THE **OFFICIAL** RASPBERRY PI MAGAZINE

The MagPi

The official Raspberry Pi magazine

Issue 57 May 2017 raspberrypi.org/magpi

AIY PROJECTS

DO-IT-YOURSELF
ARTIFICIAL INTELLIGENCE

Made by you with Google

Everything you need to bring natural voice interaction to your Raspberry Pi projects

FREE KIT NOT INCLUDED WITH DIGITAL EDITION

THE **ONLY** MONTHLY MAGAZINE WRITTEN BY AND FOR THE PI COMMUNITY

WELCOME TO THE OFFICIAL MAGAZINE

This issue marks the fifth birthday of the magazine. Happy birthday us! It's been an incredible journey from its humble beginnings as a community fanzine, to this issue's Google-powered giveaway. Over the course of the last five years we've generated more than four thousand pages of Raspberry Pi content designed to help, inspire, and entertain you on your journey with the world's favourite credit card-sized PC.

In conjunction with our friends at Google, we've got quite the birthday present for you. Those of you who read the print edition will have in your hands a brand-new kit designed specifically to help you bring Natural Language Interaction to your Raspberry Pi projects.

Over the course of 20 pages, Lucy Hattersley expertly demonstrates the power of Google's new Voice HAT, coupled with the Google Assistant API, to create a cardboard contraption capable of answering any question you put to it. That's just the beginning, though, and we can't wait to hear about the amazing projects you make!

Here's to the next five years of the official Raspberry Pi magazine. Enjoy the issue.

Russell Barnes
Managing Editor, The MagPi

THIS MONTH:

DISCLAIMER: The Google AIY Projects kit is a free gift. It is provided 'as is' without any representation or endorsement made and without warranty of any kind whether express or implied, including but not limited to the implied warranties of satisfactory quality, fitness for a particular purpose, compatibility, and accuracy. All liability arising from such issues is expressly excluded to the fullest extent permitted by law.

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

The MagPi

Available on the App Store
GET IT ON Google Play

EDITORIAL
Managing Editor: **Russell Barnes**
russell@raspberrypi.org
Features Editor: **Rob Zwetsloot**
News Editor: **Lucy Hattersley**
Sub Editors: **Rachel Churcher and Phil King**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Daiva Bumelyte, and Mike Kay**
Illustrator: **Sam Alder**

PUBLISHING
For advertising & licensing:
russell@raspberrypi.org
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
Alex Bate, Mike Cook, Tony Hansen, James Hobro, Phil King, Simon Long, Matt Richardson, Richard Smedley, and Clive Webster

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

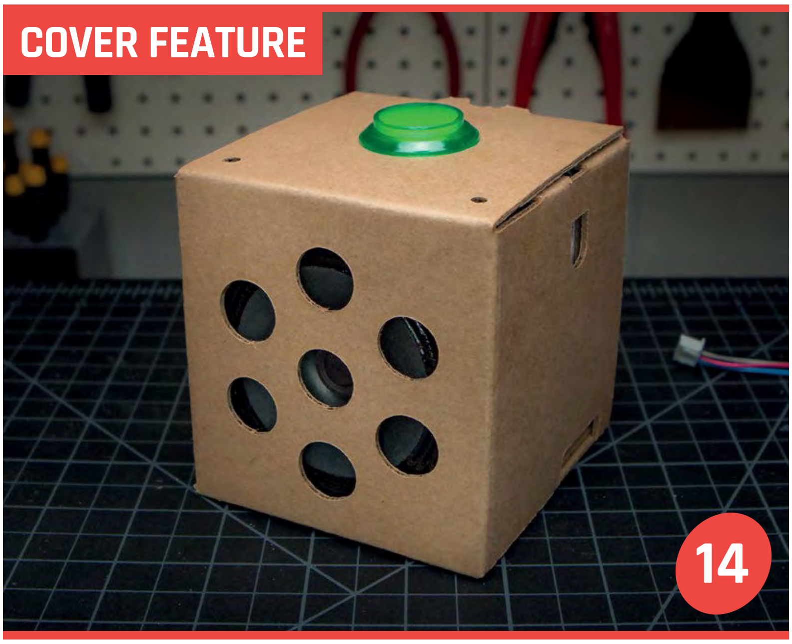
This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Contents

Issue 57 May 2017 raspberrypi.org/magpi

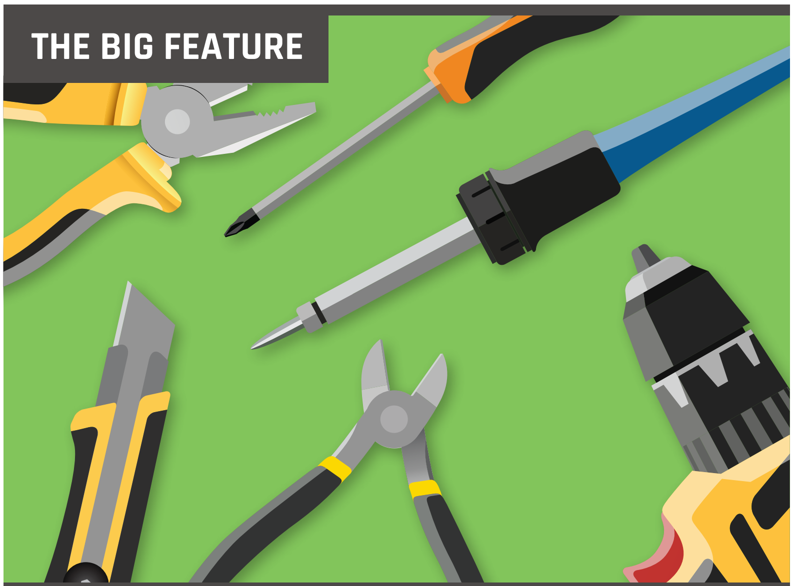
TUTORIALS

COVER FEATURE



IN THE NEWS

THE BIG FEATURE



MAKERS' TOOLKIT

Get yourself ready to make your dream project

WIN $300+ OF COMPUTER HARDWARE

WDLABS

MONSTER BORG

The PiBorg team is back with a new robot

YOUR PROJECTS

PI ORDER

How a restaurant revolutionised its ordering system using a Raspberry Pi

REGULARS

COMMUNITY

REVIEWS

ASTRO PI EU WINNERS ANNOUNCED

French astronaut Thomas Pesquet announces latest winners of Astro Pi competition

In October last year, with the European Space Agency and CNES, The Raspberry Pi Foundation launched the first ever European Astro Pi challenge.

Students from across Europe were asked to write code for the flight of French ESA astronaut Thomas Pesquet to the International Space Station (ISS).

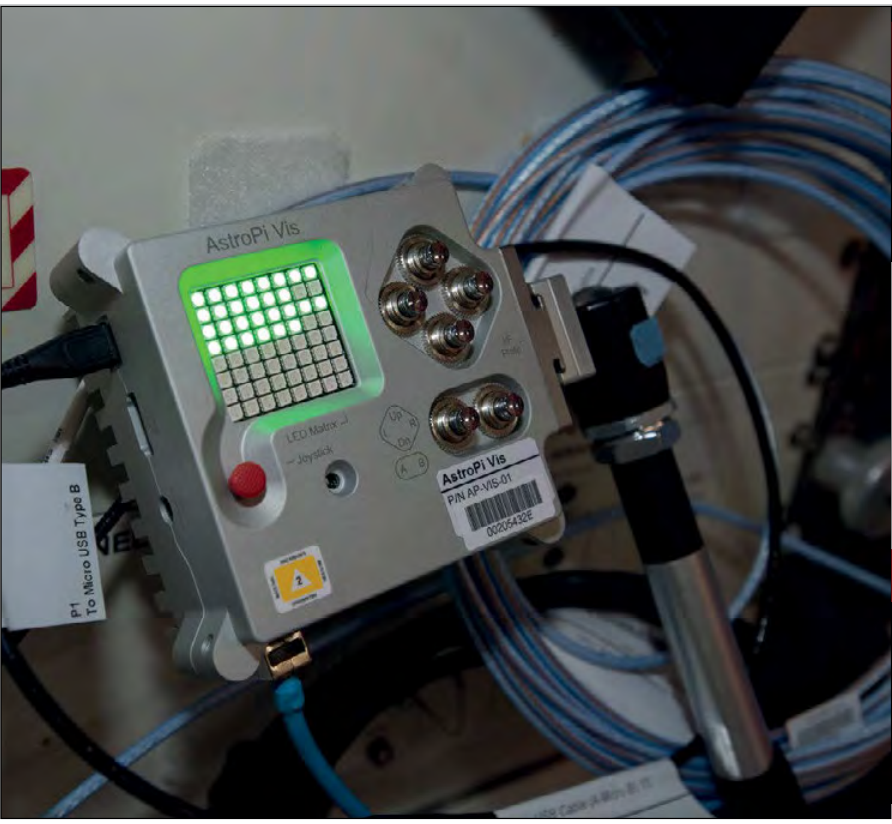
"As I have been busy doing science and engineering in space, you have been busy in your classrooms preparing science investigations," says Thomas. He then thanked makers "working on amazing computer code with your Raspberry Pi [devices]."

"This year's competition expands on our previous work with British ESA astronaut Tim Peake," explains Dave Honess, the Astro Pi Programme Manager at the Raspberry Pi Foundation. "The European Challenge provided school students from every ESA member country with the opportunity to design code to be run in space."

> "I encourage you to continue exploring using different ways to use Astro Pi"

"With so many excellent submissions, it was really hard to choose the best code to run

Below Thomas Pesquet with an Astro Pi device on board the ISS. Credit: ESA (European Space Agency)



Above Ed, one of the Astro Pi devices on board the ISS. Credit: ESA (European Space Agency)

on the ISS," says Thomas. The code runs on two space-hardened Raspberry Pi units, or Astro Pis. Affectionately nicknamed Ed and Izzy, the units were launched into space on an Atlas V rocket in 2015.

Dave reveals that the entries included a cheeky bit of coding: "I found one team who were trying to find the public IP address of the ISS. It made a call out to a web server, then they tried to email the returned page to themselves! No networking code is allowed to fly so they were disqualified, but I might email them and say 'nice try kid!' "

In the coming weeks, Thomas will be supervising Astro Pi 'Ed', continuously running code provided by students.

Astro Pi doesn't end here, though – watch out for an announcement this September. "I encourage you to continue exploring using different ways to use Astro Pi to create more amazing projects," adds Thomas. "Use your imagination, ask yourself questions, and have fun."

THE WINNERS

Everyone produced great work, and the judges found it tough to narrow the entries down. In addition to the winning submissions, a number of teams were awarded 'Highly Commended' status. These teams will also have their code run on the ISS.

The Raspberry Pi Foundation would like to say thank you to everyone who participated. ESA will be uploading your code digitally using the space-to-ground link over the next few weeks. Your code will be executed, and any files created will be downloaded from space and returned to winners for analysis.

A full list of winners and highly commended teams can be found on the Raspberry Pi Foundation website (magpi.cc/2o6mZwA).

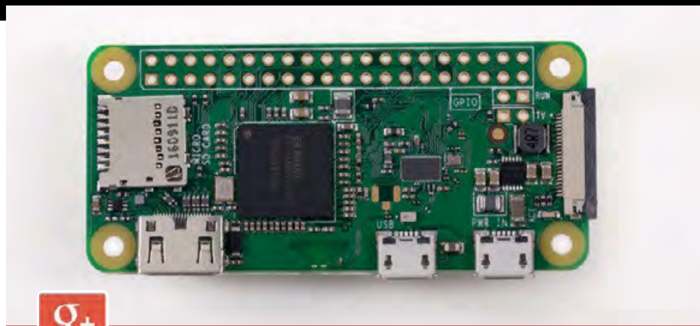
NOW TRENDING

The stories we shared that flew around the world

RX300

magpi.cc/2n5LJWX

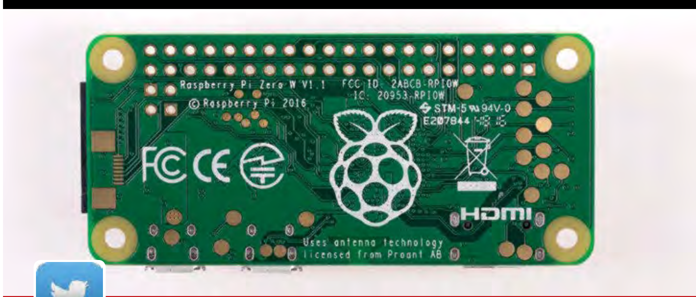
The RX300 is a thin client computer built using a Raspberry Pi. It enables people to build Windows (or Linux) virtual environments and set up offices for an affordable price.



PI ZERO W ANTENNA

magpi.cc/2lylJhX

Our story about how one smart idea made the new Pi Zero W possible intrigued the Google Plus community. Clever surface-mount antenna technology from Proant enabled Pi engineers to add wireless networking to the Pi Zero W.



FREE PI ZERO W

magpi.cc/2lwRp6N

Get a free Pi Zero W with an official case and adapter cables! Our new subscription offer was the top tweet last month.

MonsterBorg is a rugged robot with a 3 mm thick aluminium chassis

MONSTERBORG SMASHES KICKSTARTER TARGET

Monster success for PiBorg's biggest robot yet

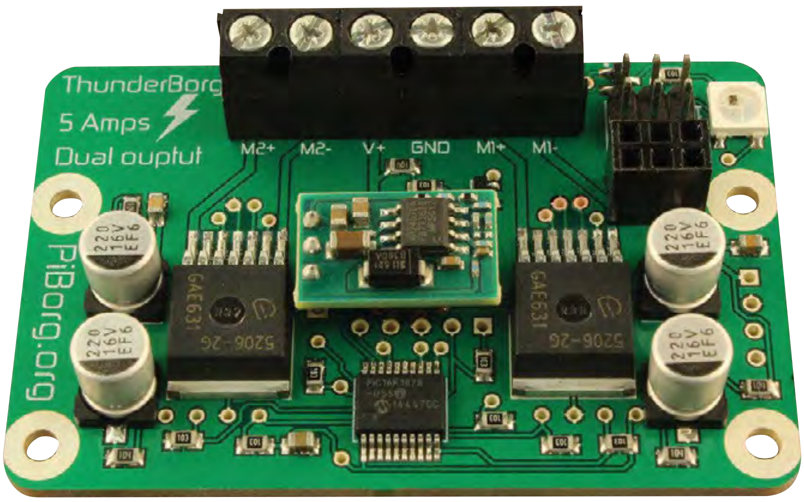
Powering forth on its 10cm wheels, the MonsterBorg has smashed through its Kickstarter target: 1749% funded, with pledges totalling £52,471 (**magpi.cc/2oBdFzg**). "We are stoked!" exclaims PiBorg's Tim Freeburn, who says the robot will soon also be on general sale from **piborg.org**.

Tim tells us the idea for MonsterBorg came about following the first FormulaPi race series using the YetiBorg. "A lot of people wanted to buy and build the robots themselves, but to also use it outdoors or on rough surfaces, so we thought we would design something that essentially was a bigger, more powerful off-road version."

> The MonsterBorg is based on PiBorg's ThunderBorg board

Equipped with four 300 rpm Zhengke 37 mm motors, MonsterBorg is PiBorg's fastest, most powerful robot yet. "It has enough torque to flip itself over when going from full forward to full reverse on flat ground!" reveals Tim. "It can tank-steer on all sorts of surfaces – grass, carpet, bitumen – and it climbs hills very well."

The MonsterBorg is based on PiBorg's new ThunderBorg motor controller board. Also available separately, this provides a dual output of up to 5 A. A running time of up to three hours (using ten AA batteries) will permit longer races in the upcoming MonsterBorg FormulaPi series, whose use of the Raspberry Pi 3 should lend itself to "more complex code such as neural networks and self-learning algorithms," says Tim. "I think we will see some serious entries and some really great lap times."



Below The ThunderBorg controller can drive larger motors than previous boards

NEW RASPBERRY JAM KIT

New resources for organisers of Raspberry Jam events

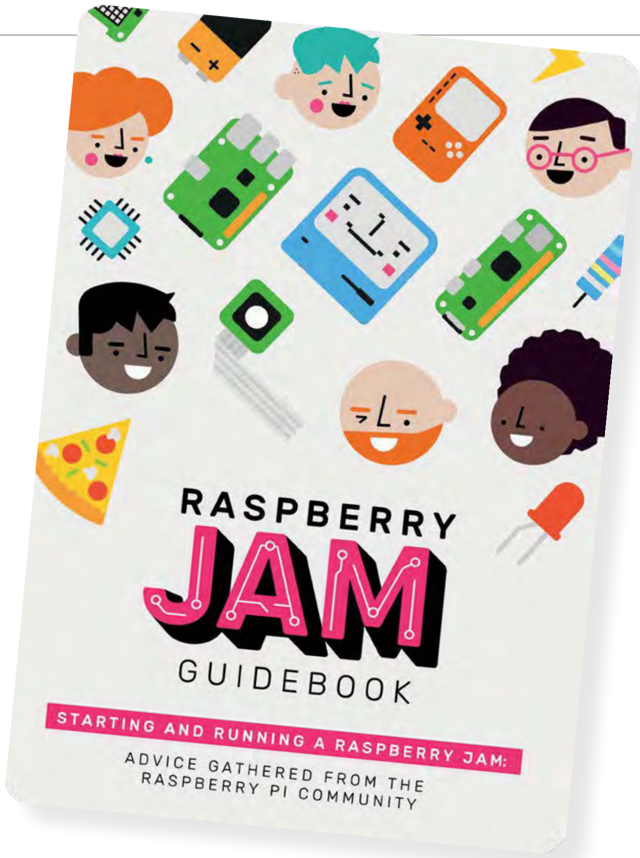
Raspberry Jams are hugely popular community events where people gather to learn digital making with Raspberry Pi. They're a great way to find out more about the Pi and meet like-minded folks.

The Raspberry Pi Foundation has released two new products to support the teams that organise Raspberry Jams.

"The community of Jam makers has so much great advice for people starting their own Jam," says Clare Sutcliffe, Executive Director of Communities and Outreach. "It's been really interesting and exciting to gather all this advice and share it in this guidebook."

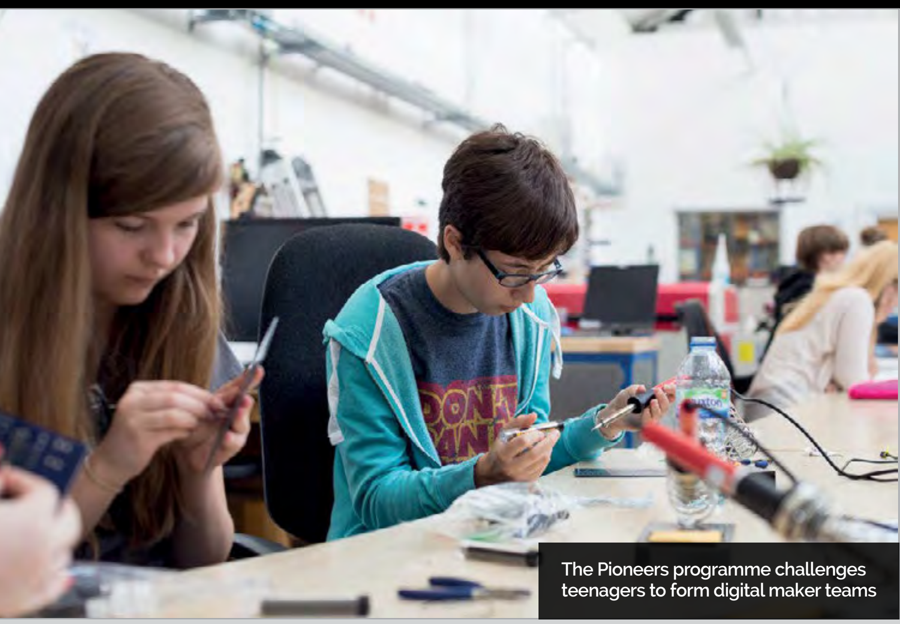
The Raspberry Jam Guidebook is a free guide to running a Jam that you can download from the Raspberry Jam website. The Starter Kit contains a projects book, printed worksheets, leaflets, stickers, and a copy of *The MagPi*. It also features a branding pack of logos, assets, and templates for you to easily make your own posters and flyers to advertise your Jam. As soon as you have submitted your Jam to the calendar on the Raspberry Pi website, you can request your Jam Starter Kit.

For more information on how to download the Jam Guidebook and Starter Pack, visit the Raspberry Jam website (**magpi.cc/28Nxeff**).



Above The new Raspberry Jam Guidebook

PIONEERS WINNERS



The Pioneers programme challenges teenagers to form digital maker teams

The Raspberry Pi Foundation has announced the winners of its first Pioneers challenge. "We laid down the epic challenge of making us laugh," says Olympia Brown, Senior Programmes Manager. "And boy, did the teams deliver! My face hurt from all the laughing on judging day."

Theme winners were The Technological Tricksters, with their Singing Potato. Heritage Hackers and their Water Pistol Trap won the Inspiring Journey award, Black Thunder and their Living Joke Robot won the Best Explanation award, while Shady Hackers and their Scare Chair Rig won the Technically Brilliant award.

The Pioneers programme challenges teenagers aged 12-15 to form teams of digital makers and share their builds online. The next Pioneers challenge will be announced shortly. Head over to the Pioneers website (**magpi.cc/2oMN2rc**) and sign up for the newsletter to be kept up to date.

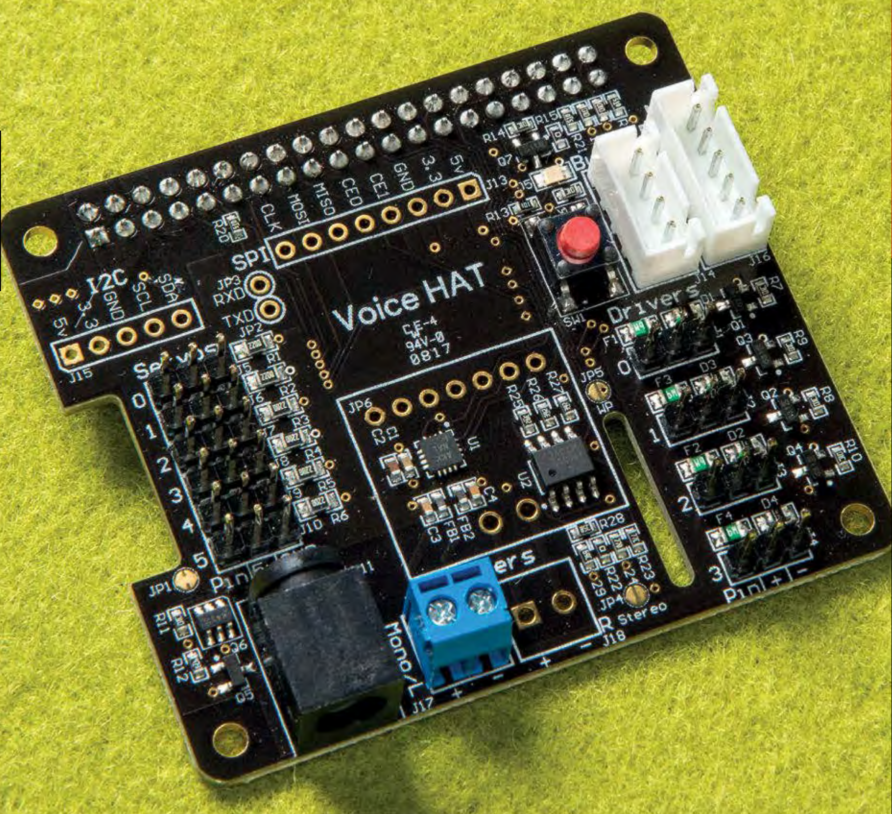
SURVEY RESULTS

Results of maker survey show keen interest in smart technology

Back in January 2017, we asked the Raspberry Pi community to respond to a Google survey canvassing opinions from makers.

"We conducted a survey to understand how makers would want to use artificial intelligence in their work," says Kristine Kohlhepp, a User Experience Researcher working on AIY Projects at Google. "These results, along with other feedback we gather from the community, will help us figure out where to take AIY Projects next."

The team at Google are busy analysing this data and plan to share more results via their website soon. In the meantime, you can get a sneak peek (to the right) of some of the results.



The Voice HAT included with this month's AIY Projects kit

MAKER SURVEY IN NUMBERS

OVER 15,000 MAKERS RESPONDED

63% INTERESTED IN MACHINE LEARNING

MACHINE LEARNING MAKERS

Over half of makers wanted to learn more. about machine learning. Of those...

80% HAVE WORKED ON AUTOMATION PROJECTS

69% HAVE WORKED ON IOT PROJECTS

57% IDENTIFY AS ENGINEERS

RASPBERRY TURK

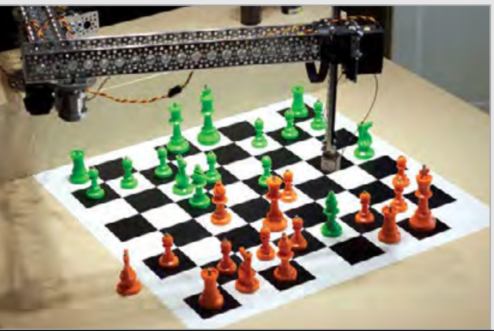
This chess robot moves the pieces by robotic hand

The Raspberry Turk makes a move using its robotic arm, which also removes captured pieces

Inspired by the 18th-century Mechanical Turk, Joey Meyer has built a chess-playing robot based on a Raspberry Pi 3 (**raspberryturk.com**). OpenCV is used to interpret the overhead view of the board from a Camera Module. A Python program determines when it's the Raspberry Turk's turn, and it then makes a move using its robotic arm, equipped with an electromagnet to lift the pieces.

After the Turk spends one second 'thinking' using the Stockfish chess engine, Joey tells us: "It takes between 20-40 seconds to make its move depending on how far it has to reach and whether or not it is capturing a piece (needs to make two trips)."

Not only does the Raspberry Turk look impressive, but nobody has managed to beat it yet. "The Stockfish engine is so powerful that even running on a computer as small as the Pi, it can play at Grandmaster level."



