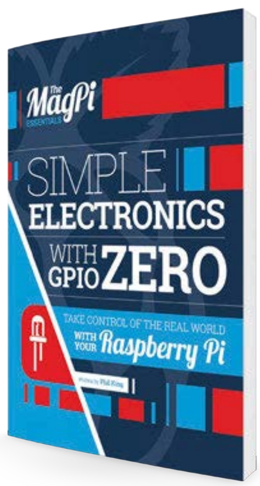
With your circuit set up, you then control the GPIO pins in a programming environment like Python or Scratch. GPIO pins are set to input or output mode. GPIO outputs are easy because the pin is switched on or off (known as HIGH or LOW in computing terms). When the GPIO pin is HIGH, voltage flows through the GPIO pin, lighting up an LED or buzzing a buzzer. Set the pin to LOW and the LED goes out, or the buzzer goes quiet.

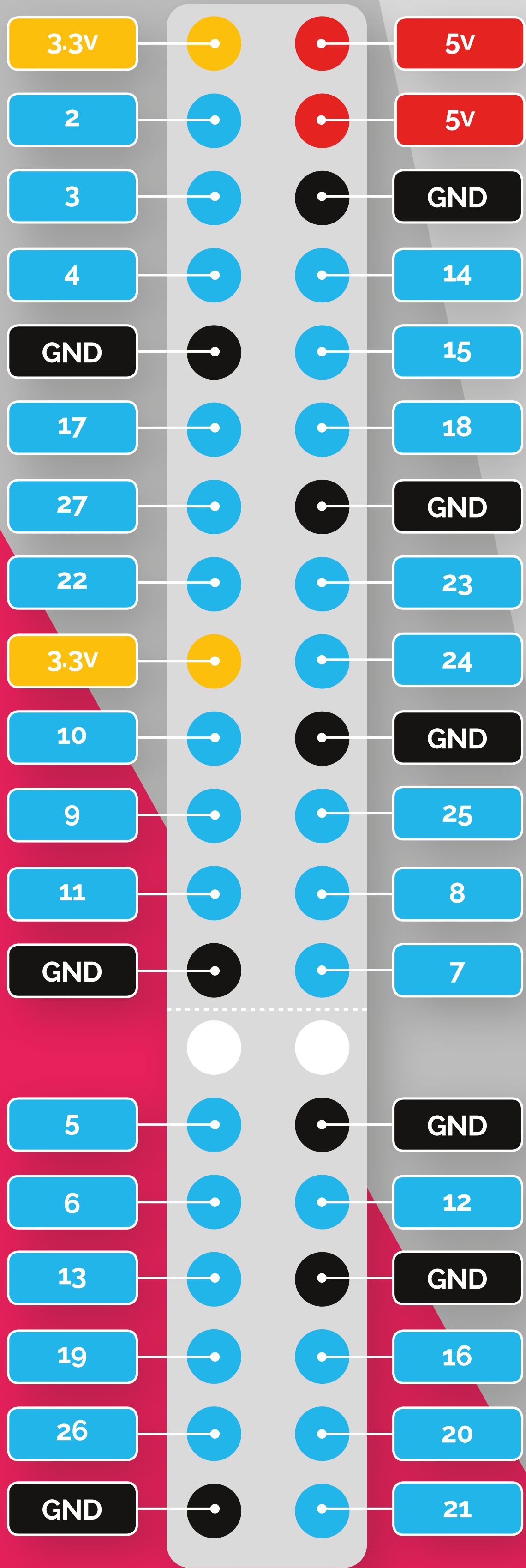
GPIO input is a bit more tricky. In this case, the GPIO pin is set to HIGH or LOW and responds to a change from a circuit. A button (or other electronic component) can change the circuit from LOW to HIGH, or HIGH to LOW, with the Raspberry Pi coded to respond accordingly. This is often referred to as 'pull up' or 'pull down'. Don't worry: if this all sounds complicated, you can get started by using GPIO Zero to make programming much easier.

Never underestimate the pure fun you can get from a little computer, a bunch of pins, and a handful of electronic components. Discovering how to use GPIO is a great way to spend your time.

GPIO ZERO ESSENTIALS

Learning to use the GPIO pins is the route to having real fun with a Raspberry Pi. It's a big subject, with lots of tricks and tinkering to discover. Our *GPIO Zero Essentials* book teaches you the basics (and beyond) of using the GPIO port with the GPIO Zero Python library. See **magpi.cc/GPIOZero-book** for more information.



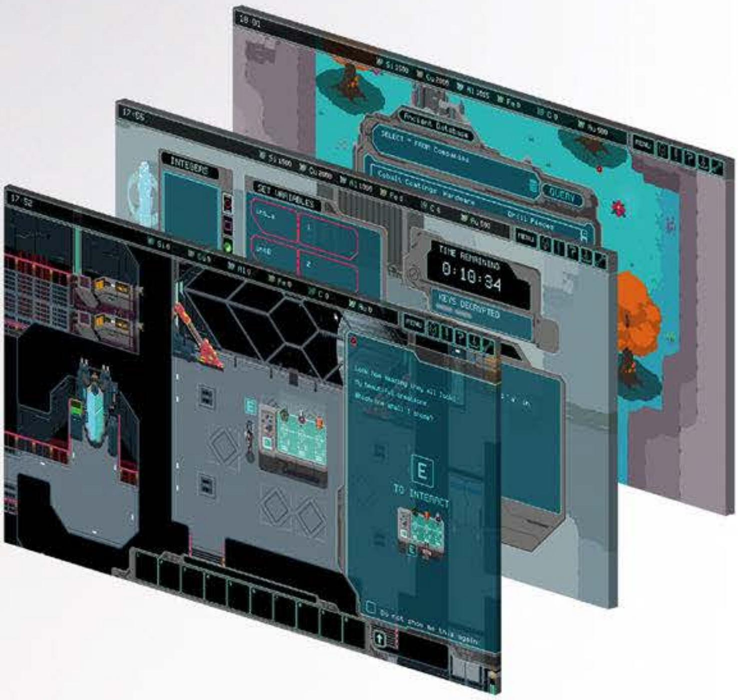


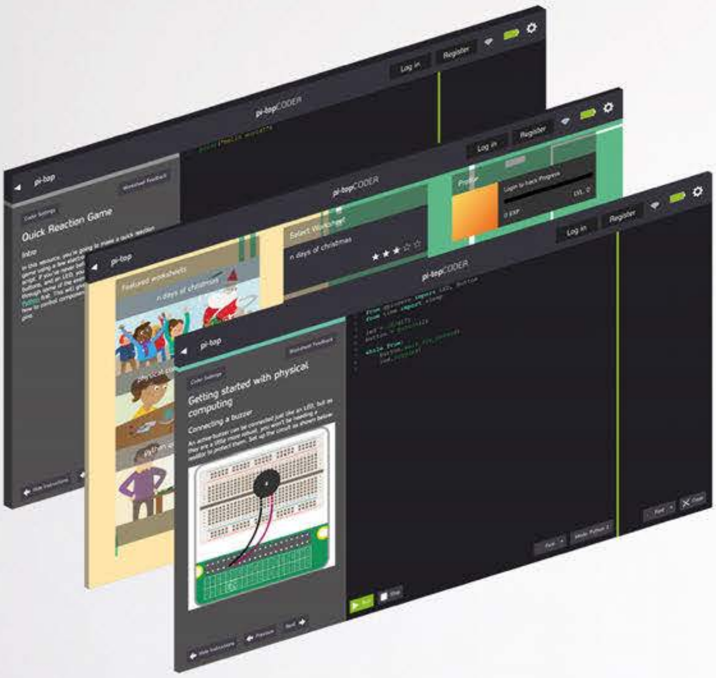
There are 40 GPIO pins, each with a specific function. Use this image as a handy guide whenever you're programming electronics

CEEDuniverse

CEED**universe** is a world of fantasy grounded in computing reality! After crash landing on a strange new planet you will first encounter 'drag and drop' coding puzzles that improve your computational thinking skills.

Discover more about the planet you've landed on and the civilisation that used to inhabit it while learning harder and more complex concepts in fun and engaging ways! Before long you'll be writing your own conditional statements, loops and functions. Find out more about CEED**universe** at www.pi-top.com





pi-topCODER

Influenced by the workflow of makers and hackers, pi-top presents **pi-top**CODER - an integrated code editor which allows you to learn, write and test code all in one view. With intuitive syntax highlighting, dynamic views and customizable themes it makes for a versatile learning tool for your projects.

pi-topCODER also has every Raspberry Pi Foundation lesson plan created and will track and save your progress as you go through dozens of fun hardware and software projects.

pi-topPROTO

pi-topPROTO is a HAT compatible Add-on Board for your **pi-top** or **pi-top**CEED that allows you to prototype electronics. Create a Weather Station, HAM Radio, Heart Rate Monitor, or integrate any Arduino based maker kits into your own Raspberry Pi compatible prototyping board!

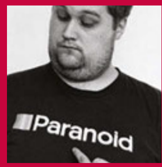
pi-topSPEAKER

Give a voice to your **pi-top** device with **pi-top**SPEAKER!

- Modular design, attach up to three in a row to give true stereo sound.
- 2W per module
- Left, Right and Mono mix selection
- High quality SPDIF digital audio from HDMI
- I^2C controlled

ZERO360

Take 360-degree panoramas with some clever Pi Camera Module placement and programming



JAMES MITCHELL

James is a software quality assurance engineer based in Berlin. He also organises the Raspberry Jam Berlin.
magpi.cc/2bgxXri

Quick Facts

- There are eight Pi Zeros and cameras
- The build took a few months
- It currently only sees 52 degrees of vertical space
- The Pi 3s actually power the Pi Zeros
- James has also taken pictures of the moon with a Pi camera

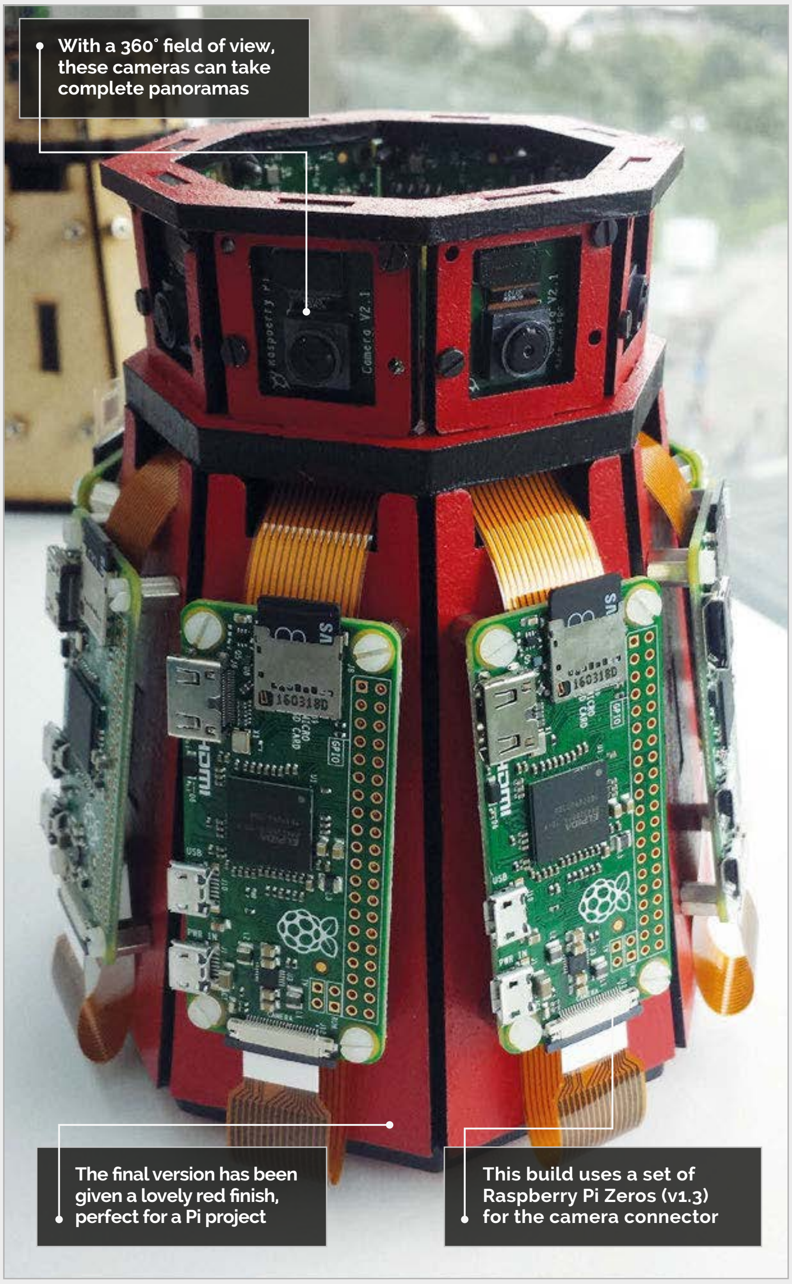
There's always some new visual technology trying to break into the mainstream, whether it's to try to improve the way we experience things or make a bit of money. The quality, however, varies wildly. At the moment, we're entering a new age of virtual reality (VR); this has created an interesting new set of visual experiences that has inspired James Mitchell.

"Recently, there has been a rush of 360-degree VR videos online," James tells us. "They're really impressive. Loving the technical side of photography and the Raspberry Pi, it seemed only logical that I would try and build something that would allow me to recreate those videos using the Raspberry Pi."

And so he did with the Zero360: a bank of Raspberry Pi Camera Modules arranged in a circle, connected to Pi Zeros. They can all take a photo at once; these are then stitched together to make a 360-degree panorama.

Why make it out of Pi Zeros, though? James explains that cost was a big factor:

"The issue is that the equipment for making 360-degree videos is extremely expensive. Using the Raspberry Pi, it's a fraction of the



With a 360° field of view, these cameras can take complete panoramas

The final version has been given a lovely red finish, perfect for a Pi project

This build uses a set of Raspberry Pi Zeros (v1.3) for the camera connector

The wood for the construction was laser-cut and was very easy to make, according to James

cost. You could argue that the Zero360 is not really that cheap when you could use your mobile phone or even a DSLR camera, but those would only take a single still image and need a user to move the camera around, whereas the Zero360 can take stills from all angles at the same time and repeatedly. Those stills can be made into a time-lapse. Also, video is an option! These features don't normally come that cheap!"

The housing for the system was quick to make, once James had managed to procure enough Raspberry Pi Zeros; however, the code took a few weeks on and off to get working. Two Raspberry Pi 3s are also used in the project to stitch the image together, and the build is otherwise just made up of Pi Zeros, Camera Modules, and power cables. "I'm using Raspbian Lite on all the Pis, with the raspistill and picamera Python libraries," James explains. "I also managed to stitch the images on the Pi 3 using Hugin."

Aside from some issues with getting the networking going, the whole project is pretty straightforward.

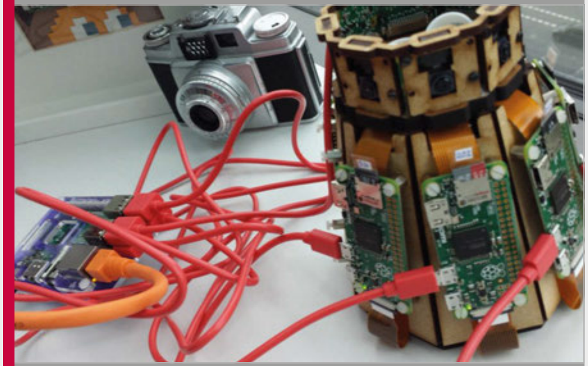
"Code-wise, there's still a lot of work to do, so I can't claim it's doing what it does efficiently," admits James. "But the final results are amazing! It's especially cool that the images are stitched together on the Pi itself!"

James has plenty of plans to improve the Zero360 in the future, so it can make even better panoramas.

MAKING A PANORAMA

>STEP-01
Relay the command

The setup has the Raspberry Pi 3s command the Pi Zeros to take their photos, rather than controlling them directly from a separate computer.



>STEP-02
Gather the photos

The photos from each individual Pi Zero are then sent over the network to one of the connected Pi 3s, rather than both of them.

>STEP-03
Stitch in time

Hugin is used on the Raspberry Pi 3 to stitch all the images together. The Pi 3 is chosen for this as it has a bit more power than the Pi Zeros.

RUSSELL GROKETT

Retired programmer/engineer Russell belongs to astronomy, amateur radio, Linux, and cloud computing clubs. In his spare time he travels worldwide, and is also an amateur caver and scuba diver.
grokett.org

EARTHQUAKE PI

This clever box of tricks rumbles to alert you to earthquakes anywhere in the world

Quick Facts

- Earthquake Pi took three days to build and program
- It took longer to write the documentation!
- The device checks for quakes every 15 minutes
- An electric toothbrush motor provides the rattle
- Earthquake sounds are played through a speaker

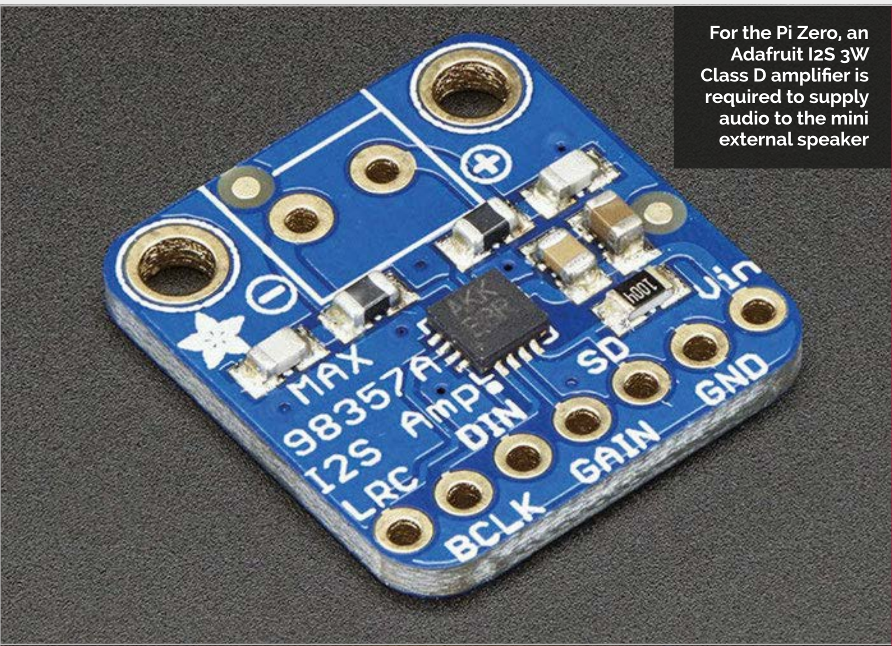
Russell Grokett has been fascinated by earthquakes and geology ever since he was a child, when his father built him a simple swinging beam seismograph. However, since Russell now lives in Florida, known for hurricanes but not quakes, he's created the Earthquake Pi (**magpi.cc/2aPNa62**) to satisfy his interest. Rather than acting as a detector of local tremors, like some Pi-powered projects, it's a neat alert system that uses real-time open data from the United States Geological Survey (USGS) to detect earthquakes around the globe.

"I had seen fancy maps and graphs of their data," explains Russell, "[but] I wanted to 'feel' (safely!) when an earthquake occurs. So I came up with the idea of taking their data and building a device that rattles and rumbles when an earthquake occurs. This is different from the typical detection on a chart or graph."

The Earthquake Pi comprises a wooden box containing the electrical components, including a Raspberry Pi Zero and a vibrating motor recycled from an old battery toothbrush to make the box rattle during an alert: "I found that just loosely taping the motor down worked best, as it bounces around a bit while running." To complete the effect, an external speaker plays earthquake sounds, while a strip of NeoPixels light up and an LCD display shows details of the seismic event.

"By default, the vibrating motor alerts run for a few seconds per magnitude: about two seconds for mag. 1 and up to about ten seconds for a mag. 9 (never heard that, luckily!)." The LCD display and NeoPixel bar graph then come on, displaying the quake location and magnitude. Lastly, the earthquake audio sound effect plays for a few seconds more. "You just set the box on your desk or table where is sits quietly… until boom! The first few times it goes off will probably scare you, as it's completely unpredictable!"

Russell's Python program includes a variable that can be set to the minimum magnitude



For the Pi Zero, an Adafruit I2S 3W Class D amplifier is required to supply audio to the mini external speaker

The Earthquake Pi – without optional NeoPixel bar graph – showing an alert for a small quake in California

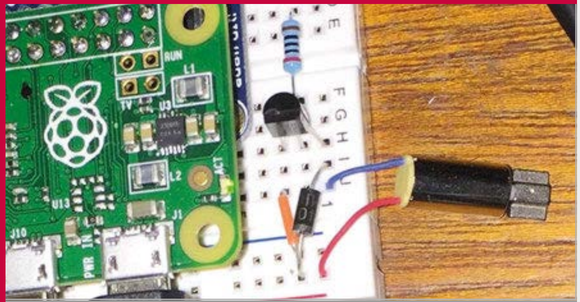
for alerts. "If you set it to alert on even the smallest (magnitude 1.0 or greater) earthquakes, then it will be going off almost every hour or so." He tells us his is set to magnitude 3.0 and higher of where many earthquakes are occurring. It's also a good geography lesson, as I've never heard of many of the cities or islands where they are, and so I look them up on a map."

> The first few times it goes off will probably scare you, as it's completely unpredictable!

and goes off a few times per day. "When I hear it rattle, I run in to see where the earthquake is located. I especially perk up if I hear it rattle for many seconds, as that means a big one occurred somewhere. After watching it for a while, you do see a pattern

To ensure the Earthquake Pi doesn't wake him during the night, Russell uses a cron job to only run the program between 8am and 11pm: "You DO NOT want to run it while trying to sleep. It would probably scare everyone in the house!"

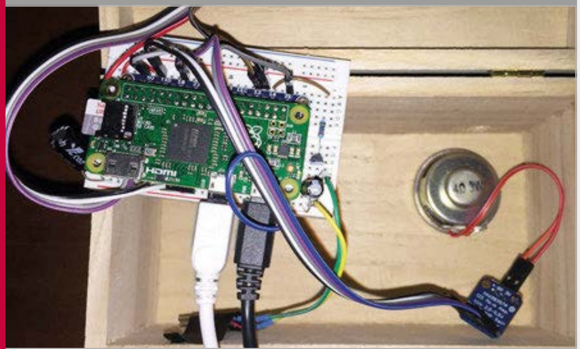
BUILDING AN EARTHQUAKE ALERT SYSTEM



>STEP-01

Vibrating motor

Taken from an old electric toothbrush, the vibrating motor is connected to the Pi Zero via a breadboard circuit, including a transistor and rectifier diode to limit the current.



>STEP-02

Internal connections

Inside the wooden box, the Pi Zero's GPIO pins are wired up to various components, including a vibrating motor, audio speaker, and LCD display (on the lid), via a cobbler kit and breadboard.

>STEP-03

LCD display

A test script is used to check the LCD display is working correctly. The Earthquake Pi requires a 20×4 screen to show all the details of the earthquake during each alert.

JONATHAN TYLER-MOORE

When he's not building awesome projects, Jonathan is a dab hand at photography, tweeting like a pro, and winning things.

@ piboyuk

THE TABLET OCARINA PROJECT

Ocarina players Robert and James sought the help of teenager Jonathan to build an interactive touch tablet for reading music

Quick Facts

- The Rebel Makers Club runs once a month
- Jonathan coded with the Adafruit Python MPR121 library
- Jonathan used his mobile phone to record notes
- The project was showcased in July
- The Blue Flash Music Trust backed the project

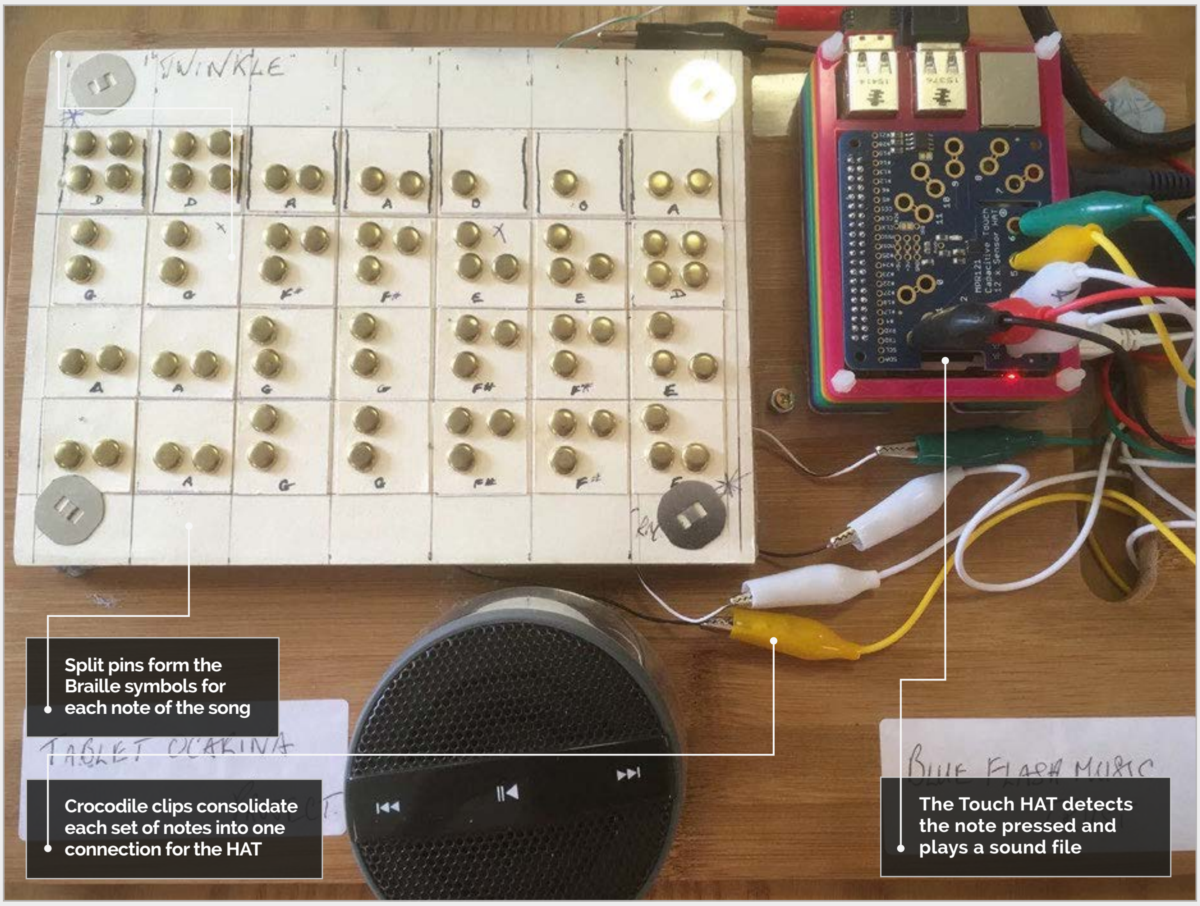
When Robert Mayfair met eight-year-old James at a party in 1994, he gave him the gift of an ocarina. James was blind, and so thankful for the gift that he later contacted Robert and asked for lessons. A new bond was instantly formed between the two.

Over the years of friendship, Robert and James have collected nearly 30 different instruments, with James's love for music ever-growing, especially toward the ocarina. The joy of learning music together, however, is often clouded by the inability to truly share the experience; resources are limited for the visually impaired.

Recently, Robert discovered that the Royal National Institute of Blind People (RNIB) had published a Braille book of ocarina music, and though this was a wonderful advancement in accessibility for the visually impaired, Robert realised that sighted people were unable to interact with the content:

"On buying the book I realised that the Braille book was of no use to the sighted person, as it was like looking at a landscape covered with snow."

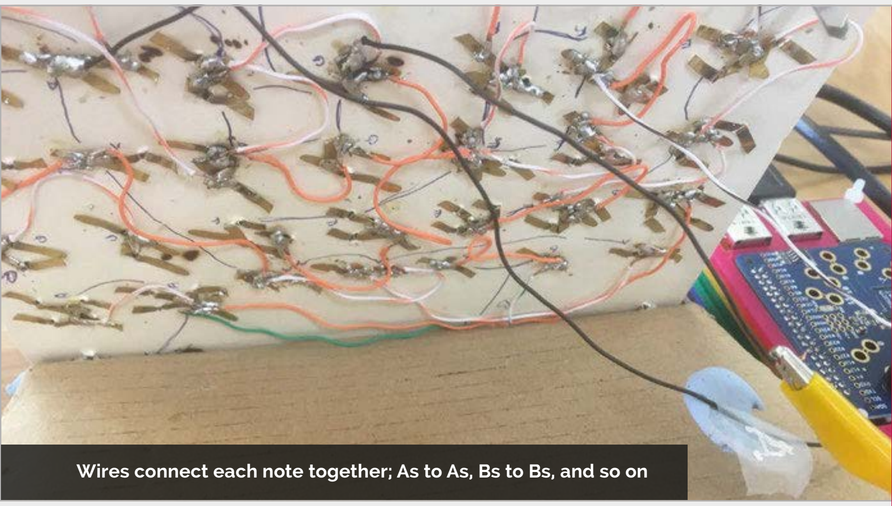
Aiming to find a solution, Robert found his answer far quicker than anticipated when he came across a HackHorsham display in a shopping centre last November. The display, using pieces of fruit to produce music via conductivity, gave him the inspiration he needed to change the way he and James read music together.



Split pins form the Braille symbols for each note of the song

Crocodile clips consolidate each set of notes into one connection for the HAT

The Touch HAT detects the note pressed and plays a sound file



Wires connect each note together; As to As, Bs to Bs, and so on

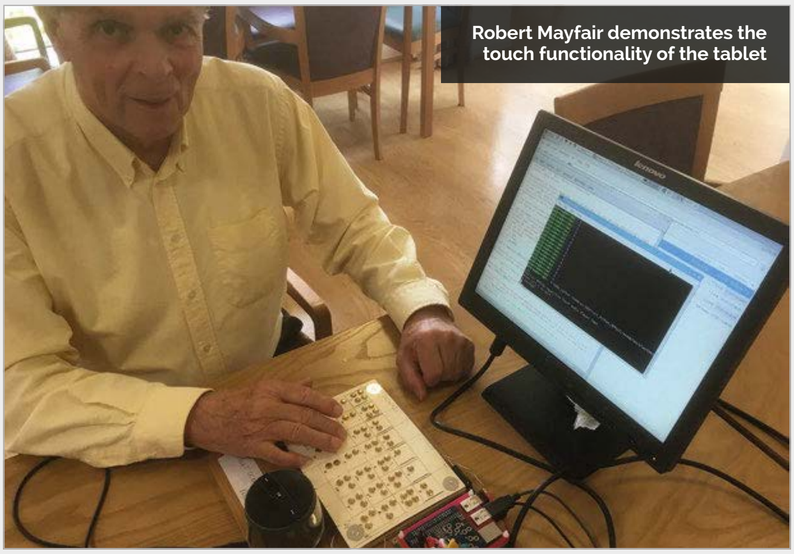
Robert produced a prototype of plastic and cardboard, and later brad nails, that James was able to interact with, recognising *Twinkle Twinkle Little Star* via touch. After a few alterations, a tablet was produced where nails formed the notes of the song in Braille, James reading them with one finger.

In April this year, Robert attended The Rebel Maker Club, a monthly event hosted by HackHorsham, and met Jonathan Tyler-Moore. Jonathan already had experience of building with Pi and finding solutions for issues using tech. So it was no surprise when the 13-year-old quickly introduced a Pi and speaker to the setup, allowing the appropriate note to be played aloud as split pins were touched on the tablet build.