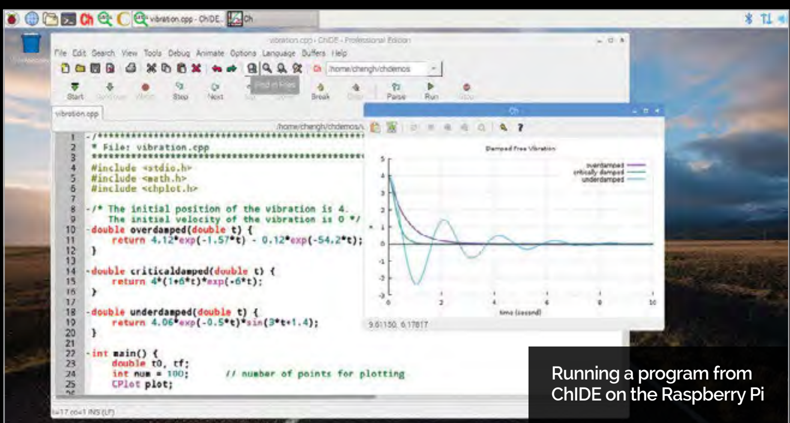
FREE C/C++ LEARNING TOOL

Cross-platform C/C++ interpreter for Raspberry Pi

SoftIntegration Inc has launched Ch 8.0, "the most complete cross-platform C/C++ interpreter", available free for Raspberry Pi users as part of the C-STEMbian OS based on Raspbian (**magpi.cc/201JVNO**). The latter also includes a WiringPi binding for Ch to enable easier access to the Pi's GPIO pins in C/C++ programs.

Equipped with a user-friendly IDE (ChIDE), Ch is designed to help students learn coding, robotics, and maths. Wayne Cheng, SoftIntegration's VP for Application and Business Development, points out: "Ch is a superset of C with salient features of C++ and many extensions including computational arrays, plotting, QuickAnimation, and advanced numerical computing capabilities. Ch is also a command shell, a very high-level language (VHLL) for shell programming and scripting for rapid prototyping. Therefore, Ch can be used not only for learning C/C++, but also for serious software development and applications."

As well as 1999 ISO C Standard (C99) and C++ classes, Ch supports many industry standards including POSIX, X11, and OpenGL. An Embedded version of Ch is also available, enabling users to embed Ch into C/C++ applications as a scripting and programming engine.



Running a program from ChIDE on the Raspberry Pi

PI WARS 2017

Roboteers from around the world gathered in Cambridge for the Pi Wars robotics challenge

Robot makers from around the UK and further afield assembled for Pi Wars 2017, the annual challenge-based robotics competition starring Raspberry Pi robots.

This year's challenge, on 1-2 April, was held at the Cambridge Computer Laboratory. Pi Wars is open to teams from around the world.

The competition typically sees around 30 teams from schools, families, and hobbyists compete for two full days of robotics fun and games.

Pi Wars 3.0 was moved from December to April, giving student teams an extra three months to build and test robots (and making it easier for students starting term in September to enter).

The event was held over two days, with the Saturday being for Schools/Kids Club; and Sunday for Beginners, Intermediates & Pros/Veterans.

Competitors took part in an Obstacle Course, Line Follower, Straight-Line Speed Test, Skittles, Slightly Deranged Golf and The Minimal Maze. There were also prizes awarded for Artistic and Technical Merit.

ExaBot came top on Schools day, while Brian Corteil's '2 wheels or not 2 wheels' won on Sunday. For full results, see **magpi.cc/2oaeUqV**.

SCHOOL WINNERS

1. ExaBot (Michael Syree)
2. Kenilworth School Computer Club (Edward Powell, Daniel Sendula)
3. Cranmere Code Club (Richard Hayler)
4. Adams' Federation (Kevin Brace)
5. QM-Pi-Bot (Jon Witts)

PROS/VETERAN WINNERS

1. 2 wheels or not 2 wheels (Brian Corteil)
2. Ipswich Makerspace / Robot Tractorbot (Keith Ellis)
3. Metabot3 (John Palombo / Lance Robson)
4. KEITH 3 – 'Mostly Harmless' (Harry Merckel)
5. Hitchin Hackspace (David Booth)

PROS/VETERAN WINNER

Brian Corteil's '2 wheels or not 2 wheels' won the Pros/Veterans event

SCHOOL WINNERS

Michael Syree and the ExaBot team won first place in the Schools day

AIY PROJECTS

FREE KIT NOT INCLUDED WITH DIGITAL EDITION

Build an intelligent device that you can talk to

This month, Google is launching a new initiative called AIY Projects to bring do-it-yourself artificial intelligence (AI) to the maker community. AIY Projects is a series of open-source designs that demonstrate how easy it is to add AI to your projects.

Then you will learn how to add voice commands to your own projects. For example, you can register for commands, such as "turn the lights on" or "robot, turn right and move forwards". In the kit, Google has included an accessory board, called Voice HAT, that is loaded with breakout pins to wire up a variety of sensors and components, as well as the microphone and speaker.

> "Google can't wait for makers to build intelligent devices that solve real world problems"

In this issue of *The MagPi*, we are thrilled to present the very first project, a free kit that lets you explore voice recognition and natural language understanding.

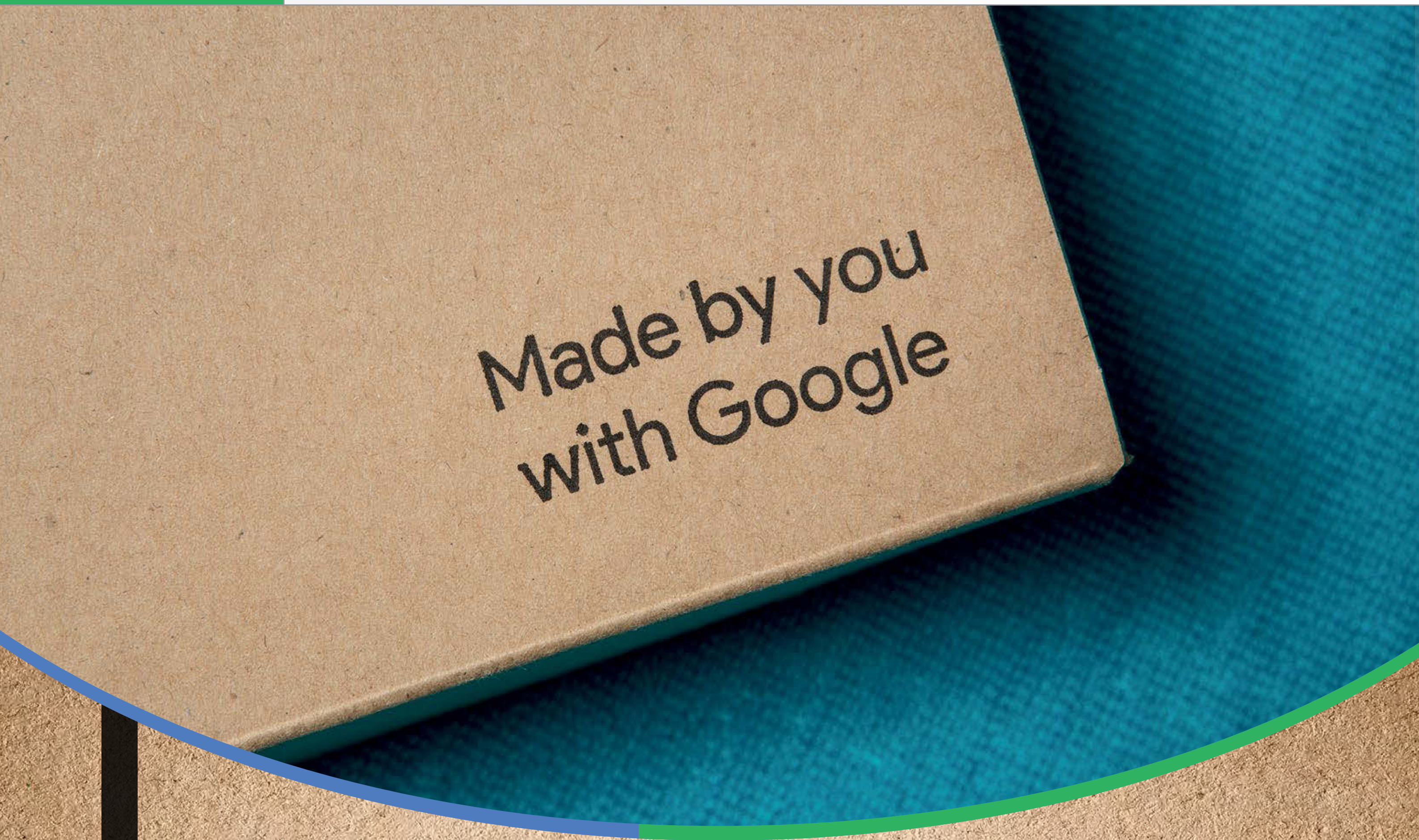
You will build a cardboard device that uses the Google Assistant to answer questions, like "how far away is the moon?" or "what is 18 percent of 92?".

Google can't wait for makers to build intelligent devices that solve real world problems and share them back to the community to inspire others.

Your kit is a fantastic way to add voice control to your projects, and start exploring what's possible with AI.

#AIY

Your AIY Projects kit comes in four colours: Red, Blue, Green, and Yellow. Tweet us yours using the hashtag **#AIY**



MADE BY YOU WITH Google

The AIY Projects team chat to us about the making of this incredible Raspberry Pi kit

"Natural Human Interaction is this idea of being able to communicate with an electronic device the same way you and I are talking right now," says Billy Rutledge, Director of AIY Projects at Google. We've caught up at Raspberry Pi Towers to discuss the AIY Projects kit, and the future of artificial intelligence with the maker community.

"We're all familiar with graphical user interfaces (GUI)", notes Billy. "Well, building a VUI is now the big thing." Voice has become "very popular" in the last year, says Billy. "Not just with consumer products, but also as a set of tools for device makers."

Google wants to help makers familiarise themselves with voice interfaces, but it's also really keen to tap into the creative prowess of the maker community.

"We're excited to put the kits out into the world and see what people make with them," says Kristine Kohlhepp, a User Experience Researcher working on AIY Projects at Google. "We've done a lot of research to make sure people can assemble the kit and figure out how to make it work."

"The initial project is just an opener," reveals Blaise Agüera y Arcas, Principal Scientist at Google. "It's fun to be able to make a cardboard kit that uses the Google Assistant, but this is about a lot more than just making a lower-cost DIY version of Home."

The future belongs to intelligent devices. Billy says: "At some point soon, we'll see a new generation of devices that you can just walk up to and ask 'what are you and what do you do?' Then you'll have a conversation with it, to use its services in a very easy-to-understand way."

Natural Human Interaction is the term used for this kind of interaction between humans and devices. "A generation or two ago, all of our devices had analogue dials and knobs," explains Billy. "Then there was a shift to digital buttons and displays. Now we are moving to a human interface where you simply have a conversation with the device."

It's also important to ease people into AI as part of their natural interaction with electronic devices, alongside touching buttons and screens. "We can easily become distracted by personification of these kinds of systems," says Blaise.

> We want to show you how easy it is to use AI

It's important for makers to realise that VUIs are something they can create, and use, in their projects. "I think letting the makers see how easy it is to put AI, specifically Natural Human Interaction capabilities, into their projects will be a great thing," says Kristine.

"We want to show you how easy it is to use AI, and then share back with us to inspire new project ideas and keep the whole cycle going," Kristine continues.

"My top-secret plan is to build more engineers," discloses James McLurkin, Senior Hardware Engineer of AIY Projects at Google. "Getting kits like this out into the world with Raspberry Pi allows us to build the things that then create more engineers." AIY Projects enables young makers to explore the possibilities with AI. "So this is very exciting for us," says James.

"What's interesting about the maker environment is what happens when we shut up, and listen, and see what people try," says Blaise. Historically, there have been many 'Hello World' types of starter projects for various programming languages and platforms, and in recent years we've seen exciting new hardware like the Raspberry Pi emerge. Now there is AI, another technology for makers and developers to add to their projects. AIY Projects brings these three things together, which will be "super interesting," reckons Blaise.

"I don't know what will come out from the mixture of those, but I'm very keen to see."

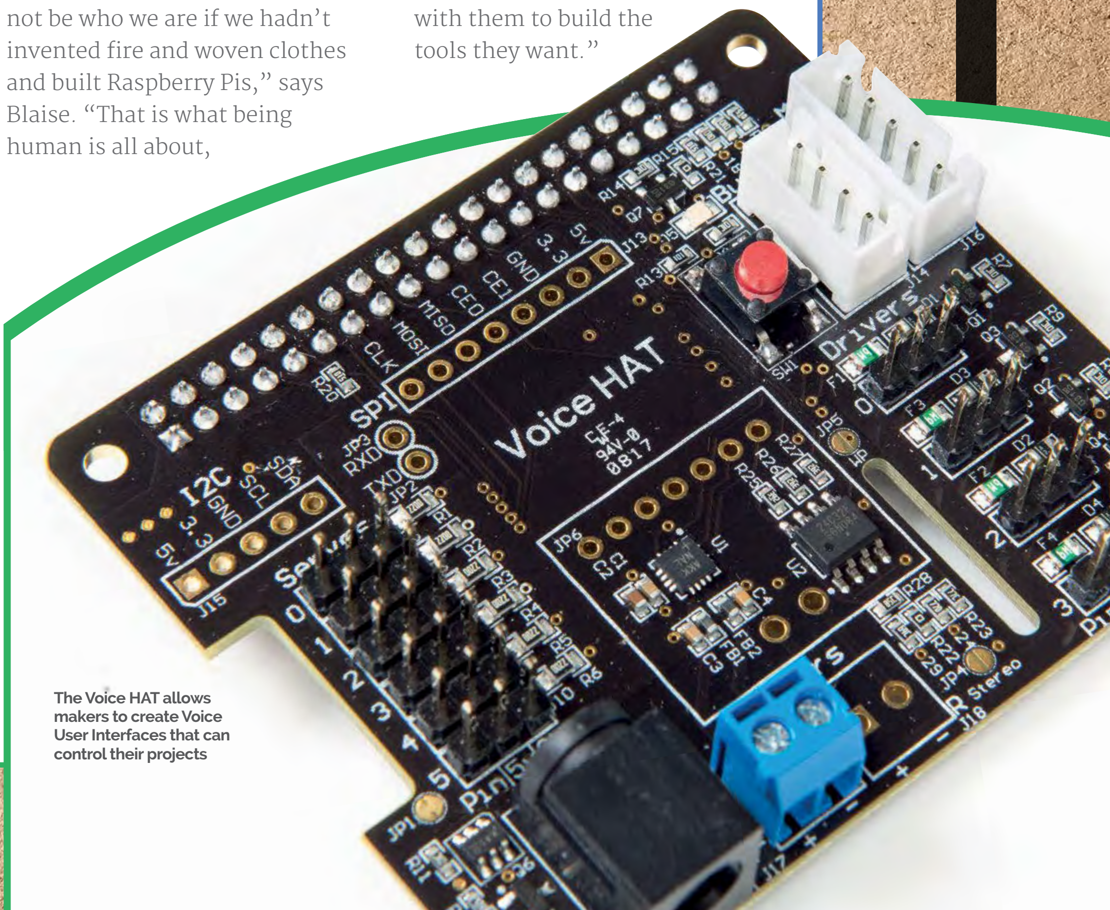
"I view this as an essential component in a maker's toolkit," Billy tells us. "We want makers to see that using AI is not hard or complex."

The Australian artist Stelarc has said that technology constructs our human nature. "We would not be who we are if we hadn't invented fire and woven clothes and built Raspberry Pis," says Blaise. "That is what being human is all about, and that's what distinguishes us from the other animals. So I don't like this idea that talks about AI as a competitive landscape of human exceptionalism, and ways that it is being eroded. That really misses the point of what all this is about."

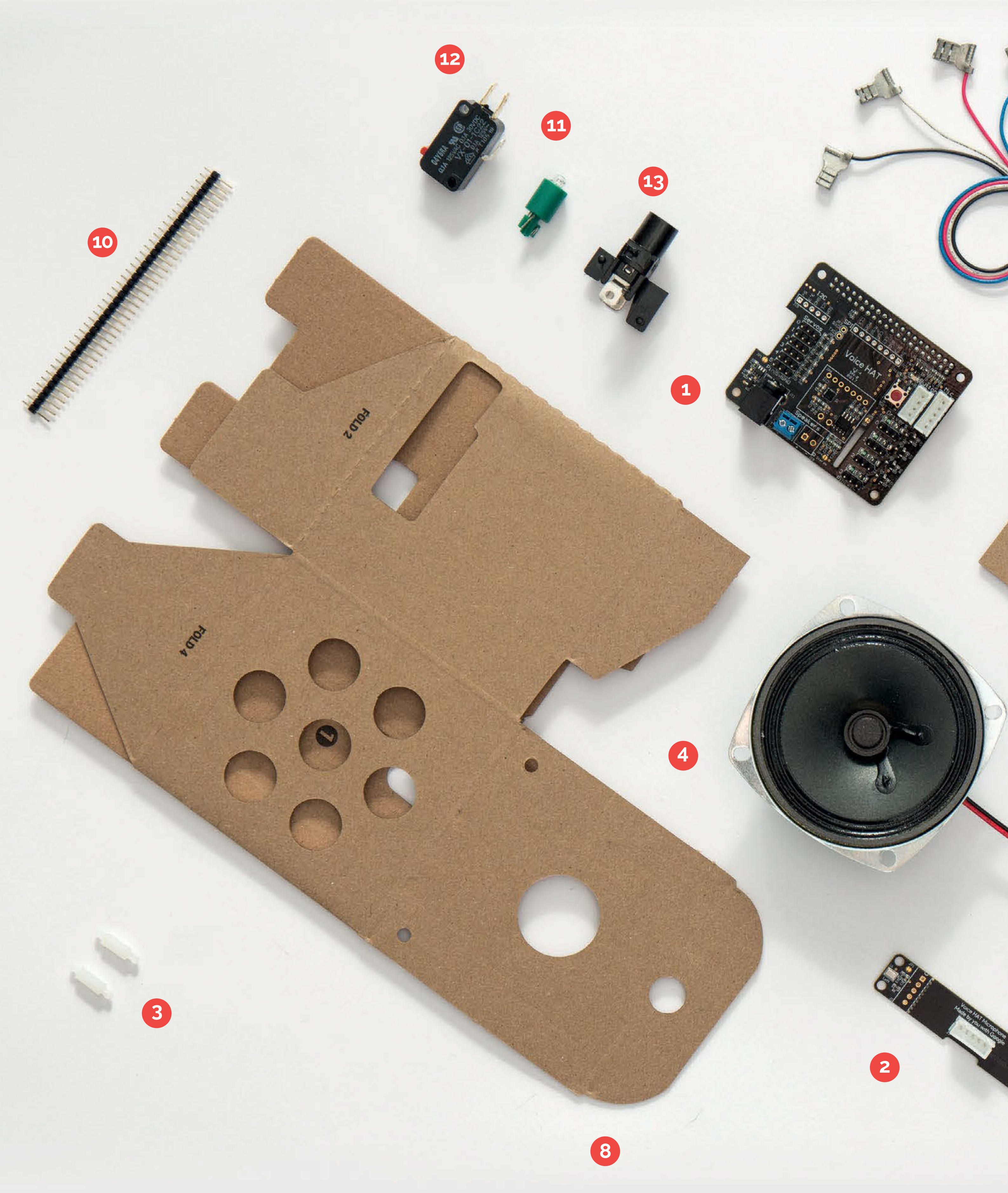
The AIY Projects team meet up with The MagPi and Eben Upton at Pi Towers

"This first kit showcasing voice is just the start of our effort to bring Google AI to the maker community," reveals Billy. "Our projects will largely focus on Natural Human Interaction." Following voice, we intend to feature projects with vision, motion, and learning."

Google wants makers to add AI to their own projects, and share their results with others. "We want to learn what this community needs," says Billy, "and then work with them to build the tools they want."



The Voice HAT allows makers to create Voice User Interfaces that can control their projects


12
11
13
10
1
4
3
2
8
FOLD 2
FOLD 4
Voice HAT

YOUR **AIY PROJECTS** VOICE KIT

Construct your free AIY Projects voice kit and explore natural language recognition

aiyprojects.withgoogle.com

Inside the kit will be the components you need to build a voice-capable device with Raspberry Pi.

Open the box and you'll find two pieces of cardboard, an arcade-style button, a speaker, and some cables, along with a HAT (Hardware Attached on Top) board and another narrower board. One is to connect all the accessories together; the other is a stereo microphone.

All of these components fit together to build the AIY Projects kit: a small cardboard device with a colourful button on the lid. You press the button, or clap your hands (or create a custom trigger), and speak out loud to ask the device a question. The speaker, at the front, then announces the answer.

Tucked inside the cardboard device are all the components. Use the Bill Of Materials list below to check you have all the components.

BILL OF MATERIALS

1. Voice HAT accessory board
2. Voice HAT microphone board
3. 2× plastic standoffs
4. 3-inch speaker (wires attached)
5. Arcade-style push button
6. 4-wire button cable
7. 5-wire daughter board cable
8. 9. Cardboard box and frame
10. 40-pin header (pre-soldered to Voice HAT accessory board)
11. Lamp
12. Microswitch
13. Lamp holder

ASSEMBLE THE AIY PROJECTS KIT

Put the parts together to build a voice-enabled device

With all your parts ready, it's time to build the AIY Projects voice kit. The aim is to assemble all the included parts (and a Raspberry Pi board) and create a small cardboard device with a button on top.

This project is a relatively easy build, and you won't need to solder any of the components. Be careful to line up the wires correctly, especially the wires for the button. It's also a good idea to take a close look at the Voice HAT accessory board (the larger board). The Voice HAT is the heart of the AIY Project kit, and everything connects to it. It also provides breakout GPIO pins, organised into two blocks: Servos and Drivers.

You'll connect the Voice HAT accessory board to your Raspberry Pi via the GPIO pins. The Raspberry Pi is the brains of the outfit: it connects to Google's cloud services through a local Python application. The Python source code is provided with the software image, as well as on GitHub.

Also take a close look at the smaller microphone board, which enables the device to hear you speak.

But first, we need to get it all assembled. The first step is to mount the Voice HAT accessory board to your Raspberry Pi, and then connect the speaker and microphone. Then you'll move on to folding the cardboard case and placing the components inside. Finally, you'll assemble the arcade-style button and secure it (and the microphone) to the case.

Ready? Let's start building your kit.

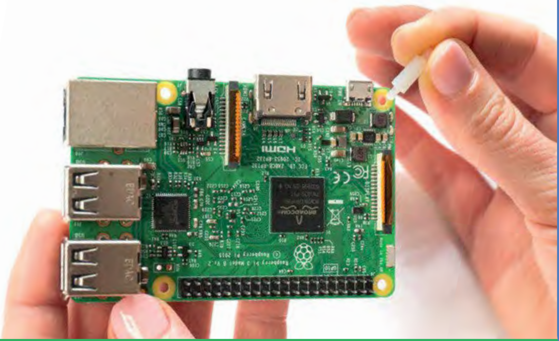
You'll Need

- Raspberry Pi 3
- Small, needle-nose pliers
- Phillips 00 screwdriver
- Scotch tape

SET UP THE VOICE HAT

1 INSERT THE STANDOFFS

Start with the two standoffs. These are the small plastic cylinders, and they fit into the yellow mount holes on the Raspberry Pi board. Insert the standoffs into the two yellow holes on the opposite side from the 40-pin GPIO header (on the same side as the HDMI connection). Push them firmly, and they will hold in place.

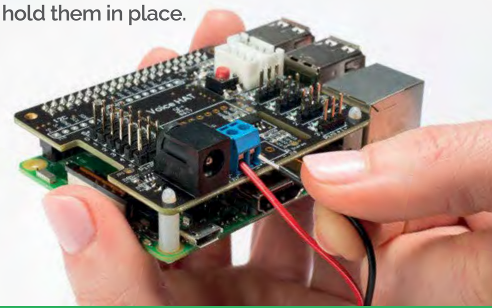


2 SECURE THE HAT

Now get the Voice HAT accessory board and attach it to the GPIO pins on the Raspberry Pi board. Carefully line up the GPIO connector on the Voice HAT accessory board with the pins of the GPIO header on the Raspberry Pi. Gently press down to make sure the Voice HAT accessory board is secure. Press down on the spacers on the other side of the board to snap the boards together.

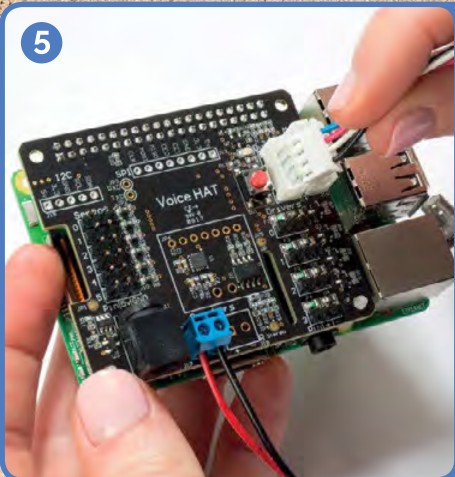
3 ATTACH THE SPEAKER WIRES

Take a close look at the Voice HAT accessory board and find the blue terminal with two small screws. This terminal is the speaker connection (it has 'Speakers' printed above it on the board). Each of the two connections has a small '+' and '-' symbol printed below. Find the speaker with the red and black wires attached. Insert the red wire into the positive '+' terminal on the Voice HAT accessory board. Now add the black wire into the negative '-' terminal. They won't be fixed yet, so hold them in place.



4 SCREW IN THE WIRES

At this point, the two wires will be sitting in the sockets unsecured. Hold the wires in place, and gently turn each screw in the socket using a Phillips 00 screwdriver. Gently tug on the wires to make sure they're secure. Now place the speaker to one side of the board so you can access the other components.

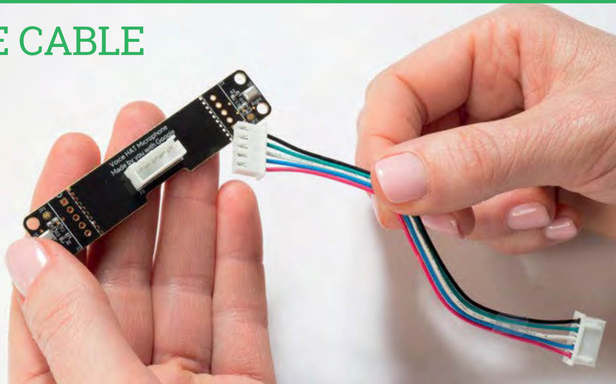


5 THE BUTTON CABLE

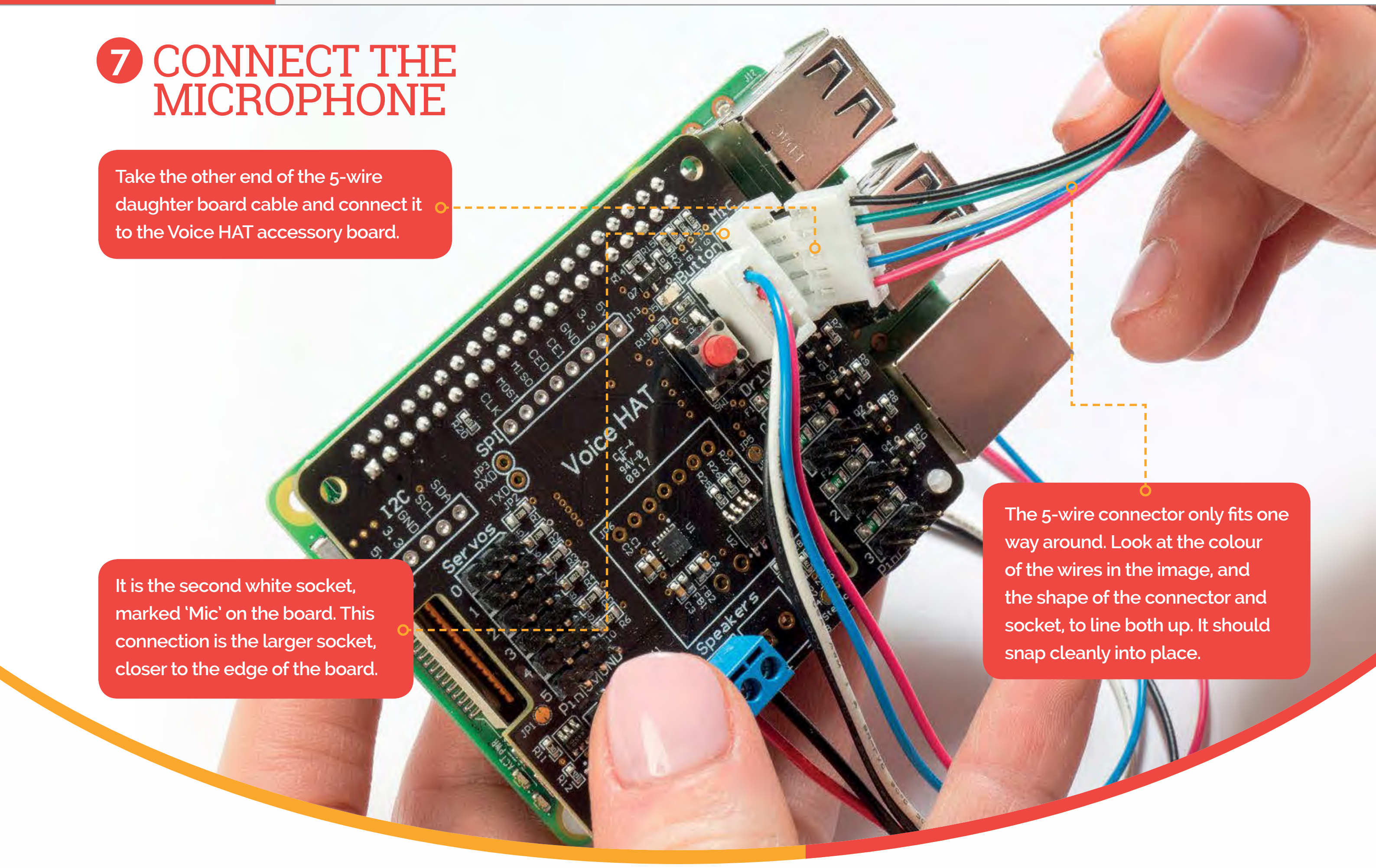
Find the 4-wire button cable: it has a white male connector on one end and four separate wires with metal contacts on the other. Insert the white plug into the matching white socket marked 'Button' on the Voice HAT accessory board (it is the one nearest to the red button). The cable will only go in one way around, so don't force it. Check that the colours of the cable match the image. Don't worry about the four separate wires with metal contacts; we'll come back to these later.

6 THE MICROPHONE CABLE

Find the Voice HAT microphone board and the 5-wire daughter board cable. The cable has matching white plugs on either end. Both ends of the cable are identical, so take either end of the 5-wire connector cable and slot it into the Voice HAT microphone board. It will only fit one way around. Snap the cable in, but don't force it.



7 CONNECT THE MICROPHONE



Take the other end of the 5-wire daughter board cable and connect it to the Voice HAT accessory board.

It is the second white socket, marked 'Mic' on the board. This connection is the larger socket, closer to the edge of the board.

The 5-wire connector only fits one way around. Look at the colour of the wires in the image, and the shape of the connector and socket, to line both up. It should snap cleanly into place.

ASSEMBLE THE BOX

FOLD THE CARDBOARD

Now let's move on to the box. Find the larger cardboard piece with a bunch of holes on one side (as shown in the image). Fold along the creases, then find the side with four flaps and fold the one marked FOLD 1.

9 SECURE THE BOX

Do the same for the other folds, tucking FOLD 4 underneath to secure it in place. Now set it aside.

10 FOLD THE FLAPS

Find the other cardboard piece that came with your kit (as shown in the picture). This piece will build the inner frame to hold the hardware. Fold the flaps labelled 1 and 2 along the creases.

11 PUSH IT OUT

The flap above the 1 and 2 folds has a U-shaped cutout. Push it out.

12 FOLD OUT THE FLAP

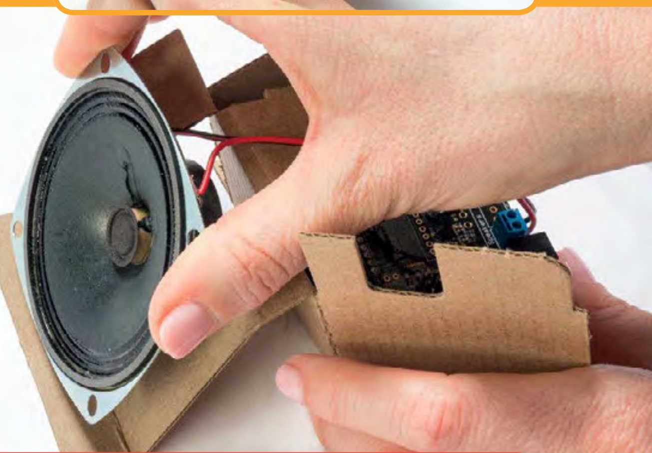
Then fold the rest of the flap outward. Fold the section labelled FOLD UP so that it is flush with the surface you're working with. There's a little notch that wraps behind the U-shaped flap to keep it in place.

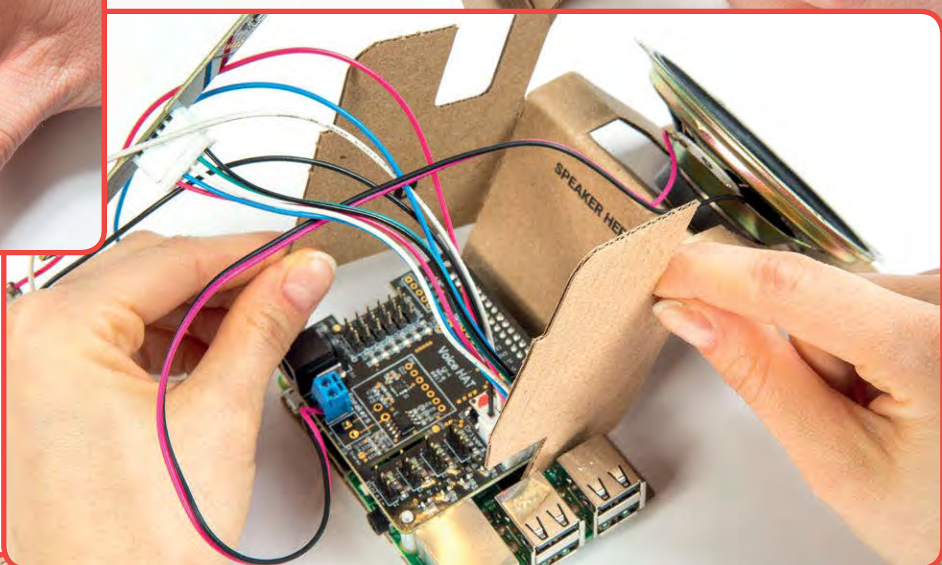
13 CHECK THE FLUSH

The U-shaped flap should lie flush with the box side. At this point, the cardboard might not hold its shape. Don't worry: it'll come together once it's in the box.

14 ADD THE SPEAKER

Find your speaker (which is now attached to your Raspberry Pi 3). Slide the speaker into the U-shaped pocket on the cardboard frame.





15 SLIDE INTO THE RASPBERRY PI

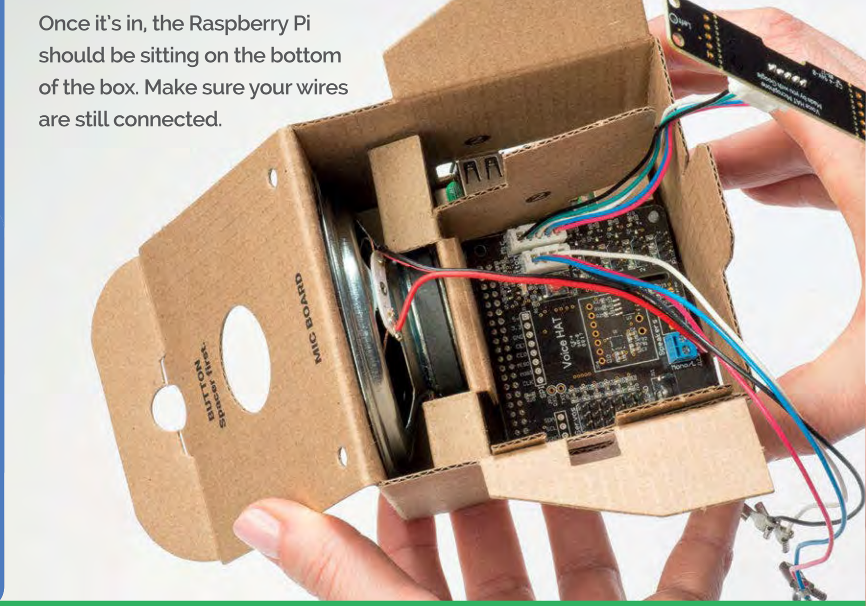
Turn the cardboard frame around. Take the Pi + Voice HAT hardware and push it into the bottom of the frame below flaps 1 and 2 (pictured). The cardboard frame should expose the USB ports of the Raspberry Pi.

16 PUT IT ALL TOGETHER

It's time to put the build together. First, remove the SD card from your Raspberry Pi to prevent damaging it. Now take the cardboard box you assembled earlier and find the side with the seven speaker holes. Slide the cardboard frame and hardware into the cardboard box. Ensure that the speaker is aligned with the box side that has the speaker holes.

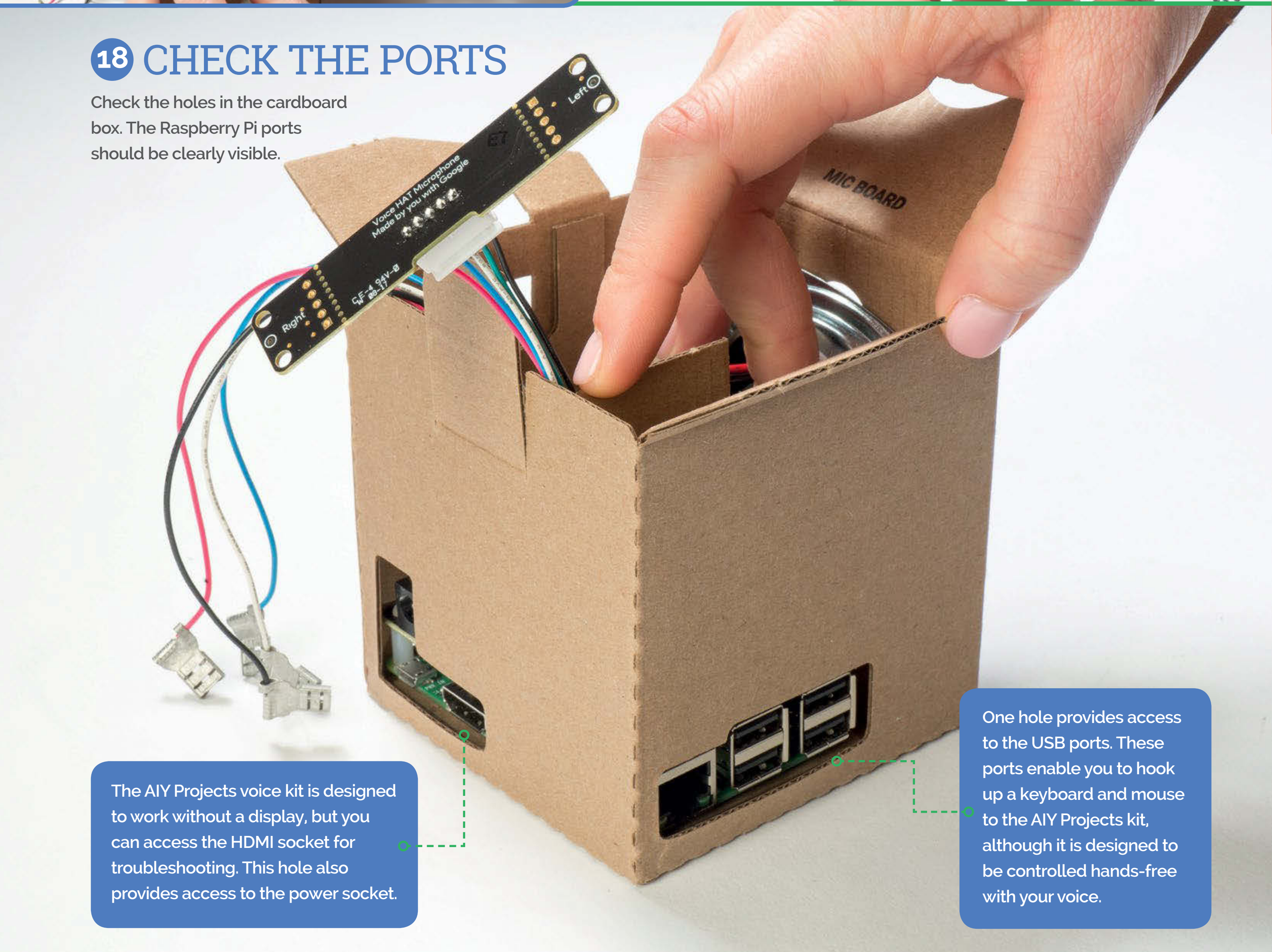
17 CHECK THE WIRES

Once it's in, the Raspberry Pi should be sitting on the bottom of the box. Make sure your wires are still connected.



18 CHECK THE PORTS

Check the holes in the cardboard box. The Raspberry Pi ports should be clearly visible.



The AIY Projects voice kit is designed to work without a display, but you can access the HDMI socket for troubleshooting. This hole also provides access to the power socket.

One hole provides access to the USB ports. These ports enable you to hook up a keyboard and mouse to the AIY Projects kit, although it is designed to be controlled hands-free with your voice.

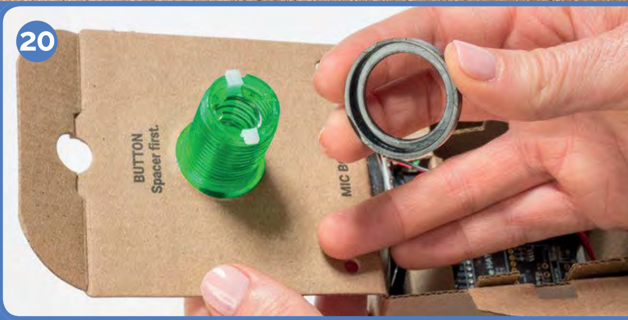
19 PACK THE ARCADE BUTTON

Find your arcade button. There will be a plastic button, a spacer, and a nut. If they're connected, unscrew the nut and spacer from the main button and place them to one side.

ADD THE BUTTON

Insert the plastic button into the top flap of the cardboard box from the outside in. The pushable button side should face outward, with the larger screw on the inside; i.e. the side marked 'BUTTON Spacer first.'

20



21 SECURE THE BUTTON

Now take the spacer and add it to the screw. Next, screw in the washer to secure the button in place. Make sure that the spacer sits between the washer and the cardboard lid.

22 THE BUTTON LAMP

Next, we're going to assemble the button lamp. This lamp lights up the button and displays the status of the device. There are three principal components to use: Lamp, Black Microswitch, and Black Lamp Holder. Make sure you have all three to hand.

23 INSERT THE LAMP

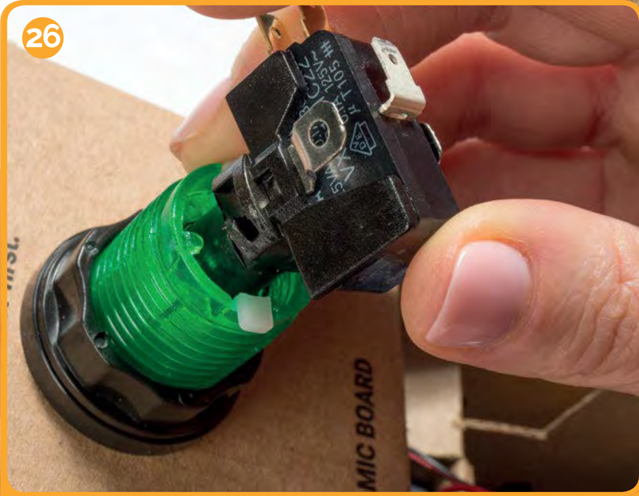
Insert the Lamp into the Black Lamp Holder. Look at the image to see which way around it fits. The light should face outwards. You may need to adjust the rotation later to get a good connection.

24 ATTACH THE LAMP

Now attach the Black Lamp Holder (with the Lamp) to the Black Microswitch component. Two protrusions on the Black Lamp Holder match the indentations on the Black Microswitch. Observe the position of the red button on the Black Microswitch component. Clip the two parts together.

25 PLACE IN BUTTON

You now have a completed lamp with a microswitch mechanism. Look at the button to see spacers that press down on the red microswitch. When you press the button, it pushes the microswitch. Insert the lamp into the button.

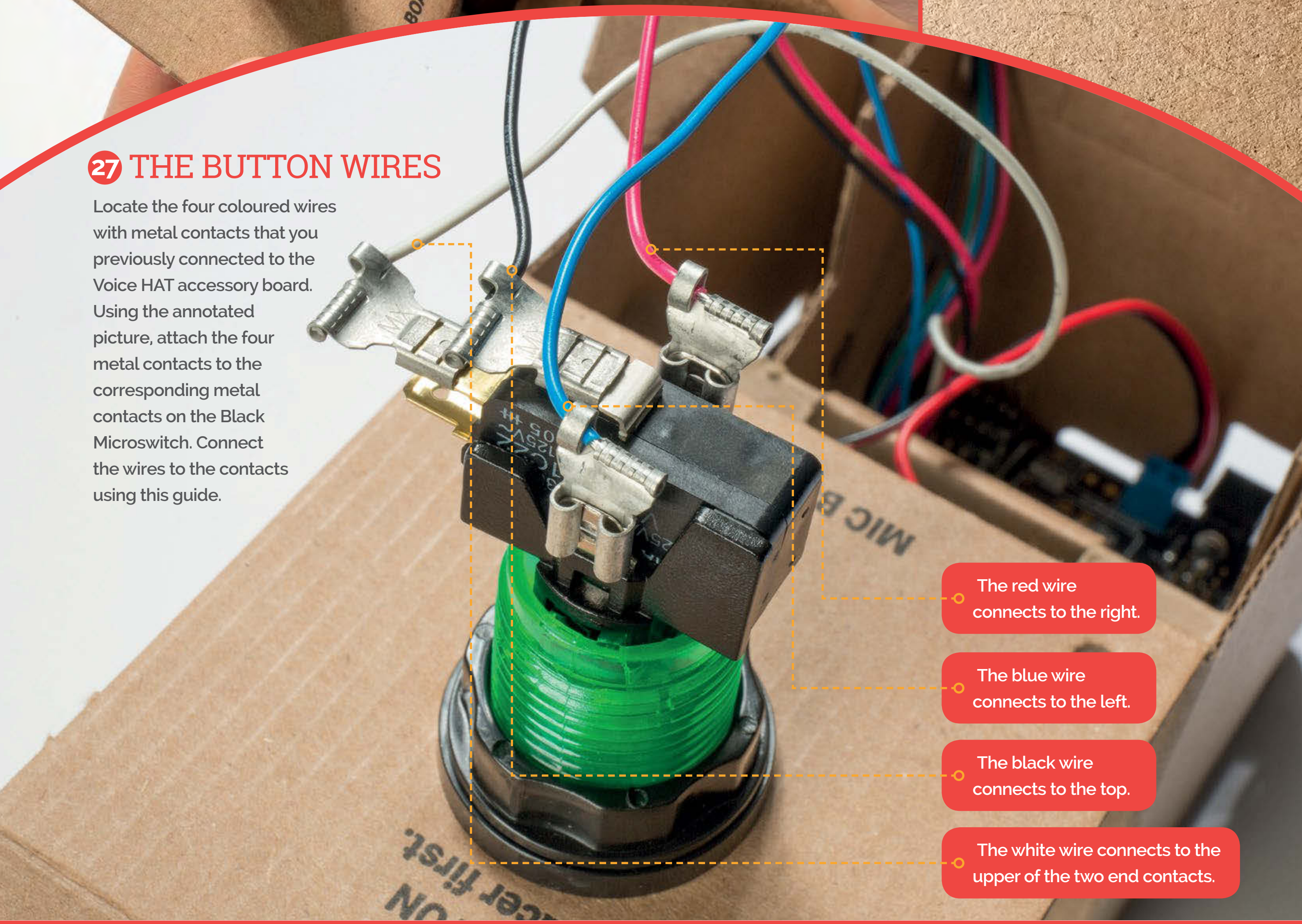


ROTATE THE LAMP

Now you secure the lamp in place by carefully rotating it clockwise (rightward). Push it firmly into place, but don't force it. It should rotate and click together to form a secure button unit.

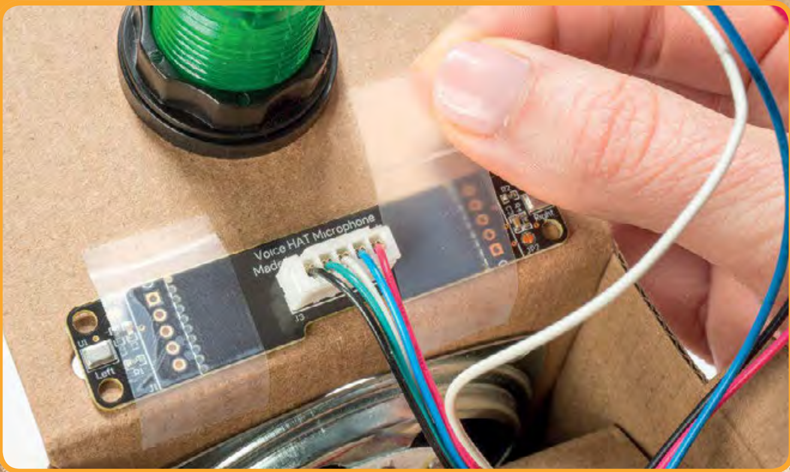
27 THE BUTTON WIRES

Locate the four coloured wires with metal contacts that you previously connected to the Voice HAT accessory board. Using the annotated picture, attach the four metal contacts to the corresponding metal contacts on the Black Microswitch. Connect the wires to the contacts using this guide.



28 TAPE THE MICROPHONE

Next, we use Scotch tape to secure the Voice HAT Microphone board to the top flap. The board sits below the button on the top flap, with the two microphones aligned with the two holes. Check that the holes, on the other side, are aligned with the two microphones before taping down the board. Use some Scotch tape to fix the microphone board to the top flap of the cardboard.