Each split pin is wired, with sets of notes connected together. All A notes, B notes and so on are then connected via crocodile clips to an Adafruit Capacitive Touch HAT. Touch an A note on the board, and the HAT recognises the connection and tells the Pi to play the appropriate sound. Jonathan used an ocarina to record each note onto his mobile phone, later copying them to the Pi as OGG files.

The build was a success; James and Robert now have access to the technology that will allow them to learn music together, through both touch and sound.

The project was finally showcased at the HackHorsham event at the Capitol Theatre in July, receiving praise from musicians and educators alike. Backed by the Blue Flash Music Trust, a community-based music charity within Horsham, the Tablet Ocarina Project is still a work in progress and a promising starting point for a broader scope of builds.



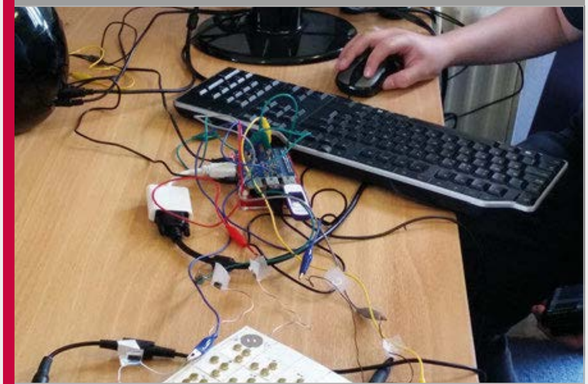
Robert Mayfair demonstrates the touch functionality of the tablet

LEARNING THROUGH TOUCH AND SOUND

>STEP-01

Building the song

The song is set out on the board, note by note, using split pins. Each set of notes is linked together into an Adafruit Capacitive Touch HAT.



>STEP-02

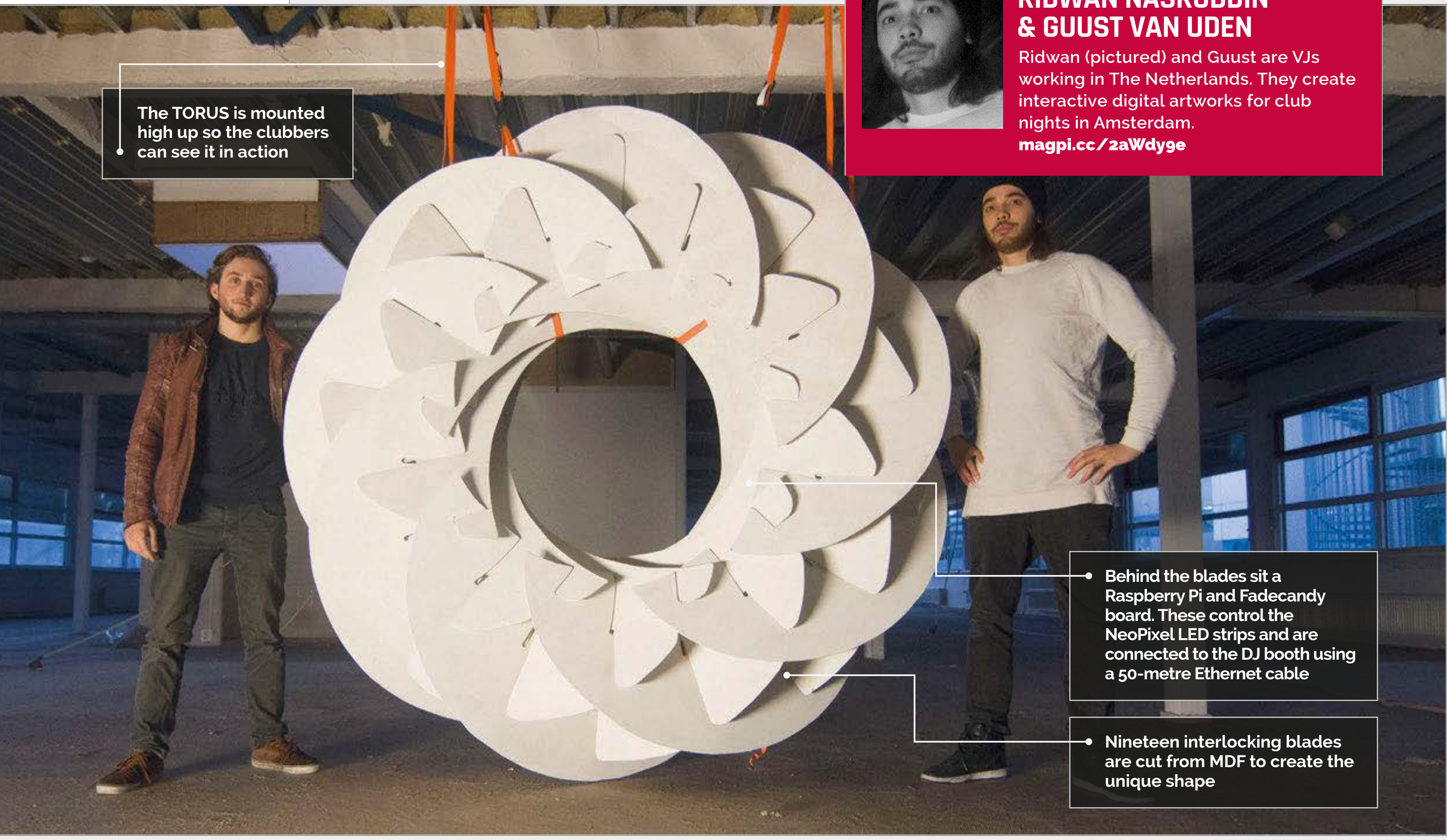
Recording the notes

Each note is recorded using an ocarina, then loaded onto the Pi as an OGG file, which is a type of audio/music file like an MP3.

>STEP-03

Learning the tune

When touching a set of pins, the HAT recognises the note and the appropriate sound is played through a speaker.



RIDWAN NASRUDDIN & GUUST VAN UDEN

Ridwan (pictured) and Guust are VJs working in The Netherlands. They create interactive digital artworks for club nights in Amsterdam.

magpi.cc/2aWdyge

Quick Facts

- When assembled, it measures two metres in diameter
- In geometry, a torus is a circle rotated around an axis
- TORUS's Fadecandy controls eight NeoPixel LED strips
- The TORUS features 400 LEDs in total
- TORUS is painted white to reflect projected film

TORUS

VISUAL MUSIC INSTALLATION

Amsterdam club nights look incredible thanks to this TORUS, a Raspberry Pi-controlled visual art sculpture

Amsterdam is famous for its party scene, but The Netherlands is also a high-tech hub with lots of creative people working in science and computer technology.

At the end of the week the Dutch like to party, and Amsterdam's clubs are full of high-tech audiovisual treats.

TORUS is a music installation piece created by Dutch visual artists Ridwan Nasruddin and Guust van Uden. It's a large sculpture based on the torus geometric shape, covered in hundreds of LEDs all controlled by a Raspberry Pi.

"TORUS started as a research project," explains Ridwan. "We were already doing visuals during club nights on a flat white screen, but we wanted to create a sculpture.

"We are interested in origami shapes and modular forms," he continues. "We stumbled upon the paper art of Yoshinobu Miyamoto [a Japanese architect]. Inspired by his art, we created the TORUS."

Built from 6mm MDF plywood, TORUS comprises 18 blades assembled in a circular pattern. The blades are covered in NeoPixel LED strips (**adafruit.com/category/168**), and the whole unit mounted in a dance club alongside a projector.

The LEDs are controlled using an Adafruit board called Fadecandy. This is a NeoPixel driver with built-in dithering, that can be controlled over USB. "We tried different ways to control the LEDs," says Ridwan, "and found out Fadecandy was

TORUS lights up Amsterdam's De Marktkantine club with its blend of film projection, light, and sculpture

the best way to send the signal from Resolume (**resolume.com**) to the Raspberry Pi and then to the LEDs."

The Raspberry Pi is connected by a 50-metre Ethernet cable, used to send the signal from Resolume (running on a laptop) to the Pi.

"Because we didn't knew a lot about how to connect LEDs to Resolume, we researched and experimented a with different boards and types of LEDs. We thought we could learn it quickly, but when we couldn't figure it out we asked around and found Nathan Marcus, a local programmer." Nathan wrote the major part of the code and with his help, they learned how to create the image they had originally envisaged.

The Raspberry Pi was added to get the LED data over a long distance, of about 50 metres. "We had already settled on using a Fadecandy because of it capabilities and ease of use," explains Nathan. The Raspberry Pi functions like a server connected over Ethernet with the laptop. It interprets the messages from the laptop and sends them to the Fadecandy.

"We had a lot of help from the guys of the club [De Marktkantine in Amsterdam] where we showed the TORUS." To hang the TORUS where everyone could see it, they used steel cables to make a hanging truss above the stage.

"The reactions of the crowd are great. We always wanted to make the experience of the music as intense as possible and it works. By creating one focus point, people really get into the vibe of the club night."

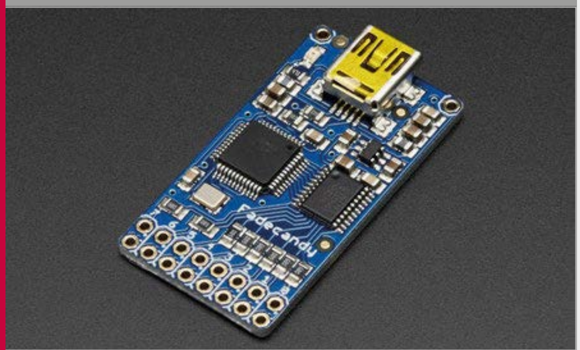
> "We always wanted to make the experience of the music as intense as possible"

BUILDING A TORUS

>STEP-01

Making the pattern

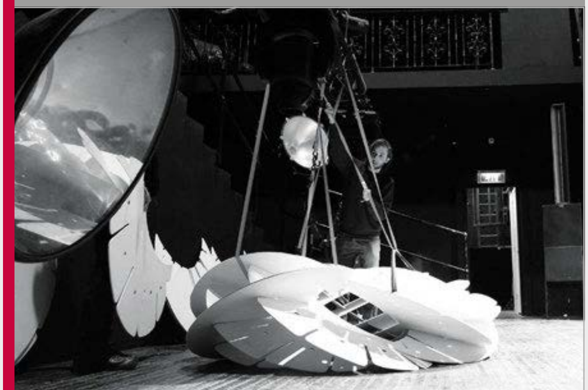
The TORUS is made from MDF cut into 19 interlocking blades. These can be assembled and disassembled, making it easy to transport to and from a club venue.



>STEP-02

Using NeoPixels

The TORUS blades are covered in AdaFruit NeoPixel LED strips. These are then controlled using Fadecandy (a custom board for controlling NeoPixel strips). A Raspberry Pi is connected to control the Fadecandy board.



>STEP-03

Assemble TORUS

The TORUS is assembled on location and the Raspberry Pi is connected to the DJ booth using a 50-metre Ethernet cable.

NESPI

MIKE GIBSON

Mike is an electronics and 3D printing hobbyist whose latest project is a miniature NES games console.
daftmike.com

We have a new king of retro consoles, thanks to NESPi. It doesn't just look like a NES: it runs mini cartridges

Quick Facts

- The cartridges contain NTAG216 NFC tags
- The reset button is used to send commands to the Raspberry Pi
- It runs RetroPie with Python code to control the NFC reader
- It's 40 percent of the size of the original NES
- Mike is designing a SNES next, and then maybe an N64

"I grew up with the NES and have really strong memories of playing it with my brothers," says Mike, the creator of NESPi, one of the most realistic retro console recreations we've seen.

NESPi started out as an attempt to 3D-print a replica NES case. "It's 40 percent the size of a real NES," Mike tells us. "It wasn't until I was a little way into researching cases that it turned into more than a replica project."

After buying a 3D printer on impulse and downloading and printing a few trinkets, he decided to make something a bit more substantial.

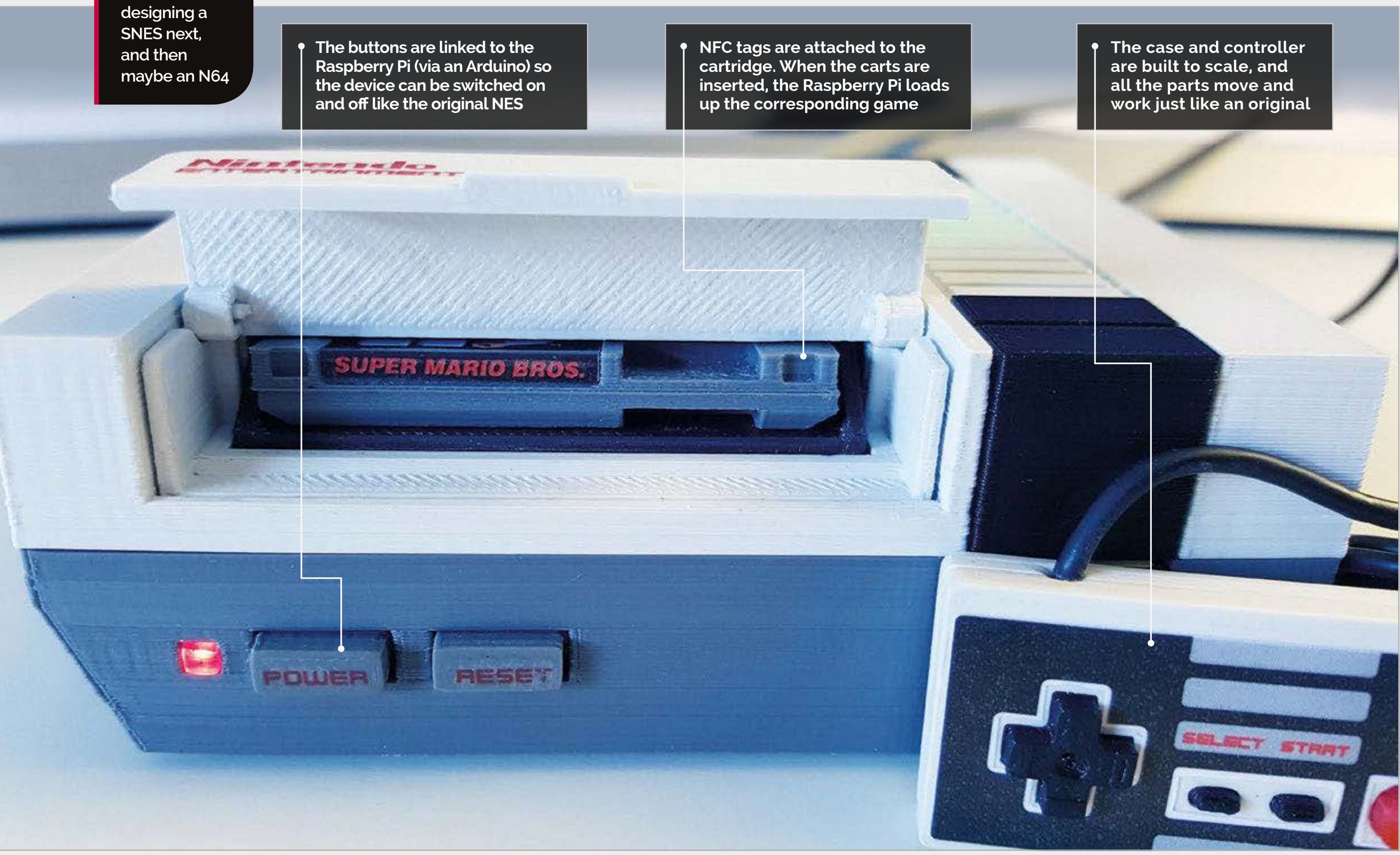
"I'd seen people build Raspberry Pis into original full-size NES cases, but I always thought that it was a shame there was so much wasted space. Part of Raspberry Pi's appeal comes from its small size; that's why I wanted mine to be as small as possible.

Mike took measurements from the real thing and tried to replicate it as best he could.

"The colour of the plastic I used also has a big role in making it look real. I researched a lot of filament suppliers before I found one that matched the shades correctly." Mike used the Monotone Mix Pack from Faberdashery (**faberdashery.co.uk**).

The NESPi goes way above and beyond most console reproductions by featuring mini cartridges to load games. These are inserted into the NESPi, which detects them using an NFC chip. "The NFC reader I used is based on the PN532 chipset," says Mike.

"The game is stored on the SD card in the Raspberry Pi, which is because



The buttons are linked to the Raspberry Pi (via an Arduino) so the device can be switched on and off like the original NES

NFC tags are attached to the cartridge. When the carts are inserted, the Raspberry Pi loads up the corresponding game

The case and controller are built to scale, and all the parts move and work just like an original

The finished result is spectacular, looking and working just like a NES in miniature

of the very small capacity of NFC. I used quite large tags, and they had just 888 bytes available to the user."

The NFC tags themselves only contain file name information; when the Raspberry Pi reads it, it looks up which game to run.

"The cartridges are 3D printed," says Mike. "I designed them in the same way as the case." He also scale NES controller, he printed an empty shell, and the Arduino Pro Micro board fit inside perfectly. He notes that while the tiny controller is fully functional, it's only really for display purposes: in practice he uses a Wii U Pro controller or Xbox joypad.

Mike is delighted with the finished project. "It looks very

> NESPi goes way above and beyond most console reproductions by featuring mini cartridges

printed little labels and covered them with sticky tape to get the right glossy look.

"The great thing about 3D printing is once your printer is set up and tuned properly, you can get very consistent results. I learned a lot about designing for 3D printing from this project."

Mike tells us the carts load very well. For extra authenticity, when the project was mostly complete, he added a cute little controller. After mocking up a 40 percent cool; inserting a cartridge and clicking it down instantly makes me nostalgic for my childhood. I have a lot of fun playing with it."

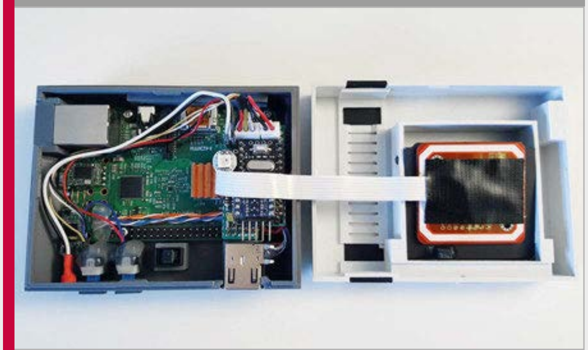
Reactions to the NESPi have also been very positive. "People love seeing it. Everyone likes the little cartridges; even those who were too young or don't remember the NES think it's really cute. Those that do remember go crazy when I show them how the cartridges work. Suddenly they're ten years old again: it's so funny."

The controller contains a Arduino Pro Micro and works just like the original

BUILDING A MINI NES

>STEP-01
3D printing

All of the parts for the NESPi are 3D printed to 40% scale, even the replica cartridges for each game. Each cartridge has an NFC tag attached to it.



>STEP-02
Pi inside

The Raspberry Pi is placed inside the NESPi case along with an Arduino and NFC reader. The Arduino is connected to the replica buttons at the front of the console.

>STEP-03
Mini NES

The final result is a miniature NES system that works just like the original. The buttons control power, and cartridges are loaded into the front.

VOICE CONTROL ON YOUR PI

Use Amazon's Alexa service on a Raspberry Pi and open up a world of voice-control projects

ROB ZWETSLOOT

Tinkerer, sometime maker, other-times cosplayer, and all-the-time features editor of *The MagPi*.

magpi.cc / @TheMagP1

You'll Need

- Alexa AVS **magpi.cc/2boDnjB**
- A constant internet connection
- External speaker
- A USB microphone

In the UK, all the *Star Trek* shows have just been put back on Netflix, reminding us of the desire to ask the computer for Earl Grey tea or Klingon coffee (we can't start the day without a raktajino, you know). So it's exciting to see Amazon's Alexa is quite readily available on the Pi now. Let's get it working, then, and make some projects.

Speak into the mic and ask Alexa for things, like a song in the key of G with a fast tempo

After analysing your request, Alexa will send a reply. It works like Google Now or Siri, mostly

The client is on the Raspberry Pi, but the heavy lifting is done in the cloud

We need to start by installing VLC. Not just normal VLC, though – we need to install this one slightly differently. Open up the terminal and enter:

```
sudo apt-get install vlc-nox vlc-data
```

This might take a while; once its done, we then need to set the environment variables so we can access VLC from Alexa properly later. Do this with:

```
export LD_LIBRARY_PATH=/usr/lib/vlc
export VLC_PLUGIN_PATH=/usr/lib/vlc/plugins
```

Next, it's time to download the Alexa files we need:

```
git clone https://github.com/amzn/alexa-avs-raspberry-pi
```

Now we need to install our dependencies: Node, JDK, and Maven. In the terminal, enter:

```
curl -sL https://deb.nodesource.com/setup | sudo bash -
```

And let it work. It will end by prompting you to install Node.js. Do that with:

```
sudo apt-get install nodejs
```

Next, use **cd** to move to **/alexa-avs-raspberry-pi/samples/companionService** and install npm with:

```
npm install
```

Once that's finished, we need to then install a specific version of the Java Development Kit (JDK). Use **cd** to move to the **alexa-avs-raspberry-pi/samples/javaclient** folder and run:

```
./install-java8.sh
```

You will get a message from Oracle Java installer that you must accept the terms of service for the Java SE platform, which you need to now do.

Once that's complete, download Apache Maven from **magpi.cc/2bDPluf**. Move to the **Downloads** folder and extract the contents with:

```
sudo tar xvf apache-maven-3.3.9-bin.tar.gz -C /opt
```

You then need to create a file with some system settings for Maven. Start by creating the file like so:

```
sudo touch /etc/profile.d/maven.sh
sudo nano /etc/profile.d/maven.sh
```

Add the following to the file you just opened:

```
export M2_HOME=/opt/apache-maven-3.3.9
export PATH=$PATH:$M2_HOME/bin
```

Save and exit the file. Reboot your Raspberry Pi before continuing.

Certification

We now need to generate self-signed certificates:

```
sudo apt-get install openssl
```

Once installed, move back to **/alexa-avs-raspberry-pi/samples/javaclient** and run the script:

```
./generate.sh
```

It will ask you to enter some information. Enter the following details exactly as shown:

```
Product ID: my_device
Serial Number: 123456
```

Just press **ENTER** when it prompts you for a password and then let it run and generate a key.

Now we can get our details for the Alexa Voice Service; this does require an Amazon account, though. Go to **developer.amazon.com** and log in – we then had to 'complete' our registration before continuing, so be prepared to do so as well.

Once you're at the dashboard, click on the Apps & Services tab, then Alexa. On Alexa Voice Service, hit Get Started. From the drop-down menu 'Register a Product Type', select Device.

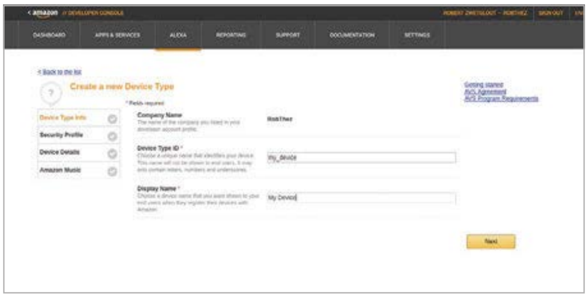
On the first page, fill in Device Type ID as **my_device** and Display Name with **My Device**. Click Next to go to the security profile. Click on the Security Profile drop-down and choose 'Create a new profile'.

Enter the following:

```
Security Profile Name: Alexa Voice Service
Sample App Security Profile
Security Profile Description: Alexa Voice
Service Sample App Security Profile Description
```

Click Next and your Client ID and Client Secret will be generated for you. Go to the Web Settings tab and make sure the security profile you just created is selected in the drop-down menu, then click the Edit button.

On Allowed Origins, click Add Another and then enter **https://localhost:3000** in the text field that appears. For Allowed Return URLs, enter **https://localhost:3000/authresponse** and then click on Next. We're now on Device Details; first set the Category as Other. Use a description of 'Alexa Voice Service sample app test' and then choose 'Longer than 4 months / TBD' for the expected timeline question. Finally, enter 0 for for the number of devices you plan to commercialise and hit Next once more. On the next tab, click for Amazon Music and hit Submit. You're done!



Above You'll need to set up the app on your Amazon developers page, which is linked to your Amazon account

Final configurations

In a browser, go to **magpi.cc/2bvWrNu**. At the top of the page, select the security profile we created and click Confirm. You'll need to enter a dummy web address for the consent privacy notice URL.

Click Save. Click on Show Client ID and Client Secret and make a note of them.

Back on the Pi, move the **alexa code** folder to the Desktop and rename it **alexa-avs-raspberry-pi-master**. In the terminal, open the following file:

```
nano /home/pi/Desktop/alexa-avs-raspberry-pi-master/samples/companionService/config.js
```

Post your Client ID and Client Secret in the fields clientID and clientSecret respectively, and then save and close the file.

Now we can get the Alexa service running. Start in the terminal with:

```
cd /home/pi/Desktop/alexa-avs-raspberry-pi-master/samples/companionService
npm start
```

Open a new terminal window. Move to **alexa-avs-raspberry-pi-master/samples/javaclient** and use:

```
mvn install
```

It will say 'build success' when done. Run it with:

```
mvn exec:exec
```

Talk to me

A window will pop up asking you to register the device. Copy the URL into a browser and log into Amazon, and click Okay on the next page to confirm everything. You'll be redirected to a page saying 'device tokens ready'.

Click OK on the original pop-up and you can start asking Alexa stuff. Hit the Start Listening button, wait for the audio cue, then ask it something like 'what is two plus two'. Hit Stop Listening and it should reply 'four'. And that's it, you're ready to play with Alexa!

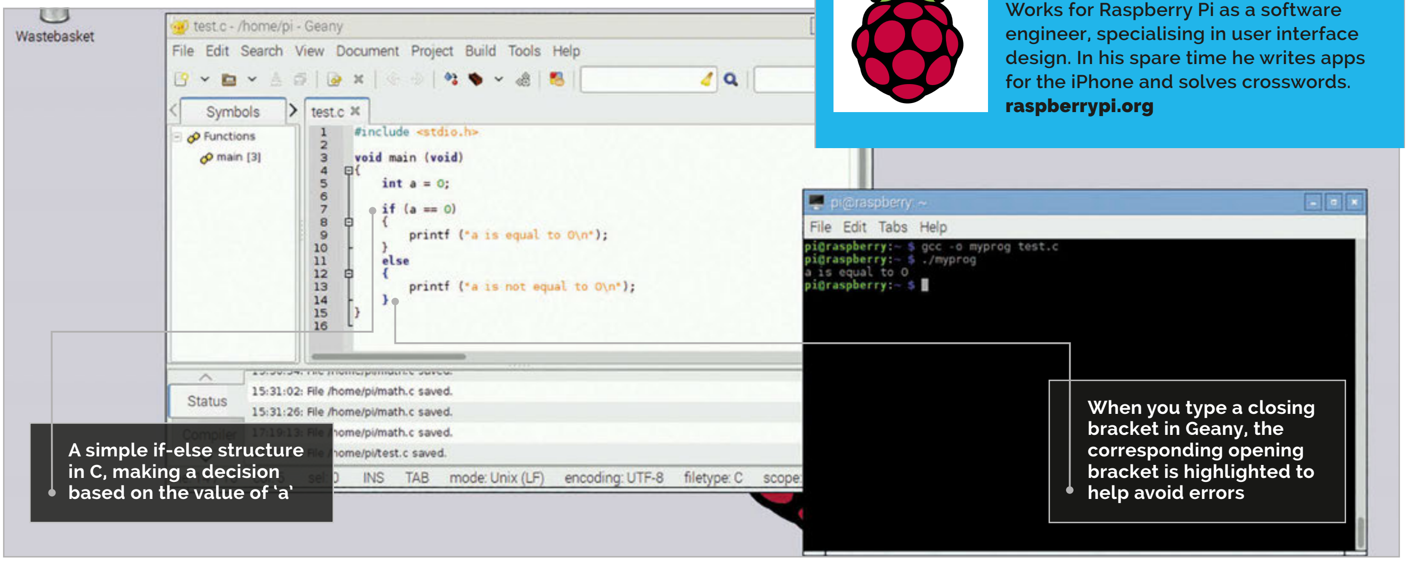
ADAPTED FROM

This tutorial was adapted from this excellent tutorial by Akash Chandran: **magpi.cc/2bljvC2**

ALEXA CONTEST!

Join the Internet of Voice challenge and create an amazing way to add Alexa to a Pi project. Find out more here: **magpi.cc/2baErdf**

SIMON LONG
Works for Raspberry Pi as a software engineer, specialising in user interface design. In his spare time he writes apps for the iPhone and solves crosswords.
raspberrypi.org



A simple if-else structure in C, making a decision based on the value of 'a'

When you type a closing bracket in Geany, the corresponding opening bracket is highlighted to help avoid errors

AN INTRODUCTION TO C PART 03
CONDITIONS & COMPARISONS

Branches and loops: controlling the flow of your C program

One of the fundamentals of any programming language is the ability to make conditional operations – to change the program's flow depending on the result of a test – and C allows you to do this. In this instalment, we'll look at how you test conditions within your C programs, and how you use the results to determine what happens next.

In C, the mechanism for controlling flow based on testing a condition is the if-else statement. Here's a simple example:

```
#include <stdio.h>

void main (void)
{
  int a = 0;

  if (a == 0)
  {
    printf ("a is equal to 0\n");
  }
  else
  {
    printf ("a is not equal to 0\n");
  }
}
```

Here, the keyword **if** is followed by a test enclosed in round brackets, in this case **(a == 0)**. If the test evaluates as true, the operations enclosed by the curly brackets after the test are executed.

This example also shows the use of an else clause. At the end of the curly brackets around the operations which you want to execute if the test is true, there is an **else** followed by another set of curly brackets, which contain the operations you want to execute if the original test evaluated as false.

Try compiling the code above, and change the value with which **a** is initialised to make sure it does what you expect.

CURLY BRACKETS

Curly brackets are used to group together a set of statements which always execute together. If your loop or if statement only needs to execute one single statement, you can leave out the curly brackets after the test, but this can make the code's purpose less obvious to a human!

= or ==

That's all fine, but what's this **a == 0** all about? Surely if we want to know whether **a** is equal to 0, we just put **a = 0**. Why the two equals signs? Well, try replacing the double equals sign with a single equals and see what happens.

This is a very important aspect of C syntax, and a common source of bugs. The equals sign is used for two different things: one is to assign a value to a variable, whereas the other is to test whether a variable is equal to a value. A single equals sign (**=**) assigns a variable; a double equals sign (**==**) tests a variable.

So the statement...

```
if (a == 0)
```

...tests to see if **a** is equal to 0. If it is, then the test evaluates as true, and the code immediately after the **if** is executed.

But the statement...

```
if (a = 0)
```

...doesn't compare **a** against 0 at all: it just sets **a** to 0. So how does the compiler decide what to do next? In this case, it just looks at the value of what's in the brackets; you've set **a** to 0, so the value inside the brackets is 0.

In C, a value of 0 is equivalent to false, and a non-zero value is equivalent to true. So by replacing the double equals with a single equals, you've changed the value of **a**, and then you look to see if the value you've set **a** to is equivalent to true or false; neither of which were what you wanted to do! If a C program is behaving strangely, check very carefully that all your tests are actually tests and not assignments: this is a very easy mistake to make.