29 CHECK THE MICROPHONE

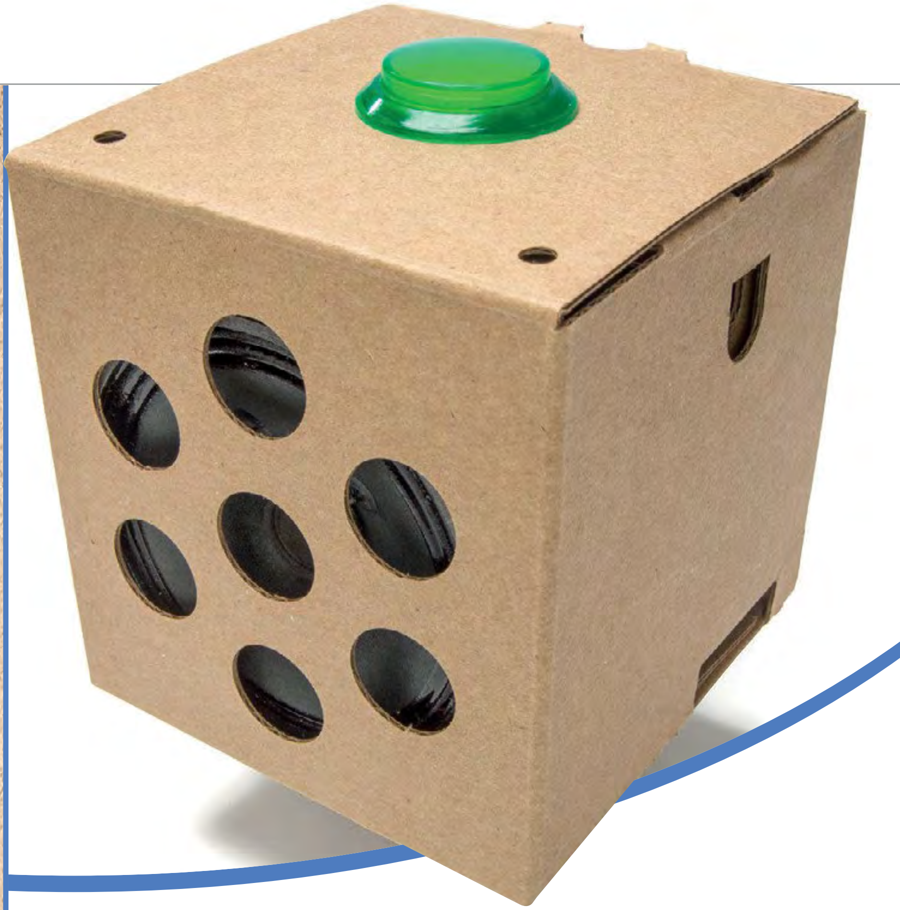
Turn the flap around, and double-check that the microphones match the cardboard holes. Correct alignment ensures that the microphone board can clearly hear you when you start issuing voice commands.

30 THE FINISHED BUILD

That's it. Your voice kit is assembled, and you can now start installing the software and using the Google Assistant to answer your questions. Fold the top flap down to close the box up and admire your handiwork.

SET UP THE AIY PROJECTS SOFTWARE

Google Assistant SDK



Info & Updates:

aiyprojects.withgoogle.com/voice/

Use the Google Assistant SDK to create a device that answers your questions and helps you get things done

You'll Need

- Assembled AIY Projects kit
- MicroSD card
- AIY Projects image file
- Google account

You now have a fully assembled cardboard device that is almost ready to respond to your questions.

To do this, you'll set up a Google Developer project and activate the brand-new Google Assistant SDK.

Google has provided a Python sample application that demonstrates how to use the Google Assistant SDK. To set it up you'll need to download the AIY Projects image file from **aiyprojects.withgoogle.com/voice/**.

Burn the image file to a microSD card using a program like Etcher (**etcher.io**). This software copies the image file to the SD card (see 'Burn SD cards with Etcher', **magpi.cc/2fZkyJD).**

>STEP-01
Set up the SD card

Insert the SD card and connect the power. The software image is based on Raspbian, so it feels familiar. Alternatively, you can replace it with a software image based on Android Things. For more information on using Android Things, visit the AIY website (**aiyprojects.withgoogle.com/voice**).

>STEP 2
Set up the network

Click on the Networking icon and choose your network. Now double-click the 'Check WiFi' icon.

>STEP 3
Check the audio

Next, double-click the 'Check audio' icon. If there is a problem, check the Voice HAT accessory board's connection to the speaker and microphone.

>STEP 4
Set up the cloud

Open the Chrome web browser and click on Google API Console in the Bookmarks bar (or enter **console.cloud.google.com** in the URL field). Enter your Google Account ID and click Next (or create a new account). Enter your password and click 'Sign in'.

>STEP 5
Create a project

Click the menu icon marked 'Project' and choose 'Create project'. Give the project a name, like 'aiyproject', and click Create.

>STEP 6
Enable the services

Click on the 'Products & services' menu and choose API Manager. Search for and click Enable API at the top of the screen. Click Google Assistant API and click Enable. While you're here, select Speech API and click Enable.

>STEP 7
Set up OAuth consent

Click on Credentials in the API Manager sidebar (on the left) and choose the 'OAuth consent' screen tab. Enter the name of the app, such as 'AIY Project', in the 'Product name shown to users' field. Click Save.

>STEP 8
Create OAuth client ID

Click Create Credentials and select 'OAuth client ID'. Choose Other and click Create (it will be named 'Other client 1' by default). A window will appear with your client ID and client secret.

>STEP 9
Download credentials

Look to the right of the 'Other client 1' item and locate the download icon. Click the download icon to save the JSON file.

>STEP 10
Rename credentials

Open a Terminal window and enter:

```
cd Downloads
ls
```

You will see a file called **client_secret_** with a long ID number after it. We need to rename the file to **client_secrets.json** and move it to the home folder. Enter this:

```
mv client_secret
```

Press **TAB** to autofill the file name and add **/home/pi/client_secrets.json** as the destination. The command looks like this:

```
mv client_secret_[123etc]
/home/pi/client_secrets.json
```

>STEP 11
Create service account

Head back to Google Cloud Console in the browser and click Create Credentials again. This time choose 'Service account key'. Click the 'Service account' menu and choose 'New service account'. Give it a name, such as 'AIY Projects', and change the Role to Project > Owner. Make sure the Key type is JSON and click Create. The key is downloaded to your computer.

>STEP 12
Create credentials

Find the file you've downloaded. You need to rename the file **credentials.json** and place it in your home folder.

```
mv aiyproject-[123etc]
/home/pi/credentials.json
```

>STEP 13
Activity controls

In your web browser, visit **myaccount.google.com/activitycontrols**. Sign in using the same account and turn each of these items on:

- **Web & App Activity**
- **Location History**
- **Device Information**
- **Voice & Audio Activity**

These are required for the Google Assistant to run. Now close the browser window.

>STEP 14
Start the voice unit

Double-click the 'Start dev terminal' icon and enter:

```
src/main.py
```

A Request for Permission window appears. Click Allow. Copy the code from the web browser and paste it to the command line in the Terminal window. Press **RETURN** and you will see 'Press the button on GPIO 23 then speak, or press Ctrl+C to quit…'

>STEP 15
Push the button

Push the button on the top of your device and it will display:

```
[INFO]main:listening...
[INFO]main:recognizing...
```

Ask the device a question, such as "what is the weather in Cambridge?"

The AIY Projects voice kit speaks out the answer. Keep pushing the button and asking questions. Press **CTRL+C** when you're done.

>STEP 16
Clap your hands

You can set up the voice-recognizer to respond to hand claps. Enter this command in the Terminal:

```
nano /home/pi/.config/voice-
recognizer.ini
```

Now remove the # before this line of code:

```
trigger = clap
```

Press **CTRL+O** and press **RETURN**, then **CTRL+X**.

>STEP 17
Listen on boot

Now set up your device to start listening automatically. Double-click 'Start dev terminal' and enter the following command:

```
sudo systemctl enable voice-
recognizer
```

The Terminal displays 'Created simlink from' messages. Finally, reboot your Raspberry Pi.

androidthings

Android Things is a new OS for connected devices that is fully compatible with the AIY Projects kit that you received. Developed by Google, it is a ready-to-use solution for building connected devices.

Developers can use existing Android development tools, security updates, APIs, resources, and a thriving developer community. It also includes new Android framework APIs that provide low-level I/O and libraries for common components like temperature sensors and display controllers.

In addition, a wide range of Google APIs and services, such as Google Play services, TensorFlow, and Google Cloud Platform are available on Android Things. Developers can push Google-provided OS updates and their own app updates, using the same OTA infrastructure used on Google's own products.

To get started on building your kit with Android Things, visit the AIY Projects website (**aiyprojects.withgoogle.com/voice**). More information about Android Things is available on the developer website (**developer.android.com/things**).

CREATE A VOICE USER INTERFACE FOR YOUR PROJECTS

Google Cloud Platform

Swap out traditional interfaces with a custom voice control using your AIY Projects voice kit

You'll Need

- Google Cloud Speech API

By now, you have built a device that embeds the Google Assistant. That's cool, but it's just the beginning.

With Google Cloud Speech API, you can create an interactive, custom Voice User Interface (VUI) for your project.

This enables you to explore a new generation of devices that you can have a conversation with, without the need for remote-control devices (such as joysticks or smartphone apps). Let's reconfigure the kit to use the Google Cloud Speech API.

> STEP 1
View the source

The source code for the voice-recognizer app is part of the image that you've just installed. You can view the Python source code in the **/home/pi/voice-recognizer/raspi/** directory.

> STEP 2
View the config

The application can be configured by adjusting the properties found in **/home/pi/.config/voice-recognizer.ini**. This file lets you configure the default activation trigger and which API to use for voice recognition. Don't worry: If you mess it up, there's a backup copy kept in **home/pi/voice-recognizer-raspi/config**.

> STEP 3
Check Cloud Speech API

You should already have the **credentials.json** file in your home directory. We installed this in the previous tutorial. This provides access to the Google Cloud Speech API. Double-click the Check Cloud icon on the desktop to test the Google Cloud Speech API. If it says 'Everything's set up to use the Google Cloud', close the Terminal window and skip ahead to Step 5.

TensorFlow

TensorFlow is an open-source software library for machine learning. It was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organisation. You can learn more about TensorFlow, and how it can be used to add intelligence to your own projects, at **tensorflow.org**. To see how you can use TensorFlow to enable on-device audio detection, visit the AIY Projects website at **aiyprojects.withgoogle.com/voice**.

GETTING HELP

With so many options to explore with this first AIY Project from Google, you should make good use of the vibrant Raspberry Pi community. The Raspberry Pi community is on hand to help you with any issues, and make suggestions for your projects. Head to the Raspberry Pi forums and find the new AIY Projects page.
magpi.cc/1NlH5rQ

HANDY COMMANDS

```
sudo systemctl start voice-recognizer
sudo systemctl stop voice-recognizer
sudo systemctl enable voice-recognizer
sudo systemctl disable voice-recognizer
systemctl status voice-recognizer
```

> STEP 4
Check credentials

If you do not have **credentials.json** in your home directory, open the API Manager in Cloud Console and choose Create Credentials > Service account key. Select your project under Service Account or choose New Service Account (give it a name and set the Role to Project > Owner). Choose JSON as the Key Type and click Create. Rename the file to **credentials.json** and place it in your home directory.

> STEP 5
Check Cloud Speech API

You need to have the Cloud Speech API enabled to use the service. Open the API Manager in Cloud Console and choose Library. Click on Speech API and click Enable (if you see Disable, it's already enabled and you can move on).

> STEP 6
Check billing

You need to have billing set up with Google to use the Cloud Speech API. Open Cloud Console. If you don't have a billing account, click 'New billing account' and go through the setup. Check your project is selected in the Projects menu at the top. Click Products & Services > Billing. To connect or change the billing account, click the three-dot button, then select 'Change billing account'.

> STEP 7
Edit config

It's time to turn on the Cloud Speech API. Editing the config file enables you to switch from the Google Assistant SDK to the Cloud Speech API. Open a Terminal window and enter:

```
nano /home/pi/.config/voice-recognizer.ini
```

Delete the # before this line:

```
cloud-speech = true
```

Press **CTRL+O**, press **RETURN**, then **CTRL+X** to save and exit. Now enter:

```
sudo reboot
```

> STEP 8
Start it up

Open 'Start dev terminal' and enter:

```
src/main.py
```

You can now issue a limited number of commands:

- **What are the three laws of robotics?**
- **Hello**
- **What time is it?**
- **Volume up**
- **Volume down**

Press **CTRL+C** to quit the interaction.

>STEP 9
Create voice commands

You can create new actions in **src/action.py**, and enable them in **src/main.py**. To control an LED that you've connected to GPIO 4 (the leftmost pin marked 0 under Drivers on the Voice HAT Action), add **action.py** code to **src/action.py**:

```
actor.add_keyword:
    actor.add_keyword('light on', action.
GpioWrite(4, True))
    actor.add_keyword('light off', action.
GpioWrite(4, False))
```

Then add the following **main.py** code to **src/main.py** under other calls to **actor.add_keyword**:

```
import RPi.GPIO as GPIO

class GpioWrite(object):

    '''Write the given value to the given GPIO.'''

    def __init__(self, gpio, value):
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(gpio, GPIO.OUT)
        self.gpio = gpio
        self.value = value

    def run(self):
        GPIO.output(self.gpio, self.value)
```

You can discover more information on using the GPIO pins at **aiyprojects.withgoogle.com/voice**.

FANTASTIC IDEAS FOR VOICE CONTROL PROJECTS

What makers around the world are building with the AIY Projects voice kit

By now you will have set up your AIY Projects voice kit, and discovered how to add Natural Human Interaction to your projects. Wow!

The question is: what are you going to build? What projects will be amazing with voice control?

The good news is you're not alone. Thousands of members of the Raspberry Pi community will be building Natural Human Interaction projects. And we want you to share them with us, and each other.

To get the ball rolling, Google has been running a Hackster community for Google Beta Testers. These lucky makers were sworn to secrecy and received an AIY Projects voice kit early.

We asked members of the Hackster group and members of the Raspberry Pi community working in voice control what they thought the best ideas for projects were. Here are their suggestions for projects they'd like to build.

Now it's your turn. Create something amazing and share it with the Raspberry Pi community on Hackster (**magpi.cc/2oPx6nb**).

hackster.io/google/products/aiy

MAGIC MIRROR

Add a Voice User Interface to a mirror display for the ultimate creative assistant

magpi.cc/2nwpxBA

Michael Teeuw's Magic Mirror came first in our Top 50 list of greatest projects.

Perhaps the only thing that could make this project more awesome would be voice control.

The good news is that Magic Mirror has a powerful community behind it. And they've already been busy working with voice interaction.

MMM-Hello-Mirror, by a user known as Matzefication, is a voice control system for the Magic Mirror based on Google Speech Recognizer (**magpi.cc/2nwyGtO**). The AIY Projects kit will make a Voice User Interface a real possibility. Your reflection has never looked smarter.

Adding voice control to the Magic Mirror makes it more naturally interactive

KIT-CAT CLOCK

magpi.cc/2nZxQIX

Kit-Cat Clocks are big in the USA, where they're an art deco classic dating back to 1932.

"I want to try to tie servos to eyes in one of those Kit-Cat clocks, so it will direct its eyes to where the voice is coming from," says maker Paul Langdon. It will detect the voice position of the speaker using the two-microphone array in the Voice HAT microphone board.

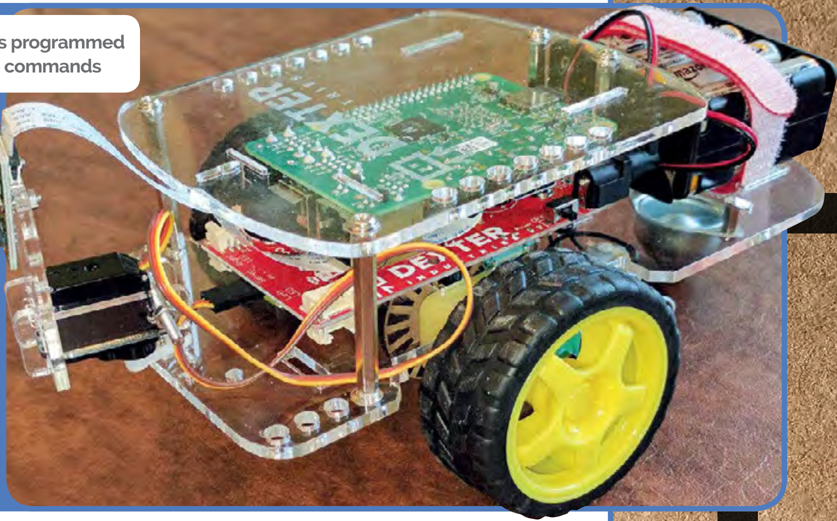
The Kit-Cat Clock is an art deco classic being given AI smarts

VOICE ROBOTICS

magpi.cc/2oLP8dG

Integrating Google Voice API with robotics projects (via the Voice HAT accessory board) will be high on the list for many makers.

Makers have been using a variety of hacks to integrate voice control to their projects. This GoPiGo was hacked by *The MagPi* reader Paul Matz to move and rotate when he speaks out directions.

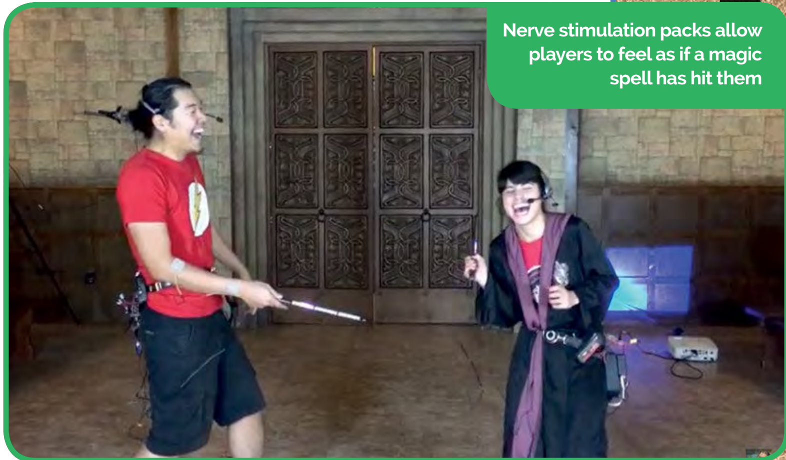


This GoPiGo robot is programmed to respond to voice commands

WIZARD WAND DUEL

magpi.cc/2omrDrh

Allen Pan's Real-Life Wizard Duel project brings wand warfare to life. The Google Cloud Speech API recognises one of five spells as the words are shouted out by the wizards. The kits use Transcutaneous Electrical Nerve Stimulation (TENS) machines to deliver realistic blows to the recipients of successful spells.



Nerve stimulation packs allow players to feel as if a magic spell has hit them

NABAZTAG IOT

magpi.cc/2nZMqAh

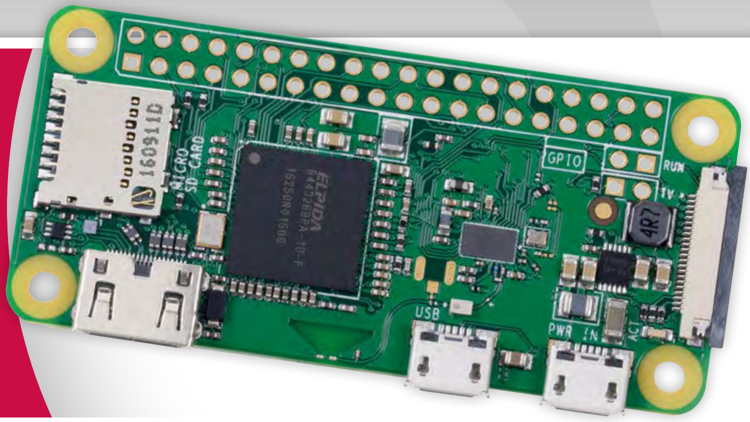
"I have one of these Nabaztag IoT devices," says Bastiaan Slee. "They were ahead of time and didn't survive. Now it's time to give it a new breath."

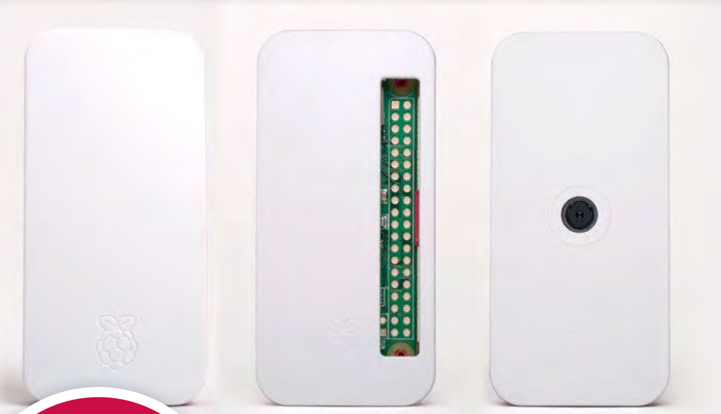
Bastiaan plans to use servos to move the ears, LED lights for notifications, and perhaps add a camera and ultrasonic sensor to sense the nearby presence of a person. "If there is still space in his body, who knows what more will be added?", he tells us.

The rabbit-shaped Nabaztag WiFi assistant will be incredible with Natural Human Interaction

SUBSCRIBE TODAY AND RECEIVE A

FREE PI ZERO W





PLUS OFFICIAL PI ZERO CASE WITH 3 COVERS

AND FREE CAMERA MODULE CONNECTOR AND USB / HDMI CONVERTER CABLES

Subscribe in print for 12 months today and receive:

- A free Pi Zero W (the latest model)
- Free Pi Zero W case with three covers
- Free Camera Module connector
- Free USB and HDMI converter cables

Other benefits:

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

Pricing

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

FREE PI ZERO W!

Get three issues:

£12.99 (UK) (Direct Debit)

$37.50 (US) (quarterly)

How to subscribe:

- magpi.cc/Subs-2 (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 and save money

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

Mag#57

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 ☐☐/☐☐/☐☐

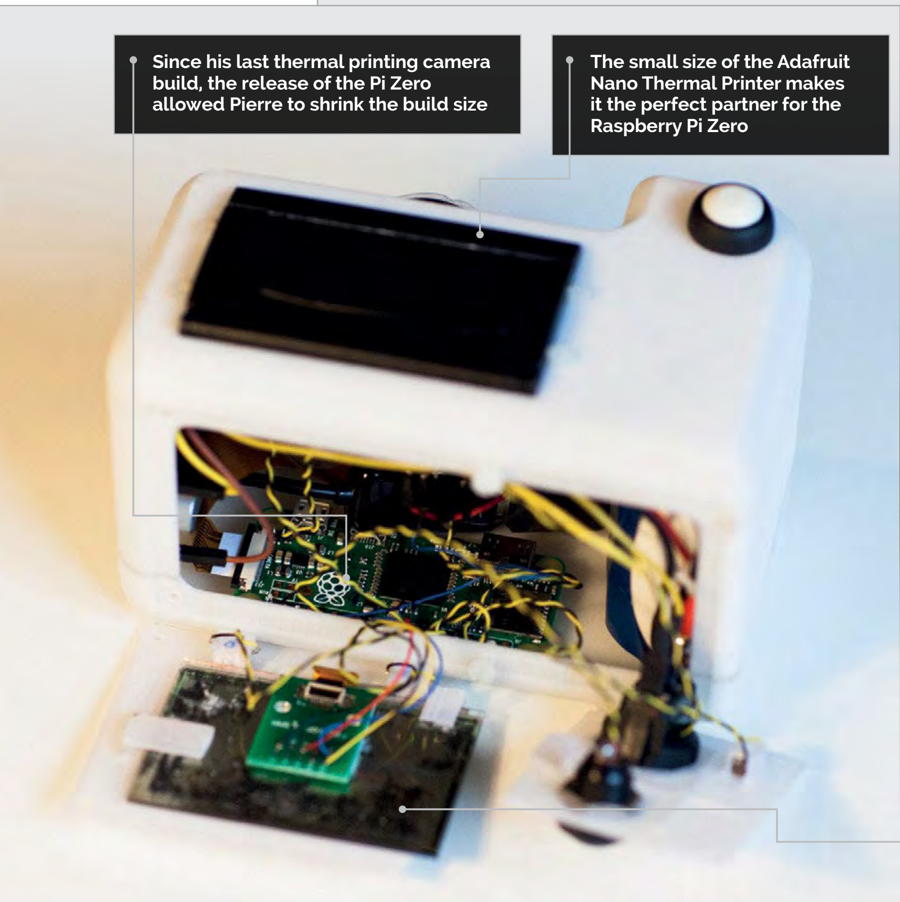
I would like my subscription to begin from issue .... (month + year)

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.



Since his last thermal printing camera build, the release of the Pi Zero allowed Pierre to shrink the build size

The small size of the Adafruit Nano Thermal Printer makes it the perfect partner for the Raspberry Pi Zero

Pierre kept the LCD display in black and white to give a better impression of the final print result

PIERRE MUTH

Pierre Muth is an electronic technician with a love for making things, thermal printers, and the Adafruit Big Red Button.
magpi.cc/2ndiKg8

POLAPI-ZERO

The tiny Zero-powered thermal-printing camera that recreates the joy of using a Polaroid

Quick Facts

- Pierre installed an on/off switch for shutdown
- He printed the case via an online 3D-print service
- Everything is available from his GitHub repo **magpi.cc/2ndslE3**
- You can navigate past images via button press
- Watch the camera in action here **youtu.be/8D-sL3GoFZM**

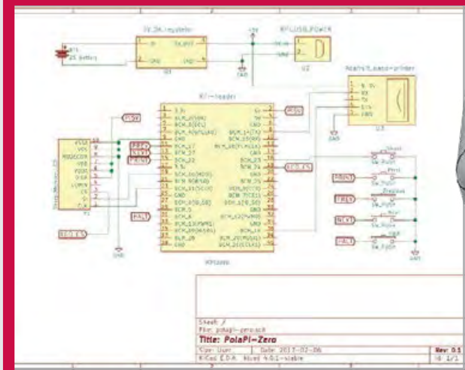
The PolaPi-Zero is the second iteration of Pierre Muth's exploration into portable photography with the Raspberry Pi and thermal printer. No stranger to thermal printing builds, he's previously created such wonders as a camera booth lottery ticket system. Take your photo and if your thermal printout displays another's face, locating them grants you both a free beer.

So while his original PolaPi model also housed a Raspberry Pi – the version 2, with a full-size casing – the newer model allows for a smaller body with its use of the Raspberry Pi Zero.

We've seen many digital camera builds using the Raspberry Pi and Camera Module. From 3D-printed cases to retrofit vintage classics, the majority act as simple point-and-shoot cameras. The PolaPi-Zero, however, takes its lead from the iconic Polaroid camera, utilising a thermal printer inside its body to deliver instant prints of your subject matter.

In his original PolaPi build, Pierre had been forced to cannibalise a retail-grade thermal receipt printer, leaving the unit bulky and weighty. With the new model, following in the footsteps of the small-bodied Raspberry Pi Zero, he managed to acquire the Nano Thermal Receipt Printer from Adafruit: a smaller device marketed specifically for use with boards such as the Raspberry Pi and Arduino. Coupled with a Sharp

BUILDING THE POLAPI-ZERO



>STEP-01
The inside

The schematics for the build are similar to Pierre's original camera. The only differences are the added LCD screen and buttons for reviewing previously captured images.

>STEP-02
The outside

Pierre designed the case and used an online 3D-print service to complete the process. The white colouring of the camera is, in part, a homage to the *White Box* artwork by Vít Hašek.

>STEP-03
The PolaPi-Zero

Pierre used the opportunity to utilise new, smaller components as a means to simplify the overall technical build and add extra features, such as the ability to view previous images.

memory LCD, the camera allows its user to see the image on screen in black and white before printing, guaranteeing the quality of the photograph before you commit to the print.

Pierre used the project as "a good excuse to start learning Python (finally)", in part due to the array of existing Python code available online. His original camera ran using Java, and though he admits to the final Python code not being "the most elegant", he provides it via both his GitHub repo (**magpi.cc/2ndslE3**) as complete code, and as a downloadable image for the Raspberry Pi Zero.

For the physical body of the camera, Pierre designed the unit in Autodesk 123D before sending it to an external 3D printing company, 3DHubs.com, for completion. Again, he provides the case 3D print files in his GitHub repo.

Completing the build with a Pi Camera Module, a 7.2V battery with voltage regulator, and a handful of buttons, the PolaPi-Zero is good to go, providing instant gratification to any user wishing to immortalise their photography on receipt paper.

So what next?

With the technology in place, Pierre started to experiment with different styles of image capture. Starting with the idea of slit-scan photography, where a movable slide with a slit cut in it is passed between the lens and subject matter, Pierre played around with a coded variant. The result is an odd, stuttered image effect that varies depending on whether the scan reads horizontally or vertically. The continuous length of the thermal camera paper allows this effect to be captured and printed.

Pierre claims to live "always with the hope to make something and not just use something", and as his interesting builds continue to wow us, we look forward to seeing what comes next.

Above and Below Examples of the slit-scan photography that Pierre has been experimenting with using the PolaPi-Zero

MICHAEL DARBY

A tech nerd from a young age, obsessed with computers and tech, with a dream to one day be like Tony Stark.

314reactor.com

WINDOWS 98 WATCH

Wind back the clock with this chunky wristwatch

Quick Facts

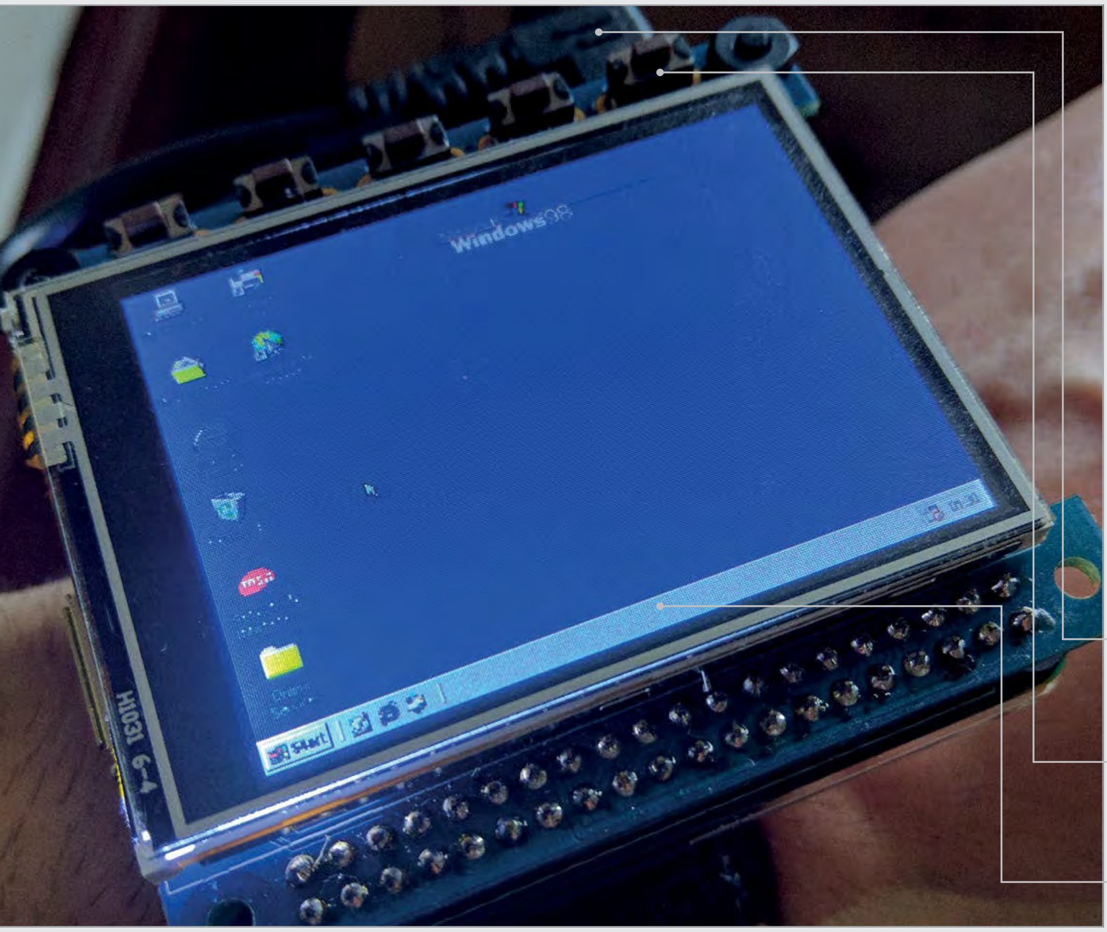
- Windows 98 is emulated using QEMU
- It takes several minutes to boot up
- The Pi A+ is overclocked at 800MHz
- Michael wants to make a Pi 3 version
- He's also working on a Robo Skull

If Microsoft had designed a smartwatch back in the late Nineties, it might have looked something like this! Michael Darby, aka 314reactor, has built a chunky, Pi-powered wristwatch running the Windows 98 operating system.

While he admits the ancient OS used to drive him mad back in the day, he has an odd nostalgia for it. "Many years later you look back on it and want to relive it. I think time has a funny way of keeping more of the good than the bad within memory."

Once he'd accrued the required components, including a Raspberry Pi Model A+, it only took a few hours to put together. There's a tutorial on his site: **magpi.cc/2nX9ss4**. The Pi A+ sits in the bottom of an Adafruit Pi Protector case with a PiTFT 2.4-inch HAT touchscreen on top. Five tactile buttons have been added to the latter, although only one is currently used – to shut down the system cleanly. The watch is powered by a slimline LiPo battery connected via a PowerBoost 500 with switch, while Velcro feet secure the watch body to a wrist strap.

Emulation is employed to get Windows 98 working on the watch, using the QEMU hypervisor running in Raspbian. "Its relatively simple," says Michael. "Once you've set up a QEMU environment on another PC and installed Windows 98 to it, it's a case of dragging the virtual hard drive



A LiPo battery, hidden underneath, powers the Raspberry Pi A+ and touchscreen

Buttons have been added to the PiTFT screen, although only one is currently used

Emulated using QEMU, Windows 98 really works, albeit slowly

Remember this? The classic 'flying windows' screensaver

file over to the Pi and running it there."

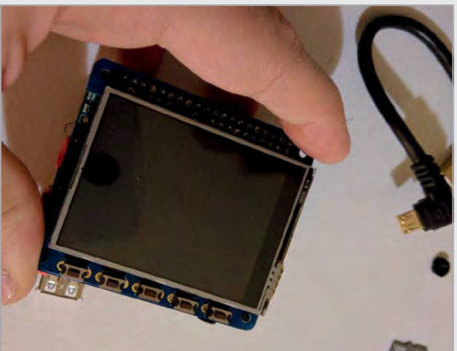
When the Windows 98 system eventually boots, it really does work, although Michael admits it's too slow to be usable and touchscreen navigation is tricky. "I need to look into calibration, I think, or it could be an issue with the way QEMU works when launched from the command line."

Even so, Michael is eventually hoping to get classic 3D shooter Doom running on the watch, once he's overcome a technical issue: "I can't find a way to create a virtual graphics card on QEMU on the Pi that will allow Windows 98 to run in 8-bit colour."

While Michael has considered using a Pi Zero for the watch, he has a different plan for an upgraded version: "It could be made marginally slimmer with a Pi Zero and it would give a bit of a speed boost… I am thinking of going the other way, though, and using a Pi 3 in future." Although this would be even bulkier, it would run a lot faster as the Pi 3 handles emulation far better. "I can potentially put a bigger screen on and have multiple operating systems selectable from the buttons, such as 95, XP, and even something out there like ReactOS or some random Linux distro."

> "I can potentially put a bigger screen on and have multiple operating systems selectable"

In the meantime, Michael has received plenty of positive feedback for his Windows 98 Watch, which he wore to Raspberry Pi's Fifth Birthday Bash. "The reaction has been crazy; I never expected it to take off like it did, but I'm very happy it did and it has inspired me to create more and work harder."

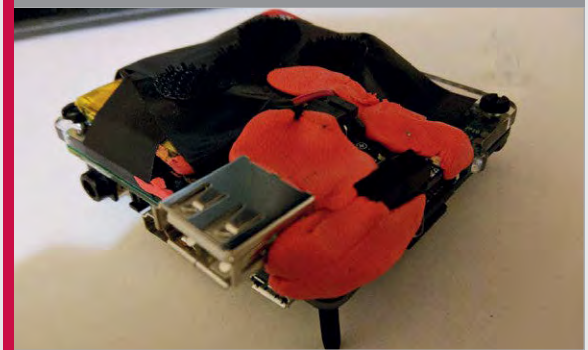


An AdaFruit 2.4˝ PiTFT touchscreen, with added push buttons, is mounted on the Pi

BUILD YOUR OWN TIMEPIECE

>STEP-01
Add a touchscreen

Equipped with a heatsink and spacers for mounting the PiTFT 2.4-inch touchscreen, a Pi A+ – with optional overclocking – runs the QEMU hypervisor in Raspbian.



>STEP-02
Power it up

Sugru and tape are used to secure the 1000mAh LiPo battery and PowerBoost to the bottom of the Raspberry Pi, powering it via a USB to micro-USB cable.

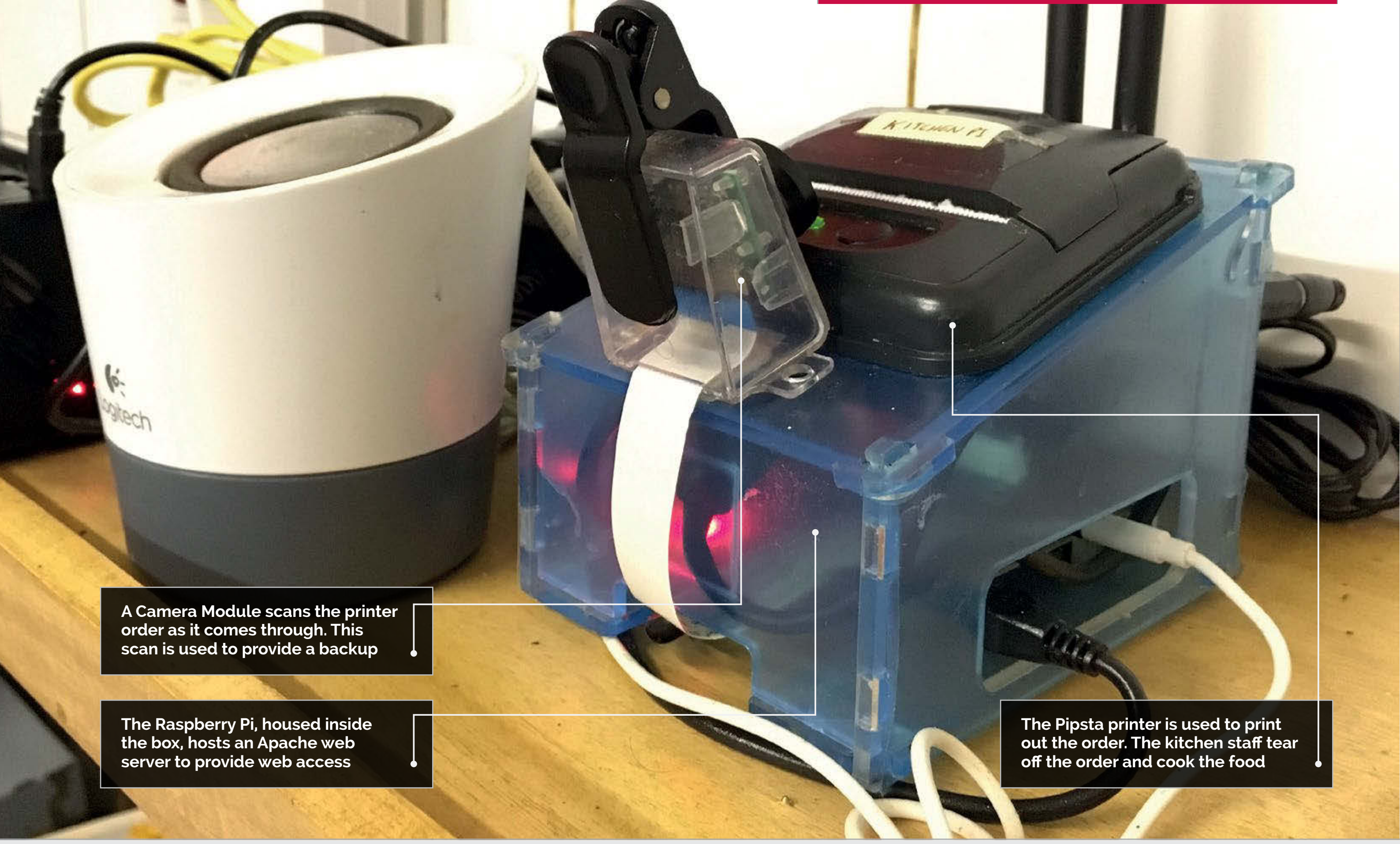
>STEP-03
Emulate Windows

Windows 98 is converted to an IMG file on a PC, then transferred to the Pi to run via QEMU. Framebuffer mirroring is used to output the display to the touchscreen.

EHSAN RAHMAN

Ehsan is a waiter at his father's restaurant and a prodigious maker.

magpi.cc/2mFCdZB



PIORDER

How one Indian restaurant used Raspberry Pi to automate its whole kitchen ordering system

Quick Facts

- Waiting staff use an Amazon Fire HD to take orders
- The Khyber Tandoori is a 50-seater restaurant
- The online system took 27 orders in its first two weeks
- The EPOS system has been in use for eight months
- Customers get live updates on waiting times

"I help at my father's restaurant on Friday and Saturday evenings," says Ehsan Rahman.

That establishment is the Khyber Tandoori, an Indian restaurant based in Kingswood, Surrey (**magpi.cc/2miqcqv**). Thanks to the Raspberry Pi, it has become a highly automated environment.

Two years ago, Ehsan became frustrated at writing orders on pen and paper. Ehsan's answer was to code and hack his way out.

The result is PiOrder, a fully automated EPOS (electronic point of sale) system. PiOrder comprises Raspberry Pis, several Pi Camera Modules, and a Pipsta thermal printer (**magpi.cc/2miwYMQ**).

The waiting staff use large Kindle Fire tablets to take orders. Two smaller tablets are kept near the phones so staff can take orders over the telephone.

In the kitchen is a Raspberry Pi board hosting the Apache website. A program written in PHP and HTML is used to provide the webpages.

Apache is used to host the webpage used by the waiting staff to take orders. It also offers online ordering for takeaway customers.

Chefs are alerted to new orders via a speaker attached to the Raspberry Pi.

The Pipsta printer also prints a hard copy of the order, and a


Raspberry Pis with Camera Modules provide CCTV for the restaurant, which can be viewed using the system

Camera Module takes a photograph of the order to ensure it has printed out correctly (and to act as a backup). More Camera Modules are used by managers to keep an eye on how busy the kitchen is.

As well as making Ehsan's waiting duties easier, PiOrder saves on costs. "Just Eat charges approximately £699 + VAT just for signing up," reveals Ehsan. Then it charges around 11% per order, an amount that rapidly racks up.

More importantly, "we have control over our software and order flow," adds Ehsan.

the system has "been resilient." There's an automated test every day at 5:30 for a single popadom, "so the chefs and waiting staff know the system is up and working," explains Ehsan.

There are even spare Raspberry Pi boards in case of failure. "But I've not needed them yet after two years," Ehsan reveals.

As a result of all this tinkering, the restaurant is incredibly high-tech. As well as the ordering system, they are using Raspberry Pi Model B boards as smart CCTV cameras. Ehsan has even set up a

> The chefs would not go back to reading handwritten orders ever again

The system is a mixture of PHP, JavaScript, and jQuery, with Bash scripting used to communicate between the Raspberry Pis. "The great thing about Unix files is just how reliable they are," says Ehsan.

The other waiting staff and Ehsan's father have completely stopped using pen and paper.

There were some teething troubles: the original WiFi system occasionally dropped the connection, and the Pipsta struggled to print large orders. But after sorting those issues,

Raspberry Pi 3 to act as a remote monitoring system, "so my father can see how busy the restaurant is from home."

Ehsan isn't finished. He plans to enable customers to order food from their table using a smartphone or tablet.

"The chefs and my father were not convinced at first, but slowly they saw the benefit." The waiting staff love the ability to update orders with just a few taps. And the chefs would not go back to reading handwritten orders ever again.

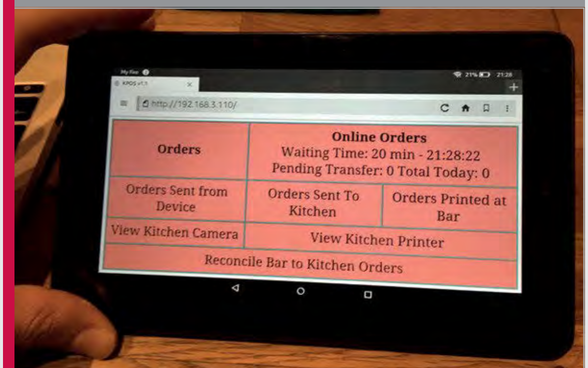
BUILD A MENU ORDERING SYSTEM

>STEP-01
Tablet interface

Waiting staff place orders on a webpage, viewed on Amazon Kindle Fire tablets. A Raspberry Pi running Apache serves up the webpage over a WiFi network.

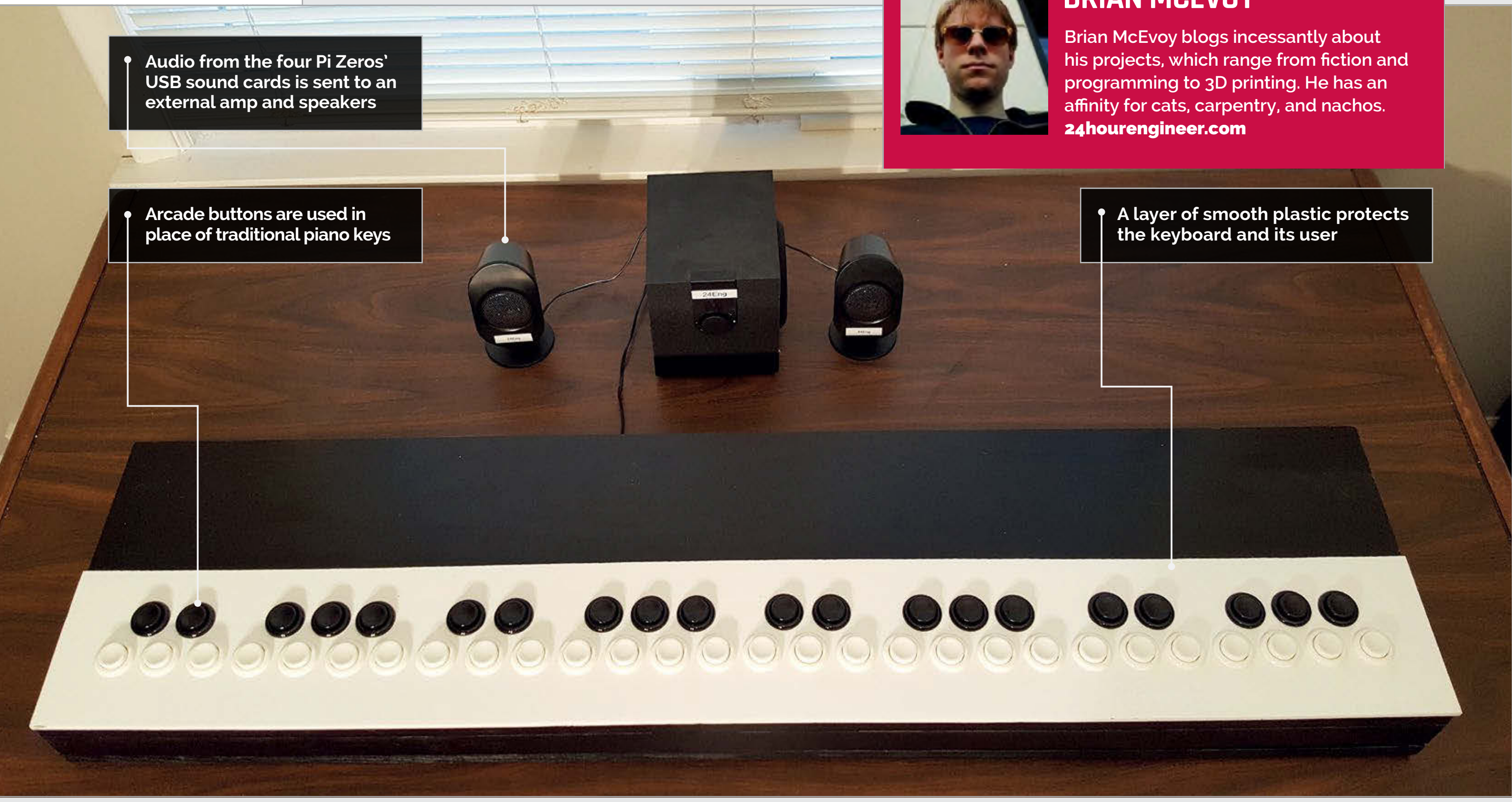
>STEP-02
Kitchen printer

The order is printed out in the kitchen using a Raspberry Pi connected to a Pipsta printer. A Camera Module scans the order as a backup and sends a push alert to the manager.



>STEP-03
Remote management

The manager can keep an eye on how many orders have been placed. The system also enables them to view the kitchen-monitoring and security cameras.



BRIAN MCEVOY

Brian McEvoy blogs incessantly about his projects, which range from fiction and programming to 3D printing. He has an affinity for cats, carpentry, and nachos.
24hourengineer.com

TOUGH PI-ANO

Hammer the keys on this incredibly robust musical instrument

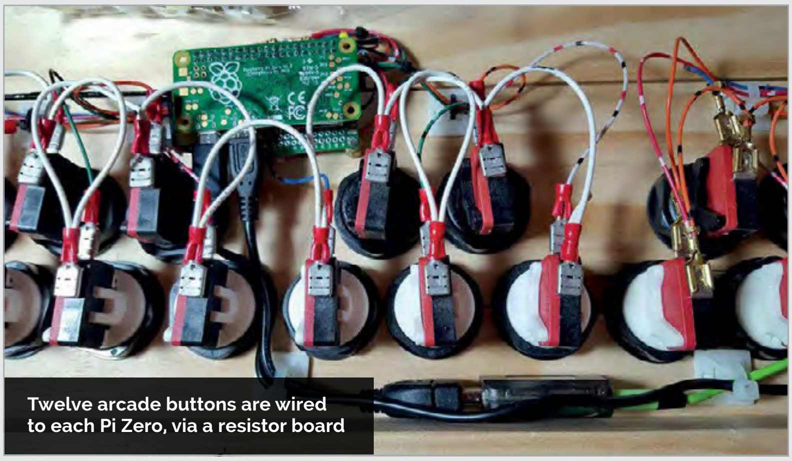
Quick Facts

- The Tough Pi-ano has 48 'keys' in total
- It will be used for music therapy
- The keys are plastic arcade buttons
- A Pi Zero controls each octave
- The USB sound cards cost $0.99 each

Brian McEvoy wanted to make a toughened electronic keyboard for his cousin, who has Down's syndrome: "He loves music therapy but he breaks pianos on a weekly basis." Brian's aunt asked him to help with therapy devices for his cousin and other such kids. "The idea was exciting because there is a whole different mindset when designing for someone who will not handle things with gentle hands. These projects have to absorb damage without hurting anyone."

The result is the Tough Pi-ano. It has no exposed metal, and the keyboard area is covered with thick plastic. For its keys it uses arcade buttons, which are inexpensive and easy to replace. Each of four octaves is powered by a Pi Zero, with a USB sound card outputting the audio to an external amplifier and speakers.

While Brian contemplated the Tough Pi-ano's design for two years, it only took him a couple of weeks to build, putting his woodworking skills to good use. "Keeping everything simple was integral to making a solid structure." Since the original design concept was for a perfect piano replica with easily replaceable octaves, he started crafting wooden piano keys. After experiencing too many problems, however, these were replaced with plastic arcade buttons. "In the end, arcade buttons were the best solution since they were easy to source and replace."