So **==** is the test to see if a value is equal to another one. There are other useful symbols that can be used in a test. The symbol **!=**, for example, means 'is not equal to'. The mathematical operators **>** and **<** are used to test for 'is greater than' and 'is less than' respectively, and they can also be combined with an equals sign to give **>=** and **<=**, the tests for 'is greater than or equal to' and 'is less than or equal to'.

You can combine tests with logical operators. The symbol **&&** is a Boolean AND (i.e. test whether both sides are true), and **||** is Boolean OR (i.e. test if either side is true). So, to execute code only if both **a** and **b** are 0, you would use **if (a == 0 && b == 0)**. To check if either **a** or **b** is 0, you use **if (a == 0 || b == 0)**.

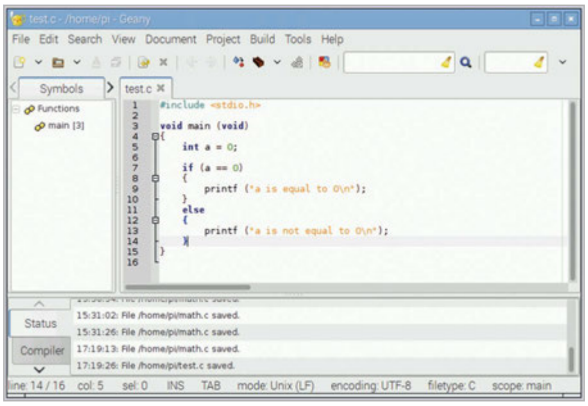
Similarly, you can use the operator **!** as a Boolean NOT to invert the result of a test, so **if (!(a == 0))** is the same as **if (a != 0)** .

Looping

The if statement is useful for making a single decision, but what if you want to do something repeatedly until a test is true or false? We use a while loop for this, and here's an example:

```
#include <stdio.h>

void main (void)
{
  int a = 0;

  while (a < 5)
  {
    printf ("a is equal to %d\n", a);
    a++;
  }
  printf ("a is equal to %d and I've
finished", a);
}
```

This is very similar to an if statement, but the code in the curly brackets is executed repeatedly for as long as the test in the round brackets is true, not just once.



Left Make sure that you use a double equals sign in the brackets after the if, not a single one!

So in our example code, **a** is initialised to 0. We enter the while loop, and test to see if **a** is less than 5, which it is, so the code inside the curly brackets is executed. The value of **a** is printed out, then we have one of C's useful shortcuts to save too much typing...

a++ is the same as **a=a+1**; the double plus means 'add one to this variable'. Similarly, **a--** means 'subtract one from this variable'; these are very commonly used to count the times around a loop. The notation **a+=1** can also be used to add a value to a variable; this also works for other arithmetic operators, so **a*=3** multiplies **a** by 3, and so on.

In the while loop, each time the code in the curly brackets has been executed, the test in the round brackets is repeated; if it's still true, the loop code is repeated again. As soon as the test is false, execution continues with the line after the closing curly bracket.

Sometimes, we might want a loop which always runs at least once before a test is made. We do this with a small modification to the syntax to create a do-while loop:

```
#include <stdio.h>

void main (void)
{
  int a = 0;

  do
  {
    printf ("a is equal to %d\n", a);
    a++;
  } while (a < 5);
  printf ("a is equal to %d and I've
finished", a);
}
```

The keyword **do** now goes before the curly bracket, and the **while** and test go after the closing curly bracket. When this runs, the code in the loop always executes once before the test; you can test this by running both the loop examples above with **a** initialised to 5 rather than 0, and seeing how the behaviour differs.

In the next instalment, we'll look at some more complex examples of looping and flow control.

ELSE-IF

You can have multiple else statements in one test. Instead of one simple **else** for one alternative, use **else if ()** with a new test for each alternative you want. We'll look more at this in the next instalment.

INFINITE LOOPS

Make sure your loops always finish! If the condition you test in a while loop never evaluates to false, your program will sit in the loop forever and never finish. If a program appears to be doing nothing when you run it, check your loop tests.

MORE ABOUT SEMICOLONS

Unlike the test in an if statement or a while loop, you need to put a semicolon after the test in a do-while loop. This indicates the end of the loop code; in a while loop, the loop code doesn't end until the last statement inside the curly brackets.

MIKE'S PI BAKERY

MIKE COOK

Veteran magazine author from the old days and writer of the Body Build series. Co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.
bit.ly/1aQqu15

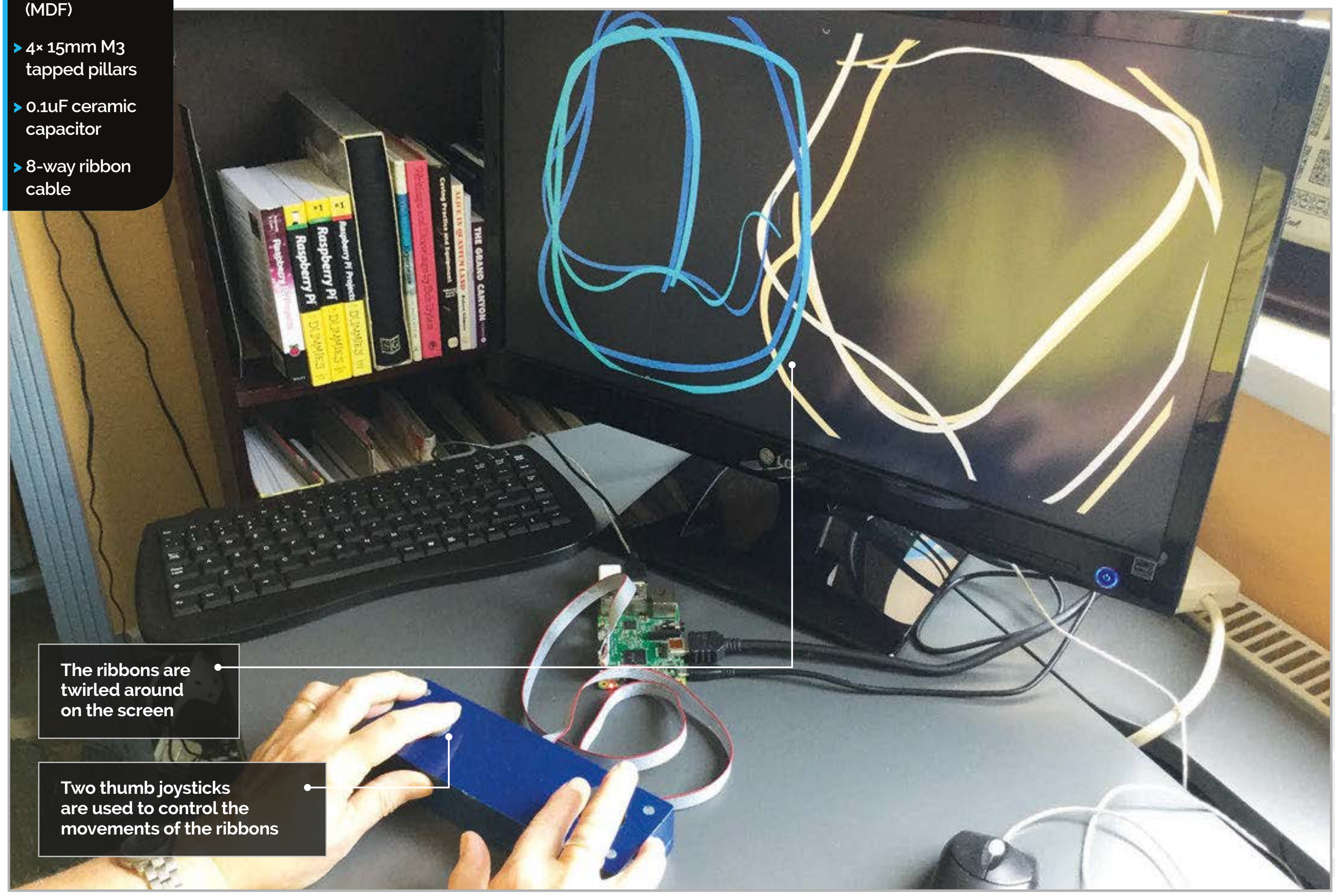
RHYTHMIC GYMNASTICS

Create hypnotic ribbon effects with this ribbon twirling simulator in honour of the Olympics

You'll Need

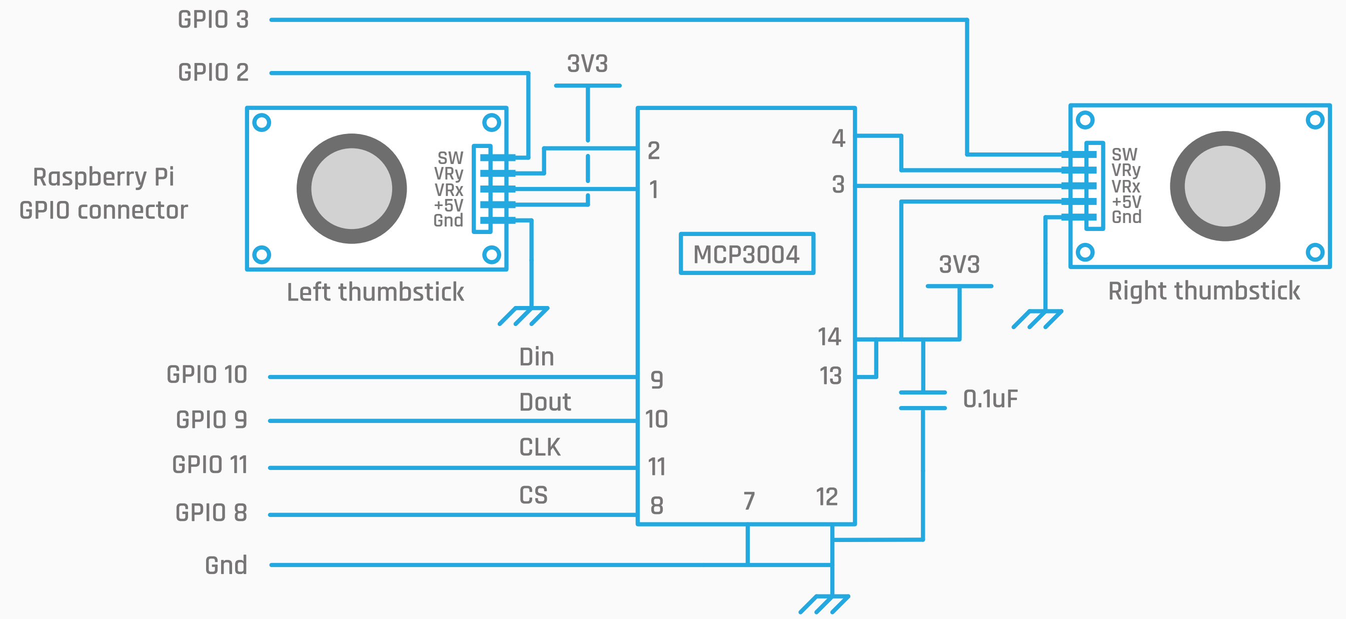
- MCP3004 – A/D converter
- 2× thumb joysticks
- 13-by-10-hole stripboard
- Wooden box (MDF)
- 4× 15mm M3 tapped pillars
- 0.1uF ceramic capacitor
- 8-way ribbon cable

This month's project, like last month's, was inspired by the recent Olympic games. We've always liked the rhythmic gymnastics, especially the section with the ribbons, so we set off to recreate this on the Pi. This project is a bit of a departure from normal Bakery stuff, in that for the first time we switch languages from Python to Processing. While Processing is a strange name for a programming language, the language itself is quite good. Basically, it's an implementation of Java, and Java is implemented with a C syntax. It's a language much beloved by the artistic, creative community, and there are plenty of stunning examples of its use. We used it for this project because there was already an excellent ribbon drawing class that makes the code writing so much easier.



The ribbons are twirled around on the screen

Two thumb joysticks are used to control the movements of the ribbons



The hardware

The two ribbons are controlled by small thumb joysticks, and are read into the Pi with an MCP3004 analogue-to-digital converter (ADC). This is the cousin of the MCP3002 chip we used in the Spectrum Display and the Hairgrip sequencer of MagPi issues 45 and 46. This chip has four analogue inputs, but uses the same SPI software commands to interface with it as the smaller chip. You could also use the eight-channel MCP3008 if you like, and there are a few pre-built Pi interfaces that use this chip. The schematic is quite simple and is shown above; we used the surface-mount version of the chip, but through-hole chips are also available. Full construction notes for the through-hole chip are given in the step-by-step guide.

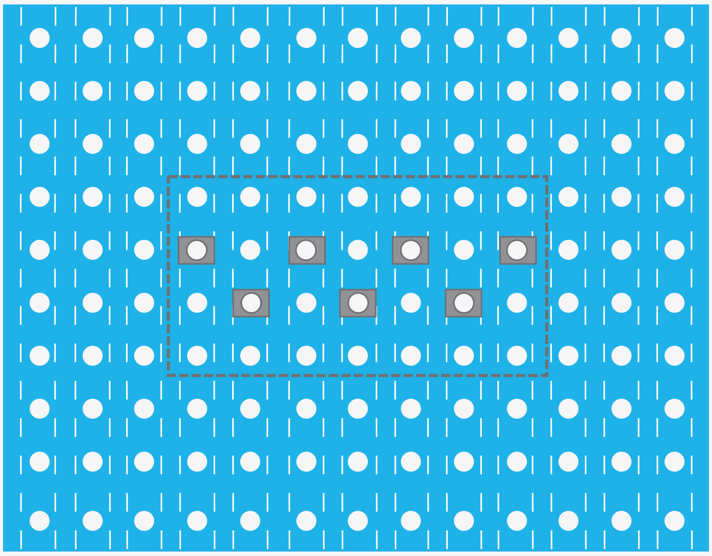
The language

As Processing has only recently been ported to the Pi, it's likely you will need to install it; this is not complex but can take a little time. You can install it from the command line by typing:

```
curl https://processing.org/download/install-arm.sh | sudo sh
```

After a reboot, it will appear in the Programming section of the desktop's Main Menu. The good news is that the Pi version of Processing has support for the GPIO pins built in; for this project we need to use the SPI port, so this needs to be enabled. These days this is easy to do: just open Raspberry Pi Configuration (under Menu>Preferences), navigate to the Interfaces tab, enable SPI, and then restart your Pi. After installing Processing, you will find a folder called **sketchbook** on the top level of your files; this is the normal place to put your Processing code. Start up Processing from the Programming menu; it's not quick to start, especially the first time, so you'll need to be a little patient. Have a look at the examples found in the File menu; most work, although some of the graphic demos don't. One favourite of ours is found at Topics>Fractals>Tree.

BUILDING THE RIBBON CONTROLLER



>STEP-01
Prepare the board

Get a piece of 13-by-10-hole stripboard and cut the tracks as shown, with the view from under the board. There's just one IC on this board; the rest of the board is used to hold the input/output wires. Solder a 14-pin socket to the other side of the board.

The software

Now take the blank program that came up on startup and save it as 'ribbons', then click the arrow next to the tab, select New Tab, and name it 'MCP3004'. Make three more tabs and call them 'Ribbon', 'RibbonManager', and 'RibbonParticle'. Now we're ready to start filling these tabs with code, so we'll start with the pre-written classes. Go to the page **magpi.cc/2aRvO9i**, copy the code from the comment `//==manager` to `//== ribbon`, and paste it into the RibbonManager tab.

Next, copy the code from `//==ribbon` to `//== particle` and paste it into the Ribbon tab.

Finally, copy the code from `//==particle` to the end of the file and paste it into the RibbonParticle tab.

Now for our code. Type the **Ribbons.pde** listing into the 'ribbons' tab and **MCP3004.pde** into the MCP3004 tab. This last class was part of the Processing distribution, but it contained an error that took about a day to track down; make similar changes to the MCP3008 class if you want to use that.

Now we need to make some minor changes to the RibbonManager class. Find the part of the file near the end that starts `void setDragFlaire` and after that line, type in the new method `setNewColour` found in the **Change_colour.pde** listing.

The last tweak is in the Ribbon tab. Change:

```
float radiusMax = 8;
```

to

```
float radiusMax = 12;
```

One final thing to do is to take two JPEG images and save them in the **ribbons** folder of the **sketchbook** folder. These should be called **swatch_01.jpg** and **swatch_02.jpg** and are the images that will be used to generate the colours of the ribbons. Two random pixels are chosen from each for the colours. These are best as just tiny images, like a 16-by-16 pixel image of colours that go together, but any image file will do.

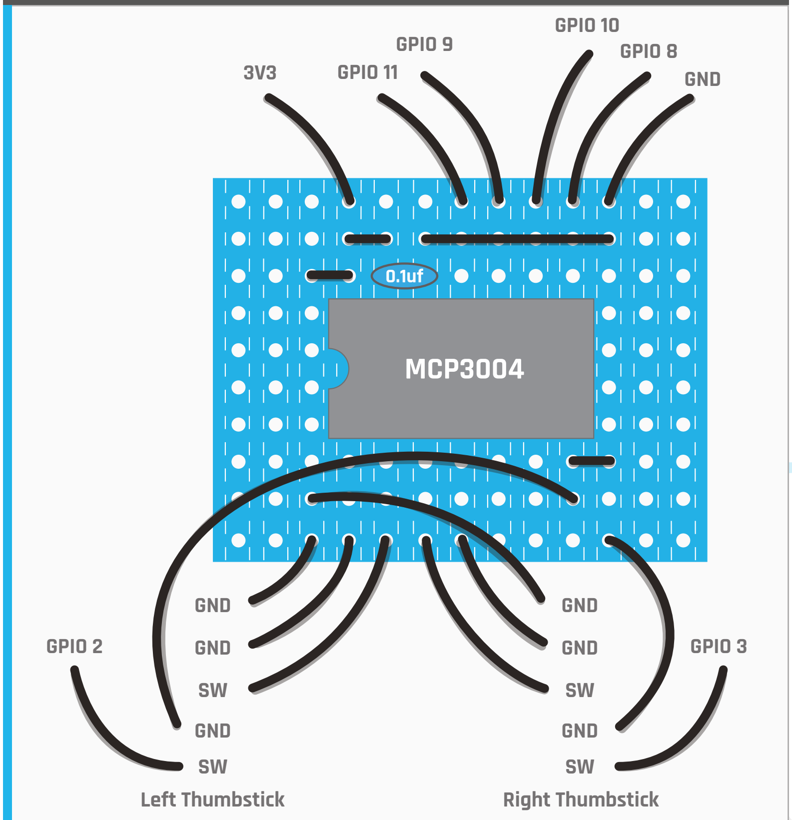
Running the code

Click the triangle in the top-left corner of the code window to run the code; it's set up to run in full screen mode, although you can set a window size if you want, by commenting out the full screen and uncommenting the line below. The code will run faster in a small window than full screen. Note that in Processing the double slash `//` is the comment symbol. Moving the joypads will move the ribbon round the screen, and pressing on the joypad will click a switch and change the ribbon colour.

All the code, ready to run and with two swatch images, is available in our GitHub repository.

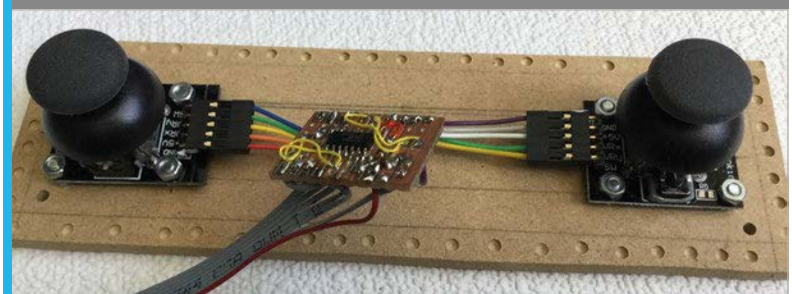
Taking it further

There are lots of default parameters you can change about the ribbons, either in the Ribbon tab or when you instantiate the ribbon in the main 'ribbons' tab. For example, you can change the ribbon parameters of `friction`, `gravity`, `dragFlare`, and `ribbonAmount`. This last one controls the number of strands in a ribbon. Also, making small swatch images is a good way to change the colours in a way you like. Happy twirling!



>STEP-02
Mounting the components

Solder solid wire links between the holes as shown and fit the 0.1uF decoupling capacitor. Wire the two thumb joysticks to the board. Wire the connections to the Pi using a length of 8-way ribbon cable. We used a 26-way socket to plug onto the GPIO pins; that way it will work with all models of Pi.



>STEP-03
Construct the box

The base plate is drilled with shallow holes to allow the Gorilla Glue to foam into them, and the sides glued on are made from 17mm strip pine. Make a top with two 25mm diameter holes for the joysticks and fasten to the base with tapped pillars. Make a small notch in the top side to allow the ribbon cable to come through.

Ribbons.pde

```
// Ribbons by Mike Cook August 20 16
// with credit to http://www.zenbullets.com
import processing.io.*;
MCP3004 adc;

int ribbonAmount = 2; // number of ribbon strands
int ribbonParticleAmount = 20;
float randomness = .2;
RibbonManager ribbonManager1;
RibbonManager ribbonManager2;
float xPad1 = 0.120, yPad1 = 0.120;
float xPad2 = 0.120, yPad2 = 0.120;
boolean rightClick = false;
boolean leftClick = false;

void setup()
{
  fullScreen();
  //size(600, 450);
  frameRate(30);
  background(0);
  GPIO.pinMode(2, GPIO.INPUT);
  GPIO.pinMode(3, GPIO.INPUT);
  ribbonManager1 = new RibbonManager(ribbonAmount,
ribbonParticleAmount, randomness, "swatch_01.jpg");
  ribbonManager2 = new
RibbonManager(ribbonAmount,
ribbonParticleAmount, randomness,
"swatch_02.jpg");
  adc = new MCP3004(SPI.list()[0]);
}


void draw()
{
  fill(0, 255);
  rect(0, 0, width, height);
  doClick();
   xPad1 = 0.5 - (adc.getAnalog(0)/2.0);
   yPad1 = adc.getAnalog(1);
   xPad2 = 0.5 + (adc.getAnalog(2)/2.0);
   yPad2 = adc.getAnalog(3);
   stroke(255,255,255);
   ellipse(xPad1*width, yPad1*height, 15, 15);
   ellipse(xPad2*width, yPad2*height, 15, 15);
  ribbonManager1.update(int(xPad2*width), int(yPad2*height));
  ribbonManager2.update(int(xPad1*width), int(yPad1*height));
}

void doClick(){
  if (GPIO.digitalRead(2) == GPIO.LOW && !rightClick) {
    print("right press ");
    rightClick = true;
     ribbonManager2.setNewColour();
  }
  if (GPIO.digitalRead(2) == GPIO.HIGH && rightClick) {
    println("release ");
    rightClick = false;
  }
  if (GPIO.digitalRead(3) == GPIO.LOW && !leftClick) {
    print("left press ");
    leftClick = true;
    ribbonManager1.setNewColour();
  }
  if (GPIO.digitalRead(3) == GPIO.HIGH && leftClick) {
    println("release ");
    leftClick = false;
  }
}
```

Language
>PROCESSING 3.1.1

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS
Check out Mike's Bakery videos at:
magpi.cc/1NqJnTz

MCP3004.pde

```
import processing.io.SPI;

class MCP3004 extends SPI {

  MCP3004(String dev) {
    super(dev);
    super.settings(500000, SPI.MSBFIRST, SPI.MODE0);
  }

  float getAnalog(int channel) {
    if (channel < 0 ||  channel > 3) {
      System.err.println("The channel needs to be from 0 to
3");
      throw new IllegalArgumentException("Unexpected
channel");
    }
    byte[] out = { 0, 0, 0 };
    // encode the channel number in the first byte
    out[0] = (byte)(0x18 | channel);
    byte[] in = super.transfer(out);
    int val = ((in[1] & 0x3f)<< 4 ) | ((in[2] & 0xf0) >> 4);
    // val is between 0 and 1023
    return float(val)/1023.0;
  }
}
```

Change_colour.pde

```
void setNewColour() {
    for (int i = 0; i < ribbonAmount; i++) {
      int xpos = int(random(img.width));
      int ypos = int(random(img.height));
      color newColor = img.get(xpos, ypos);
      ribbons[i].ribbonColor = newColor;
    }
  }
```

IOANA CULIC

Ioana is an Internet of Things specialist and has written several IoT tutorial books and articles. She focuses on IoT in education.
wyliodrin.com

BUILD A CAR MONITORING SYSTEM

Thinking about monitoring cars passing by? Prototype your own number plate recognition system with a Camera Module connected to your Pi

You'll Need

- Wyliodrin STUDIO **magpi.cc/1Q5i4il**
- SS441A Hall sensor
- 220Ω resistor
- 16×2 LCD
- Potentiometer
- Jumper wires
- Breadboard

Ever wondered how the systems monitoring the traffic flow work? This article will guide you through the first part of building a mock-up system that monitors cars passing by your Camera Module. By using a simple magnetic sensor, you can detect the presence of a magnet and display its traffic rate on an LCD, all by using a data flow approach which makes programming fun and intuitive.

>STEP-01

Connect the Hall sensor

The SS441A sensor is a device that reacts to magnetic fields; its behaviour can be compared to that of a push button. The sensor outputs HIGH in the normal state and LOW when a magnet is near it.

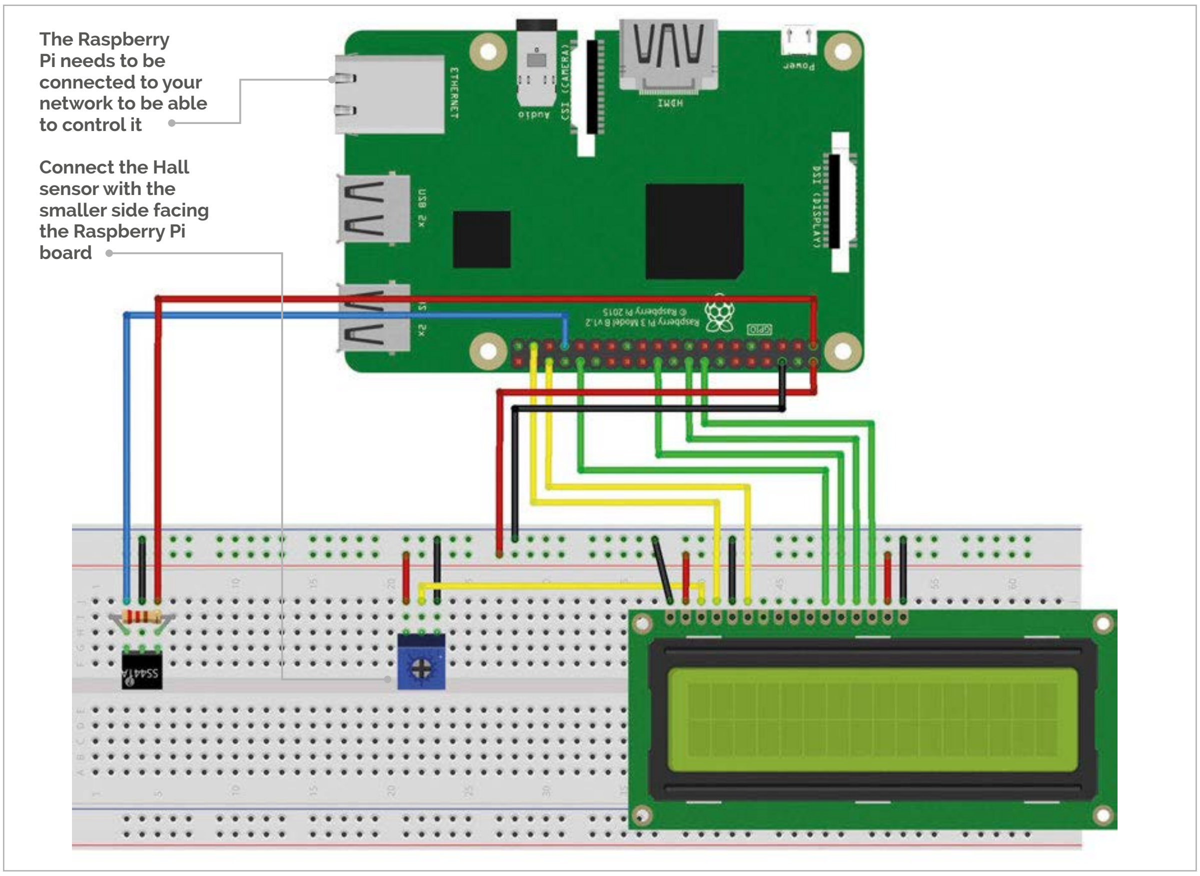
We insert the sensor into the breadboard with the smaller side facing the board, then connect the right leg to the 3.3V pin, the middle one to the GND, and the left leg to a GPIO pin of the Raspberry Pi.

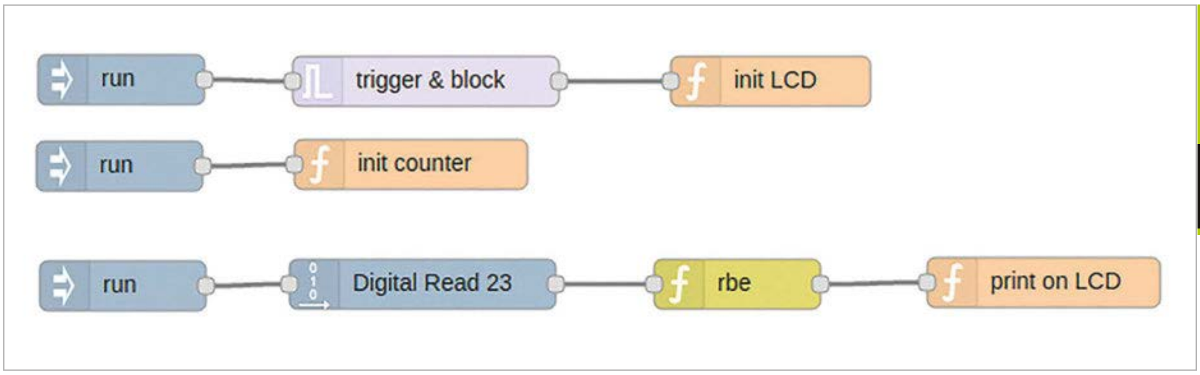
We also need to connect the 220Ω resistor between the 3.3V pin and the GPIO pin. Refer to the Fritzing diagram below for how to do this.

>STEP-02

Connect the LCD

For the LCD, the first two pins on the right are used to power it up. The next four pins, the ones connected by green cables, are the four data pins, and finally there are two control pins.





There are also three pins used for contrast. Two of them are used to power on the backlight and there's one more pin directly connected to the potentiometer, which will output a different voltage depending on its angle. This way, we can control the contrast just by rotating it. Again, you can refer to the Fritzing diagram to make sure it's wired up properly.

>STEP-03

Create a Node-RED application

We will program the Raspberry Pi using Wyliodrin STUDIO. First, we must download the application (**magpi.cc/1Q5i4il**) and follow the steps to connect the Raspberry Pi. For more details on how to get started with Wyliodrin STUDIO, read the Pi thermometer article from issue 45 of The MagPi (**magpi.cc/1rBnGgA**).

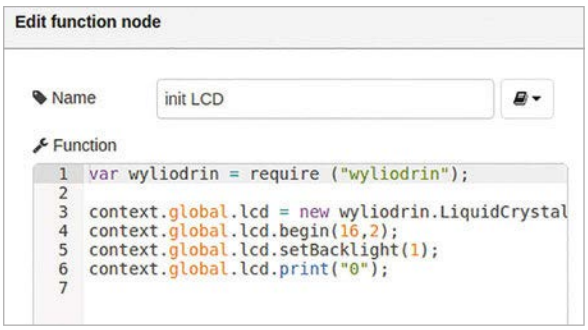
Once connected, we create a new Streams application, which allows us to write applications using Node-RED. This language is based on events; it uses nodes which are connected and transmit messages to each other. Once a node receives a message, it processes it and sends it forward.