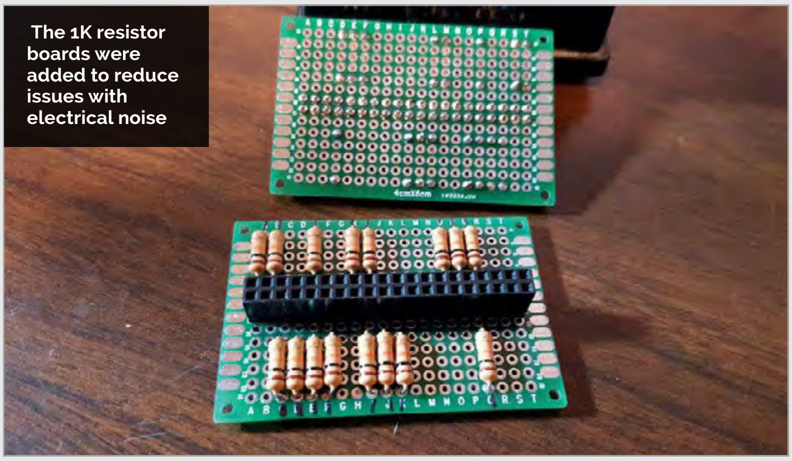
Twelve arcade buttons are wired to each Pi Zero, via a resistor board

Brian wrote a Python program using the Pygame library to read the arcade button presses and play WAV piano samples. While he considered adding a 3.5mm jack to each Pi Zero, he eventually opted to use cheap USB sound cards to output the audio. The biggest problem was electrical noise. "The first octave I built didn't have any pull-up resistors on the keys so they were, of course, prone to floating. At that point, I had inadvertently built a touchless piano," recalls Brian. 1K resistors were added to each input to solve the problem. "There is still some noise on the audio despite using a clean power supply and line filter. Maybe using $0.99 USB audio cards is to blame."

While Brian admits it would have been possible to use a single Raspberry Pi and remote I/O to power the piano, the use of one Pi Zero per octave has some benefits: "If any part was to break, it would be possible to shuffle working hardware around for a three-octave Tough Pi-ano, at least until repairs could be made. So there is still an advantage to the redundancy."

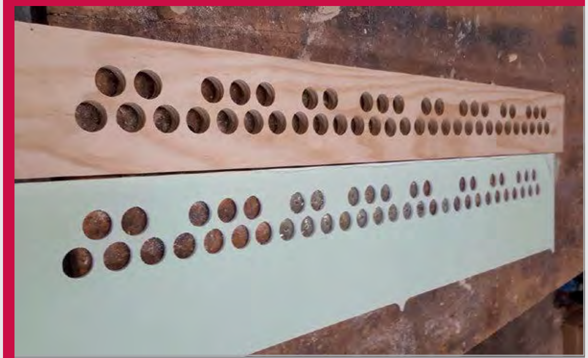
Brian's Tough Pi-ano is now set to be used in his aunt and uncle's new centre for local families with kids on the autism spectrum and those with Down's syndrome. While he doesn't plan to build another piano, he has some advice for would-be makers. "I would recommend building the speakers into the enclosure, but be sure to ventilate the amplifier. Also, buy one of the tools used to fasten arcade button washers, or you'll regret it when you get to the 50th button and your knuckles are bloody."

> "In the end, arcade buttons were the best solution since they were easy to source and replace"



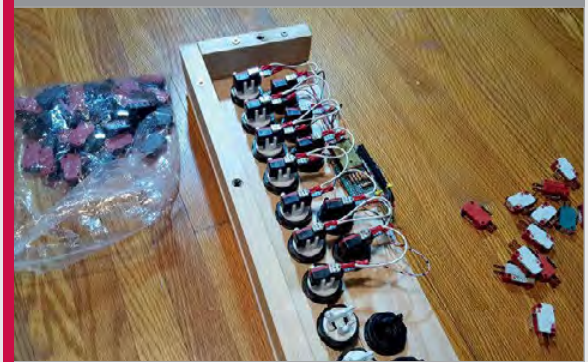
The 1K resistor boards were added to reduce issues with electrical noise

MAKE A TOUGH PIANO



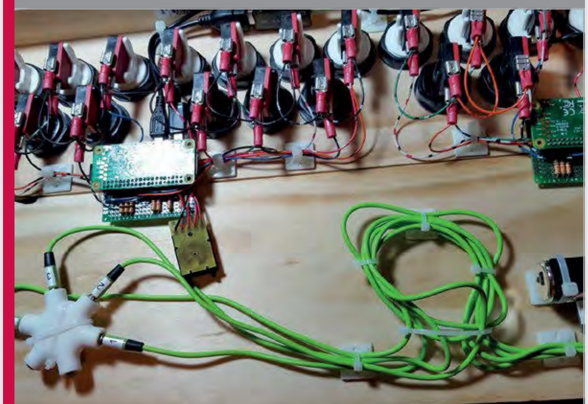
>STEP-01
Wooden chassis

Brian constructed the chassis from pine lumber, and drilled holes in both the wooden panel and the protective plastic layer to house the arcade buttons.



>STEP-02
Buttons for keys

While the original design featured conventional wooden piano keys, these were replaced with heavy-duty arcade buttons and switches.



>STEP-03
Wiring it up

For each octave, the arcade buttons are wired to a resistor board, which is connected to a Pi Zero. A USB sound card is used to output the audio.

GETTING STARTED WITH MINECRAFT: PI EDITION

Get off to a good start with Minecraft: Pi Edition. Play the game and write your first program using the API.

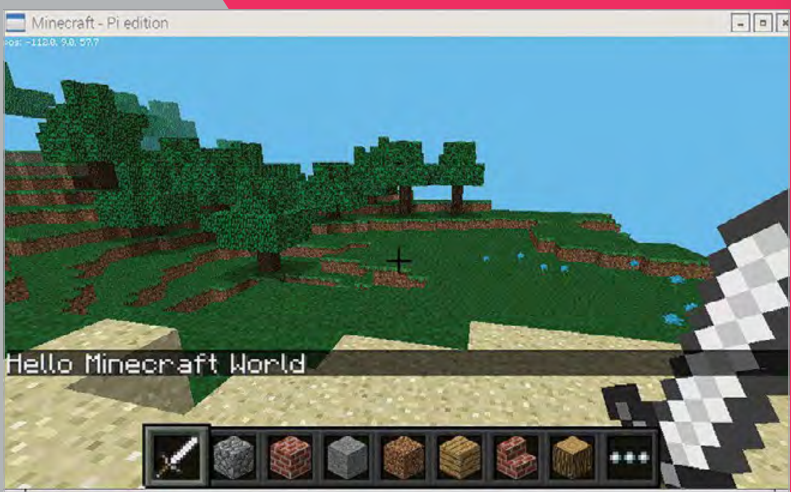
If you've never played Minecraft and you want to be a master block builder, we'll help you get stuck into the game, build a house, and get started with the API.

Minecraft is a game which has achieved monumental success; more than 120 million copies have been sold across all its versions. Not bad for a game which doesn't really have a point! If it does have a point, as an indie sandbox game, it's to make stuff. And people have really made stuff, from fully functioning computers to scale models of the Starship Enterprise.

The best things about Minecraft: Pi Edition are that it's free, and that it comes with an API. You don't get this with any other version of Minecraft.

Minecraft is installed by default on Raspbian. If you have an older version, you can get it by opening a Terminal (Menu > Accessories > Terminal) and typing these commands, pressing **ENTER** after each:

```
sudo apt-get update
sudo apt-get install minecraft-pi
```



Use the API to write a 'Hello Minecraft World' program

Playing the game

Click Menu > Games > Minecraft: Pi Edition to run the game. Minecraft: Pi Edition offers one playing mode, Classic, which is all about exploring and building. Click Start Game, then click Create New (or choose an existing world) to enter a world.

- **The mouse** changes where you look
- **Holding the left button** destroys blocks
- **Right button** places blocks
- **W, S, A, D** move you forward, backward, left, and right
- **1, 2, 3, 4, 5, 6, 7, 8** change what you are holding
- **E** opens the inventory
- **ESC** takes you back and to the **Menu**
- **SPACE** is jump. Double-tapping it makes you fly or stop flying

The API

The API (application programming interface) allows you to write programs which control, alter and interact with the Minecraft world, unlocking a whole load of Minecraft hacking. How about creating massive houses at the click of a button; writing a game which uses a LED and buzzer to help you find a block; or recreating Nintendo's Splatoon in Minecraft?

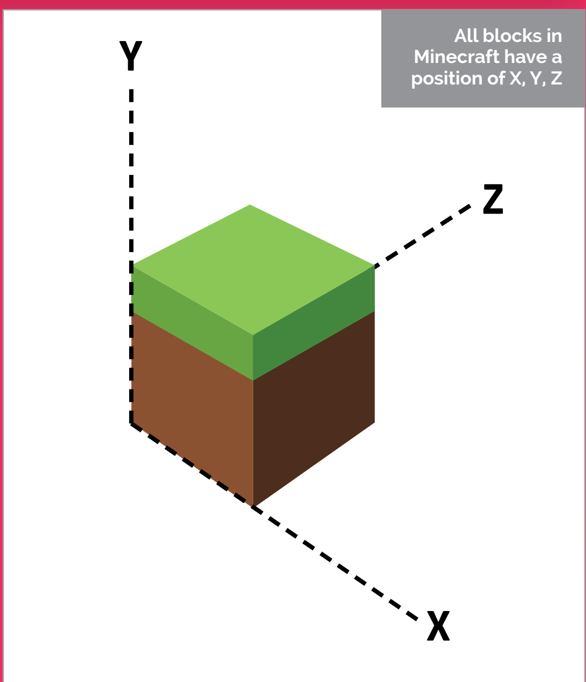
The API works by changing the world as the game is being played, allowing you to:

- Get the player's position
- Change (or set) the player's position
- Discover the type of block at a specific location
- Change a block
- Change the camera angle
- Post messages to the player

Hello Minecraft World

The first program all programmers create when learning something new is called Hello World, which puts the phrase 'Hello World' on the screen. You're going to do the same, but in Minecraft:

01. Go to the Minecraft menu with **ESC**, but leave the game running (you can minimise it).
02. Open IDLE by clicking Menu > Programming > Python 3 (IDLE).
03. Use File > New File to create a new program and save it as **hellominecraftworld.py**.
04. At the top of your program, type the following code to import the `minecraft` module, which will allow you to use the API and talk to the game:
    `import mcpi.minecraft as minecraft`
05. On the next line, create a connection from your program to Minecraft and call it `mc`:
    `mc = minecraft.Minecraft.create()`
06. Use your Minecraft connection and the function `postToChat()` to put a message in the chat window on a third line:
    `mc.postToChat("Hello Minecraft World")`
07. Run your program by clicking Run > Run Module.



All blocks in Minecraft have a position of X, Y, Z

Minecraft: Pi Edition has an API you can use to program the game

Switch back to Minecraft, click 'Back to game', and you should see the message 'Hello Minecraft World' on the screen. Be quick, though, as the message only stays on the screen for ten seconds.

Any errors will appear in red text in the Python Shell window. Check your code carefully for spelling mistakes, and ensure that you have used the right upper- or lower-case letters.

When you have successfully made the message appear on the screen, try changing it and running the program again.

WATCH FOR RED TEXT

Any errors in your program will appear in the Python Shell in red text.

Teleportation

Using your new Python programming skills and the Minecraft API, you can teleport Steve around the world by adding just one more line of code to your program.

Minecraft is a world of blocks, all about 1m × 1m × 1m. The player and every block in the world has a position made up of x, y, and z: x and z are the horizontal positions and y is the vertical. By changing the player's x, y, and z position, you can teleport them wherever you want.

The player starts at position x = 0, y = 0, z = 0, which is the spawn point, and the player's current position is shown at the top left of the screen.

Add the following code to your 'Hello Minecraft World' program to teleport the player to position x = 0, y = 50, z = 0, which will put your player 50 blocks up in the air:

01. Teleport the player by setting their position:
    `mc.player.setPos(0, 50, 0)`
02. Run your program by clicking Run > Run Module.
03. Quickly switch back to Minecraft to see your player fall to the floor (unless in flying mode).

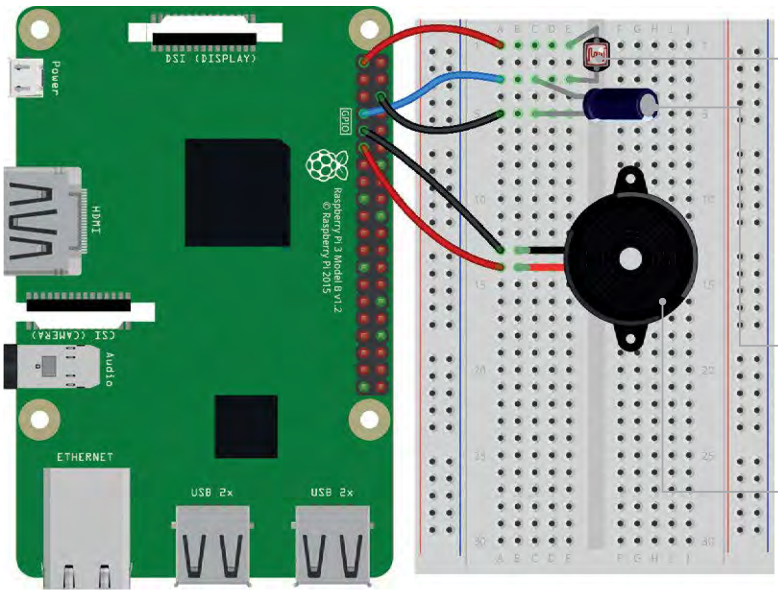
PYTHON IS CASE-SENSITIVE

Beware of upper- and lower-case letters: 'Minecraft' and 'minecraft' are different things to Python.

Try changing the values in `setPos()` to teleport your player to different places around the world. Use values -125 to 125 for x and z and -64 to 64 for y, otherwise the player will be teleported outside the world.

PHIL KING

When not writing and sub-editing on *The MagPi*, Phil loves to work on Pi projects. This article is based on an official resource by Marc Scott.
@philking68 @Coding2Learn



The light-dependent resistor (LDR) has lower resistance when the laser is shining onto it

By timing how long it takes to charge the capacitor, we can work out the light level on the LDR

A simple piezo buzzer is made to beep whenever the laser tripwire beam is broken

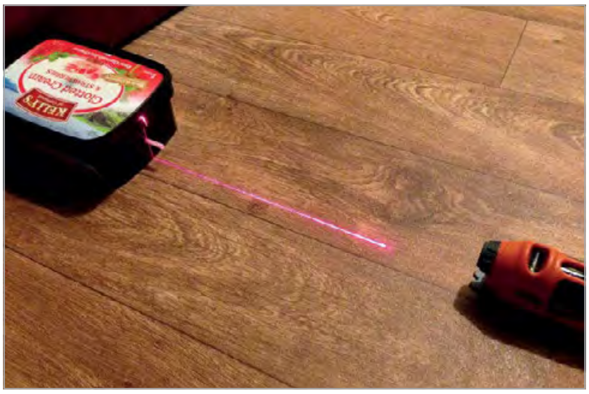
MAKE A LASER TRIPWIRE

Learn how to use an LDR to detect a laser pointer beam

You'll Need

- GPIO Zero
- 1× solderless breadboard
- 1× light-dependent resistor (LDR)
- 1× 1μF capacitor
- 1× laser pointer
- 5× male-to-female jumper wires
- 5× female-to-female jumper wires (optional)
- 1× drinking straw
- 1× plastic box

The Raspberry Pi can easily detect a digital input via its GPIO pins: any input that's approximately below 1.8V is considered off, and anything above 1.8V is considered on. An analogue input can have a range of voltages from 0V up to 3.3V, however, and the Raspberry Pi is unable to detect exactly what that voltage is. One way of getting around this is by using a capacitor, and timing how long it takes to charge up above 1.8V. By placing a capacitor in series with a light-dependent resistor (LDR), the capacitor will charge at different speeds depending on whether it is light or dark. We can use this to create a laser tripwire!



Above Place your laser tripwire across a doorway; when someone breaks the beam, the alarm will sound

>STEP-01

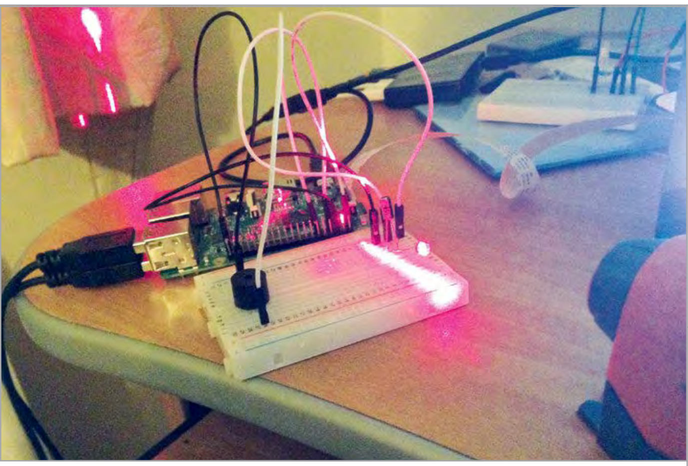
Connect the LDR

An LDR (also known as a photocell) is a special type of electrical resistor whose resistance is very high when it's dark, but reduced when light is shining on it. With the Raspberry Pi turned off, place your LDR into the breadboard, then add the capacitor. It's essential to get the correct polarity for the latter component: its longer (positive) leg should be in the same breadboard column as one leg of the LDR. Now connect this column (with a leg of both components) to GPIO 4. Connect the other leg of the LDR to a 3V3 pin, and the other leg of the capacitor to a GND pin. Your circuit should now resemble the diagram above.

>STEP-02

Test the LDR

On the Pi, open IDLE from the Main Menu: Menu > Programming > Python 3 (IDLE). Create a new file by clicking File > New File, enter the code from **ch8listing1.py**, then save it. At the start, we import the `LightSensor` class from GPIO Zero. We then assign the variable `ldr` to the LDR input on the GPIO 4 pin. Finally, we use a never-ending `while True:` loop to continually display the current value of the light sensed by the LDR, which ranges from 0 to 1. Try running the code and then shining your laser pointer on it to vary the light level.



Above Shining the laser onto the LDR in a darkened room will dramatically affect the measured light level

>STEP-03
Enclose the LDR

Unless you're working in a darkened room, you'll probably notice little difference between the measured light level when the laser pointer is directed onto the LDR and when it's not. This can be fixed by reducing the amount of light that the LDR receives from other light sources in the room, which will be essential for our laser tripwire device to work effectively. We'll achieve this by cutting off a short section – between 2cm and 5cm – of an opaque drinking straw, and inserting the head of the LDR into one end. Now try the test code again and see how the measured light level changes when you shine the laser pointer into the other end of the straw. You should notice a larger difference in values.

>STEP-04
Wire up the buzzer

To create an audible alarm for our laser tripwire, we'll add a piezo buzzer to the circuit. Again, the polarity has to be correct: connect the column of the buzzer's longer leg to GPIO 17, and the shorter leg to a GND pin. Let's test whether it is working. In IDLE, create a new file, enter the code from **ch8listing2.py**, and save it. At the top, we import the **Buzzer** class from GPIO Zero. Next, we assign the **buzzer** variable to the buzzer output on GPIO 17. Finally, we use **buzzer.beep** to make the buzzer turn on and off repeatedly at the default length of one second. To stop it, close the Python shell window while it is off.

>STEP-05
Test the tripwire

We'll now put it all together so that laser pointer shines at the LDR through the straw, and whenever the beam is broken, the buzzer sounds the alarm. In IDLE, create a new file, enter the code from **ch8listing 3.py**, and save it. At the start, we import the **Buzzer** and **LightSensor** classes from GPIO Zero. We also import the **sleep** function from **time**; we'll need this to slow the script down a little to give the capacitor time to charge. As before, we assign variables for the buzzer and LDR to the respective devices on GPIO pins 4 and 17. We then use a **while True:** loop to continually check the light level on the LDR; if it falls below 0.5, we make the buzzer beep. You can change this number to adjust the sensitivity; a higher value will make it more sensitive. Try running the code. If you break the laser beam, the buzzer should beep for eight seconds. You can adjust this by altering the **buzzer.beep** parameters and **sleep** time.

>STEP-06
Package it up

Once everything is working well, you can enclose your Raspberry Pi and breadboard in a plastic box (such as an old ice cream tub), with the drinking straw poking through a hole in the side. If you prefer, you can remove the breadboard and instead connect the circuit up directly by poking the legs of the components into female-to-female jumper wires, with the long capacitor leg and an LDR leg together in one wire end, connected to the relevant pins. Place the tub near a doorway, and place the laser pointer on the other side, with its beam shining into the straw. Run your code and try walking through the doorway: the alarm should go off!

ch8listing1.py

```python
from gpiozero import LightSensor

ldr = LightSensor(4)

while True:
    print(ldr.value)
```

ch8listing2.py

```python
from gpiozero import Buzzer

buzzer = Buzzer(17)
buzzer.beep()
```

ch8listing3.py

```python
from gpiozero import LightSensor, Buzzer
from time import sleep

ldr = LightSensor(4)
buzzer = Buzzer(17)

while True:
    sleep(0.1)
    if ldr.value < 0.5:
        buzzer.beep(0.5, 0.5, 8)
        sleep(8)
    else:
        buzzer.off()
```

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/2bhxwxC

CLIVE WEBSTER

A professional tinkerer since 2004, Clive just keeps seeing more uses for Raspberry Pis around the house. How many Pis is too many?

@clivewriting

HACK AN AMAZON DASH BUTTON

Rescue Amazon's £5 smart button from the monotony of ordering loo roll, and use it to do anything you like.

You'll Need

- Dash Buttons **magpi.cc/2mUKaqT**
- A smart thing
- Some patience

The Internet of Things has two flaws: the name, and the need to get your phone out of your pocket, unlock it, swipe to the appropriate app, and wait for the app to load before you can do perform a basic task, such as turning on a light. Buttons are handy for a reason: they're exactly where they need to be, and you can just press them. So if we're going to have smart things, we need smart buttons.

Typically these cost £40, but Amazon's Dash Button costs a fiver. With a little tinkering and subversion, it can power anything from smart bulbs to alerts on your family's smartphones to logging your billable hours.

Head over to Amazon to buy some Dash Buttons (you'll need to be a Prime subscriber) and you'll be baffled by the options: you can emergency-order raw virgin coconut oil (only £16.62 for 1.2kg), 20-sheet binding machines, or Nerf Darts with a Dash Button.

The catch with these Dash Buttons is that you can only order certain products from certain firms. This matters, as Amazon will discount the price of a Dash Button from the first purchase made with it. Make sure you buy a button that lets you order something you'd buy anyway; that way, your Button is free.

Once the Dash Button arrives, use the Amazon app (Android and iOS only) to set it up and order that first item. There are decent instructions at **amzn.to/2mNhAqt**. Once you've ordered your product, go back into the Dash Devices and disable it. Then – bear with us – enable the Button again, but this time don't select a product. Instead, just quit the app (don't quit the process, close the whole app). This will have copied your WiFi credentials to the button without re-establishing the link to Amazon. It is now an unshackled smart button ready to be repurposed.

Jessie games

We used Jessie Lite on our Raspberry Pi Zero W, as it will be running as a headless server with no GUI required. Once in, follow the usual update procedure: **`sudo apt-get update && apt-get -y dist-upgrade`**. As your Dash Button has already accessed your router, its MAC address should be listed in your router's logs or DHCP tables. The location differs according to manufacturer. The button will show up as 'Internet Device' or similar.

Now we're ready to code. The idea behind this hack is to use the Pi as an interpreter. It constantly monitors your network for the appearance of the Dash Button, then uses that appearance as a trigger for some other action. We're going to use Python to intercept those button presses, but this requires an extra module:

```
sudo apt-get install -y pip
sudo pip install scapy
```

Pip is a Python-specific installer, and Scapy is the module we'll use to 'sniff' for the Dash Button's MAC address (**`pkt[ARP].hwsrc`**) appearing on the network. As the Dash Button only powers up when you press its button, its MAC address will only appear after a button press.

Button whole

Now that we can intercept a button press, let's do something with it. We've got a LIFX smart light, so we need the LifxLAN Python module: **`pip install lifxlan`**. Open the trigger script and add the LifxLAN module's functions to the script with the line **`from lifxlan import *`**.

Now we need to find the MAC and IP address of the smart bulb from our router, and to name the bulb in the Python script using the 'Light' object of the LifxLAN module: **`bedroom = Light('xx:xx:xx:xx:xx:xx', '192.168.1.xxx')`**. We need to find the current power level of the bulb in order to toggle it, which we do with the **`current_state = bedroom.get_power()`** variable and the **`bedroom.set_power()`** commands of our **`if, else`** statements. Annoyingly, the Dash Button sends two ARP packets every time it is pressed, so we need to ignore the second ARP packet by using the **`second_arp`** Boolean variable.

Once done, save your script, make it executable (**`chmod +x lights.py`**) and add it as a cronjob (**`crontab -e`**) to make sure it runs every time your Pi boots: **`@reboot sudo python /home/pi/lights.py`**.

Now we've got a smart button that we can stick to any wall, table, desk or bookcase to turn on our smart lights. And if a LIFX bulb sounds steep at £60 each, it would cost a lot more to install a new 'dumb' light switch once you've factored in gouging cable runs in the wall, hiring plasterers to make good and repainting a whole room or hallway. People have made Dash Buttons work with Philips Hue and Samsung SmartThings devices, too.

BASIC CODE

trigger.py

Language

>PYTHON

DOWNLOAD:
magpi.cc/DashButtonCode

```
from scapy.all import *

def arp_detect(pkt):
  if pkt[ARP].op == 1: #network request
  if pkt[ARP].hwsrc == 'xx:xx:xx:xx:xx:xx'
    return "Button detected!"

print sniff(prn=arp_display, filter="arp", store=0)
```

EXAMPLE CODE

lights.py

```
from scapy.all import *
from lifxlan import *

#Buttons
andrex = 'xx:xx:xx:xx:xx:xx'

#Lights
bedroom = Light('xx:xx:xx:xx:xx:xx', '192.168.1.xxx')
second_arp = False

def arp_detect(pkt):
  if pkt[ARP].op == 1: #network request
    if pkt[ARP].hwsrc == andrex:
      current_state = bedroom.get_power()
      if current_state == 0:
        bedroom.set_power("on")
      else:
        bedroom.set_power("off")

if second_arp == False:
  sniff(prn=arp_detect, filter="arp", store=0)
  second_arp = True
else:
  second_arp = False
```

Shoot for the (Button) Moon

So, what else could you press your Dash Button to do? Hook your Python script into an online spreadsheet and you can log the time between button presses – useful for a musician logging practice sessions, or a freelance accountant logging billable hours. Tired of shouting up the stairs for your kids to come down for dinner? You could go the JavaScript route to connect an SMS messenger: press your Dash Button and they'll receive a text message (**magpi.cc/2mt3zmo**). Or use a service like Pushover (**pushover.net**) to make a smart doorbell, sending alerts to your smartphone or watch.