>STEP-04

Initialise the LCD

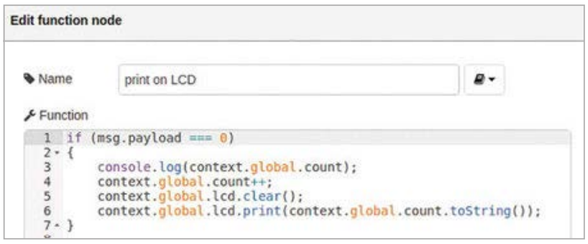
The first stream of nodes initialises the LCD connection. This stream needs to function only once. It starts with a run node which we double-click and set to Fire once at the start. Then, we drag the trigger node and set it to wait until reset.

Next, the function node initialises the LCD. First of all, we must require the **wyliodrin** module.



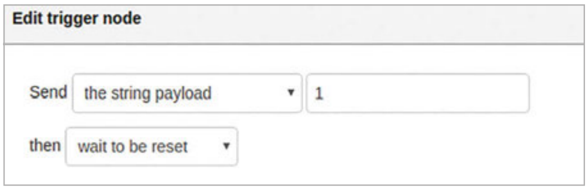

```
var wyliodrin = require ("wyliodrin");

context.global.lcd = new wyliodrin.LiquidCrystal
context.global.lcd.begin(16,2);
context.global.lcd.setBacklight(1);
context.global.lcd.print("0");
```

Above The init LCD function creates the global lcd object based on the pins the LCD is connected to, and prints 0 on the display




```
if (msg.payload === 0)
{
    console.log(context.global.count);
    context.global.count++;
    context.global.lcd.clear();
    context.global.lcd.print(context.global.count.toString());
}
```

Left The print LCD function verifies the incoming value and if the value is 0, increases the counter and displays the new value



Left The trigger node is set to wait to be reset before sending another message

Afterwards, we call the **LiquidCrystal** constructor, which gets the following pin numbers as parameters: RS, Enable, D5, D6, D7, and D8 (the four data pins).

Finally, we call **begin** and **setBacklight** and the LCD is ready to be used.

>STEP-05

Display the number of cars

The second stream of nodes runs every 10 minutes, Fires once at the start, and sets **context.global.count** to 0. Any variable prefixed by **context.global** can be accessed from any node.

For the third stream, we set the run node to send a message every 0.1 seconds and activate the digital read node, which reads the value coming from pin 23. We only want to take into account the changes in the system, which is why we use the rbe node.

Finally, we implement the 'print on LCD' function which verifies if the value read is 0, increases the counter, and displays the value on the LCD.

>STEP-06

Count the cars

All that's left to do is to run the code and simulate the passing cars by bringing a magnet towards the Hall sensor. To create a system that really monitors the traffic, the Hall sensor needs to be replaced by a coil which is usually located under the pavement.

In the second part of the tutorial, we will take a picture of the cars passing by and use a web service to identify the number plate. Look out for it next month!

π))) Sonic Pi PART 13

SAM AARON

Sam is the creator of Sonic Pi. By day he's a research associate at the University of Cambridge Computer Laboratory; by night he writes code for people to dance to.
sonic-pi.net

FIVE LIVE-CODING TECHNIQUES

Wave your hands in the air with **Sam Aaron** and get the party jumping as you live-code

You'll Need

- Raspberry Pi running Raspbian
- Sonic Pi v2.9+
- Speakers or headphones with a 3.5mm jack
- Update Sonic Pi: `sudo apt-get update && sudo apt-get install sonic-pi`

In this month's tutorial we're going to take a look at how you can start treating Sonic Pi like a real instrument. We therefore need to start thinking of code in a completely different way. Live coders think of code in a similar way to how a violinist thinks of their bow. In fact, just like a violinist can apply various bowing techniques to create different sounds (long slow motions vs short fast hits) we will explore five of the basic live-coding techniques that Sonic Pi enables. By the end of this article, you'll be able to start practising for your own live-coded performances.

Memorise the shortcuts

The first tip to live-coding with Sonic Pi is to start using the shortcuts. For example, instead of wasting valuable time reaching for the mouse, moving it over to the Run button and clicking, you can simply press **ALT** and **R** at the same time, which is much faster and keeps your fingers at the keyboard, ready for the next edit. You can find out the shortcuts for the main buttons at the top by hovering the mouse over. See section 10.2 of the built-in tutorial for the full list of shortcuts.

When performing, one fun thing to do is to add a bit of flair with your arm motion when hitting shortcuts. For example, its often good to communicate to the audience when you're about to make a change – so embellish your movement when hitting **ALT+R**, just like a guitarist would do when hitting a big power chord.

Below Sam Aaron and Ben Smith performing at Cambridge Junction as Poly Core

Manually layer your sounds

Now you can trigger code instantly with the keyboard, you can apply this skill for our second technique, which is to layer your sounds manually. Instead of 'composing' using lots of calls to **`play`** and **`sample`**, separated by calls to **`sleep`**, we will have one call to **`play`** which we will trigger manually using **ALT+R**. Let's try it. Type the following code into a fresh buffer:

```
synth :tb303, note: :e2 - 0, release: 12,
cutoff: 90
```

Now, hit **Run** and whilst the sound is playing, modify the code in order to drop down four notes by changing it to the following:

```
synth :tb303, note: :e2 - 4, release: 12,
cutoff: 90
```

Now, hit **Run** again, to hear both sounds playing at the same time. This is because Sonic Pi's **Run** button doesn't wait for any previous code to finish, but instead starts the code running at the same time. This means you can easily layer lots of sounds manually, with minor or major modifications between each trigger. For example, try changing both the **`note:`** and the **`cutoff:`** opts and then re-trigger.

You can also try this technique with long abstract samples. For example:

```
sample :ambi_lunar_land, rate: 1
```

Try starting the sample off, and then progressively halving the **`rate:`** opt between hitting **Run**, from **`1`** to **`0.5`** to **`0.25`** to **`0.125`**, and then even try some negative values such as **`-0.5`**. Layer the sounds together and see where you can take it. Finally, try adding some FX.

When performing, working with simple lines of code in this way means that an audience new to Sonic Pi has a good chance to follow what you're doing and relate the code that they can read to the sounds they are hearing.

Language

>RUBY

Above Sam Aaron live coding at Moogfest, USA

Master live loops

When working with more rhythmic music, it can often be hard to manually trigger everything and keep good time. Instead, it is often better to use a **live_loop**. This provides repetition for your code whilst also giving the ability to edit the code for the next time round the loop. They also will run at the same time as other **live_loop**s, which means you can layer them together both with each other and manual code triggers. Take a look at section 9.2 of the built-in tutorial for more information about working with live loops.

When performing, remember to make use of **live_loop**'s **sync:** opt to allow you to recover from accidental runtime mistakes which stop the live loop running due to an error. If you already have the **sync:** opt pointing to another valid **live_loop**, then you can quickly fix the error and rerun the code to restart things without missing a beat.

Use the master mixer

One of Sonic Pi's best kept secrets is that it has a master mixer through which all sound flows. This mixer has both a low-pass filter and a high-pass filter built-in, so you can easily perform global modifications to the sound. The master mixer's functionality can be accessed via the fn **set_mixer_control!**. For example, whilst some code is running and making sound, enter this into a spare buffer and hit **Run**:

```
set_mixer_control! lpf: 50
```

After you run this code, all existing and new sounds will have a low-pass filter applied to them and will therefore sound more muffled. Note that this means that the new mixer values stick until they are changed again. However, if you want, you can always reset the mixer back to its default state with **reset_mixer!**. Some of the currently supported opts are **pre_amp:**, **lpf:**, **hpf:**, and **amp:**. For the full list, see the built-in docs for **set_mixer_control!**.

Use the mixer's ***_slide** opts to slide one or many opts values over time. For example, to slowly slide the mixer's low pass filter down from the current value to 30, use the following:

```
set_mixer_control! lpf_slide: 16, lpf: 30
```

You can then slide quickly back to a high value with:

```
set_mixer_control! lpf_slide: 1, lpf: 130
```

When performing, it's often useful to keep a buffer free for working with the mixer like this.

Practice makes perfect

The most important technique for live-coding is practice. The most common attribute across professional musicians of all kinds is that they practise playing with their instruments – often for many hours a day. Practice is just as important for a live coder as a guitarist. Practice allows your fingers to memorise certain patterns and common edits so you can type and work with them more fluently. Practice also gives you opportunities to explore new sounds and code constructs.

When performing, you'll find the more practice you have done, the easier it will be for you to relax into the gig. Practice will also give you a wealth of experience to draw from. This can help you understand which kinds of modifications will be interesting and also work well with the current sounds.

Bringing it all together

This month, instead of giving you a final example that combines all the things discussed, let's part by setting down a challenge. See if you can spend a week practising one of these ideas every day. For example, one day practise manual triggers, the next do some basic **live_loop** work, and the following day play around with the master mixer. Then repeat. Don't worry if things feel slow and clunky at first – just keep practising and before you know it you'll be live-coding for a real audience.

WESLEY ARCHER

Self-taught Raspberry Pi enthusiast, founder of Raspberry Coulis, and guide writer for Pi Supply and Cyntech.
raspberrycoulis.co.uk
@RaspberryCoulis

Our 7-inch screen mounted on the RaspCade cabinet panel – it looks fantastic when playing a game!

We've added the RaspCade logo to our designs, but you can add your own if you prefer

BUILD YOUR OWN RASPCADE: DISPLAY

PART 03

In this third part of the build, we'll be showing you how to assemble your display as part of your RaspCade homebuild!

You'll Need

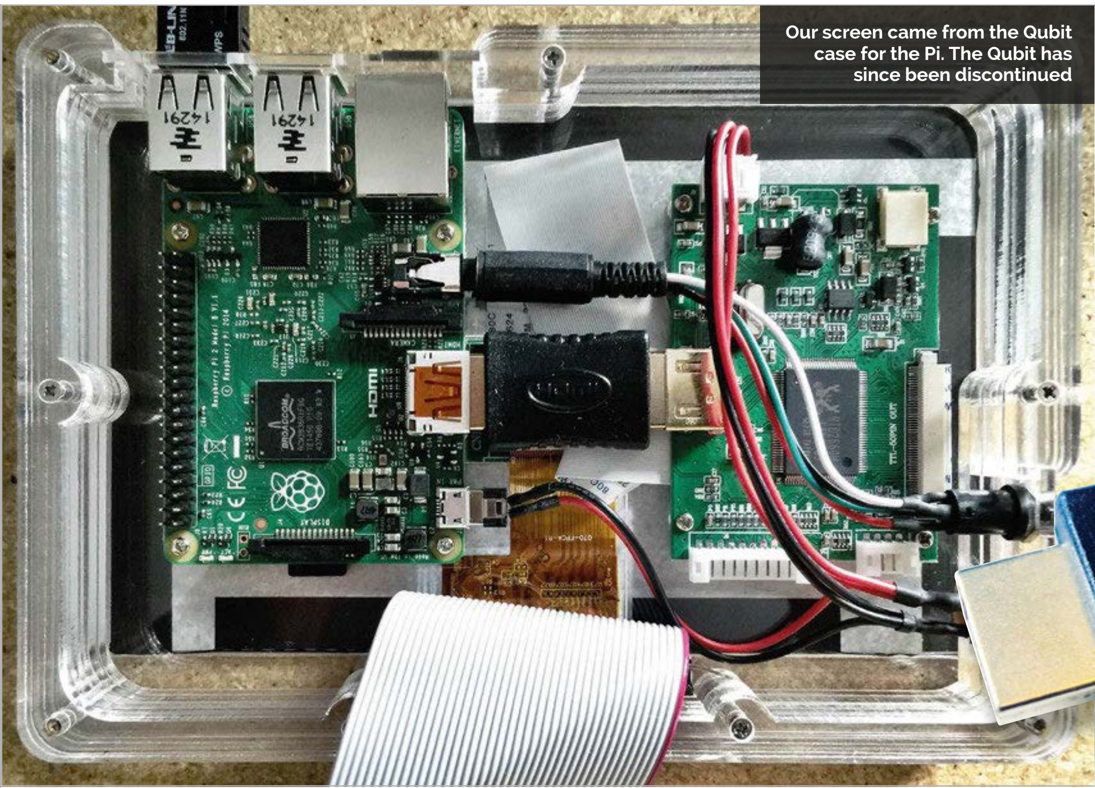
- 7-inch LCD screen (**Search '070-FPCA-R1' on eBay**)
- HDMI LCD driver board (**Search 'PCB800168' on eBay**)
- Male-to-male HDMI coupler or short HDMI lead
- Insulation tape
- 8× standoffs (any size) (**modmypi.com**)

Now that you have your controls sorted, it's now time to get our display up and running as without this, the RaspCade is nothing but a box with fancy controls! You could hook your Raspberry Pi up to a TV, but where's the fun in that? You can use pretty much any display in your RaspCade, but in our build we used a 7-inch screen and the RaspCade cabinet is designed with this in mind. The trickiest part in this guide is mounting the screen, but as you will see, this doesn't need to be elegant. Let's go!

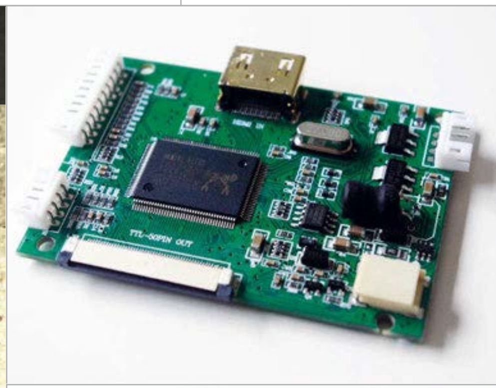
>STEP-01

The screen

The screen we're using was part of a now discontinued case for the Raspberry Pi called the Qubit. It's a 7-inch IPS screen with a maximum resolution of 1024×600, and looks fantastic when up and running. If you'd like to use the same display, then look on eBay for a 070-FPCA-R1 screen and you should find plenty. However, you can use any 7-inch display, as our cabinet has been designed to accommodate screens of this size. Just remember



Our screen came from the Qubit case for the Pi. The Qubit has since been discontinued



that you'll need an HDMI LCD driver board to make your screen work!

>STEP-02
The HDMI LCD driver board

The driver board makes the screen work properly and allows you to connect it to your Pi. Our screen uses the PCB800168 HDMI LCD driver board, which can be found on eBay, although it did come with the Qubit case. The screen connects to the driver board using a 50-pin flex cable (similar to the one on the Raspberry Pi Camera Module), and the driver board connects to your Pi using an HDMI lead. To save space in our build, we used an HDMI coupler adapter, but any HDMI lead will work.

>STEP-03
Mounting the driver board

As we're connecting the driver board to our Raspberry Pi using a HDMI coupler adapter, we need to make sure it's as close to the Raspberry Pi as needed. We connected the HDMI to the Pi and the driver board, then used a pencil to mark eight holes on the bottom panel of our RaspCade, in line with the mounting holes on the driver board and the Pi. We then drilled eight holes in the panel, and used the standoffs to secure both the Pi and driver board in place.

>STEP-04
Mounting the screen

Before you secure the screen in place, it's a good idea to check that it lines up nicely first. We drew around the screen on the back of the RaspCade panel with a pencil when we were happy, so we could line up the screen when taping it in place. We then applied insulation (or electrician's) tape around the edges of the screen, to ensure that it was held securely in place. Once the cabinet is assembled, nobody will see this, so it doesn't have to be perfect! This way, we could keep the design simple and cut out unnecessary parts.

>STEP-05
Powering the screen

As the screen needs to be powered separately to the Pi, the Qubit included a barrel jack connector that also provided power to the Pi, via a micro USB connector: see the close-up of the Qubit to learn how this was done. As we used the barrel jack connector, our power supply also uses this rather than micro USB, but it's relatively simple to convert an existing Pi power supply to barrel jack instead of micro USB if you wanted to. Just be sure to use a decent power supply to ensure you have enough power: we recommend using at least 2.5A.

>STEP-06
Connect everything together

Once you're happy that your screen is mounted in the right place and that your driver board and Pi are too, it's now time to connect everything together. Connect the 50-pin flex cable from your screen to the driver board (the same way you would for the Camera Module), and then connect your HDMI adapter/lead from your driver board to your Pi. You'll then need to connect the power leads (one to the driver board and the other to the Pi), and then connect your power supply. We're almost ready to power up!

Above top This is our HDMI LCD driver board. You'll need a driver board to get your screen working properly

Above We used this HDMI coupler adapter to save space in our build, but any HDMI lead will work

BE CAREFUL WITH THE SCREEN!

The screen is delicate as it doesn't have any case protecting it. Just be careful when handling it.

GOOGLE IS YOUR FRIEND

Don't panic if you get stuck finding what you need. A quick Google search often gives you the answer!

The players move and are your opponents. If you bump into them, they score a point

This is your character; you move it, try to dodge the other players, and score a goal

SCRATCH FOOTBALL

This tutorial will teach you how to impress your footy friends with a game you can make in Scratch

LUKE CASTLE

Luke likes coding and taking old *Space Invaders* games apart. He was in the top 50 for Astro Pi and created Pi fitness app PiMetre.

thepigeek.wordpress.com

You'll Need

- Football player **magpi.cc/2b55HYF**
- Football **magpi.cc/2b560Ts**
- Football pitch **magpi.cc/2b55sg4**

TEMPLATES OR NOT?

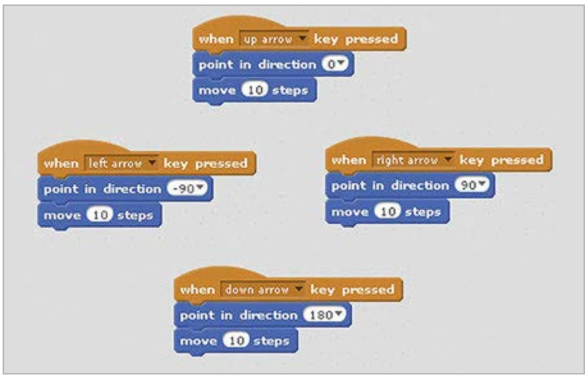
You can create your own ball, players, and/or pitch if you wish, but the code definitely works with the assets provided.

Whether you like playing *FIFA* or coding games, this is the tutorial for you. In it, we will be making a basic football game using colour coding with Scratch. Scratch is a child-friendly programming environment which is developed by MIT. You can create games, music or even apps in the platform, and it's well used in ICT lessons in schools. For this tutorial you will need to follow some easy steps to create a simple but effective game.

Download the assets from the list on the left and import the football pitch. You will need to add three football player sprites; you can choose or create any football costume you want, but make sure they have a yellow outline.

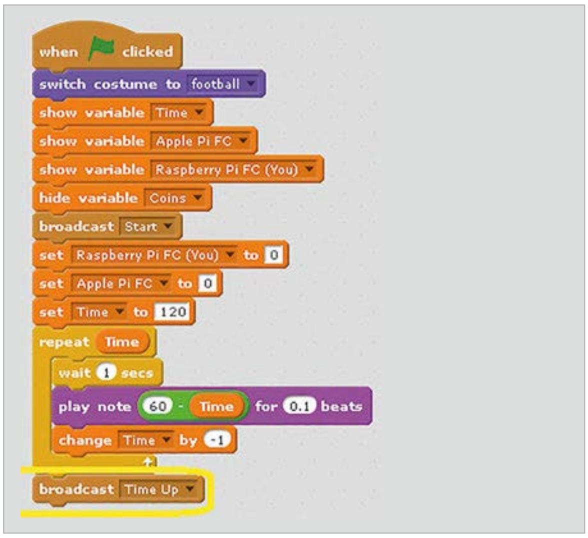
Every single football game needs a football; you can make your own or use the downloadable asset. Now we've finished designing and downloading all of our costumes and backgrounds, we can focus on making the code behind it. The football will be our player, and will be controlled using the arrow keys on your keyboard. To start, you need to get a `when up arrow key pressed` block, then attach a `point in direction 0` block followed by a `move 10 steps`. It should look like this:

You need to do the same for the down arrow, but with a direction of 180. After that, do it again for the left arrow (-90), and finally the right arrow (90) like so:

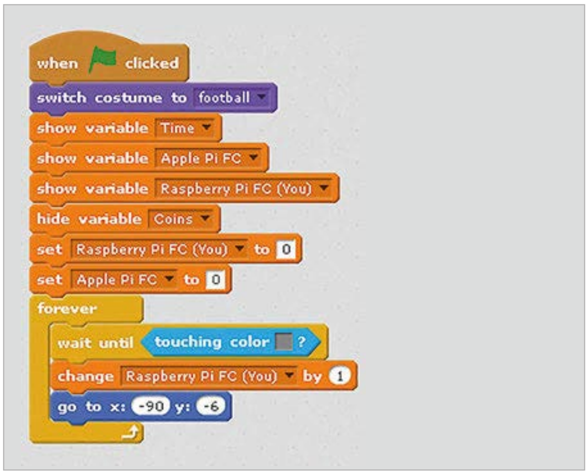


Next we need to create the following four variables: Coins, Raspberry Pi FC (You), Apple Pi FC, and Time. They should all come up in a list; you need to untick the Coins variable so that the players don't see it. You will now need to add the following script:

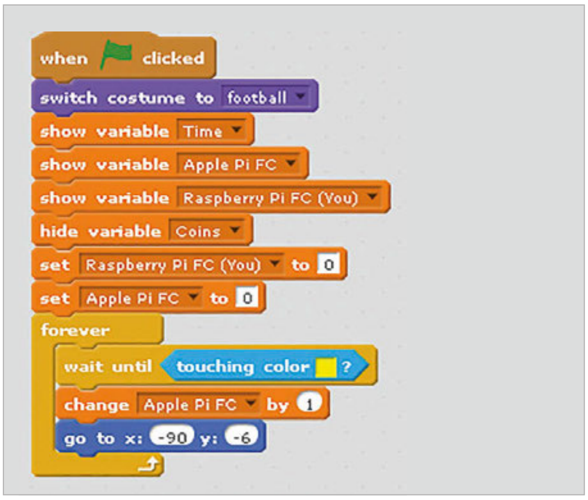
This shows that when the green flag is clicked, it will switch the ball costume to 'football'. It will also show the variables Time, Apple Pi FC, and Raspberry Pi FC (You); however, it will hide the variable Coins. We then broadcast 'Start', which broadcasts a message all around the program and can be used to trigger other parts of the code. After this, we set the variables Raspberry Pi FC (You) and Apple Pi FC to 0. The amount of time for this game is set to 120 seconds, shown in the last block of code above, but you can change that as you wish. You can add sound effects to the game by having a beep play for every second left, which we can set to slowly become deeper. Do this by adding the following to the bottom of the code:



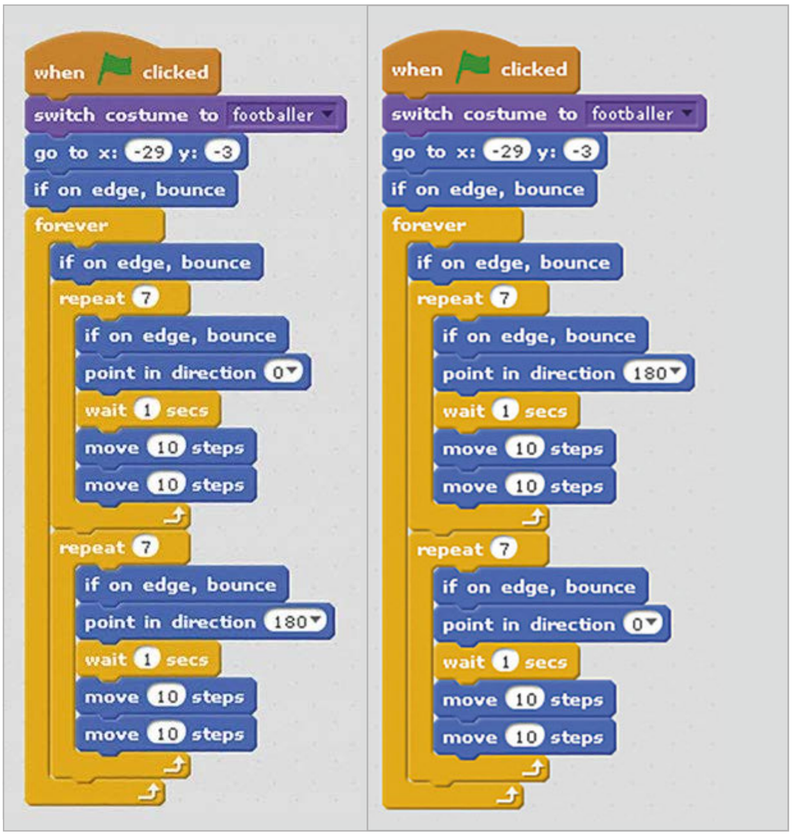
The next script should look like this one: This makes sure that when the football is on the



goal colour (in this case grey), it will add a point for your player (in this case Raspberry Pi FC). It will then reset the football to the middle of the pitch, which is at the coordinates 'x = −90 and y = −6'. You'll need to create another script for the football; this will be the last one for the football, and it will also use the colour coding technique described above.



This script tells the game that if you go out of bounds or hit a player, it should give the opposite team a point. You will then go back to the centre of the pitch at the aforementioned coordinates. Earlier on, we mentioned a yellow outline: the code uses colour coding to tell Scratch that yellow triggers scoring for the opposite team. Now we need to go to our first player and complete their code (below left).



> "It may not be FIFA, but it's a football game you've made all by yourself"

This code will make the football player go up and down. We use angles for this by setting coordinates and then a direction from 0-180.

Now we move on to our other football player: their script is very similar to the first player's, with opposite directions specified for a slightly different obstacle. The code is shown above right. We'll keep the final player stationary, so he doesn't need any code written for him.

And you're done! It may not be *FIFA*, but it's a football game you've made all by yourself. To improve the game, you could add a start menu or add code for the Coins variable, so that you can collect coins and use them to upgrade your football. It's all up to you, and we hope you have fun tinkering!

Language

>SCRATCH

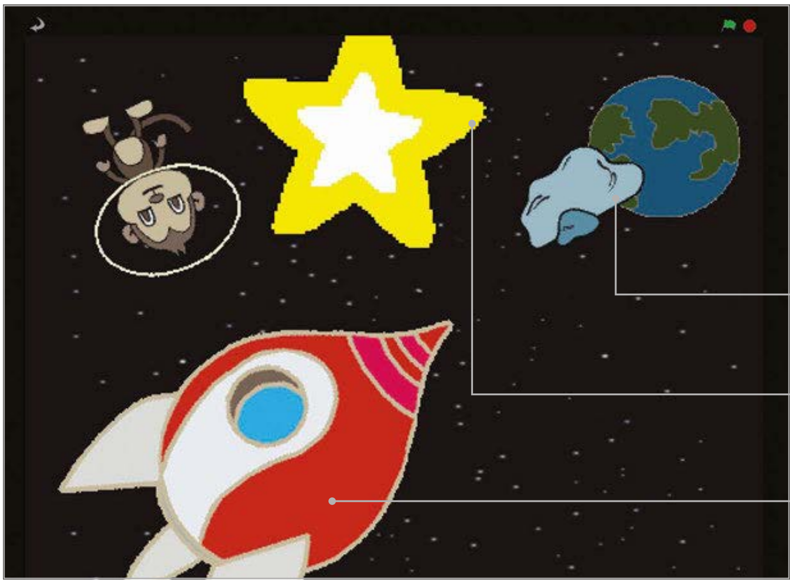
DOWNLOAD:
magpi.cc/2AG9LNF

TWO PLAYERS?

Can you figure out a way to make a two-player mode, perhaps using the stationary player?

PHIL KING

When not sub-editing *The MagPi* and writing articles, Phil loves to work on Pi projects, and to help his six-year-old son learn Scratch coding.
@philking68



This space rock floats around and bounces off the edges of the screen

The star is given a twinkling effect by scaling its size up and down repeatedly

At the start of the animation, the spaceship takes off vertically before being told to point towards the Earth

LOST IN SPACE

Program your own animation of a spaceship heading for Earth, using a scaling effect to make the ship smaller as it moves into the distance

You'll Need

- Scratch
- Art assets **magpi.cc/scratch_art**
- A thirst for space adventure

In this tutorial, you'll be creating an animation sequence which, perhaps unexpectedly, involves a rotating space monkey! This project will show you how to move, rotate, and scale sprites. This is something which will also come in very handy for other projects and games. So, start a new Scratch project and get ready to do some animating. If you need any help navigating the Scratch menus, refer to our Scratch Essentials book listed below.

>STEP-01

Prepare your artwork

After deleting the cat (right-click and Delete), it's time to import a new stage background and sprites. Let's begin by creating our space scene, changing the stage to a field of stars: click Stage in the Sprite List (bottom-right), select the Backgrounds tab (top-middle), then click Import and navigate to 'stars' in the Nature folder. Since none of the sprites used in this project are in the Scratch 1.4 library, you can download them (**magpi.cc/scratch_art**). First, let's import the Earth and Spaceship sprites: for each, click the star/folder above the Sprite List, then navigate to the folder where you've stored your sprites.

Above: In the Paint Editor, draw an ellipse around the monkey's head to give him a space helmet

>STEP-02

Move the spaceship

Click the Spaceship sprite in the Sprite List to select it, then click the Scripts tab. **Listing 1** shows the script you need to add to this sprite to make it move. First, we point it upwards (**`point in direction 0`**) and tell it to **`go to x: -150 y: -150`**, near the bottom-left corner. After waiting one second, we use the handy **`point towards`** Motion block to point it at our Earth sprite. We then use a **`repeat`** loop to keep moving it towards Earth, two steps at a time.

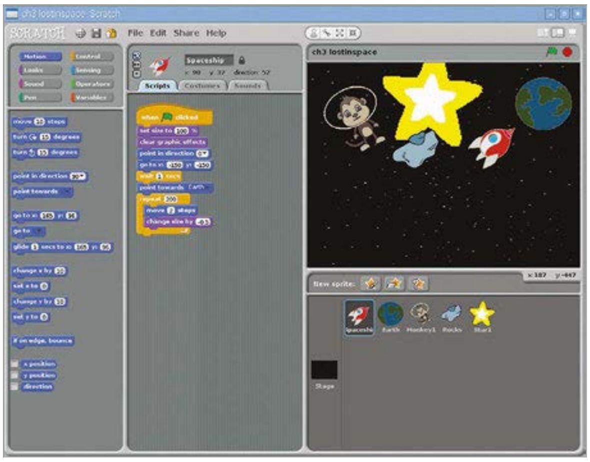
>STEP-03

Scale the ship

To simulate the spaceship moving further away from us, we need to gradually reduce its size as it moves towards Earth. This is easily achieved by adding a single extra block to its existing script. Click the Looks button in the top-left pane, drag a **`change size by`** block and drop it just below your **`move 2 steps`** block, within the **`repeat`** loop. Change the 10 of the

{code club}

This tutorial was adapted from a Code Club project (**codeclub projects.org**) and you can find more in *Learn to Code with Scratch*: **magpi.cc/Scratch-book**



Above: The spaceship points towards Earth, and is gradually moved and shrunk within a repeat loop

`change size` block to -0.5. The code should look like **Listing 2**. Now, try clicking the green flag to see your space rocket hurtle towards Earth, getting smaller all the time.