Aaron Bell has detailed how to make a hacked Dash Button into an IFTTT trigger (**magpi.cc/2msVvC5**). We'd also thank Ted Benson for being one of the first Dash hackers (**magpi.cc/2msUzhh**), although updated Amazon firmware is incompatible with his code.

TONY HANSEN

Tony is an 'experienced' software and computer engineer, meaning he's been doing this for quite some time. He rediscovered his love for the smell of solder a few years ago.
magpi.cc/2nCOVrA

ADD A SAFE OFF-SWITCH TO POWER DOWN YOUR RASPBERRY PI

Add a switch to your Raspberry Pi to safely shut it down without pulling the power

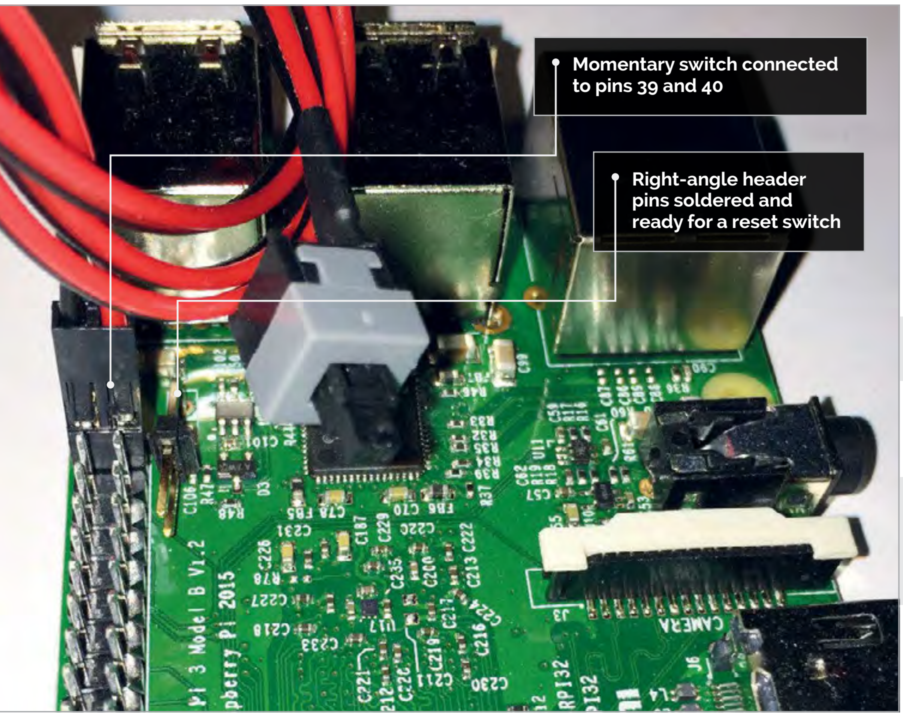
You'll Need

- Raspberry Pi (any model)
- Momentary push button switches, such as: **magpi.cc/2oCMKCm**

To keep prices down, the Raspberry Pi is missing something that most electronic devices come with: a switch to turn it on and off. That's OK, you say, we'll just pull the plug to turn it off. Unfortunately, this can lead to corruption problems with the SD card. All the instructions say you should run the shutdown command before pulling the plug, but this is not always possible, particularly when your Raspberry Pi is running headless without a connected keyboard and monitor, and possibly even without a network connection. So, what can a self-respecting DIYer do? The answer, of course, is 'add your own switch'!

Lots of articles are available to tell you how to use a breadboard to connect a button or LED to a Raspberry Pi's GPIO pins. This article focuses on doing something useful with those switches and LEDs.

The safe off-switch is complementary to a reset switch, which is the best method for starting the Raspberry Pi up again. Issue 52 of *The MagPi* featured an excellent article on how to connect a reset button.



Using GPIO Zero

With the GPIO Zero library, the Python code to deal with a button press becomes extremely simple. Assuming your button is connected between GPIO 21 and GND (ground), the code is nice and easy. You can download it from **magpi.cc/2nD29F7** as **shutdown-press-simple.py**.

This code creates a button on GPIO 21, waits for it to be pressed, then executes the system command to power down the Raspberry Pi. GPIO 21 is nice because it's on pin 40 of the 40-pin header and sits right next to a ground connection on pin 39. This combination makes it difficult for an off-switch to be plugged in incorrectly. On a 26-pin header, GPIO 7 is similarly situated at the bottom, on pin 26, next to pin 25's ground connection.

Create the script on your Raspberry Pi using your favorite text editor (e.g., nano, Vim or Emacs), as in:

```
$ nano ~pi/shutdown-press-simple.py
```

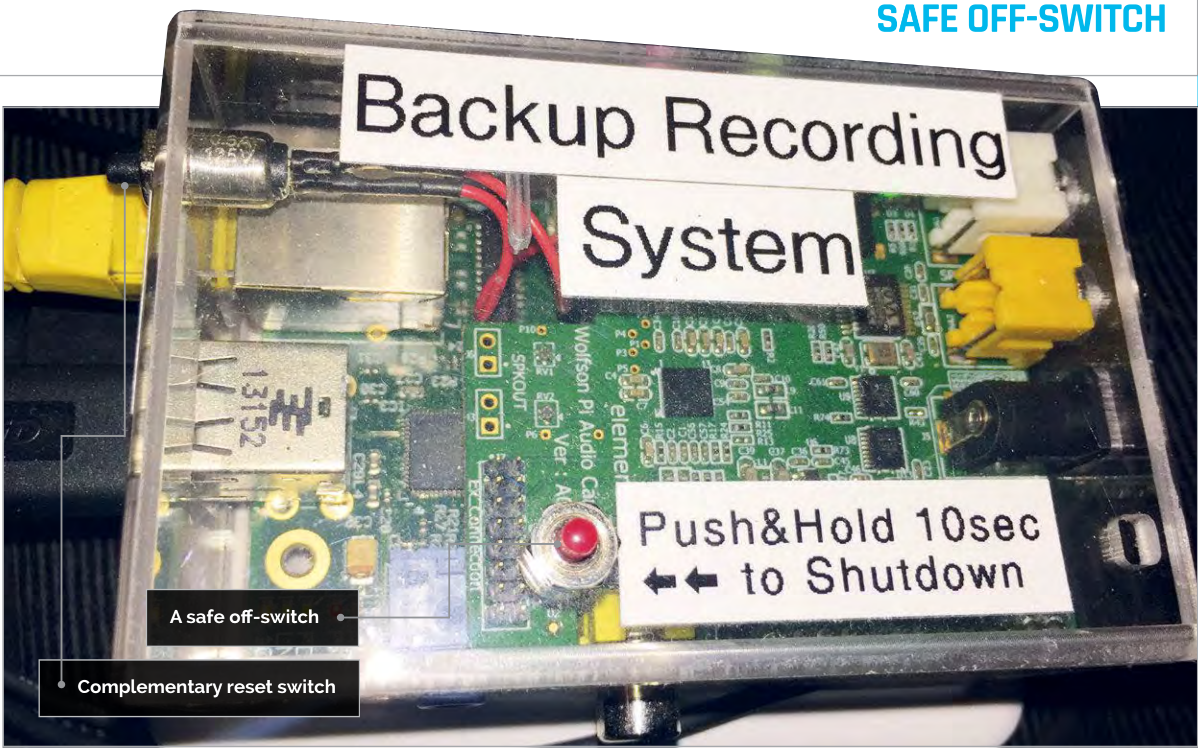
Then add a line to the end of **/etc/rc.local** to run it at boot time:

```
$ sudo su
# echo "~pi/shutdown-press-simple.py &" >> /
etc/rc.local
```

Now, after rebooting, your script will be running and listening for a button (connected between GPIO 21 on pin 40 and ground) to be pushed.

Language
>PYTHON

DOWNLOAD:
magpi.cc/2nD29F7



Preventing accidental button pushes

One major drawback of the previous code is that any accidental push of the button will shut your Raspberry Pi down. It would be better if you needed to hold the button down for several seconds before everything powers down. Check out **shutdown-with-hold.py**.

Instead of hard-coding the GPIO number 21 and the hold time, this code does a few things differently. First, it defines variables to hold these numbers at the top of the code. For a program this small, declaring the values at the top is not necessary, but it is good practice to declare any configurable variables near the top of the code. When making changes later, you won't have to hunt through the code to find these variables. Secondly, it allows the GPIO number and hold time to be overridden on the command line, so that you can change them later without modifying the program.

We then define a function named **`shutdown()`** to execute the **`poweroff`** system command. The button is also assigned to a variable for use in the next statement. This time, we are also specifying that the button must be held down, and when the hold time (6 seconds) has passed, any function assigned to the **`when_held`** event will be executed. We then assign that event to the **`shutdown()`** function we defined earlier. The call to **`pause()`** is needed to cause the script to wait for the button presses.

If you look at the examples that come with the GPIO Zero source, you'll find a script very similar to this one (**magpi.cc/2lANxVA**).

Feedback while pressing the button

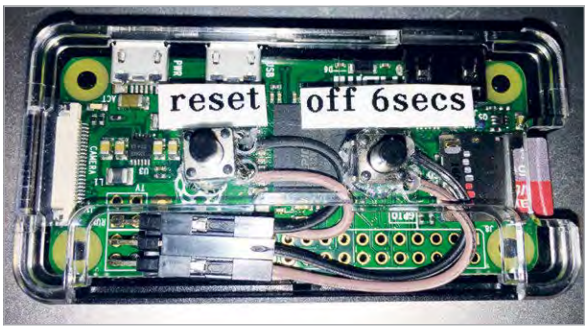
We can do better. The major thing lacking with the above code is any sort of feedback. It is hard to tell that anything is really happening while you have the button pressed down. Fortunately, GPIO Zero allows us to do much more with a button press, such as turning an LED on and off or setting it blinking, by attaching this to the button's **`when_pressed`** event. We need to ensure that the LED is turned off if the button is not held down for the entire length of time. This can be accomplished by attaching to the **`when_released`** event.

As before, the important work has been moved into functions named **`when_pressed()`**, **`when_released()`**, and the same **`shutdown()`** function we used before. These are assigned to their corresponding button events.

Going further

Can you think of other ways to provide feedback while pressing the hold button, or alternative ways to signal that it is time to turn off? How about using a buzzer, or popping up a message on a screen? You could also use the on-board activity LEDs, making them blink faster and faster as it gets closer to shutdown time. Or even play an audio clip, such as "I'm melting". The GitHub repository contains additional examples such as these. How about watching the 'low battery' signal from a battery pack as a signal to shut down? Let your imagination run wild.

Now, which of your projects are you going to add shutdown and reset buttons to?



Left Example of both an off and reset switch on a Raspberry Pi Zero. Right-angle headers are used for a compact connection. The switches are mounted directly onto an Adafruit case

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*.

magpi.cc/259aT3X

DIY CUT-OUT THEATRE

Create your own cut-out theatre controlled from your mobile device

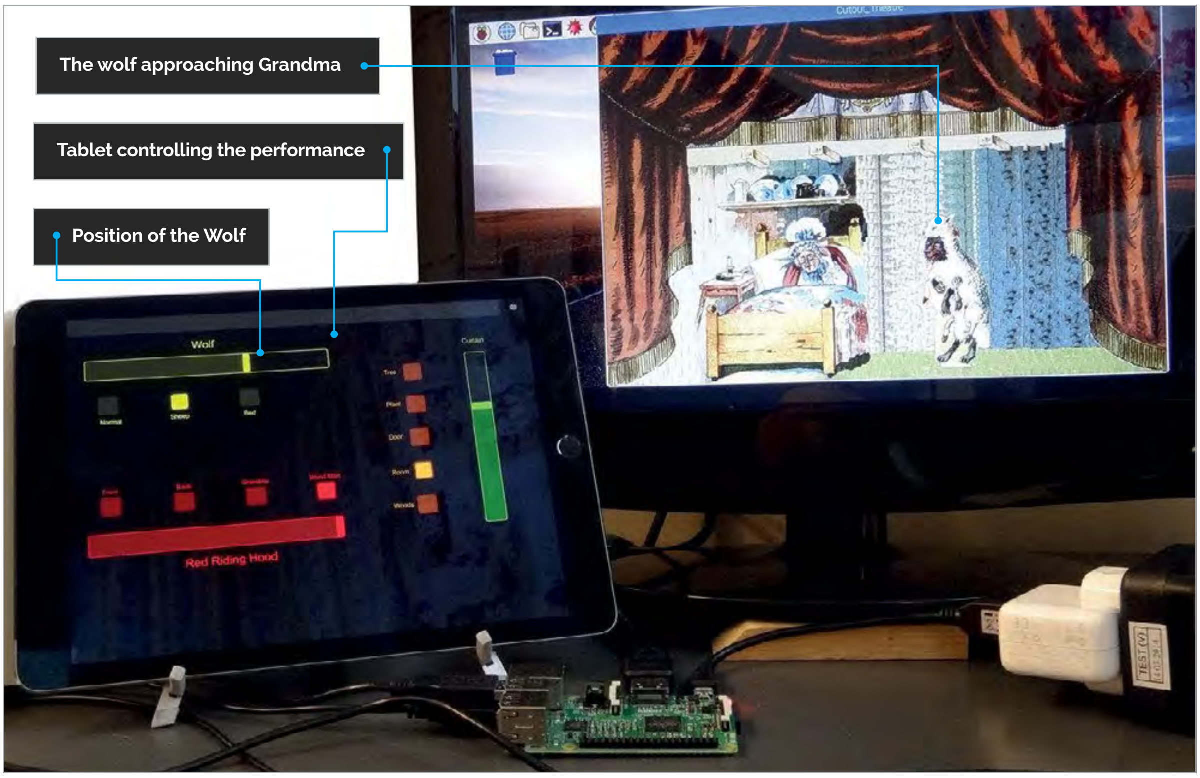
You'll Need

- Apple or Android mobile device
- TouchOSC app
- Raspberry Pi 3

The idea for this project is to replicate a cut-out theatre using the Raspberry Pi to show your production. As the Pi's screen will display the finished performance, that leaves no space on the screen for your controls. You might think that you could use a keyboard, but that would not provide proportional control and would be difficult to drive. The alternative, of using external hardware, could get expensive. A cheap solution, if you already have a smartphone or tablet, is to use that to control the theatre. All you need is an interface you can customise, and a way to send and receive messages. Enter OSC. OSC stands for Open Sound Control, and is an open, extendable method of exchanging messages. While it was originally designed for sound applications like mixers and effects units, it's flexible enough to be used for anything. TouchOSC is an excellent application that turns your mobile device into an OSC command-and-control station. It is hard to convey how satisfying it is to control the performance from a tablet!

TouchOSC

While it is not a free app, at less that £5/$6 it is not very expensive. There are two parts to it: the app that runs on your mobile device, and an editor for designing the custom control screens on your laptop. Go to the TouchOSC webpage at **magpi.cc/2nSa69S** to download the app and the editor. There are plenty of controls on offer, including faders, pads, rotary controls, and push buttons. Follow the instructions in 'Creating the custom performance' to create a simple custom interface for this project.



Language
>PROCESSING

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS

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

OSC messages

Each OSC message consists of two parts: an address and a payload. The address part is identical to the path of a file on the Pi, the only difference being that there are no directories, and you can build your own hierarchical structure to suit your own needs. On TouchOSC, each window can have a number of tabs, each tab bringing up a different screen. The first default hierarchical level is the tab number, preceded by a forward slash. There is only one screen for this project so all messages start with **/1**. The next level defines the control device, so a fader might have the address **/1/fader1**, or a push button **/1/push4**. That is as far down the hierarchy as we need to go for this project, but you can go further if you need to. The payload is the data that is delivered to that address, and can be in the form of an integer, a floating point number, text, or a Boolean logic value. When you send a message, you have to specify both the address and the payload.

At the receiving end, the message address is normally parsed to see how to apply this number to the program you want to control. In our application, the payload will be used to move graphics around on the screen. You can send messages back to the TouchOSC controller, where they can be used to change the controls without user intervention, or to add feedback or acknowledgement of commands.

Implementation

We did hope to implement the OSC messages in Python, as there are a few libraries to do this. Unfortunately this proved impossible, because the ones we tried had poor documentation that was out of date, and either fell over when trying to cope with three faders, or required complex multithreading programming with no practical examples available online. Instead, we turned to Processing, which has a simple and reliable interface for incoming OSC messages. We used Processing in the Ribbons project

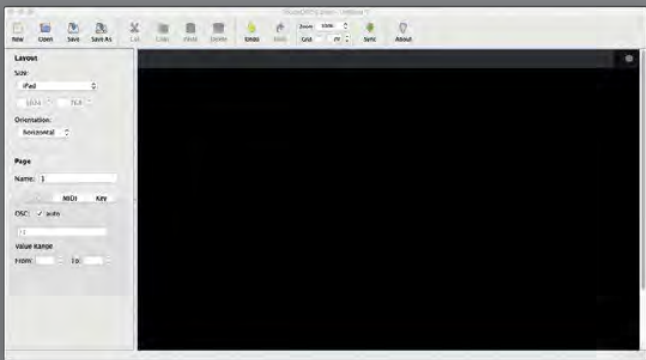
CREATING THE CUSTOM INTERFACE

>STEP-01

Installing the editor

First, download the free TouchOSC editor. There are versions for Mac OS X and Windows, and a Linux version, which will run on the Raspberry Pi. Download the ZIP file by going to **magpi.cc/2mPqfsz**. Unzip it using the Archiver tool under Accessories in the main menu. You'll find a file called **TouchOSCEditor.jar**. Move it into your project directory. To run the editor, navigate to the folder from a Terminal window and type:

```
java -jar TouchOSCEditor.jar
```



>STEP-02

Setting up the faders

Select the name of your device from the Size option, and set the orientation to Horizontal. Now right-click in the layout window and select 'H fader'. You might be surprised to see a vertical fader, but this is because you have selected the horizontal orientation for the screen. Resize this fader so that it is 50 by 415 and place it at coordinates 872, 89. Select the colour as green and the range from 0 to -380. Now create two V faders, resize them to 530 by 50m, and set the range to -240 to 900. Then put fader 2 at the x/y location 34, 61 and make it yellow, and fader 3 at 34, 462 and make it orange.

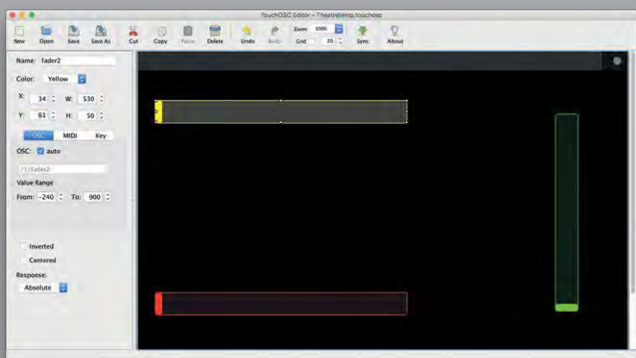


Fig 1:
HOST: Set this to the address of the Raspberry Pi. Hover the pointer over the WiFi icon to find it.

LOCAL IP ADDRESS: The address to put in the software is:
`String iPadIP = "192.168.1.69";`

in *The MagPi* #49, but in case you didn't install it then, it can be installed from a command line by typing:

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

After reboot, it'll then appear in the Raspbian menu under Programming. As with all network programs, you need to configure everything for your setup. In this case you only need two IP addresses. In the Processing code you need the IP address of the tablet, and on the tablet you need to set the IP address of the Raspberry Pi. This is shown in **Fig 1**. As in a network, the IP address can change from day to day, and it can be tedious always having to change the configuration before anything will work. You can get round this by accessing your router and telling it to always give a fixed IP address to both your Pi and your tablet.

The theatre

The theatre is simply a collection of graphics for the backdrops, props, and actors. These ware taken mainly from the story of Little Red Riding Hood in the Boston Sunday Globe of 1895. Curiously, this did not include an axe-carrying woodsman for the final scene, so we had to take one from another story and eliminate his wife by judicious use of the clone tool in our photo processing package! **Fig 2** shows the results of this process.

There are two actors in this story, and each one has its own slider control. Each actor also has a number of different costumes, with Riding Hood, in an amazing show of virtuosity, also playing the part of the woodsman, and Grandma's head. This saves on sliders and makes the control panel cleaner. There are two 'heads' for Grandma: one for talking to the Wolf, and the other for when the Wolf is pretending to be her. The movement controlled by the sliders is restricted to a simple nod, which is operated by tapping on opposite ends of the slider.

The elements are stacked up in a fixed order from back to front, defining what will show in front of what. A more complex setup could control this order, at least for the actors and props, but we're keeping things simple. The main task in adapting the graphics is to isolate the elements against a transparent background. You need to use a PNG type file and not a JPEG to define the transparent part. The image files must be placed inside the **data** subfolder of your Processing sketch's folder, in the **sketchbook** folder.

The software

The structure of the **Cutout_Theater.pde** sketch is quite simple. The **`Draw`** function runs repeatedly, and if a change has been signalled by the **`DisplayUpdate`** variable, it draws the new window. This prevents a lot

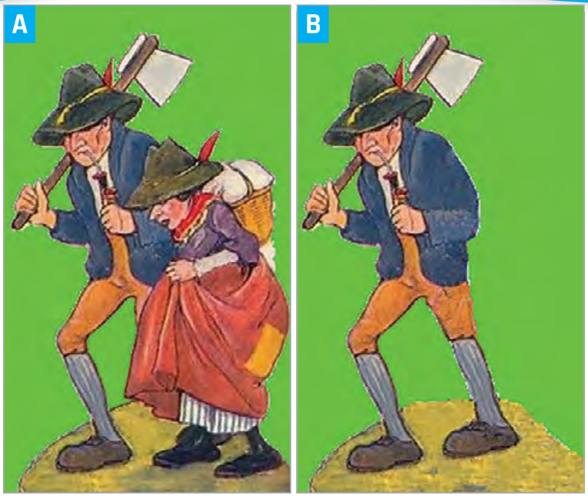


Fig 2 Removing the wife with the clone tool. It doesn't work in real life

of unnecessary drawings when nothing has changed. The **`Setup`** function defines the ports used for the OSC messages and loads in all the individual images. By far the longest function is **`oscEvent`**, which handles the incoming messages. This looks in turn at the two active control elements, push buttons and faders. The push buttons work as toggle buttons for the props and background by sending back a confirmation of the push, and fixing the colour of the button. This shows you that a message has got through, because occasionally OSC messages, like all slip-formatted messages, can go astray. The actors' costume buttons are implemented as radio buttons, so you can't have more then one selected at a time. The controls for the faders simply pass on their value to the appropriate variable. This is all that is needed because the range of the fader has been defined in the TouchOSC setup.

When the program first starts, the curtain is closed. A message is sent to the tablet to move the faders to reflect this. The initial elements for the first scene are also set up. There are a few other small functions that help with the parsing.

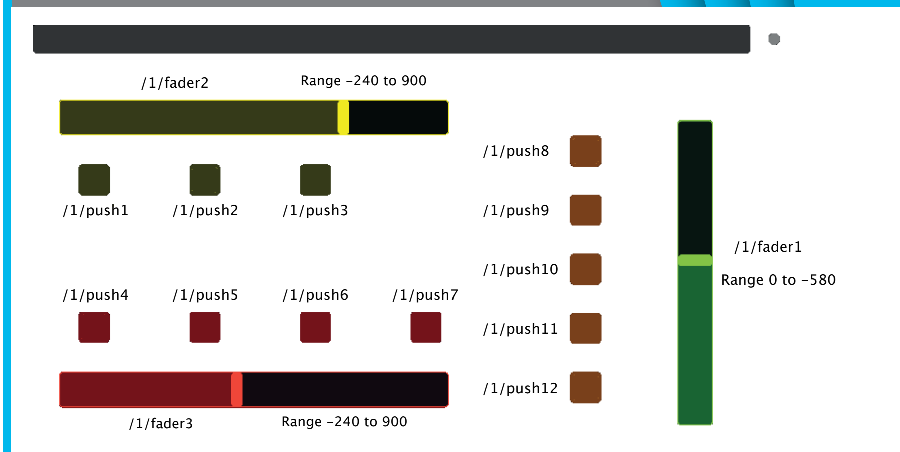
Taking it further

You can set up another play with your own characters, and once you see how it works, change the controller to cope with more variables. However, be careful not to make the control too cluttered. You could use some controls to trigger sound effects or animations, like the curtain automatically rising, or a prop falling over. You can replace the slider controls for the actors with a small XY pad for flying objects, or to put a bit of bounce into a walk. You could replace the single graphic of an actor with a sequence to show things like walking. Finally, you can add special effects, like lightning or a character dissolving in a teleporter.

>STEP-03

Add the push buttons

Right-click and select a push button. Resize it to 45 by 45, place it at 60, 148, and make it yellow. Click on the copy and paste icons and set the x position to be 210. Repeat this for x equals 360. Repeat this, move it to 60, 380 and make it red. Again, copy this and make three more buttons at x equals 210, 360, and 510. Paste again, change it to orange and place at 726, 109. Make four more copies at the same x position, but with y positions of 189, 269, 349, and 429. Check the controls have the addresses shown in the picture. You will need to click on each control in turn to see this information.



>STEP-04

Transfer the layout to the tablet

Finally, create labels for each element. The Wolf and Red Riding Hood should have a size 25 font. Save the layout, then open OSC on your tablet. Go to the layouts page and click on Add. Then click on the Pi, and click on the green Sync icon. You will see the Raspberry Pi appear on your tablet. Click on this name to download the layout, then click Stop Sync on the Pi. Go back to the configure page on the tablet and click Done to see your layout and try it out. If the devices don't see each other on the network, try power-cycling your router.

JAMES HOBRO

James is a versatile Pi hacker who specialises in multi-core programming. He has an alter ego as a research scientist at Schlumberger Gould Research in Cambridge.