>STEP-04

Add a space monkey

Now let's add a few extra features to our space scene. For a bit of fun, we'll add a floating monkey who's lost in space. Click on the star/folder icon again and navigate to your Lost In Space sprites folder, then select Monkey. As with any sprite, you can adjust its size using the Grow/Shrink sprite icons above the stage. Now let's give our monkey a space helmet! Select it in the Sprite List, then click the Costumes tab and the Edit button. In the Paint Editor, select the Ellipse tool, the outline option (on the right) below the tools, then a yellow colour from the palette. Now draw a yellow ellipse around the monkey's head for a helmet. To make things more interesting, we'll make our monkey spin around by adding the simple looping script in **Listing 3**.

>STEP-05

Bounce and shine

Finally, we'll add a shining star and bouncing rock. Import them both from your Lost In Space sprites folder, then position and scale them on the stage to your liking. For the star, add the code from **Listing 4** (two **`repeat`** loops inside a **`forever`** one), to repeatedly scale it up and down in size. Add the **Listing 5** code to the rock to get it moving, including a special block to make it bounce off whenever it reaches the edge of the stage.

>STEP-06

Taking it further

Your animation should look pretty cool by now. Try playing around with various parameters to see how they affect the speed, movement, and scaling of the objects. You could also add your own touches, such as using a **`change color effect`** block to give the spaceship a fancy disco-light effect as it moves!

.01

```
when [green flag] clicked
set size to 100 %
clear graphic effects
point in direction 0
go to x: -150 y: -150
wait 1 secs
point towards Earth
repeat 200
    move 2 steps
```

.02

```
when [green flag] clicked
set size to 100 %
clear graphic effects
point in direction 0
go to x: -150 y: -150
wait 1 secs
point towards Earth
repeat 200
    move 2 steps
    change size by -0.5
```

.03

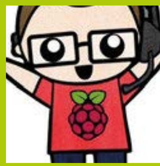
```
when [green flag] clicked
forever
    turn ↻ 1 degrees
```

.04

```
when [green flag] clicked
forever
    repeat 20
        change size by 2
    repeat 20
        change size by -2
```

.05

```
when [green flag] clicked
point towards Earth
forever
    move 2 steps
    if on edge, bounce
```



SPENCER ORGAN

Chemistry and physics teacher, Raspberry Pi enthusiast and Certified RPi Educator from the West Midlands with a passion for running workshops and building fun, educational, and practical things with the Pi!
magpi.cc/2bkQ53q / @ mruktechreviews

PROGRAMMABLE MOTION TIME-LAPSE CAMERA RIG

Take stunning motion-controlled time-lapse frames with your Raspberry Pi and Arduino wherever you go

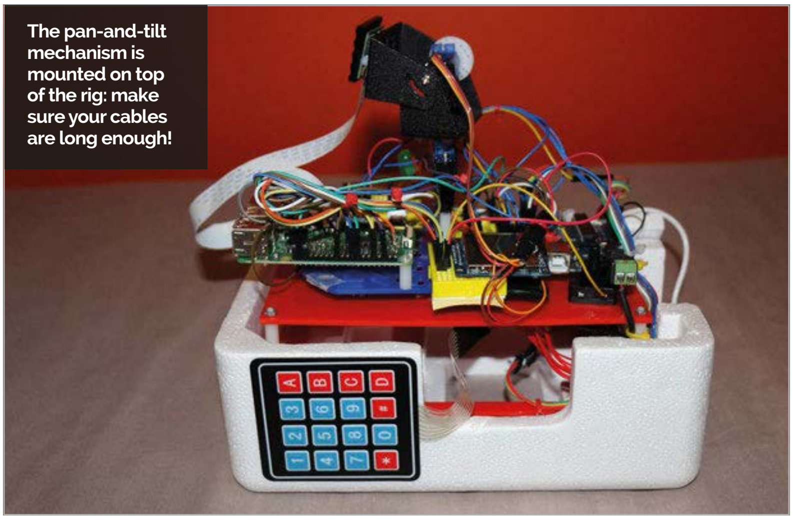
You'll Need

- Camera Module
- Arduino Uno **magpi.cc/2bkQ93a**
- 16×2 character I^{2}C LCD display **magpi.cc/2bkQcMh**
- Nokia 5110 screen (optional) **magpi.cc/2bkQvGZ**
- Adafruit mini pan-tilt kit - assembled with micro servos **magpi.cc/2bkPsa7**
- Membrane 4×4 button pad **magpi.cc/2bkQDpX**
- 5V mini DC relay, LED, 270Ω resistor, male-to-male and male-to-female jumper cables, various breadboards

The small form factor, lower power use and the high-quality camera on the Raspberry Pi makes it an ideal platform for capturing time-lapse frames. In this project, we'll use an Arduino Uno to control the motion of the Raspberry Pi Camera Module and to trigger the photos being taken.

>STEP-01
Connect pan-tilt kit

We start the project by connecting the Adafruit pan-tilt kit to the Arduino. Use a breadboard to connect a common 5V and ground line from the Arduino. Connect the red power cables on each of the servos to the common 5V line, and the brown wire to the common ground. Use jumper cables to connect the orange signal wire of the pan servo to digital pin 9 on the Arduino, and the orange signal wire of the tilt servo to digital pin 8.



The pan-and-tilt mechanism is mounted on top of the rig: make sure your cables are long enough!

>STEP-02
Connect the I^{2}C display

Once we have connected the servos, we can add the I^{2}C LCD display to the Arduino. For this, we'll need four jumper cables. Connect the GND pin to the lower ground connector, then connect the VCC pin to the common 5V line we were using just now for the servo motors. Connect the SDA pin to the analogue A4 connector, and the SCL pin to the analogue A5 connector. We'll need to download the libraries for the I^{2}C LED display for the Arduino. The latest libraries can be downloaded from here: **magpi.cc/2bkO5br**.

>STEP-03
Connect the keypad

Connecting the keypad can be one of the trickiest parts of the build. There's very good documentation for setting up and using a keypad here: **magpi.cc/2baan3b**. We'll also need to download and install the keypad libraries from this page. We have connected the rows to digital pins 5, 4, 3, and 2, and the column pins to 13, 12, 11, and 10. If you find incorrect characters being displayed when you press the keys, you'll need to try reversing the order of the row and column pins; with a bit of trial and error, this should be easy to fix.

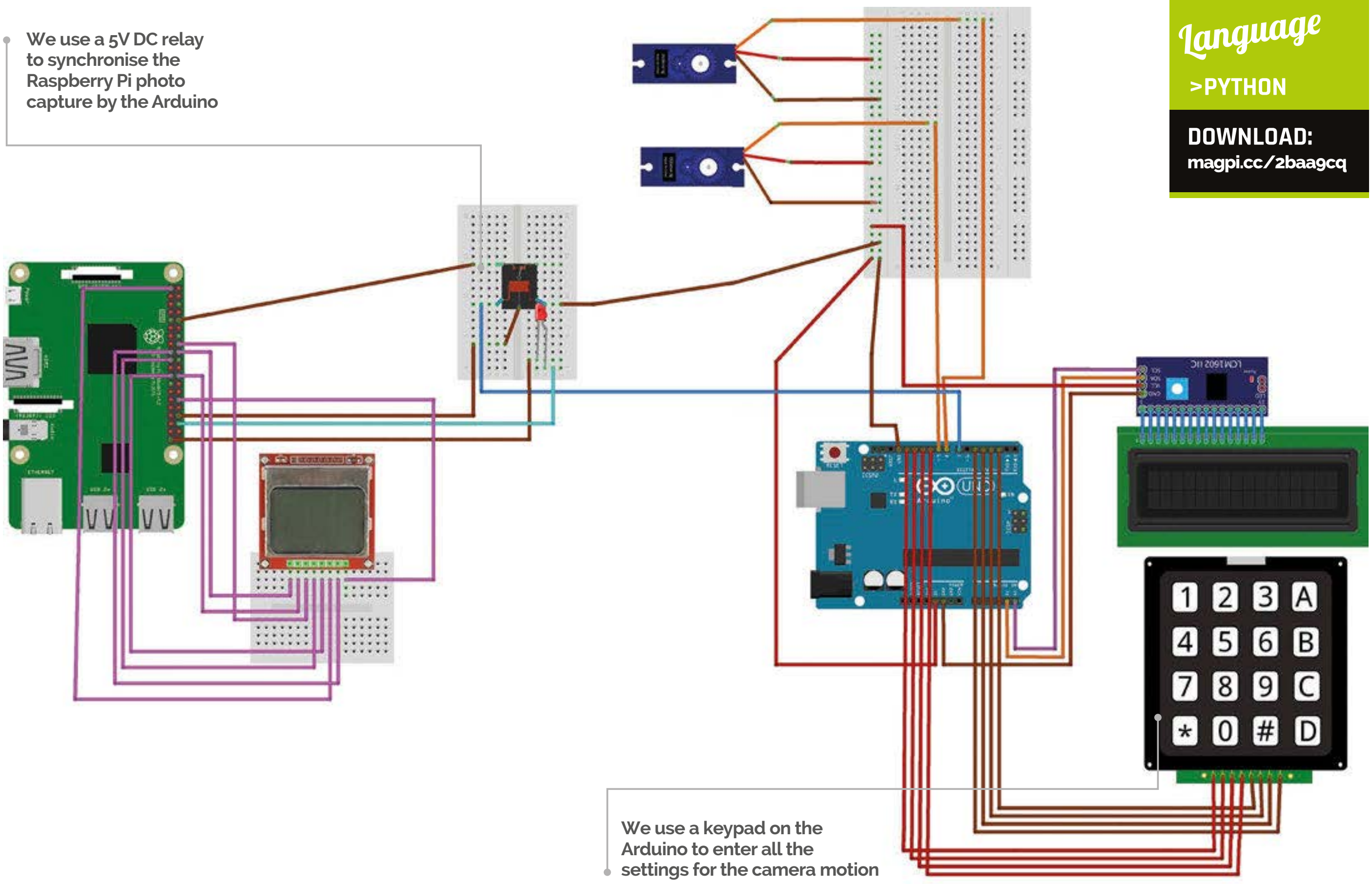
>STEP-04
Mount the Pi camera

Mount the Camera Module onto the pan-and-tilt mechanism. We have found that a longer camera connector cable works better and prevents the camera getting stuck. Connect the other end of the camera connector to the Raspberry Pi, with the printed side of the ribbon cable pointing towards the USB ports.

Language

>PYTHON

DOWNLOAD:
magpi.cc/2baagcq



We use a 5V DC relay to synchronise the Raspberry Pi photo capture by the Arduino

We use a keypad on the Arduino to enter all the settings for the camera motion

>STEP-05

Connect the Nokia 5510 screen (optional)

This optional LCD display allows you to see how many pictures have been taken. There are two versions of this display, so we advise you to check the pin layout and adjust as necessary. Excellent documentation and links to the Python libraries can be found in this PDF guide: **magpi.cc/2bkPU8g**.

LCC VCC is connected to 3.3V
LCD GND is connected to a ground pin
LCD D/C is connected to GPIO 23
LCD RST is connected to GPIO 24
LCD CS is connected to SPI CE0
LCD CLK is connected to SPI SCLK
LCD DIN is connected to SPI MOSI

There is an optional Backlight pin, which can be powered off the 5V from the Raspberry Pi or from the Arduino. If powering it from the Arduino, you'll also need to connect the LCD GND to a ground pin on the Arduino.

>STEP-06

Adding a relay and LED

We now come to the stage were we connect the Raspberry Pi and the Arduino together. The Arduino controls the movement of the servo motors, moving the camera as well as telling the Raspberry Pi when to take a photo. To ensure that the camera isn't moving, we have included a three-second countdown and delay in the Arduino code before each photo is taken. The Raspberry Pi simply waits for a switch to be closed between GPIO 15 and ground. This switch is provided by a small 5V relay powered by digital pin 7 on the Arduino.

Depending on the relay you have purchased, you'll first need to connect the coil to digital pin 7 and ground on the Arduino. The switch part of the relay can then be connected to ground and GPIO 15 on the Raspberry Pi. You should hear a satisfying click every time a photo is taken, as the relay closes and then opens again.

One final step is to add an LED to give a confirmation that a photo has been taken successfully. Connect the longer LED leg (positive) to a 270Ω resistor and then to GPIO 16, and the short leg to ground.

Above Create beautiful time-lapse videos: see an example at: magpi.cc/2bagMOQ

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved...

NEED A PROBLEM SOLVED?

Email **magpi@raspberrypi.org** or find us on **raspberrypi.org/forums** to feature in a future issue.

RASPBERRY PI RASPBIAN SOFTWARE

WHAT IS THE EASIEST WAY TO INSTALL SOFTWARE?

From the repos

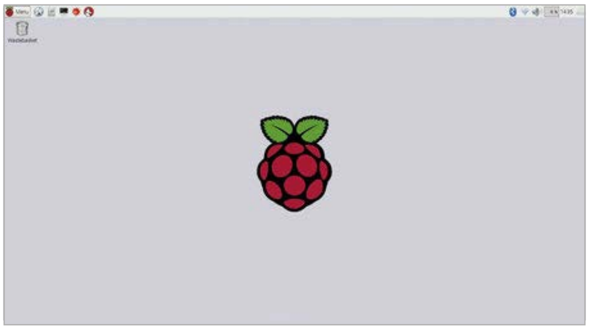
Raspbian, based on Debian, has access to a huge amount of software that can be quickly downloaded and installed to your Raspberry Pi. It's a bit like a mobile phone app store compared to an install wizard.

Find the software

You need to install software via the terminal, but you need to know the package name to install it. For example, Google Chrome is chromium-browser. Use Google to find out the package name of the software you want to use.

Install via the terminal

Once you know what you want to install, you need to open up the terminal or go to the command line and use a command like the following: `sudo apt-get install [package]`. Replace [package] with the name of the software and follow the on-screen instructions to install.



Right Install extra software to improve the Raspbian experience

HOW DO I INSTALL A BINARY?

Binary explanation

Binary files are installable pieces of software that work a lot like Windows or OS X software; in many cases they're exactly the same as the file you'd be installing from the repository, only you need to download and install this one manually.

Getting a binary

Not all binary files will work on Raspbian; they're usually built for specific versions of Linux. Generally, a compatible binary will either tell you it's for Raspbian, or at the very least be a .deb file for ARM systems.

Install the binary

Download the binary to the Raspberry Pi and open up the terminal. From there, use `cd` to move to where the binary was saved and then use: `sudo dpkg -i [binary name]`. Follow this with `sudo apt-get install -f`.

HOW CAN I COMPILE SOFTWARE?

Get the source code

If the software is not available in the repos, or isn't the right version, you can always download the source code and compile the software yourself manually. Put the source code in its own folder in the home folder and make sure it's unzipped.

Follow the instructions

There should be a README file telling you how to compile that particular bit of software. Depending on how big it is and what Pi you use, compiling may take a while. Follow the instructions closely or it may not install properly.

Fixing dependencies

During compiling, it may stop due to the program needing extra software to build. It will tell you what software is missing, which you will then need to install via the repos in the terminal, like we did in the first section. Once they're installed, start the latest step again.

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

What operating system does the Raspberry Pi use?
There are several official distributions (distros) available on our downloads page. New users will probably find the NOOBS installer the easiest to work with, as it walks you through the download and installation of a specific distro. The recommended distro is Raspbian, which is specifically designed for the Raspberry Pi and which our engineers are constantly optimising. It is, however, a straightforward process to replace the root partition on the SD card with another ARM Linux distro, so we encourage you to try out several distros to see which one you like the most. The OS is stored on the SD card.

Does it have an official programming language?
The Raspberry Pi Foundation recommends Python as a language for learners. We also recommend Scratch for younger kids. Any language which will compile for ARMv6 (Pi 1) or ARMv7 (Pi 2/3/Zero) can be used with the Pi, though, so you're not limited to using Python. C, C++, Java, Scratch, and Ruby all come installed by default on the Raspberry Pi.

Will it run Wine, or Windows, or other x86 software?
In general, this is not possible with most versions of the Raspberry Pi. Some people have put Windows 3.1 on the Raspberry Pi inside an x86 CPU emulator in order to use specific applications, but trying to use a version of Windows even as recent as Windows 98 can take hours to boot into, and may take several more hours to update your cursor every time you try to move it. We don't recommend it! As of summer 2015, a version of Windows 10 is available for use on the Raspberry Pi 2. This is an entirely new version of the operating system designed exclusively for embedded use, dubbed the Windows 10 Internet of Things (IoT) Core. It doesn't include the user interface ('shell') or the desktop operating system.

"This is an entirely new version of the operating system"

THE MAGPI APP

Having trouble with *The MagPi* on the App Store or Google Play? Here are your most common questions answered:

How do I find *The MagPi* on Google Play or the App Store?
All you have to do is go to the search bar and type 'The MagPi' or 'Raspberry Pi' to find us.

I've subscribed to the digital edition and I can't sign in to restore my purchases. Please help!
Since your *The MagPi* purchases are linked to your Google or Apple accounts, there's no need to sign in at all. If you'd like to re-download your purchases on your current device, or make your purchases available on other devices, all you need to do is press 'Subscribe' on the home screen, then 'Restore Purchases' on the next screen.

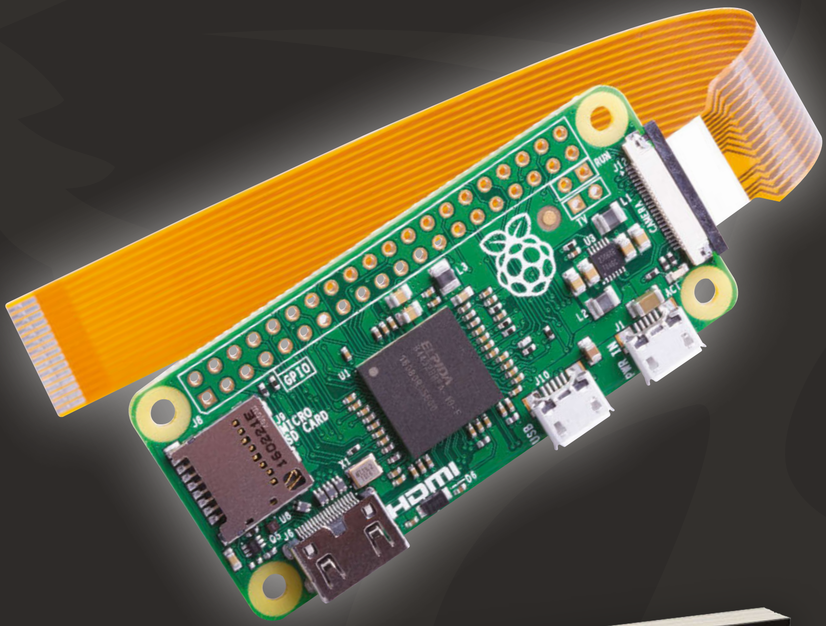
How can I search the digital magazine for keywords?
Finding direct references is really easy with *The MagPi* app: all you have to do is tap the screen to get the app's GUI to show, and then press the small magnifying glass icon in the top-right corner of the screen. Just type in your search term to find the relevant results.

GET IT ON Google Play

Available on the App Store

FREE PI ZERO!

Subscribe in print for six or 12 months to receive this stunning free gift



Subscribe today & receive:

- A free Pi Zero v1.3 (the latest model)
- A free Camera Module connector
- A free USB & HDMI cable bundle

Delivered with your first issue!

Other benefits:

- Save up to 25% on the price
- Free delivery to your door
- Exclusive Pi offers & discounts
- Get every issue first (before stores)

SAVE UP TO **25%**

Pricing

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

FREE PI ZERO!

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

FREE PI ZERO!

Get three issues:

£12.99 (UK) (Direct Debit)

$37.50 (US) (quarterly)

How to subscribe:

- **magpi.cc/Subs1** (UK / ROW)
- **imsnews.com/magpi** (USA)
- Call +44(0)1202 586848 (UK/ROW)
- Call 800 428 3003 (USA)

Available on the App Store

GET IT ON Google Play

SUBSCRIPTION FORM

YES! I'd like to subscribe to The MagPi magazine & save money

This subscription is: ☐ For me ☐ A gift for someone*

Mag#49

YOUR DETAILS Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name .... Surname ....

Address ....

....

Postcode .... Email ....

Daytime phone .... Mobile ....

**If giving The MagPi as a gift, please complete both your own details (above) and the recipient's (below).*

GIFT RECIPIENT'S DETAILS ONLY Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name .... Surname ....

Address ....

Postcode .... Email ....

PAYMENT OPTIONS

1 DIRECT DEBIT PAYMENT £12.99 every 3 issues (UK only)

Instruction to your bank or building society to pay by Direct Debit

DIRECT Debit

Please fill in the form and send to:
The MagPi, Select Publisher Services Ltd,
PO Box 6337, Bournemouth BH1 9EH

Service user number 8 3 8 7 7 3

Name and full postal address of your bank or building society:

To: The Manager Bank/building society ....

Address ....

....

.... Postcode ....

Name(s) of account holder(s) ....

Branch sort code ☐☐☐☐☐☐ Account number ☐☐☐☐☐☐☐☐

Reference ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ (Official use only)

Instruction to your bank or building society

Please pay Select Publisher Services Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit Guarantee. I understand that this instruction may remain with Select Publisher Services Ltd and, if so, details will be passed electronically to my bank/building society.

Signature .... Date ☐☐/☐☐/☐☐

Banks and building societies may not accept Direct Debit instructions for some types of account.

SUBSCRIPTION PRICING WHEN PAYING BY CHEQUE OR CREDIT/DEBIT CARD

6 ISSUES ☐ UK £30 ☐ Europe £45 ☐ Rest of World £50

12 ISSUES ☐ UK £55 ☐ Europe £80 ☐ Rest of World £90

2 CHEQUE

I enclose a cheque for .... (made payable to Select Publisher Services Ltd)

3 CREDIT/DEBIT CARD ☐ Visa ☐ MasterCard ☐ Maestro ☐ Switch

Card number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiry date ☐☐ ☐☐ Valid from ☐☐ ☐☐ (if shown)

Issue number ☐☐ (if shown) Security number ☐☐☐
(last 3 digits on the back of the card)

Signature .... Date ☐☐/☐☐/☐☐

I would like my subscription to begin from issue .... (month + year)

RETURN THIS FORM TO:
MagPi Magazine Subscriptions, Select Publisher Services Ltd, PO Box 6337, Bournemouth BH1 9EH

☐ Please tick this box if you DO NOT want to receive any other information from Select Publisher Services Ltd.

☐ Please tick this box if you DO NOT want to receive any other information from other companies.

☐ Please tick this box if you DO NOT want to subscribe to The MagPi newsletter.

The Saturn V rocket is one of the most powerful vehicles of all time, and necessary for us to get to the moon

APOLLO PI

Emulate the Apollo mission computers on the Raspberry Pi and make your own small step to the moon

The Apollo space programme is the stuff of legend. Six manned moon landings resulting in 12 people walking around on a completely different celestial body to the Earth. The only 12 people in history to have done so – and they did it nearly 50 years ago.

The legacy of these historic missions is felt and revered to this day, and it's safe to say the world would be a different place if they hadn't happened. Will we ever return? Hopefully one day, and it might be our stepping stone to the rest of the solar system and eventually distant stars.

For now, though, let's honour the Apollo legacy by investigating the computers on board these incredible spacecraft, and how we can make our own Apollo computer on a Raspberry Pi.

Buzz Aldrin stands on the moon in one of the most famous photos of all time from a legendary mission

The Apollo code, printed out and stacked, next to Margaret Hamilton who was the director of software programming for the Apollo missions

GET THE SOURCE CODE ON GITHUB!
magpi.cc/2abpPcb

MOON CODE

The computers on the Apollo spacecraft needed programming as well

You've probably heard someone say before how modern pocket calculators are more powerful than the Apollo spacecraft; they're mostly correct, although it's tricky to properly compare. The Apollo Guidance Computer (AGC) was created for the Apollo program, which featured a 1.024MHz clock speed, 16-bit word length, and 2,048 words of RAM. Not bits or bytes, words.

As 'primitive' as it may seem 50 years later, it was powerful enough for the task. Of course, the computer needed more than power and that's where the code comes in. Programmed during the 1960s, the project was fundamental in creating what we know of today as software engineering.

The code is written in assembly, which is a much 'lower-level' programming language to something like Python, but was much more common in the Sixties, when programming computers was a fairly new concept.

It was a marvel for its time. Now, as with all of NASA's work, it's open to the public. While you may have been able to access it in some way for a few years now, the Apollo 11 version of the code is now up on GitHub. Modern-day code collaboration software being used to house and distribute the code that got humans to the moon – an incredible time for computer science.

All images courtesy of NASA

MOONPI

From space to your Raspberry Pi

The code for the moon landings is an amazing piece of history, but what does that have to do with the Raspberry Pi? A couple of years ago, the AGC code was ported to various versions of the operating system Linux to create a virtual AGC that people could use and learn from. It's not a simulator in any sense of the word, but it can give you an idea of how working the computers in space might have gone.

Raspbian, the main Raspberry Pi operating system, is a version of the OS called Debian that has been tweaked to work on the Pi. Debian itself is a popular distribution of Linux and the virtual AGC worked on normal Debian, so getting it working on Raspberry Pi is quite simple! Over the next few pages we'll teach you how to get it working on the computer powering the Astro Pis currently up in space, in homage to the Pi's moon-landing ancestor.

The AGC computer and its control pad. Computers were very different in the Sixties, relying mostly on magnetic ribbon for storage

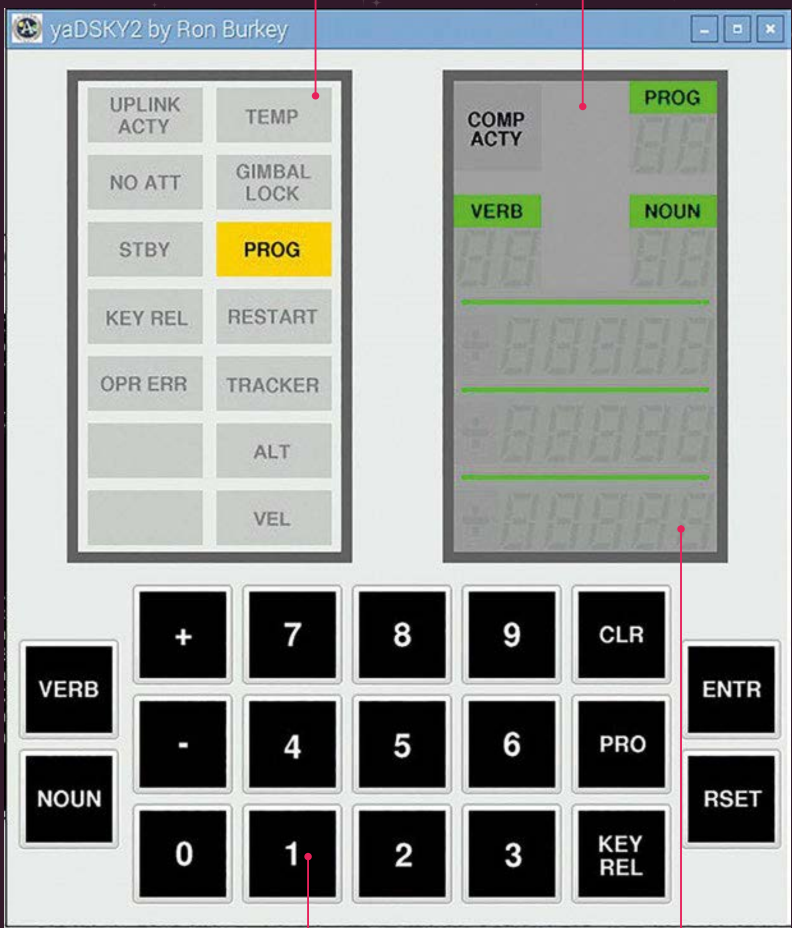
THIS VIRTUAL AGC WAS CREATED BY RON BURKEY: magpi.cc/2b2oasx

SET UP YOUR APOLLO PI

Make your Raspberry Pi ready to pilot a spacecraft straight to the moon

These fixed information indicators light up as different operations where performed by the computer

Programs were run on this display, with the VERB and NOUN and PROG boxes showing what was being run by the astronaut



These fixed information indicators light up as different operations where performed by the computer

The results from the programs were shown on the last three lines – luckily all the results were numbers so they could be easily read

It's actually quite simple to get the virtual AGC running on Raspberry Pi – all we need is a few libraries and the specific code. Luckily, Dave Honess of the Raspberry Pi Foundation has already built the code so we can download it and run it on the Pi without having to build it from scratch ourselves.

>STEP-01

Train up your Raspberry Pi

We'll need the latest version of Raspbian. If you've not reinstalled Raspbian in a while it may be best just to do a fresh install of Raspbian jesse to your SD card. You can find the latest image of Raspbian here: **magpi.cc/1MYYTMo**

If you're installing fresh or not you'll have to make sure your Raspberry Pi is up-to-date. You can do this by opening the terminal and using the following:

```
sudo apt-get update
sudo apt-get upgrade
```

>STEP-02

Launch prep

For the code to work, we need some extra software on Raspbian. You can install this with the following command in the terminal:

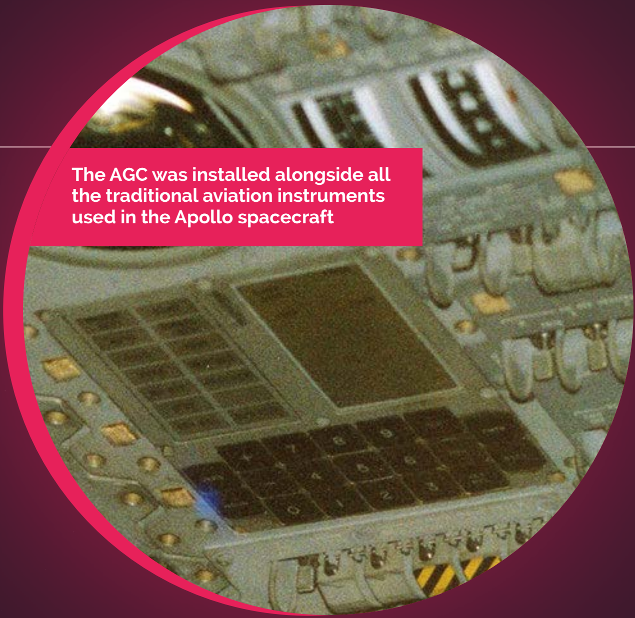
```
sudo apt-get install wx2.8-headers
libwxgtk2.8-0 libsdl1.2debian libncurses5
```

Thw wx2.8-headers and libwxgtk packages allows us to use the graphical interface that's been created for the virtual AGC, which we'll discuss over the page. The libsdl and libncurses packages lets the AGC have better access to Raspbian so it can work properly.

>STEP-03

Ignition sequence

Once everything is installed, it's time to download the code. You can either open a browser on your Raspberry



The AGC was installed alongside all the traditional aviation instruments used in the Apollo spacecraft

and go to **magpi.cc/2b5QZ4B** to get the zip file, or you can download it in the terminal with:

```
wget https://dl.dropboxusercontent.
com/u/14125489/RaspberryPi/agc.zip
```

You'll need to unzip the file once it's downloaded (unzip agc.zip if you're using the terminal). Move it to its own folder in the home directory to make sure it's all nicely contained before unzipping if you wish.

>STEP-04

Blast-off!

This part you need to do in the terminal within the desktop environment. If you're in the command line, use startx and then open a terminal window.

From there use cd to move to the lVirtualAGC folder that you unzipped (e.g. cd IVirtualAGC). After thatm cd into the bin folder within and run the Virtual AGC with:

```
./VirtualAGC
```

The option interface will start up. Select Apollo 11 Command Module, click on Full on the DSKY option in the right hand column, and finally hit Run to use the AGC.

LUNA PROGRAMMING

Operating the AGC is quite different to how we use computers today. Calculations and queries were made using a verb and a noun code – two-digit numbers that told the computer what to do. The verb was the action that the astronaut wanted the computer to do, while the noun was the data that the action needed to be done on. For example, pressing VERB and then 05 followed by NOUN and 09 and then hitting Enter will display (the action) the alarm codes (the data) if there's any problems with the AGC. In short hand this **is referred to as V05N09E.**

IMPORTANT CODES

verbs:

05 Display Octal Components 1, 2, 3 in R1, R2, R3.
06 Display Decimal (Rl or R1, R2 or R1, R2, R3)
25 Load Component 1, 2, 3 into R1, R2, R3.
27 Display Fixed Memory
37 Change Programme (Major Mode)
47 Initialise AGS (R47)
48 Request DAP Data Load Routine (RO3)
49 Request Crew Defined Maneuvre Routine (R62)
50 Please Perform
54 Mark X or Y reticle
55 Increment AGC Time (Decimal)
57 Permit Landing Radar Updates
59 Command LR to Position 2
60 Display Vehicle Attitude Rates (FDAI)
63 Sample Radar Once per Second (R04)
69 Cause Restart
71 Universal Update, Block Address (P27)
75 Enable U, V Jets Firing During DPS Burns
76 Minimum Impulse Command Mode (DAP)
77 Rate Command and Attitude Hold Mode (DAP)
82 Request Orbit Parameter Display (R30)
83 Request Rendezvous Parameter Display (R31)
97 Perform Engine Fail Procedure (R40)
99 Please Enable Engine Ignition

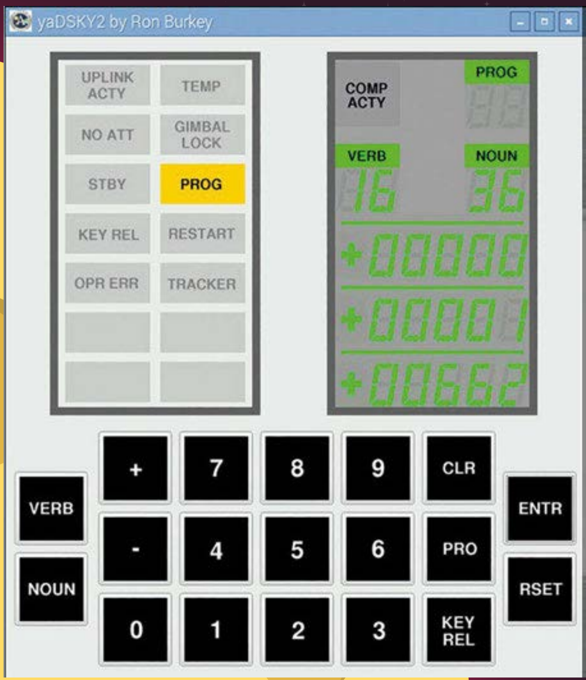
nouns:

11 TIG of CSI
13 TIG of CDH
16 Time of Event
18 Auto Maneuvre to FDAI Ball Angles
24 Delta Time for AGC Clock
32 Time from Perigee
33 Time of Ignition
34 Time of Event
35 Time from Event
36 Time of AGC Clock
37 Time of Ignition of TPI
40 (a) Time from Ignition/Cutof
(b) VG
(c) Delta V (Accumulated)
41 Target Azimuth and Target Elevation
42 (a) Apogee Altitude
(b) Perigee Altitude
(c) Delta V (Required)
43 (a) Latitude (+North)
(b) Longitude (+East)
(c) Altitude
44 (a) Apogee Altitude
(b) Perigee Altitude
(c) TFF
45 (a) Marks
(b) TFI of Next/Last Burn
(c) MGA
54 (a) Rang
(b) Range Rate
(c) Theta
61 (a) TGO in Braking Phase
(b) TFI
(c) Cross Range Distance
65 Sampled AGC Time
66 LR Slant Range and LR Position
68 (a) Slant Range to Landing Site
(b) TGO in Braking Phase
(c) LR Altitude-computed altitude
69 Landing Site Correction, Z, Y and X
76 (a) Desired Horizontal Velocity
(b) Desired Radial Velocity
(c) Cross-Range Distance
89 (a) Landmark Latitude (+N)
(b)Longitude/2 (+E)
(c)Altitude
92 (a) Desired Thrust Percentage of DPS
(b) Altitude Rate
(c) Computed Altitude

APOLLO 11 MADE ITS HISTORIC MOON LANDING AT 20:18:04 ON 20 JULY 1969!

MOON TIME

Check the time since launch and set yourself up an Apollo clock on your Raspberry Pi



Right: The hours, minutes, and 100ths of seconds are listed on the display

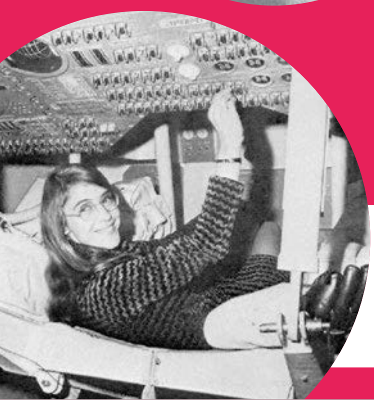
One of the most basic functions of the AGC was for the computer to keep track of the time. It was also an important function, aiding with mission planning and also figuring out if it's too early in San Francisco to give someone a call.

The virtual AGC keeps track of time since launch, or in this case time since the AGC was turned on. We can check this time by keying in **V16N36E**. From the list of codes, this means we're asking for the time (V16) of the AGC clock (N36). You might see this split into LGC, which is the lunar guidance computer that would have been the computer in the Lunar Module, or CGC which is the Command Module's computer. Both use the same AGC hardware and code.

After typing in the code, you'll get three lines of numerical readouts. The top display will be hours, the second display is the minutes, and the third display is in 100ths of a second. The display is updated by the second, so you don't need to keep repeating the code to keep an eye on the time since launch.

CODING IN THE SIXTIES

Pioneering software engineering at the dawn of computing



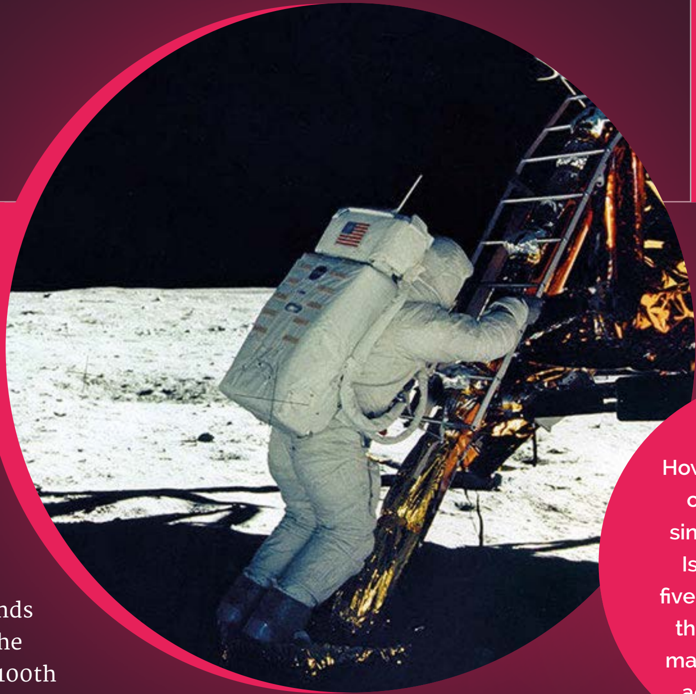
Top: James Lovell (of Apollo 13) can be seen here taking a star reading during Apollo 8 – next to him is the AGC's control pad

Left: Margaret Hamilton, then lead software designer on Apollo, tried to give legitimacy to software engineering in a time when it was looked down upon

The Apollo missions were a huge undertaking and the brightest minds in the United States were called upon to help on any relevant area. This means when the computer was to be designed and programmed, NASA went to MIT. In 1962, the project began and paved the way not only for modern computers but also modern software.

In the Sixties, the term software was not as widespread as it was today – it was only really known to those who made it or were very close to the projects that required it. Coming off the back of older computers, the concept of software to the hardware engineers was foreign and distrusted as it wasn't a physical thing they could see, even if it was a fundamental necessity.

The whole thing was written in assembly language, as discussed earlier, but many new programming techniques were invented to make sure the whole thing would work. Software could



A giant leap that has a legacy which can still be felt nearly 50 years later
Image courtesy of NASA

SPACE CLOCK

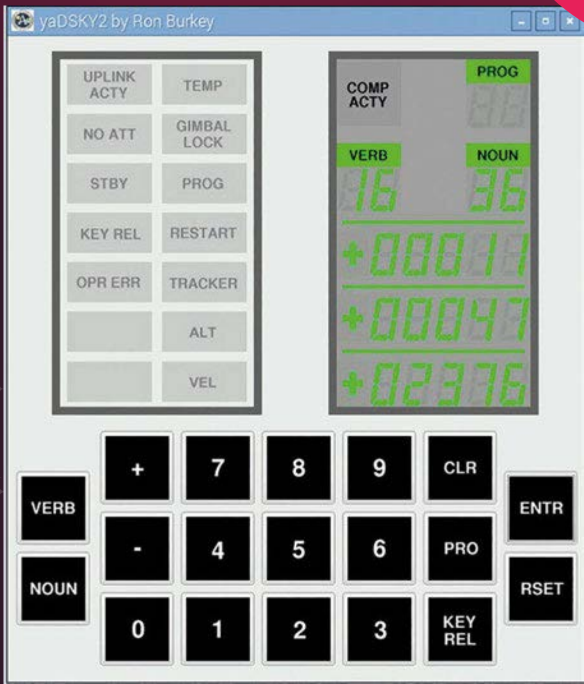
The AGC is a programmable computer, so it stands to reason we can reprogram the clock to show the current time. N36 can be modified down to the 100th of a second and we can modify it using V25; this verb allows us to load a component (change the number) in the readout of the noun, in this case the clock.

On the AGC, use **V25N36E** and the top line (R1) will clear and you can change it to be the current hour by pressing **+** on the virtual keyboard and then using the numpad to key in the time. If you make a mistake, you can press **CLR** to start again, but once you're happy you can press **ENTER** and move onto the middle line (R2) and set the minutes the same way. Remember, for the seconds it's in 100th of a second increments so 5 seconds would 500, 10 would be 1,000, etc.

Use **V16N36E** to display the current time from this edited state. This will update every second like it did before and allow you to use the AGC as a clock. With a smaller screen and some inventive setting customisation, you can make it your main clock somewhere in your house. If you want to find out the time since bootup, you can always use a different key combination of **V25N65E**, and then return to your clock with **V16N36E**. When you restart the AGC, you'll need to reset the clock, though.

Challenge!

How would you go about calculating the time since Apollo 11 landed? Is it possible with the five-digit display to count that many hours? How many years can the AGC actually count up to?

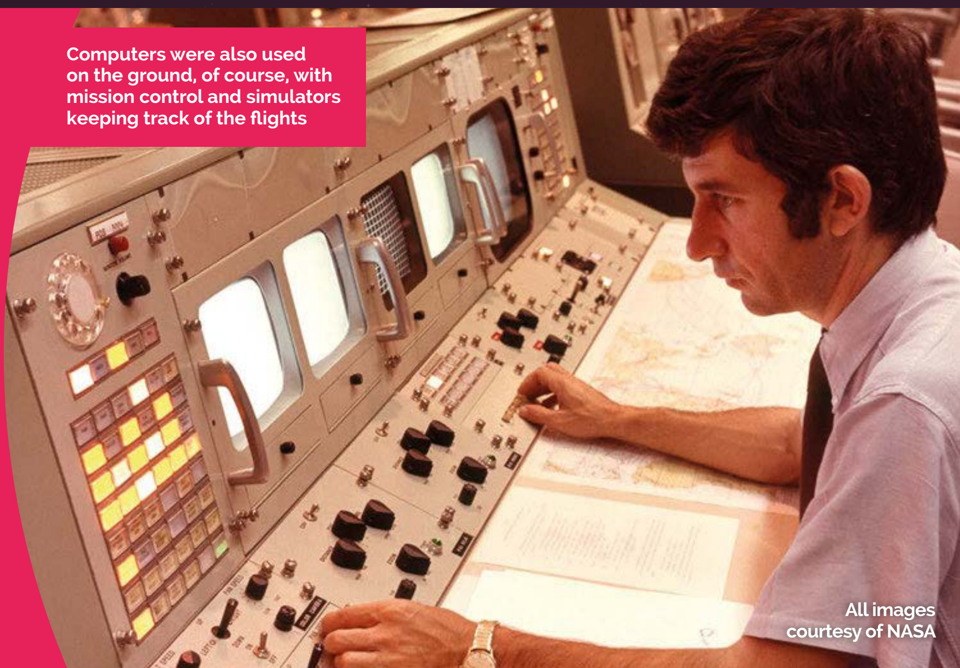


Left: The clock is set to your specific time. It's not a 24-hour clock, though, so you may need to reset it sometimes

be run asynchronously, and a priority scheduler allowed tasks for the computer to be executed when they were needed.

These innovations were key to the successful landing of Apollo 11 on the moon: due to faulty power supplied to the Lunar Module's rendezvous radar (for the return journey), the AGC was overloaded with interrupts and an abort was nearly made. Due to the scheduling system and asynchronous program running, the computer was able to cope with the extra load, resulting in Apollo 11 landing safely on the surface of the moon.

The software was continually updated and worked on throughout the rest of the Apollo missions. To work around the limitations, many little tricks were employed and in some cases the readability of the code suffered – a great reminder to always document your code!



Computers were also used on the ground, of course, with mission control and simulators keeping track of the flights

All images courtesy of NASA

The crew of Apollo 13 required a special startup process for their Command Module computer during the final stages of their fateful return home

SPACE TESTS

Perform the vital tests needed to start up your AGC and get to the moon

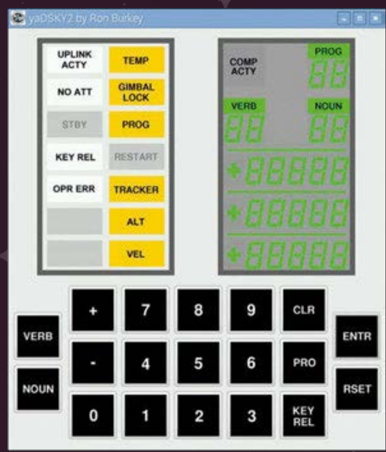
Consider the situation – you've just launched into space on a Saturn V rocket on your way to the moon. Your spacecraft has performed its docking operation between the Lunar Module and the Command Module and you're well on your way. This is when you need to check to make sure your computer is working properly – you don't want any problems when you're 100,000 miles from the nearest layby. Check the status of your on-board computer by using Apollo 13 Lunar Module and then follow these steps:

>STEP-01

Lamp check

CODE: V35E

There aren't any LEDs as this is 1969, so to start the test we need to turn all the indicator lamps on to make sure they're all working. If one is burnt out and it will tell you something important, you need to know.

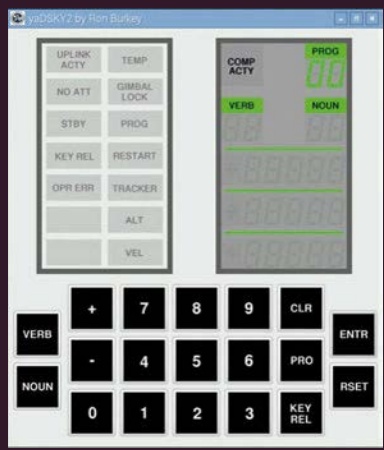


>STEP-02

Start the main program

CODE: V37E, 00E

Program P00, affectionately named Pooh after the bear, is one of the main programs for the AGC. PROG in the top right should show 00; this means your software is initialising and ready to work.



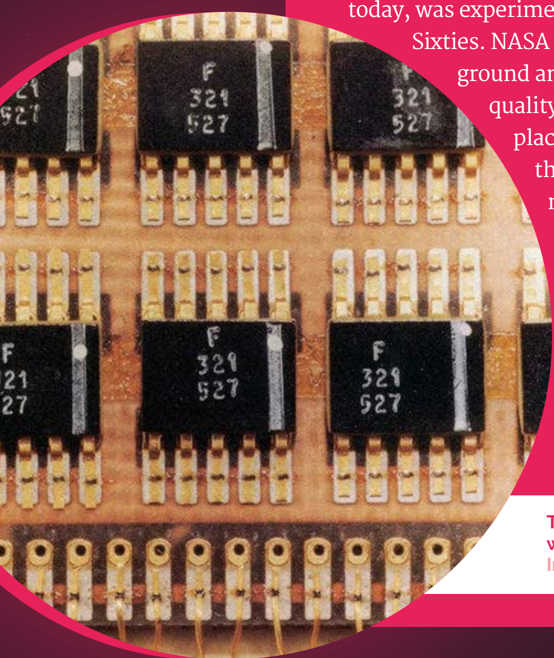
COMPUTERS IN THE SIXTIES

Revolutionising hardware to send men to the moon

One of the biggest problems with sending men to the moon was that all the equipment needed to do it didn't exist yet. There were no rockets powerful enough to get that far and computers were the size of a room. In order to fit a computer into the confines of a very tightly designed spacecraft, a new technology needed to be invented: the microchip.

Fairchild Semiconductor, which still operates today, was experimenting with the idea in the Sixties. NASA was keen to get them off the ground and make sure they were high quality and well researched, so they placed an order for a million of them, knowing they would only need a few hundred.

It worked and the integrated circuits were able to reduce the size of the computer down by a sizeable amount, allowing it to be small and light enough to fit in the craft and not hamper the flight to the moon.

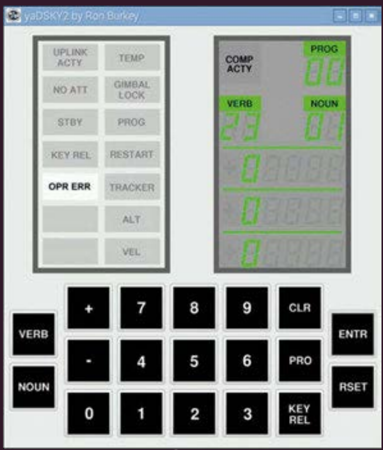


These microchips used in the AGC were some of the first
Image courtesy of NASA

>STEP-03

Error counting

CODE: V25E, N01E, 01365E, 0E, 0E, 0E

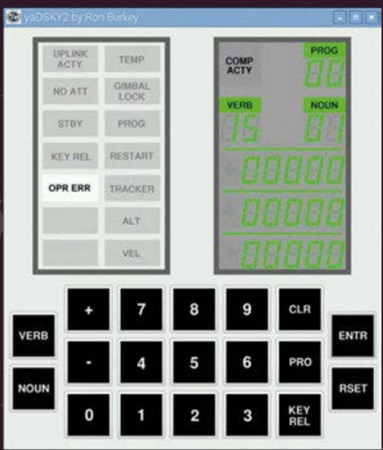


Before we begin the tests, we need to set the count of total failed self-tests, total started self-tests, and successfully completed division tests to 0. We want to make sure we know exactly how many errors we get in this test alone.

>STEP-04

Monitor the test

CODE: V15, N01E, 01365E

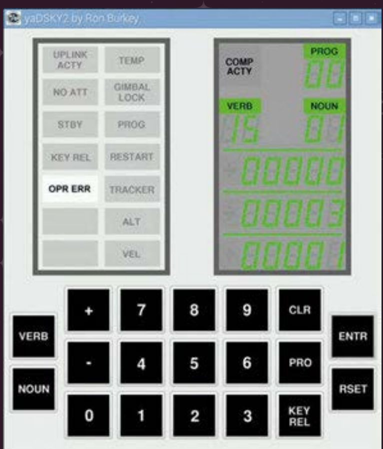


We've reset the counts; now we get ready to monitor the tests. We have to set up the three lines of output to do this first. The first row (R1) shows the number of failed tests, R2 displays how many test have actually been made, and R3 shows the number of completed division tests.

>STEP-05

Begin the tests

CODE: V21N27E, 10E

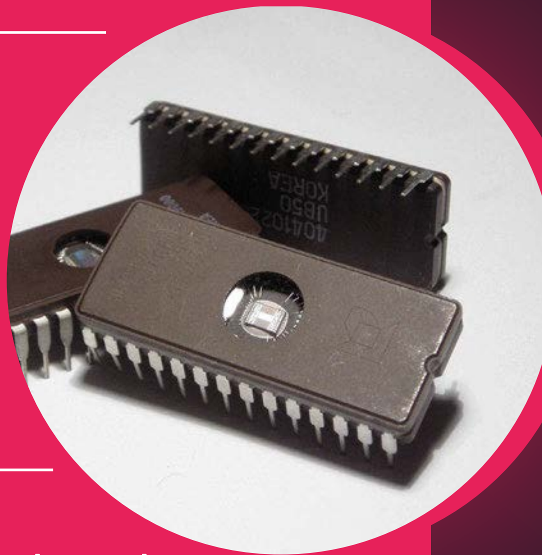


The tests will start and go through the computer. These will continue on as long as you want them to and you can stop them with **V21N27E** followed by **0E**. Hopefully your computer will be fine and you'll be on your way to the moon!

APOLLO COMPONENTS USED TODAY

Integrated circuits

These were a revolution at the time, heralding a new future for computers. These are still widely used in almost all electronics in varying ways. You can also use a few with the Raspberry Pi on a breadboard, such as an analogue-to-digital converter chip.



Number pad

The calculator-style interface on the AGC was the first of its kind to use a number pad. As well as the calculators it inspired, you can see a very similar evolution of it on the number pad found on the side of a full computer keyboard.

Image courtesy of NASA

Seven-segment display

While it didn't change the world like the integrated circuit did, the seven-segment display for showing numbers is still used today – in fact, patents for it go back to 1908. So when you plug one into your breadboard, remember this is a technology that is a century old!

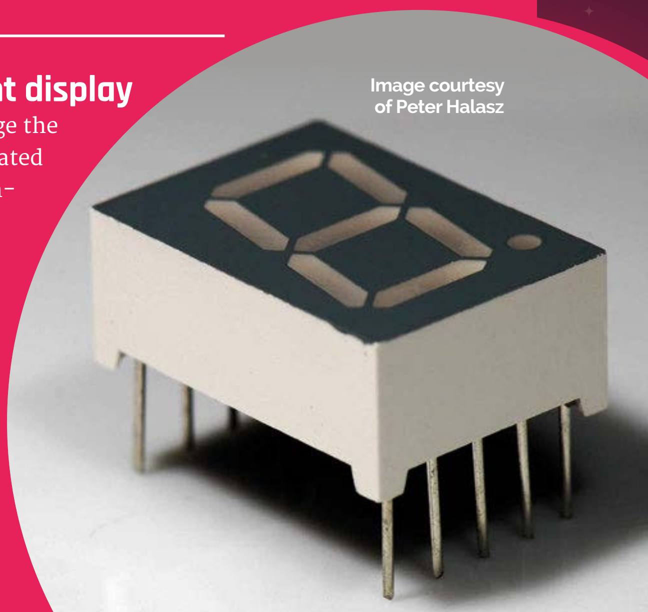
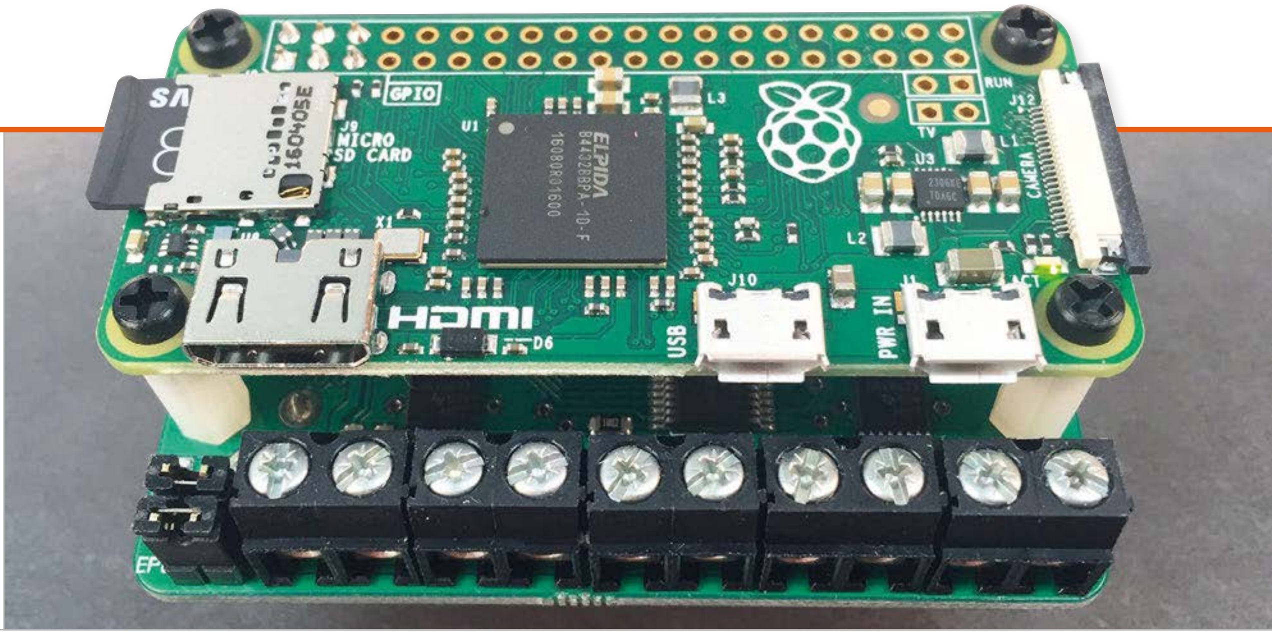


Image courtesy of Peter Halasz



Maker Says

"Opens up endless possibilities for tiny robot designs"

PiBorg

ZEROBORG

Control four motors independently with this versatile Zero-size board from the robot experts at PiBorg

Raspberry Pi robotics specialists PiBorg have turned their attention to the Pi Zero and the possibilities of using it to make very small robots. The result is the ZeroBorg, a diminutive motor controller board that's only marginally wider than the Zero itself. When mounted to the rear of the Pi Zero, the whole setup (including optional 9V battery) weighs a mere 65g. It's so lightweight and nifty that PiBorg are using it to control the YetiBorg racing robots in their upcoming Formula Pi series: see this issue's news section for more details.

The inclusion of four H-bridges means that the ZeroBorg can control four standard DC motors independently. Add some special Mecanum wheels and you can get your robot to scuttle sideways like a crab! Even when using standard wheels, the ZeroBorg offers extra control since the bidirectional PWM (pulse-width modulation) signal sent to each of the four wheels can be varied precisely. Each H-bridge can deliver 2A peak or 1.5A RMS current, so it should work with most small motors. Alternatively, the board can be used to run two four-, five-, or six-wire stepper motors.

Stacks of fun

One curious aspect of the ZeroBorg is that it's designed to be connected to a Pi Zero that has an unpopulated GPIO header. Instead, it's supplied with a small female header to fit to the rear of the Zero, at the 3V3 end of the GPIO header; into this you slot the ZeroBorg's six pins, two of which connect to SDA and SCL for I²C communication. Now, while it's possible to do this without soldering the small header to the Pi Zero, and instead simply holding the two units together firmly using the supplied standoff screws, we were unable to get this method to provide a reliable enough connection. Once we'd soldered the header to the Pi Zero, however, everything worked absolutely fine, so we'd strongly advise doing this. Alternatively, if your Zero already has a full GPIO male header attached, you could always use two 3-pin female-to-female connectors to connect it; this method would also enable you to use the ZeroBorg with any other Raspberry Pi model.

It's important to note that the ZeroBorg comes in three main versions. While the basic KS1 model comes pre-assembled, the KS2 adds a DC/DC regulator and battery clip (supplied loose or pre-soldered) so that the ZeroBorg, motors, and Pi Zero

Related

MOTOZERO

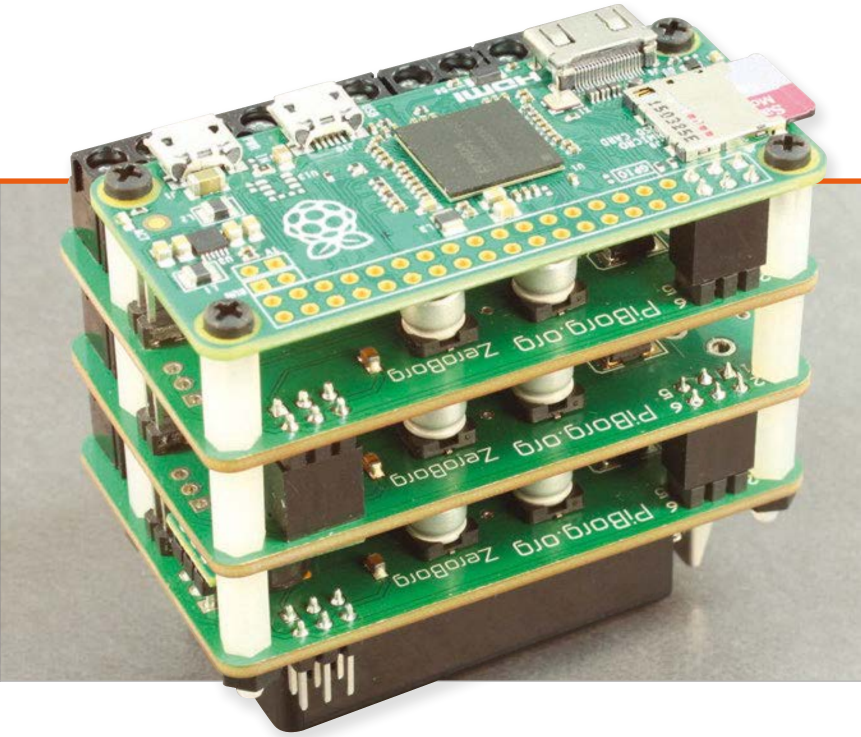
Resembling an exposed engine, it can control four motors independently, though it lacks any sensor inputs.

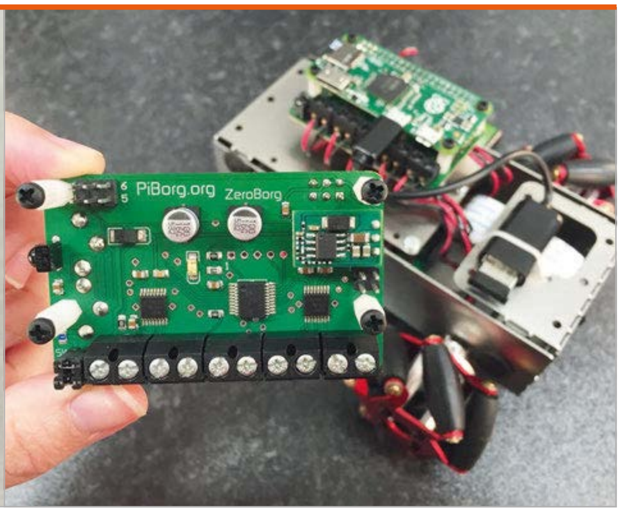
£10 / $13

magpi.cc/1XRfqGQ

magpi.cc/2aFKJ60

From £18 / $23





Above left The ZeroBorgs are designed so they can be stacked on top of one another

Above Build full robots with the tiny ZeroBorg

can all be powered by a standard 9V PP3 battery. Alternatively, an external power source such as a battery pack can be attached to two of the ZeroBorg's terminals, enabling you to mount it flat. The KS2 model also includes an infrared sensor (more on that later) and a second six-pin male I²C header for daisy-chaining with other add-on boards, including the UltraBorg, PicoBorg Reverse, or another ZeroBorg. Indeed, the KS3 option comprises a stack of three ZeroBorgs, the middle of which features two female I²C headers to allow communication between the three boards. While overkill for your average robot, this version could prove particularly useful for animatronics projects or running multiple servos in a CNC machine, for instance. All ZeroBorg models also include two analogue inputs (plus power and ground) for attaching sensors.

Motoring on

We tested a pre-soldered KS2 ZeroBorg for this review, so all we needed to do was solder the female header to the Pi Zero, screw in the standoffs, insert the battery, and we were ready to roll. Well, almost. First, you need to ensure I²C is enabled on the Pi, then install the ZeroBorg software using a single terminal command. It's then just a matter of wiring up your motors as usual; the terminals are all located on one edge of the ZeroBorg, which isn't quite as intuitive as on the rival MotoZero, but they're nice and chunky so they should prove durable. In addition, the ZeroBorg features short circuit prevention to prevent any damage from incorrect connections, along with overheat protection, under-voltage lockout, and a fast-blow 5A fuse.

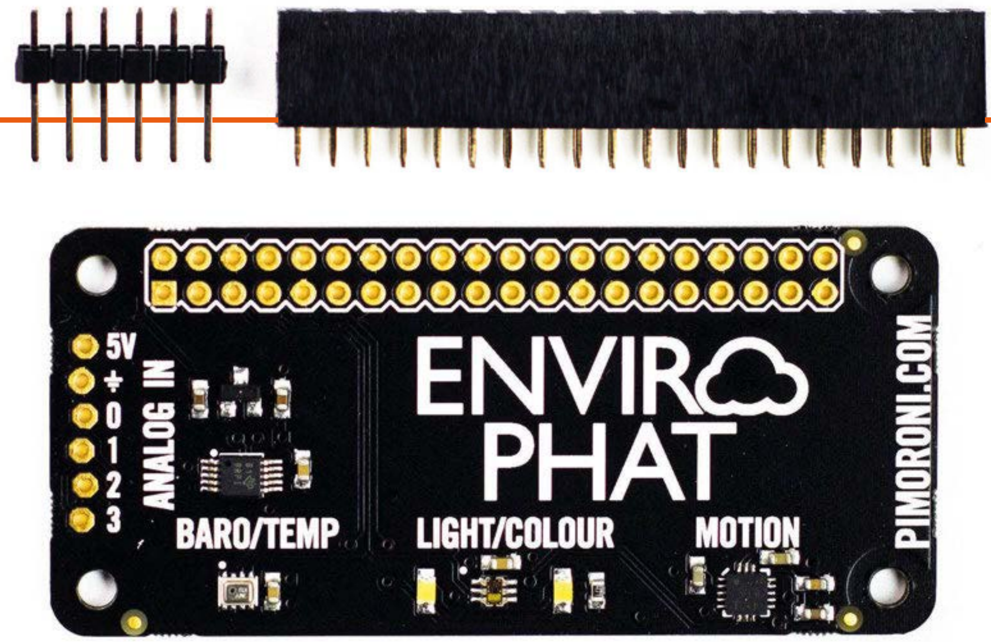
> "ZeroBorg software includes a special Python library, along with numerous examples"

The ZeroBorg software includes a special Python library, along with numerous examples to get you started. It's all fairly straightforward: for example, the `ZB.SetMotor1(1)` command is used to supply maximum speed to motor 1. Use a lower number for less power, zero to stop it, and a negative value to reverse. The examples include joystick control, stepper motor sequence, analogue inputs, and control using an infrared TV remote; if yours isn't supported by default, it's easy to record and save the raw IR codes and add them to the main script. We were soon using a TV remote to control our swiftly assembled 'Tubbybot', made from a small plastic storage tub to which we strapped four micro metal-gear motors and wheels. While not the fastest off the blocks, Tubbybot was able to do some nifty spin-turns by powering one pair of wheels forwards while reversing the others.

Last word

While its connection method is a little unorthodox, the ZeroBorg is a mini marvel for motor control. The ability to power both the motors and Pi Zero using a single 9V battery should prove particularly useful when designing small robots, while the daisy-chaining options offer extra flexibility for other possible uses.

★★★★★



Maker Says

It's ideal for monitoring conditions in your house, garage or galleon

Pimoroni

ENVIRO PHAT

This Zero-size add-on features four built-in sensors plus analogue inputs

While not an official Raspberry Pi standard, Pimoroni's pHAT class of half-size add-on boards are great fun and match the Pi Zero's form factor perfectly, although they'll work with any 40-pin Pi model. The latest addition to the line is the Enviro pHAT, which is all about taking environmental and motion measurements. Along with several built-in sensors, it features four analogue input channels to connect your own external sensors. In effect, the Enviro pHAT is Pimoroni's Flotilla weather, colour, and motion modules rolled into one, with the addition of an analogue-to-digital (ADC) converter.

First things first: the Enviro pHAT comes in kit form, so you'll need to get your soldering iron out to attach the 2×20-pin female header and six male pins for the analogue inputs. Alternatively, you could even solder the pHAT straight onto the GPIO pins of a Pi Zero, if you wanted to use them together as a permanent room-monitoring or motion-measuring device.

Once the pHAT is assembled and mounted on the Pi's GPIO header, installing the software requires just a single command in the terminal. Assuming your Pi already has I²C enabled, you're then able to start coding to obtain readings from the sensors, using the pHAT's own Python library. The latter is partitioned into five separate modules: **`light`**, **`weather`**, **`motion`**, **`analog`** (inputs), and **`leds`**.

Modular sensors

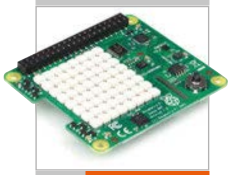
The **`light`** module offers two main methods for reading the built-in TCS3472 sensor, which monitors four different values: clear, red, green, and blue. As well as an ambient light level reading using **`light.light()`**, you can obtain RGB colour values with **`light.rgb()`**, for a tuple which can easily be split into separate values. As you can see, the function naming structure used by the library couldn't be simpler, so it's all very easy to code. To aid accuracy of colour readings, the board has two small white LEDs located on either side of the light sensor, which can be switched on and off using the **`leds`** Python library module. Even so, the colour values produced are for a duller shade than the real item analysed, so may require some calibration.

The library's **`weather`** module enables you to obtain temperature and barometric pressure (in hPa) readings from the Enviro pHAT's BMP280 sensor, but it doesn't measure humidity. Since the sensor is mounted on the PCB rather than remotely, its temperature reading is greatly affected by the heat of

Related

SENSE HAT

As used in the Astro Pi devices aboard the ISS, the Sense HAT features multiple built-in sensors and an 8×8 LED matrix display.

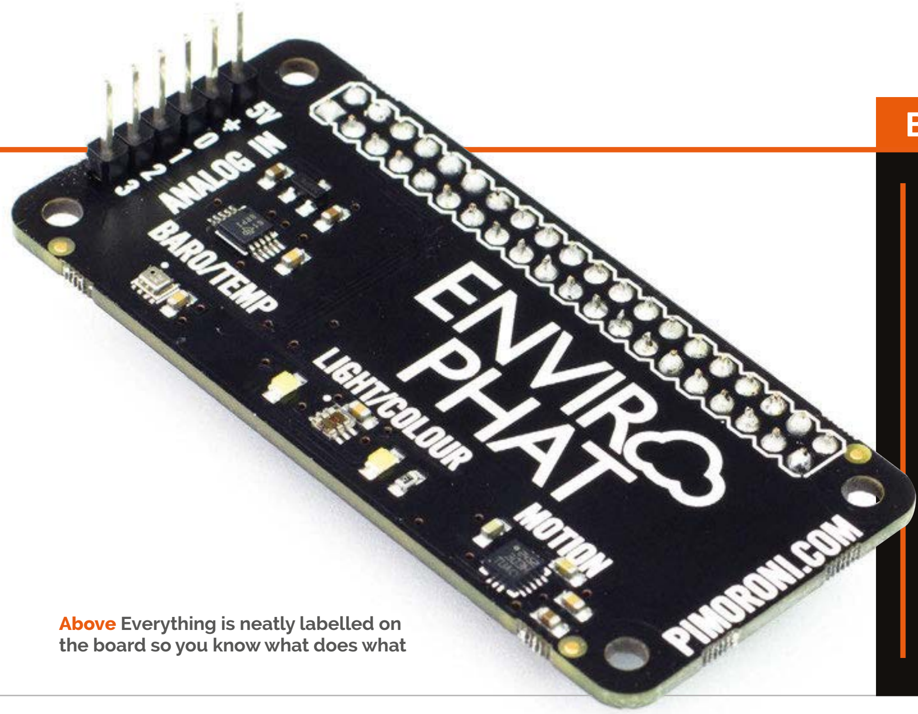


£29 / $38

magpi.cc/1TGGUt5

magpi.cc/29NHB3T

£16 / $21



Above Everything is neatly labelled on the board so you know what does what

BUILT-IN SENSORS

Light: The highly sensitive TCS3472 colour sensor enables you to measure the ambient clear light level and RGB colour values, aided by twin LEDs to illuminate objects.

Weather: A BMP280 sensor measures atmospheric pressure and temperature, although the latter is affected by the heat produced by the pHAT.

Motion: The built-in LSM303D 3D accelerometer/magnetometer can detect the board's orientation, motion, and compass heading.

Analogue Input: An ADS1015 ADC enables it to convert analogue readings from external sensors on four channels.

the Raspberry Pi CPU beneath it. Therefore, you'll need to calibrate it by comparing the real ambient temperature, using a standard thermometer, to discover the difference; for us it was around 7°C, but it may vary depending on the setup. For a more accurate reading, you could always use a remotely placed temperature sensor connected to the pHAT's analogue input section: more on that later.

Detecting motion

The Enviro pHAT includes an LSM303D accelerometer/magnetometer for detecting the board's motion through three axes (pitch, roll, and vertical) and its compass bearing. The latter can easily be calibrated to north, so long as you already know where that is; it's done by setting a variable to its value and subtracting it from the reading (with modulo 360) to get the correct compass heading in degrees. Meanwhile, the `motion.accelerometer()` tuple can be split into three variables, one for each axis. You can also obtain the raw magnetometer data if you prefer. Since the combination of Enviro pHAT and Pi Zero has such a small form factor, it's ideal for measuring the motion of people carrying it or objects attached to it, although it'll require a portable power source such as a phone charger.

Last but not least, the Enviro pHAT features an ADS1015 ADC for reading external analogue sensors. Located on a short edge of the board are six pins: 5V power output and ground, plus four input channels to take readings from sensors. Note that the input pins are designed to measure signals between 0 and 3.3V, so if your sensor's output is 5V you'll need to create a voltage divider, using three identical resistors on a breadboard, to lower it to 3.3V. While Pimoroni says that in its tests, running 5V into the ADC inputs didn't cause any adverse effects, the readings won't be reliable unless you use a voltage divider. It's not much of a hurdle, though, and the inclusion of an analogue input section for connecting extra sensors is a major bonus.

Overall, with its similar functionality, the Enviro pHAT is a cheaper, more portable alternative to a Sense HAT, although without the LED matrix and a few other features, but with the addition of analogue inputs for extra sensors. The Python library is very intuitive and easy to use, aided by an online tutorial (**magpi.cc/29maHZT**) to get you started and a few helpful code examples in the GitHub repository (**magpi.cc/29M8bdD**). With its small form factor, we can see the Enviro pHAT being used with a Pi Zero to create IoT devices for monitoring room temperature, light levels (to possibly trigger electric lighting), and various other remote uses. By using a stacking header, it could also be combined with another Pimoroni pHAT, such as the Scroll pHAT with its LED matrix, to display its readings in situ.

Last word

For portable projects requiring sensor data, the Enviro pHAT could prove particularly useful. You could just mount it on a Pi Zero and leave it on a shelf to monitor room conditions, for instance, logging its readings into a file or database. The inclusion of an ADC and analogue inputs for external sensors is a bonus for what is a fun, easy-to-use add-on with plenty of possibilities.

★★★★★

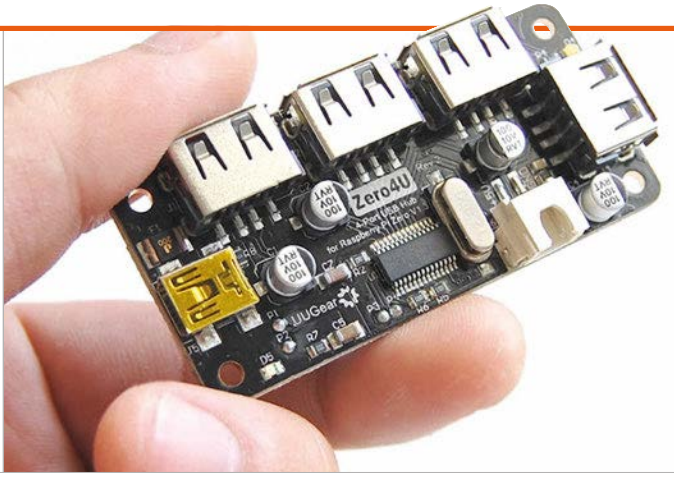
magpi.cc/2aFCXtY

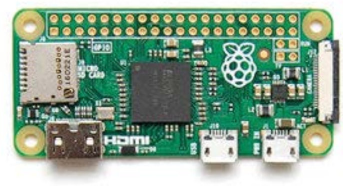
£6 / $8

Maker Says

"Can be mounted to Raspberry Pi Zero back-to-back"

UUGear





ZERO4U

Adding four USB ports to the Pi Zero, can it replace a USB hub?

While the Raspberry Pi Zero's compact nature makes it ideal for many projects, the downside is that it only offers a single micro USB port for connecting peripherals. So, to use it with a keyboard and mouse, for instance, you'll need a USB adapter and a standard USB hub. Well, not any more…

Designed by UUGear in the Czech Republic, the Zero4U is a four-port USB hub that's mounted on the rear of the Pi Zero. Its four pogo pins connect to the tiny PP1 (+5V), PP6 (GND), PP22 (USB D+), and PP23 (USB D−) testing pads on the Pi Zero. This enables it to take its power from the latter, in which case it can output up to 2A current to all four USB ports.

Since the pogo pins are only in surface contact with the pads, they need to be kept firmly in place by securing the Zero4U to the Pi Zero using the plastic standoff screws and spacers supplied. We were slightly concerned about the pins maintaining a reliable contact, but didn't experience any problems. One detail to note is that since the testing pad positions are slightly different on the two Pi Zero models – the original v1.2 and new v1.3 with camera connector – there are two versions of the Zero4U to suit, so you need to ensure you order the correct one. Either way, the Zero4U can also be used with any other Raspberry Pi model via its mini USB input, although the power output is reduced in this case unless you power it independently via its JST XH2.54 port.

Once the Zero4U is piggybacking the Pi Zero and powered on, a blue LED lights up to show that it's operating. In addition, each port has a white status LED that's lit whenever a device is connected to it, which is a nice touch. All four ports operate at standard USB 2.0 speed (480Mbps). The only caveat is that if you insert a USB 1.1 device, they'll all be slowed down to 12Mbps, since the hub has a single transaction translator, but it's not a major problem.

Related

THREE-PORT USB HUB WITH ETHERNET

You'll need a micro USB adapter to plug it into the Pi Zero, but it has the bonus of an Ethernet port for wired connectivity.

£10 / $13

magpi.cc/2aFJkx9

Last word

The Zero4U is an ingenious solution to the lack of standard USB ports on the Pi Zero. There's no soldering required and it's relatively easy to attach to the rear of the Zero, which means the GPIO header is kept free and unobstructed. As a bonus, the device can also be used as a standard USB hub for other Raspberry Pi models.

★★★★☆

thepihut.com

£7 / $9

Maker Says

The ZeroView is a clever window/ glass mount for your Pi Zero and Camera Module

The Pi Hut

ZEROVIEW

Stick your Pi Zero with Camera Module to a glass window with this suction cup mount

One of our favourite uses for the Pi Zero is recording time-lapse video with the new camera connector (found on the newer Pi Zero v1.3).

So we were delighted to get hold of the ZeroView. This simple board provides a suction cup mount for the Pi Zero, so you can stick it to glass.

It's ridiculously easy to set up the Pi Zero to record pictures, videos, or capture time-lapse photography. A device that effectively mounts the Pi Zero and holds the Camera Module comes in useful in a range of projects, from home-built in-car dash cams to time-lapse fish tank recordings.

The Pi Zero is mounted using plastic screws. Inside the pack you get a PCB (but just a plain board with no electronic components), two suction cups, and spacers, screws and nuts to mount both the Pi Zero and Camera Module. It took us about five minutes to screw it all together following the PDF instructions at **thepihut.com/products/zeroview**.

The Camera Module is mounted and the cable tucked between the Pi Zero and ZeroView. The end result is a compact, self-contained camera device that can be stuck to any glass surface. Combine it with a battery pack, and set up a script to automatically start recording, and you get a neat camera package.

We've had trouble with suction cups before, where devices have dropped. With this in mind, we stuck the ZeroView to a window to capture a time-lapse video, and started a stopwatch to see how long it lasted. After an hour, we decided that it was going to be there all day and stopped the test.

"We've hunted down the best quality suction cups we could find," says The Pi Hut, "using only the best 'Adams' cups made in the USA. We're so impressed with the performance of these suction cups that we just couldn't use any other brand."

Whether it's the high-quality cups or the general lightness of the package, it's hard to fault the ZeroView. It's easy to set up, looks cool, and sticks around all day.

Related

RASPBERRY PI CAMERA MOUNT

A cheaper option is to buy a mount for just the Camera Module, but this doesn't provide a combined package.

£3 / $4

thepihut.com

Last word

A neat product that transforms the Pi Zero and Camera Module into a portable, stickable camera package ideal for time-lapse and slow-motion photography projects.

★★★★★

Maker Says

"Build your computer through Minecraft"

Piper

PIPER

Build a computer and then keep building it as you play through a Minecraft adventure

Once again, we've come face-to-face with a crowdfunded Raspberry Pi laptop. With the pi-top not even a year old, it's interesting to see something that, on paper, is a competitor for the same space. A 'build-it-yourself' laptop that gamifies learning computing through a custom operating system, the Piper is very different from the pi-top when it comes down to it, however.

First of all, construction of the laptop is very different. While the pi-top feels like you're assembling the components for a real laptop, Piper feels like putting together a Meccano kit or wooden model. Laser-cut, engraved wooden sections slot into place, held together by the odd screw. There's a big sprawling poster with the steps needed to put the box together, with the engravings giving you some visual clues on what goes where. The poster is a little unwieldy and you need lots of space for it, but construction is fairly simple, if not a little lengthy. We sat through at least a couple of episodes of *Star Trek: Deep Space Nine* getting it built, so it took about 90 minutes.

Some assembly required

The final build is chunky and sturdy. The computer parts include a nice 7″ LCD display in the top, a Raspberry Pi, a USB mouse, and a portable power bank to power the whole lot. This makes it quite mobile, although you'll need to remember to charge up the power bank and keep an eye on its levels.

The most ingenious thing about the Piper, though, is that you can carry all the electronics pieces, speaker, and mouse inside the laptop. It's not really so much of a laptop as a digital toy chest, with all your Power Rangers (buttons) and Barbies (jumper wires), and whatever kids actually play with these days (Star Wars figures?) kept inside, latched up and ready to take with you wherever you go. The only thing it's really missing is a carry handle, although we really wouldn't want to be swinging it around with loads of bits inside.

The initial instructions take you as far as getting the case built, and the Raspberry Pi and screen working. Plug it all into the battery pack and you boot up into the Piper's OS. This starts with a fun little video before launching you into the Minecraft adventure that helps you continue to build your laptop, adding the extra buttons

Related

PI-TOP

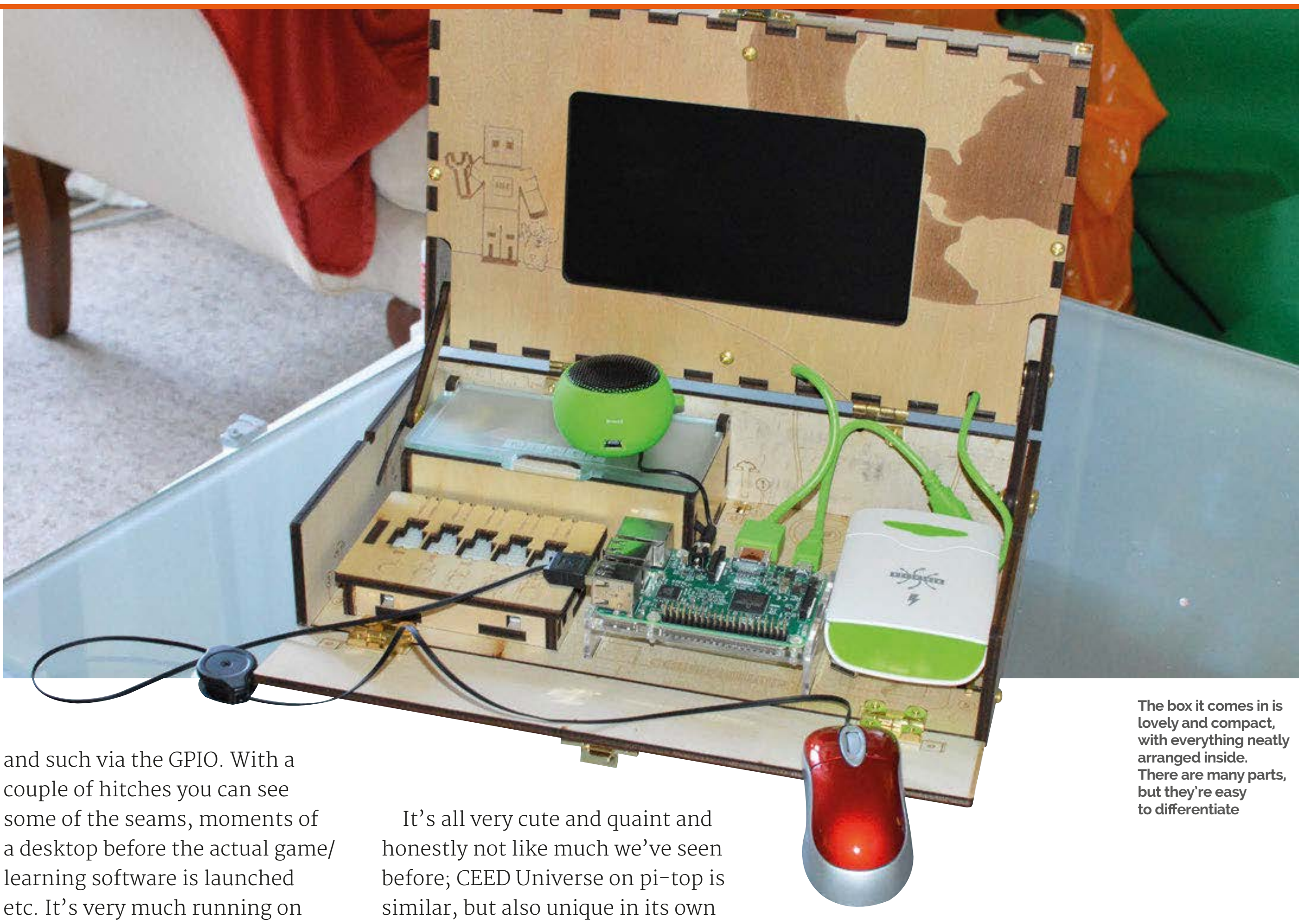
A similar concept but a very different execution, the pi-top may be more suitable for older kids and young adults.

£230 / $300

pi-top.com

playpiper.com

£230 / $299



The box it comes in is lovely and compact, with everything neatly arranged inside. There are many parts, but they're easy to differentiate

and such via the GPIO. With a couple of hitches you can see some of the seams, moments of a desktop before the actual game/learning software is launched etc. It's very much running on Raspbian, but you'll never see it through normal use.

Know your craft

PiperCraft is the name of this game, a modded Minecraft Pi that gives you challenges to complete and in the process teaches you some real-world physical computing. Each section is presented by machinima-style cut scenes, presumably filmed in full Minecraft, which are also fully voiced in an adorable fashion. Guide PiperBot to save Earth from Mars, with only a witty assistant and many Minecraft blocks to help you. There are multiple levels and apparently more are being made, which will be free to download as they become available; people can also create levels and share them.

It's all very cute and quaint and honestly not like much we've seen before; CEED Universe on pi-top is similar, but also unique in its own way beyond just being 'gamified computing education'.

Let's return to the concept of it as a laptop, though. As we've said before, the version you're supposed to build and play with is not really a proper Pi laptop in a traditional sense. You don't have a keyboard, for starters. However, it can easily be modified to be a more normal laptop. You can take out the Piper SD card and make a normal Raspbian one for yourself. The screen connects via HDMI, so it doesn't require any extra software to get running. And if you take out the little component chest and the breadboard, there's enough space to store a little USB or Bluetooth keyboard within the case. The version that's shipping to consumers will come with a Pi 3 so you can connect to wireless, so really it's very little effort to do a 'conversion' if you wish.

It's a really fun, excellent kit. The build, the game, and the possibilities for it are great, even if it's perhaps more suitable for younger kids than the 'all ages' for which it's being marketed.

Last word

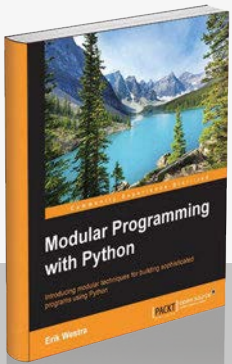
The price may be a little steep, but it's a really fun educational computer kit that should really impress those who love Minecraft and building stuff. You can also take it almost anywhere!

★★★★★

RASPBERRY PI BESTSELLERS PACKT PYTHON

The best of this summer's Packt Python books promise you an autumn of learning

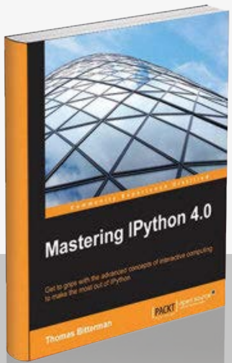
MODULAR PROGRAMMING WITH PYTHON



Author: Erik Westra
Publisher: Packt
Price: £25.99
ISBN: 978-1785884481
magpi.cc/2aUBKho

Get organised with this succinct guide to making your code modular, which takes in Python's extensive import system, testing your modules, and even preparing your modular code for sharing on GitHub.

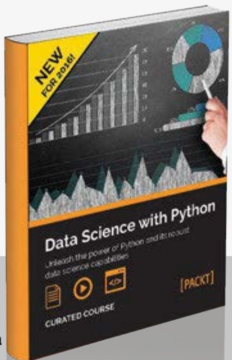
MASTERING IPYTHON 4.0



Authors: Thomas Bitterman
Publisher: Packt
Price: £31.99
ISBN: 978-1785888410
magpi.cc/2aUC5QZ

Get interactive with IPython, not just as a rich workbook interface to scientific computing, but for developing for parallel and high-performance computing. The book covers testing and working with R, Julia, and JavaScript.

PYTHON: REAL-WORLD DATA SCIENCE



Authors: Dusty Phillips et al.
Publisher: Packt
Price: £49.18
ISBN: 978-1786465160
magpi.cc/2aUBZsl

Packt's comprehensive curated course combines works to give you 1,250 pages of intensive data science learning and practical Python coding, taking in NumPy, Matplotlib, Redis, and MongoDB along the way.

BUILDING THE WEB OF THINGS

Author: Vlad M Trifa & Dominique D Guinard
Publisher: Manning
Price: £21.99
ISBN: 978-1617292682
magpi.cc/2aUEsTC

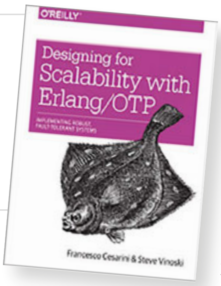
Competing standards and fragmentation – IT's traditional curse – have led to the Internet of Things (IoT) being more a collection of isolated Intranets of Things. Guinard and Trifa's solution is to integrate the fragmented parts with the most successful application layer of them all, the web, using its loose coupling and simply defined programming model as the basis of clean web APIs to build a scalable Web of Things (WoT).

The book, accessible to anyone with basic programming and web skills, is split into two parts. The first introduces the basics: the concept, a device to work on (enter the Raspberry Pi), and using JavaScript and Node.js to glue things together. A hands-on walkthrough in chapter two gets readers comfortable with using the Pi as a remote, web-connected device.

The second section also combines the theoretical and the practical, as APIs and protocols are introduced then used to build interactive WoT projects, and the reader is drawn from data security to scalable physical mash-ups of devices. As long as competing IoT devices and networks can be interacted with through the web, at least through some gateway, all things are possible. This will ready you for tomorrow, while others are still arguing over standards.

Score ★★★★★

DESIGNING FOR SCALABILITY WITH ERLANG/OTP



Author: Steve Vinoski & Francesco Cesarini
Publisher: O'Reilly
Price: £33.50
ISBN: 978-1449320737
magpi.cc/2aUDTta

Writing this as a sequel to O'Reilly's *Erlang Programming*, veterans Cesarini and Vinoski deliver the ideal next step to anyone who's completed any introductory work on the language and is ready to tackle a project that demands the distributed language's key benefits: scalability, reliability, and availability.

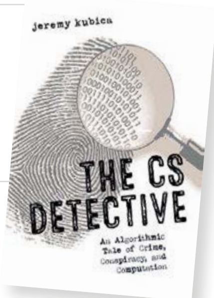
The introduction helps to define the problem space, and the tools and libraries available, as well as the principles of the OTP environment. It's followed by an Erlang refresher, or an introduction for those brave enough to start their Erlang journey here. Next, design patterns and behaviours: client server examples are developed, broken into parts, packaged into library modules, and migrated to OTP-based generic server behaviour. Then it tackles finite-state machines and event handlers, using a straightforward telephony example.

Next, there's monitoring and handling errors with supervisors, packaged into the building blocks of applications, and then non-standard behaviours and building robust applications. This is hard going for readers, as something of a shift in thinking is involved to turn out programs in such a form, but this book will help you understand the whys and hows of OTP. Treating the full trade-offs of developing, deploying, and working with code in scalable, distributed applications makes up a very useful final section.

Score ★★★★☆

THE CS DETECTIVE

Author: Jeremy Kubica
Publisher: No Starch
Price: £12.99
ISBN: 978-1593277499
nostarch.com/searchtale



"Meet Frank Runtime. Disgraced ex-detective. Hard-boiled private eye. Search expert." Search expert? Yes, Runtime uses search algorithms, in a novel designed to introduce computational thinking to a wider audience. Although most useful to learner programmers of all ages – each chapter ends with lecture notes on the concepts covered therein – the detective stories are entertaining enough to stand on their own for anyone who'll get some of the references.

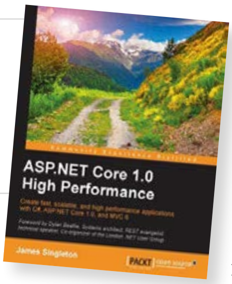
Runtime, the loner who doesn't follow the rules, is a familiar figure in fiction, and a Sam Spade-style gumshoe in a pre-industrial world is found everywhere from the 1999 computer game Discworld Noir, to Lindsey Davis's ancient Roman detective Falco. The key to making it work is to keep the humour light and the prose terse, which Kubica does. Take a look at his popular Computational Fairy Tales blog if you'd like a preview of his style.

Thanks to courses like Police Procedures and Data Structures, Runtime is able to find the best search algorithm; everything from best-first and depth-first search, to iterative deepening, parallelising, and binary search, is covered in this entertaining and educational read that should give you enough background to pursue your learning further.

Score ★★★★☆

ASP.NET CORE 1.0 HIGH PERFORMANCE

Author: James Singleton
Publisher: Packt
Price: £34.99
ISBN: 978-1785881893
magpi.cc/2aUG4x2



From 'Why Performance is a Feature', the first chapter, this is a book that encourages caring about how your code performs, to the ultimate benefit of the end user, using profiling to eliminate bottlenecks in C# applications on MS's latest web application framework. Singleton's introduction to getting the best performance on .NET Core 1.0 is not your average web application development book; performance implications of architecture are weighed, with the Raspberry Pi explicitly considered. Yes, the Pi running .NET, and not necessarily with Mono.

.NET Core, unlike traditional Microsoft products, is open-source and cross-platform. In the spirit of this, it's not an MS-centric book; other platforms (Mac, Linux, and of course the Pi), other services (RabbitMQ recommended as far better than Microsoft Message Queuing), and other tools are given a fair examination, and many so-called ALT.NET choices are recommended for working with the new ASP.NET.

After measuring, optimising, and even searching for bottlenecks in the network stack, the author gives a good look at the downsides of your improvements: there are always trade-offs, and the burden of managing complexity and caching and debugging issues is considered. Essential reading for anyone working with ASP.NET Core 1.0.

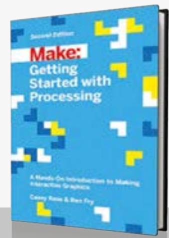
Score ★★★★☆

ESSENTIAL READING: PROCESSING

Learn to code with the open-source language designed for the visual arts

Make: Getting Started with Processing

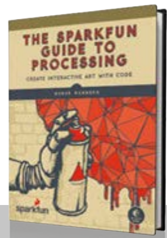
Author: Casey Reas & Ben Fry
Publisher: Maker Media
Price: £17.99
ISBN: 978-1457187087
magpi.cc/2aUHSpE



Very highly regarded introduction to working with Processing, teaching core programming concepts to coding newbies.

The Sparkfun Guide to Processing

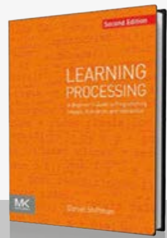
Author: Derek Runberg
Publisher: No Starch
Price: £21.50
ISBN: 978-1593276126
nostarch.com/sparkfunprocessing



Project-based intro that oozes creativity, supported by a strong educational framework.

Learning Processing 2nd Edition

Author: Daniel Shiffman
Publisher: Morgan Kaufmann
Price: £30.99
ISBN: 978-0123944436
learningprocessing.com



Well-regarded and comprehensive intro, updated for compatibility with Processing 3 with new chapters on video, sound, data visualisation, and networking.

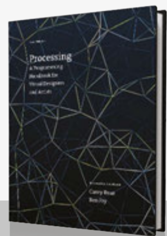
Welcome to Processing 3

Author: Daniel Shiffman
Publisher: N/A
Price: Free
ISBN: N/A
vimeo.com/140600280

Inspiring look at what's new in Processing 3 (more online resources are linked from **processing.org**).

Processing: A Programming Handbook for Visual Designers

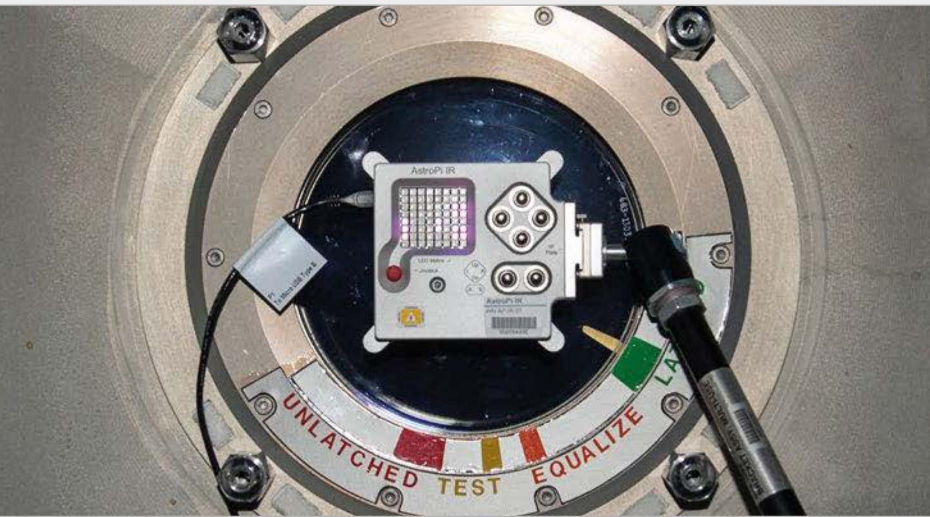
Author: Casey Reas & Ben Fry
Publisher: MIT Press
Price: £55.95
ISBN: 978-0262028288
magpi.cc/2aUL2d4



Covering Processing 2.0 and 3.0, and updated for the new syntax, the definitive reference from Processing's co-founders.

THE MONTH IN RASPBERRY PI

Everything else that happened this month in the world of Raspberry Pi



MORE ASTRO PI RESULTS!

Two issues ago we published our Astro Pi special, where we covered some of the results from the experiments performed by code written by schoolkids, for use on the International Space Station (ISS) by British ESA astronaut Tim Peake.

We really only scratched the surface with the results we were able to show off within the confines of the magazine. Fortunately, they are all now available for people to view online on the Astro Pi website: **magpi.cc/2bjX0gW**.

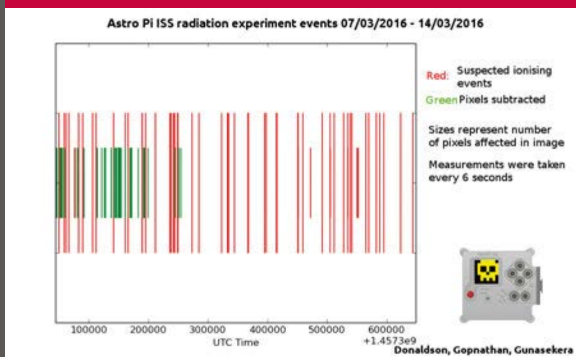
"One of the main things we've learnt from running Astro Pi is that the biggest motivational factor for young people is the very tangible goal of having their code run in space," Dave Honess writes in his blog post (**magpi.cc/2bv27rF**) detailing the ninth Astro Pi mission update on the Raspberry Pi website. While this brings to an end the Astro Pi mission as it was originally set out, there's still a bright future for Astro Pi: it will be used by other ESA countries with their own competitions in the future. Read issue 47 of *The MagPi* to find out more.

SPACE SCIENCE

Here's two more experiments from the recently ended Astro Pi mission. View them here: **magpi.cc/2bjX0gW**

RADIATION

Using image recognition software, the Radiation experiment covered the camera lens so that visible light couldn't get through. Radiation was, however, able to hit the sensor and create flashes on the sensors which can be observed. Unfortunately, due to the thickness of the flight case, the results were a little less than the team had hoped for. They are trying to decipher the data they have, but there may have been some damage to the camera during operation.



FLAGS

As the ISS flies high above the Earth, it passes over many countries, sometimes very quickly. Flags was programmed to show what country the ISS was currently over, by displaying that country's flag and a short phrase in its language. Unfortunately, by the time the code went up to space, minor course corrections on the ISS made the code out of date. However, Tim fixed it in his spare time and it started showing the correct countries again.

CROWDFUND THIS!

The best crowdfunding hits this month for you to check out…

I'M BACK

kck.st/2aDOYdR

Yes, that's its name. This is one of the most unusual projects using the Raspberry Pi that we've seen for a while: it's a special case for the Pi which allows you to attach an old 35mm camera, and use the lens and such to take photos using a Pi Camera Module. We think it's a remarkable idea, and clearly a lot of people agree: it's already funded and the campaign is just a few days old. If you want to meld classic photography with modern digital photography, give it a look.

THE VILLAGE PC PROJECT

magpi.cc/2bjzhOo

This campaign is aimed towards sending 100 Raspberry Pi computer kits, including a screen and input devices, to different villages around Cambodia so that the kids there have a chance to learn about computing. They're not just sending computers, though: the goal of the project is also to send people to help teach the kids and the schools about computing, so that they can have a bit of a head start with it. They hope to help several hundred disadvantaged kids living in rural Cambodia with this project.

WITH ICE CREAM

kck.st/2aKUVsQ

It's not a very descriptive name, but at its simplest it's a case for the Raspberry Pi inspired by the US and UK version of the NES games console. The pitch revolves around using it to play classic games, presumably via RetroPie. To this end, the case has been arranged so that you can access the SD card slot from the lid, and four USB ports have been installed on the front of the console so you can connect controllers; the standard Pi USB ports are on the back.

BEST OF THE REST

Here are some other great things we saw this month

magpi.cc/2bmA0tp

SMART SHOE RACK

An Internet of Things shoe rack may seem like a joke, but Reddit user zealen's project scrapes the internet for weather information, and then tells you what the best pair of shoes are for the day by lighting it up. Just tap the top for it to make a selection. Magic.

magpi.cc/2buA6jv

ALTOIDS COMPUTER

When the Pi Zero was first released, Matt Richardson from Raspberry Pi tried to make a computer out of a Zero and an Altoids tin, without much success. Reddit user RealSlimCadey took a slightly different approach and created this.

COMMUNITY **PROFILE**

ZACH IGIELMAN

The 16-year-old, piano-wielding, Pi-building entrepreneur looking to educate the world on the importance of tech

Zach

Name: Zach Igielman
Category: Maker
Day job: Student
Website: magpi.cc/2aXfyUY

Below The monthly *MagPi* magazine selfie has become standard on Twitter
Below right Zach was one of several young makers on the Young Person's Panel at this year's Raspberry Pi 4th Birthday Bash

You may recognise the name Zachary Igielman from issue #38, where he was mentioned during our review of the exciting Pimoroni Piano HAT. The Piano HAT, for those unaware, was inspired by Zach's own creation, the PiPiano, a successful crowdfunded add-on board that hit 184% of funding two years ago. At age 14, Zach had decided to incorporate his passions for making, engineering and music, building himself a PCB that could use physical keys to control electronic sound files and Sonic Pi code. The PCB, he explains, is a great classroom tool, educating students on the fundamentals of physically building digital tech and soldering, through to understanding sound generation through PWM frequencies.

Zach began to teach himself code at age 11, soon discovering the Raspberry Pi and, later, the Cambridge Raspberry Jams. It was through this collective of like-minded individuals that Zach was inspired to broaden his making skills, moving on to create line-following robots that avoided objects through sensors.

Moving forward, Zach visited the Raspberry Pi offices for work experience, continuing to work on and study robots and

Above At 16, Zach has already made major contributions to the Raspberry Pi community and beyond

robotic guides, working alongside our engineers to build upon his knowledge.

It was around this same time, in October 2014, that Zach met Frank Thomas-Hockey via Twitter. Frank was looking for help in creating the first London Raspberry Jam and Zach built a fast friendship online, lovingly referring to him as a fellow "computer geek". The two have worked on projects together, including several websites, and spent time

> "Zach began to teach himself code at age 11, soon discovering the Raspberry Pi"

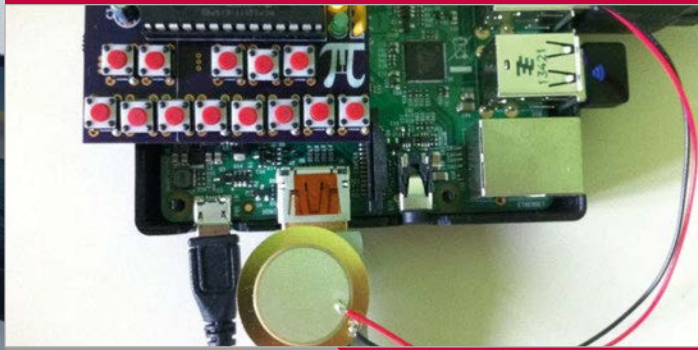
was more than willing to lend a hand. Between them, they set up the Covent Garden Jam, welcoming over 100 visitors to their first event. Their most recent Jam – now with the additional help of volunteers Ben, Paul, and Joseph – allowed them to simultaneously run workshops on soldering, Sonic Pi and Minecraft, while also highlighting maker projects through show-and-tell and talks.

Finally finished with his GCSE exams and about to begin his sixth-form studies in Maths, Further Maths, Physics and Computing, Zach now has the time to continue his recent collaboration with friend Jake Blumenow. Zach met Jake and travelling, bouncing ideas off one another with the aim to create something important. It's their most recent venture that is worthy of recognition.

"At Google Campus, we developed our business model – we believe people of all ages have the right to understand how the technological world around us works so they can modify and create their own technology."

Between the two of them, they aim to create complete Raspberry Pi education kits, inviting beginners in making and coding to create functional projects, such as an alarm system, thus cementing the pair's desire to highlight the day-to-day importance of tech in our lives.

HIGHLIGHTS



PIPIANO

magpi.cc/2bBCNni

Zach taught himself how to build a PCB in order to bring the PiPiano to life. Using Indigogo to fund his project, Zach hit 184% of target before approaching Pimoroni to hand over the design. And from his homemade PCB, the Piano HAT was born.

COVENT GARDEN RASPBERRY JAM

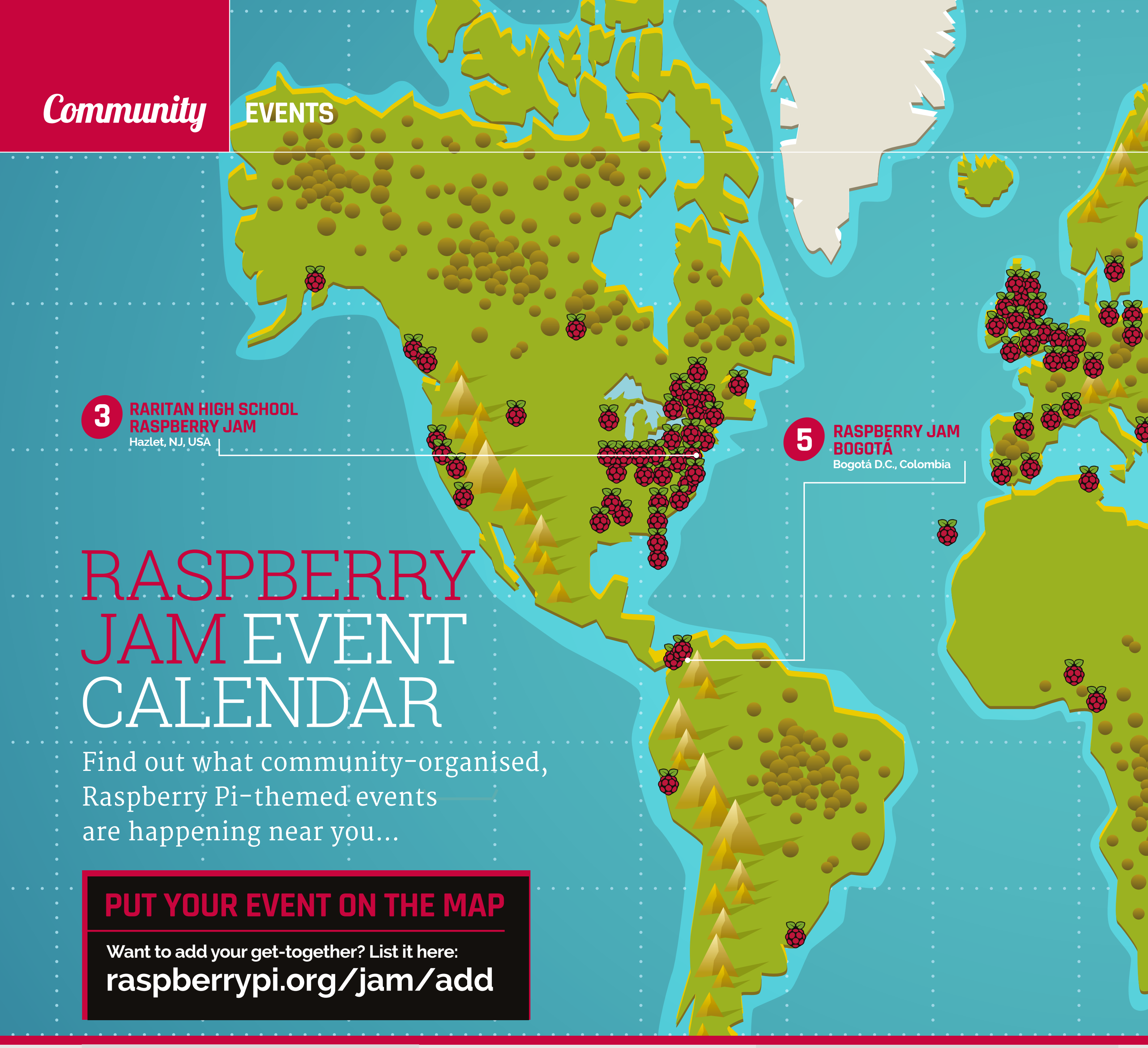
cgjam.rocks

Through Twitter, Zach met Frank in 2014, a like-minded Pi enthusiast looking to start a London-based Raspberry Jam. Between the two of them, they launched the first event at Dragon Hall, continuing the success of the Jam to now include multiple workshops, show-and-tell, and talks.

COLLABORATION WITH JAKE BLUMENOW

magpi.cc/2bmAkbZ

Zach and Jake believe everyone has the right to understand how technology builds the world around them. With this in mind, they formed a partnership, working to create Raspberry Pi educational kits, starting with a DIY alarm system.



RASPBERRY JAM EVENT CALENDAR

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

PUT YOUR EVENT ON THE MAP

Want to add your get-together? List it here:

raspberrypi.org/jam/add

1 HULL RASPBERRY JAM

When: Saturday 10 September
Where: Malet Lambert School, Hull, UK
magpi.cc/2bcgofO
An event to bring people together and discover the exciting possibilities of Raspberry Pi.

2 CORNWALL TECH JAM

When: Saturday 17 September
Where: Pool Innovation Centre, Pool, UK
magpi.cc/2bmRKW8
Come along and celebrate Software Freedom Day, for anyone of all ages and abilities interested in technology.

3 RARITAN HIGH SCHOOL RASPBERRY JAM

When: Saturday 17 September
Where: Raritan High School, Hazlet, NJ, USA
magpi.cc/2bmRIxx
Helping beginners and those interested in the Pi to exhibit any projects, create, and share ideas.

4 TORBAY TECH JAM

When: Saturday 17 September
Where: Paignton Library and Information Centre, Paignton, UK
magpi.cc/28KHl8v
Fun, informal, and family-friendly. Aims to inspire people to get into code and take up STEM subjects.

5 RASPBERRY JAM BOGOTÁ

When: Saturday 17 September
Where: Avenida El Dorado Carrera, Bogotá D.C., Colombia
magpi.cc/28Kavk3
Following on from July's first ever South American Raspberry Jam is the third Jam in Colombia's capital city, Bogotá.

6 RASPBERRY JAM PRESTON

When: Monday 3 October
Where: Media Innovation Studio, Preston, UK
magpi.cc/2bmQZfA
Learn, create, and share the potential of the Raspberry Pi at a family-friendly event.

7 RASPBERRY JAM LEEDS

When: Wednesday 5 October
Where: Swallow Hill Community College, Leeds, UK
magpi.cc/2bmQXEI

Everyone is invited for a couple of hours of computing fun, talks, demonstrations, and hands-on workshops.

8 RASPBERRY JAM IPSWICH

When: Saturday 8 October
Where: University Campus Suffolk, Ipswich, UK
magpi.cc/2bcigVI

Join the growing community and learn how to make cool gadgets and how to code.

DON'T MISS: CORNWALL TECH JAM

When: Saturday 17 September **Where:** Pool Innovation Centre, Pool, UK

There are many Jams to choose from on 17 September, as it's also being celebrated as Software Freedom Day. Cornwall Tech Jam offers a wide variety of activities, not just Raspberry Pi-based, that involve STEM subjects, coding, and other forms of tech. There'll be plenty of help in terms of learning to code in Python and Scratch, as well as being able to hack Minecraft Pi. You can find out more information on the event's page here: **magpi.cc/2bmRKW8**

YOUR LETTERS

Virtual HATs

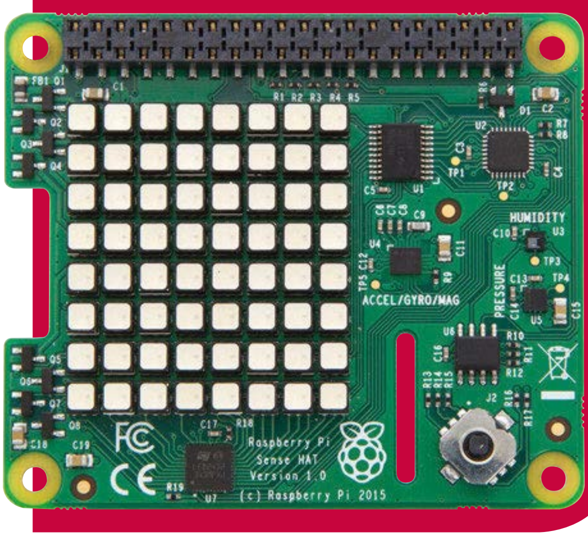
One of the few pieces of Raspberry Pi equipment I don't have is the Sense HAT. It very much interests me, but I've been to a few (admittedly small) local Raspberry Jams and haven't been able to have a go on one. I'd like to see how it works before I buy one; do you know of any way I can do this? I just want to make sure my programming knowledge is good enough to get using it!

Thanks,

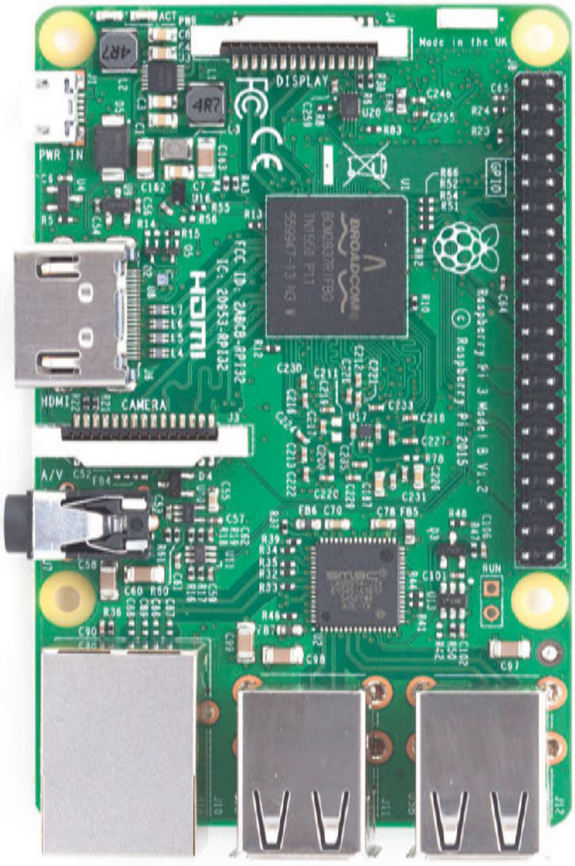
Ellie

Well, the Sense HAT is very easy to program with, we can tell you that much. If you can use GPIO Zero and some other basic Python, you should be able to work your way around it. There's an easy way to test this out, though, as the Raspberry Pi Foundation recently released a Sense HAT emulator for free. Read more about it here: **magpi.cc/2bscvQk.**

Basically, it allows you to write and test code on a virtual Sense HAT. It will allow people to code stuff for any future Astro Pi competitions, without needing a Sense HAT or even a Raspberry Pi. If you want to learn and find your way around it, we also have a special Essentials book all about using the Sense HAT, which might come in handy if you actually buy one. You can find it here: **magpi.cc/Sense-HAT-book**



Right The Sense HAT, as used in the Astro Pi; there's now an emulator for it



Above The Raspberry Pi 3's Compute Module will be very developer-friendly when it's released

Technical specifications

I need to obtain a more detailed specification sheet for the Raspberry Pi 3 B for a project I'm working on. The requirement I have is specifically to determine what voltage tolerance the I^2C interface has (i.e. minimum and maximum voltage). Do you have access to one or do you know what it is at all? Hope you can help!

Andrew Linahan

We reached out to the people who work on Raspberry Pi hardware about the tech specs, and they told us some were on the way for when the Compute Module 3 is properly released. You may remember that during the Raspberry Pi 3 announcement back in February, Eben Upton also announced an updated Computer Module which would use the same hardware as the Raspberry Pi 3, specifically the BCM2837 chip. The numbers you'll need, and much more info, will be available on this sheet when it becomes available; keep an eye on the social channels and blogs for when it comes out.

FROM THE FORUM: BOOK REVIEWS

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community – join in via **raspberrypi.org/forums**

I published a free book on the Raspberry Pi family which is available for everyone to download. I'm wondering if it could be submitted for review within *The MagPi* magazine? Is there a process for submission?
kolban

When it comes to book reviews, we obviously have the two-page spread at the end of the review section, and we're always happy to hear about books that might belong in there. The easiest way to let us know about a book you might want us to look at is to email us at **magpi@raspberrypi.org** and we'll get back to you about it. Usually it's as simple as us sending it to our book reviews man Richard Smedley so he can add it to his pile for reviews, but we can always work out the details through email.

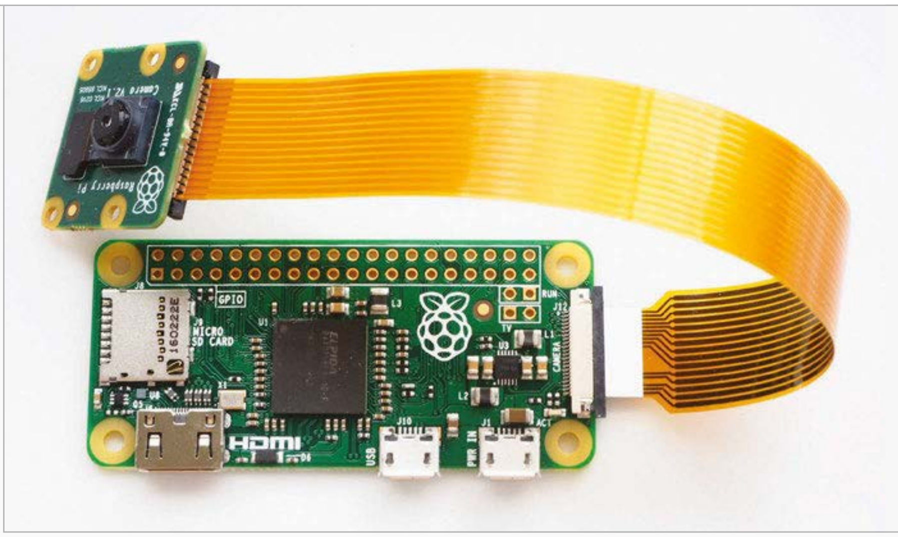
WRITE TO US

Have you got something you'd like to say?
Get in touch via **magpi@raspberrypi.org** or on The MagPi section of the forum at: **raspberrypi.org/forums**

Which Pi Zero is which?

Hi there, MagPi! I've been reading you online for ages now and really enjoying the magazine, and I thought what better way to support you and the Raspberry Pi Foundation than to subscribe to the magazine, especially with such an attractive subs offer going on!

I did just want to check with you what version of the Pi Zero you get in the subscription. Is it the new v1.3 one that has a camera connector, or the original Pi Zero? I'll probably get it anyway, but I did just want to check!
Ken B.



We actually get asked this question a lot and we're always happy to confirm that yes, the new subscription offer comes with a brand new, free Pi Zero v1.3 with the added camera connector. Just to remind everyone else reading, the offer also gets you an HDMI adapter, a USB adapter, and the cable required to attach both versions of the Pi Camera Module to the Pi Zero. It's open to people taking out six- and 12-month subscriptions of the print version of the magazine, and we'll ship worldwide as well. You can check out our subs at **magpi.cc/Subs1**.

Above Subscribe today to get the offer straight to your door and never miss another issue

READ US ANYWHERE

FREE: DOWNLOAD ALL 30 ORIGINAL ISSUES

The MagPi Magazine

Available now

for smartphones & tablets

Available on the App Store

GET IT ON Google Play

Subscribe from

£2.29 or **£26.99**

rolling subscription | full year subscription

Download it today – it's free!

- Get all 30 legacy issues free
- Instant downloads every month
- Fast rendering performance
- Live links & interactivity

WIN A PIBORG YETIBORG

Drive away with the star of Formula Pi in this month's competition!

In association with piborg.org

WHAT IS THE NAME OF **THE FORMULA RACING SERIES** WITH FULL-SIZE **ELECTRIC CARS?**

Tell us by 26 September for your chance to win!

Simply email **competition@raspberrypi.org** with your name, address, and answer!

Terms & Conditions

Competition closes 26 September 2016. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. 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 (unless otherwise stated upon entry). 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.

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

BEYOND THE BOOKCASE

Matt Richardson explores Raspberry Pi's role in the transformation of libraries in the digital age

Before I became a part of the maker movement, my impression of a library was mostly formed by my childhood experiences there. Both my school and local public library were places for books, magazines, newspapers, and research. In short, it was a place for quiet reading. Libraries today look and sound a lot different than I remember. Many now include makerspaces, tools for connected learning, and spaces for community gathering.

But if you take a closer look at what these institutions set out to accomplish in the first place, then the reason they've transformed becomes clear. Take, for instance, the mission of the Seattle Public Library, which is to "[bring] people, information, and ideas together to enrich lives and build community." The mission of the library isn't directly related to reading, even though reading can be a big part of achieving that mission.

A few years ago, I had the opportunity to visit the central branch of the Seattle Public Library. The fifth floor is called 'The Mixing Chamber' and is a designated location where people, information, and ideas can come together. Of course, there's plenty of material to read at the main branch of the Seattle Public Library, but this building in particular makes it very clear that they're about more than just reading.

As another indication of this, we see a lot of interest in Raspberry Pi from librarians. A group of us recently visited the annual conference of the American Library Association in Orlando, and the reaction to our presence there was incredibly positive. Not only have many librarians heard of Raspberry Pi, but they also use it in so many ways.

Of course, library makerspaces use Raspberry Pi just like any other makerspace would: as a platform for DIY projects. There are even many libraries that create Raspberry Pi checkout kits so that their patrons can experiment with Raspberry Pi in their own time, either in the library or at home.

And just as Raspberry Pi is used in the classroom to learn about computing, it's also being used in the library for the very same reason. We've had many librarians come to our Picademy educator professional development programme to learn about teaching people with digital making and computing. These librarians have gone on to share their knowledge and our learning resources with their patrons. Librarians especially love that our content, including this very magazine, is available online entirely for free and is Creative Commons licensed.

Multitasking

What I especially like about the librarians I've encountered is that they don't just put Raspberry Pi in the hands of their patrons, but they use our computers as a tool for their own work. For instance, I recently met Richard Loomis from the Somerset County Library System in New Jersey. He uses Raspberry Pis for networked digital signage across a few different branches. And John Jakobsen from the Palos Verdes Library District recently shared how he set up Raspberry Pis as terminals for their public access catalogue, replacing old and expensive computers. So librarians don't just talk the talk: they also walk the walk.

I'm optimistic that libraries will continue to thrive as technology changes. At the Raspberry Pi Foundation, we're delighted to see that libraries all over the world use our computers for digital making, education, and utility. Our organisation's connection with libraries will always be rich and meaningful, not only because of the way they use Raspberry Pi, but because we have something critical in common with them: we deeply value accessibility and community.