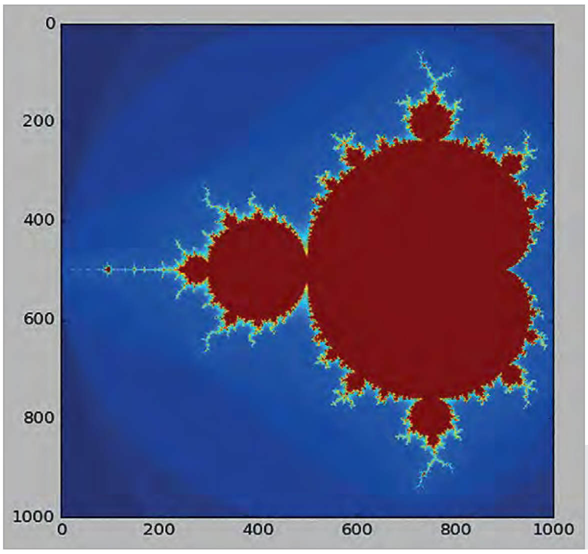
github.com/jhobro

MULTIPROCESSING WITH PYTHON

Power up your Pi from Python by running loops in parallel using the multiprocessing module (fractals are included)

Modern computers are good at multitasking. Quad-core devices like the Pi 2 and 3 can share work between their cores, and your programs can run up to four times faster if you know how to code them in parallel. However, this is not always easy. Programming languages like Python are sequential, executing instructions one at a time. What if you want to use all four cores? Luckily, there is help from the **multiprocessing** module, which allows parts of your program to run in parallel. To demonstrate how it works, we will adapt a program so that its central part runs in parallel, creating some beautiful fractal images in the process. First, you may need to install the **matplotlib** module, using:

```
sudo apt-get update
sudo apt-get install python-matplotlib
```



Right An image of the entire Mandelbrot set, as produced by the program mandelbrot.py

Embarrassingly parallel fractals

The program **mandelbrot.py** plots an image of the Mandelbrot set, which has an infinitely complex shape based on a surprisingly simple mathematical rule. At its core, it performs a simple but potentially expensive calculation for every pixel. These calculations are all independent, so they can be run in any order or in parallel (this is called an 'embarrassingly parallel' problem). Take a look at the program. The work is done by a pair of nested **for** loops (labelled 'main loops') which call the function **mandelbrot()**, defined a few lines earlier, for each pixel in the image. This is the time-consuming part. As with all basic loops in Python, the calculations are performed sequentially, or one at a time. Try running the program from the command line (unfortunately, multi-process programs cannot be launched from IDLE):

```
python mandelbrot.py
```

It should generate a Mandelbrot set image like the one on the left. You can resize it and use the controls on the bottom left of the window to zoom and examine it in detail. This program uses only one processing core and runs in about 40 seconds on a Pi 3. We would like to run the main loops in parallel. This is relatively simple, thanks to **multiprocessing**.

Introducing multiprocessing

Before we can parallelise the loops, we need to make a small change. Replace the main loops with:

```
Z = [complex(x,y) for y in Y for x in X]
N = map(mandelbrot,Z)
```

Now run the program again. It should take a similar amount of time to run. What has changed? Instead of calling the **mandelbrot()** function many times within the nested loops, we now build up a list of arguments for these calls in advance (**Z**) and use the built-in **map()** command to call the function on each entry in the list, placing the results in a new list (**N**). This does exactly the same thing as before, but our code is now in a form that we can parallelise more easily. Add the following line at the top of the program:

```
from multiprocessing import Pool
```

and change the main loop section to:

```
p = Pool()
Z = [complex(x,y) for y in Y for x in X]
N = p.map(mandelbrot,Z)
```

This is where **multiprocessing** works its magic. It creates a multi-process pool (**p**) and uses it to call a special version of the **map()** command. It makes all the same calls to **mandelbrot()** as before, but this time the work is split up and distributed in parallel using the pool. The number of processes in the pool matches the number of cores you have available (four in the Pi 2 and 3) by default. The results are collected back together into a single list (**N**) at the end. This program should run about twice as fast as before. If you run the **top** command in another Terminal window while the program is running, you will see the four extra Python processes that make up the pool. We have doubled the speed with a very simple change to our code.

Bunch them up

Can we do better? It turns out that the calls to **mandelbrot()** are too brief to run efficiently in parallel across the pool. For better performance we must bunch them together in groups. We can do this easily by returning to our two nested loops. For the inner loop, which produces the image along a single line in the **x** direction, define this function just below the definition of **mandelbrot()**:

MANDELBROT SET IMAGES

The Mandelbrot set is the set of complex numbers c for which the function $f(z) = z^2+c$ does not diverge for increasing z. In our plots, each pixel corresponds to a different complex number c. For every pixel, we determine whether the function diverges by iterating up to **maxiter** times. If we reach **maxiter** without diverging, then the point is deemed to be inside the set. Otherwise we record the number of completed iterations. Translating these numbers to a colour scale gives the patterns that we see surrounding the Mandelbrot set.

mandelbrot.py

Language
>PYTHON

DOWNLOAD:
magpi.cc/2nD2PtR

```
from numpy import linspace, reshape
from matplotlib import pyplot

xmin, xmax = -2.0 ,0.5      # x range
ymin, ymax = -1.25,1.25     # y range
nx  , ny   =  1000,1000     # resolution
maxiter    =  50            # max iterations

def mandelbrot(z): # computation for one pixel
  c = z
  for n in range(maxiter):
    if abs(z)>2: return n   # divergence test
    z = z*z + c
  return maxiter

X = linspace(xmin,xmax,nx) # lists of x and y
Y = linspace(ymin,ymax,ny) # pixel co-ordinates

# main loops
N = []
for y in Y:
  for x in X:
    z  = complex(x,y)
    N += [mandelbrot(z)]

N = reshape(N, (nx,ny)) # change to rectangular array

pyplot.imshow(N) # plot the image
pyplot.show()
```

```
def compute_all_x(y):
    Z = [complex(x,y) for x in X]
    return map(mandelbrot,Z)
```

This runs the inner loop sequentially, not in parallel. Now, for our main loop, we call this function in parallel, spreading the calls across the pool:

```
p = Pool()
N = p.map(compute_all_x,Y)
```

This maps the work in larger chunks (whole inner loops rather than individual pixels). This program should now run roughly four times as fast as the original. With some simple changes and the use of **multiprocessing**, we have distributed our time-consuming loop over four cores instead of one. Our code is still simple and readable, although slightly more complex than before. For advanced programmers, there are much faster methods of plotting fractal images. The graphics processor on the Pi can outperform its CPU, as demonstrated by the Mandelbrot set program at **magpi.cc/2mTVIKY**. This program, written in C and OpenGL, is much faster than ours, but it is also far more complicated and nearly 20 times as long.

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

AN INTRODUCTION TO C PART 11

MORE ABOUT TYPES AND VARIABLES

Global variables, type definitions, enumerations, and structures

In this instalment, we are going to look at some of the more advanced uses for variables and types, including the difference between local and global variables, defining new types, and the use of enumerations and data structures.

In the examples in this series, we have always put variables inside function definitions. These are therefore *local variables*; that is, variables which are local to those functions and have no meaning outside the function.

Global variables

C also allows *global variables*; that is, variables which are defined outside all functions. These have global *scope*, which means that they can be read and written from any function within the program. Let's look at an example:

```
#include <stdio.h>

int result;

void add (int a, int b)
{
  result = a + b;
}

void main (void)
{
  add (3, 4);
  printf ("The result is %d\n", result);
}
```

In this example, the variable **result** is a global. It can be read or written within both the **add** function and the **main** function. We write a value to it in **add**, and read it back in **main**, so we don't need to return a value from **add**.

This looks easier than passing values about all over the place, so why not just do this all the time? The answer is memory. Local variables in functions are temporarily allocated space while the function is running, and the memory is freed up as soon as the function ends. But global variables are allocated space when the program starts, and that space isn't freed until the program ends – if you allocate enough of them, you can run out of memory on some systems.

Type definitions

In a previous instalment, we looked at the range of variable types in C: **char**, **int**, **float**, and so on. C also allows you to define your own types, with what is known as a *typedef*. Here's an example:

```
typedef unsigned char BYTE;
```

This defines a new type called **BYTE**, which is another name for an **unsigned char**. (It is conventional to give user-defined types names in capital letters, to distinguish them from variables.)

When we say this defines a new type, what it really does is create an alias to an existing type. This seems a bit pointless, but it can help in two ways. First, it can make it more obvious what your code is doing if you make the type names specific to your program's data. Second, by defining specific types, you can get the compiler to warn you if you use the wrong type for a function argument or variable.

There are a couple of cases where typedefs are particularly useful – these are enumerated types and data structures.

Enumerated types

Sometimes we want a variable which can only take one of a few possible values. C provides a type called **enum** for this purpose, which defines an integer with a fixed set of named values. Here's an example:

. VS ->

When accessing the elements of a structure, make sure you are using the right symbol. A **.** is used if your variable is an instance of the structure itself; a **->** is used if your variable is a pointer to an instance of the structure. Using the wrong one will usually give an error from the compiler.

NEW TYPES INSIDE STRUCTURES

A structure can contain other new types (either plain types, enums or indeed other structures). Just make sure the typedefs for them occur before the typedef of the structure in which you want to include them.

NUMBERED ENUMS

When you create an enum, the compiler assigns a numeric value to each of the possible values. By default, it numbers the first in the list as 0 and counts up from there. You can override this by putting an equals sign after each named value and setting it to the value you want.

```
#include <stdio.h>

typedef enum {
  false,
  true
} BOOLEAN;

void main (void)
{
  BOOLEAN b_var;

  b_var = false;
  if (b_var == true)
  {
    printf ("TRUE\n");
  }
  else
  {
    printf ("FALSE\n");
  }
}
```

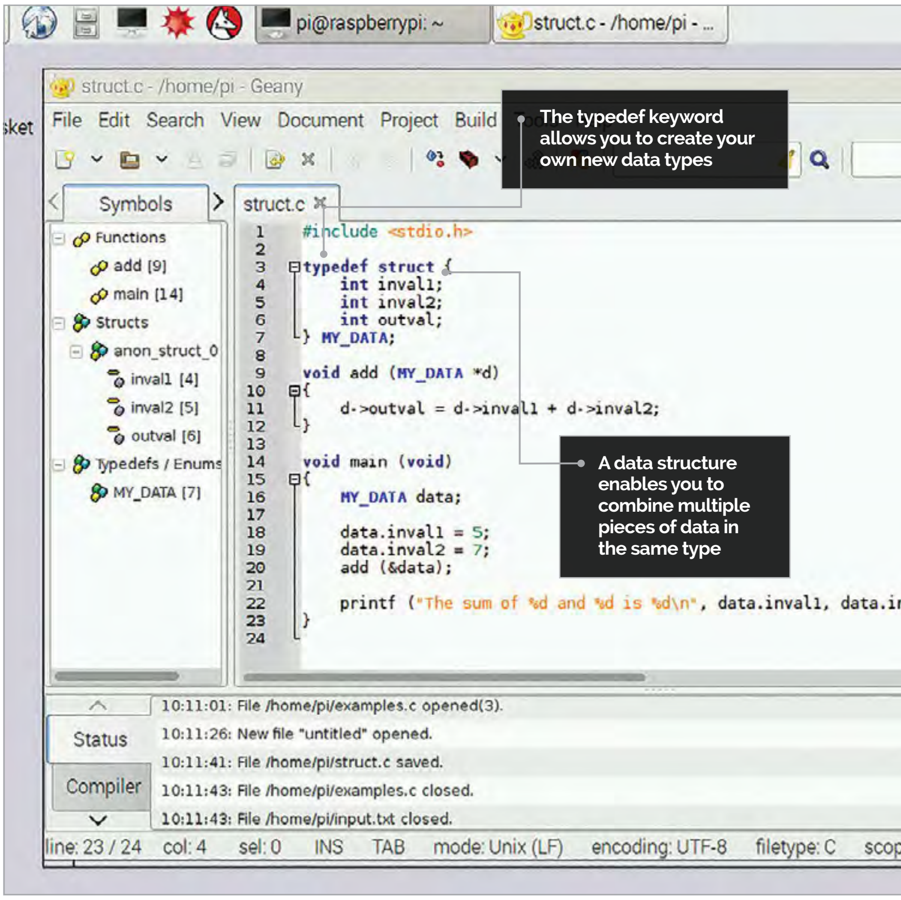
The named values of the enumerated type are used instead of numbers. This can make code easier to understand, and is a good way of preventing errors, as an enumerated variable can only ever be set to a valid value.

Structures

The other useful thing you can do with typedef is to use it to define a *data structure*, a collection of individual variables which are grouped together.

Here's an example:

```
#include <stdio.h>

typedef struct {
  int inval1;
  int inval2;
  int outval;
} MY_DATA;

void add (MY_DATA *d)
{
  d->outval = d->inval1 + d->inval2;
}

void main (void)
{
  MY_DATA data;

  data.inval1 = 5;
  data.inval2 = 7;
  add (&data);

  printf ("%d + %d = %d\n", data.inval1,
data.inval2,  data.outval);
}
```



We use a typedef to create a data type called **MY_DATA**. The definition of the structure consists of the keyword **struct** with a list of variables enclosed by curly brackets. In this case the structure consists of three integer variables.

In the **main** function, we declare an instance of the structure as a variable called **data** of type **MY_DATA**. We then access the individual elements of the structure by giving the name of the structure variable (**data**), a full stop (**.**), and the name of the specific element. So the line **data.inval1 = 5** sets the value of the element **inval1** of **data** to 5, and so on.

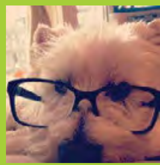
The function **add** takes a pointer to a **MY_DATA** structure as its only argument. As ever, a function cannot change the values of its arguments, but can change values pointed to by its arguments, so we pass a pointer rather than the structure itself.

To access the elements of a structure from a pointer to it, we replace the full stop with an arrow made up of a minus sign and a greater-than sign (**->**). So the **add** function reads the values of **inval1** and **inval2** in the structure pointed to by **d**, and then writes the result back to **outval** in the same structure.

Structures are useful if you need to pass a lot of data around between functions. They are more memory efficient than having large numbers of global variables, as you only need to create the structure when it is needed rather than taking up memory all the time.

USE DIFFERENT NAMES

While it is perfectly valid to give a local variable the same name as a global variable in the same program, don't do it! If you have a global and a local with the same name, the local version is used in the function in which it is declared, and the global is used everywhere else – this can result in unexpected behaviour.



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.
magpi.cc/2iUTZXz

SCRATCH GAME: PARKING MANIA

Create a Scratch game that tests your skills at parking. Will you get your Scratch driving licence?

You'll Need

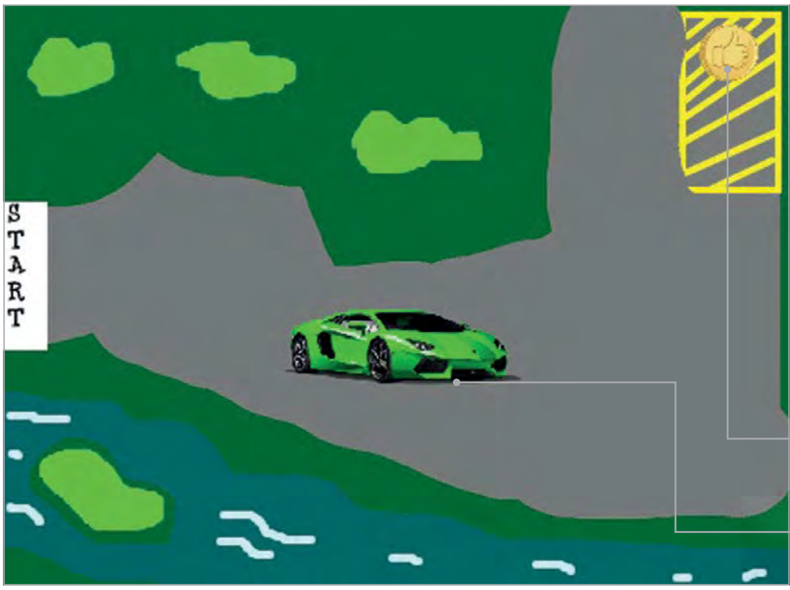
- Car sprite **magpi.cc/2kiV2gc**
- Car park backdrop **magpi.cc/2kj3sEB**
- Coin sprite **magpi.cc/2kj3pbR**

In this tutorial, we'll make a basic parking game using colour-based collision sensing with Scratch. Scratch is a child-friendly programming environment, developed by MIT. You can create games, music or even apps in this platform, and it is used in ICT lessons in schools. For this guide, you will follow some easy steps to create a simple but effective game. We will also be using angles, thanks to Miss Skelton, the author's maths teacher.

>STEP-01

Design

Download the assets from the list on the left and import the car park backdrop. Next, add the coin and car sprites. They'll be way too big to start with, so you will need to shrink each of them to the desired size by repeatedly clicking the 'four inwards arrows' icon above the stage.



This is the main objective of your game: get the coin and earn your driving licence, otherwise you'd better catch the bus

This is your car that you steer using the mouse pointer. Try to stay on the road!

Alternatively, you could create your own sprites and backdrop to make your game unique. Maybe you want to replace the blue river on the backdrop with a pink river, or replace the trees with footballs? If so, it is easy to edit or paint a new sprite costume using Scratch's built-in editor, so be creative.

>STEP-02

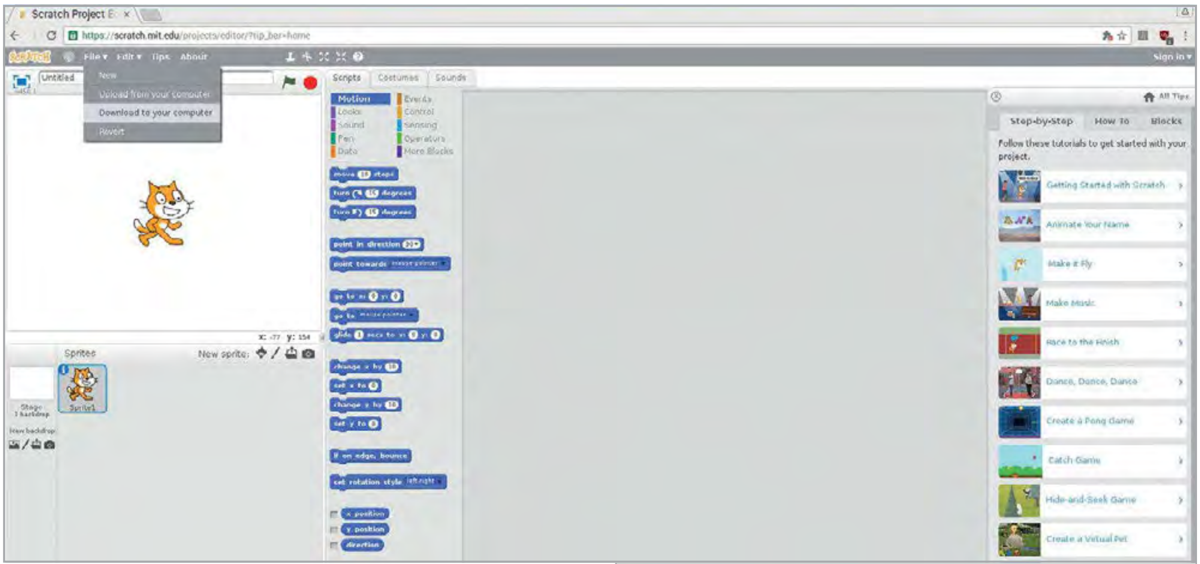
Start!

Let's make sure the car doesn't start near the coin or on the grass – we all want a fair chance of earning our driving licence. Instead, it should appear at the start line on the left. With the car sprite and its Scripts tab selected, you will need to grab a `when green flag clicked` Events block and drag it into the scripts area. Then attach a `go to` Motion block under it and replace the number in x with -229 and y to 24, so the car starts at the middle of the stage's left edge. In addition, we need to ensure the car points to the right following any failed or successful attempt, so add a `point in direction 90` Motion block.

>STEP-03

On your marks...

Let's get the car on the road. Under the previous blocks, attach a `wait 5 seconds` Control block. This will give the player five seconds to get the mouse ready at the start of a game. Next, add a `forever` Control block. Inside the latter, place a point `towards mouse-pointer` Motion block, then a `move 5 steps`



Left You'll need to code this in Scratch 2.0, so open up your browser and go to scratch.mit.edu

Motion block. This bit of code means the car will forever head towards the mouse-pointer. You can change the number of steps in the move block to alter its speed if you like.

>STEP-04

Crash detection

Next, we'll create some code to check whether the car has hit the grass. Under the previous blocks, still within the **forever** block, add an **if then** Control block. In its small hexagonal field, add a **touching color** Sensing block. Click its blue square, then click on the green grass of your course on the stage to set the correct colour. In the gap of the **if then** block, place a **say Game Over for 2 seconds** Looks block, then attach a **stop all** Control block – this will stop all scripts running when the car crashes.

>STEP-05

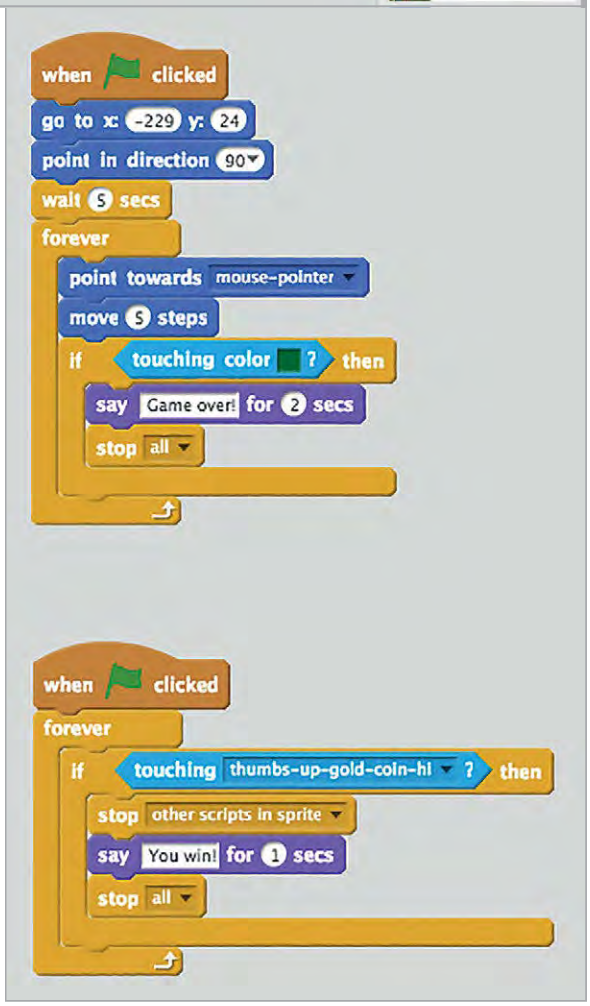
Grab the coin

We now need to create a second script for the car, to detect whether it is touching the coin in the parking space. Add another **when green flag clicked** block to the scripts area, then attach a **forever** block. Inside the latter, place an **if then** block. In its hexagonal field, add a **touching** Sensing block, and click its drop-down to change it to Sprite 2 (or the name of your coin sprite). In the gap of the **if then** block, place a **say You Win for 1 seconds** block, then attach a **stop all** block. This script makes sure that after the green flag is clicked, if the car is touching the coin, the car will say 'You Win' and the game will stop.

>STEP-06

Play it

You have now created a very basic parking game. We've made it fairly simple, so you should be able to code it and play it within an hour. Have a go on the game and see whether you can earn your driving licence – or should you just stick with a bus pass?



Left This is the only piece of code you will need to create your basic game

Now it's your turn. Here are some challenges for you to complete. Can you get the car to restart following a crash? How about creating a 5-4-3-2-1 countdown at the start of the game? Maybe you could add a timer? Can you create multiple levels? Why not add a coin variable? Instead of the car saying you have lost or won, why not make a special backdrop for the announcement? Why not, when you pass, create a driving licence? Otherwise, change the game in any way you want.

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 SPEEDS

HOW FAST IS THE RASPBERRY PI?

Processor

This is a bit different depending on the Raspberry Pi. The Model A, B, A+, and B+ are 700 MHz single-core processors. The Pi Zero and Pi Zero W have the same processor but it clocks at 1 GHz. The Raspberry Pi 2 and 3 are both quad-core, with the Pi 2 at 900 MHz and the Pi 3 at 1.2 GHz. The Pi 3 processor is also 64-bit, but it is used in 32-bit mode.

USB

The USB ports on all models are USB 2.0, which means they are limited to about 280 Mbit/s. On models with wired internet (Model B, B+, 2 and 3), the network cable is routed through the USB ports.

Networking

As the Ethernet is routed through the USB ports, you cannot make full use of Gigabit LAN on the Raspberry Pi. The networking is limited to 100 Mbit/s maximum speed. The wireless networking on the Pi 3 and Pi Zero W is 802.11n operating at 2.4 GHz, which means it operates at a maximum of 150 Mbit/s.

Right How quickly do all the pieces of a Pi communicate?

CAN I INCREASE THE SPEED?

Processor

You can overclock the Raspberry Pi (increase the speed of the CPU), although with the Pi Zero, Pi Zero W and Pi 3 you do so at your own risk. Other models support official overclocking modes in the config menu, increasing the CPU speed to up to 1 GHz. Any overclocking outside these menu options is at your own risk.

USB

You cannot increase the USB speed, but using fewer USB devices at any one time will help to improve performance. Make sure your USB devices are USB 2.0 Hi-Speed, and that they are properly powered.

Networking

You can increase the speed of your wired network by using a USB Gigabit LAN adapter. While it won't be a full Gigabit, it will be somewhat faster than the standard Ethernet connection on the Raspberry Pi.

WHAT ABOUT OTHER PORTS?

GPU

The VideoCore IV for the Raspberry Pi is rated at 250 MHz and can do 24 GFLOPS. The Raspberry Pi 3's VideoCore has been overclocked slightly so it runs between 300 MHz and 400 MHz depending on whether it is working on 3D or video.

CSI

This is the connector for the Camera Module. It has a fairly high data rate of about 1 Gbit/s maximum per lane. There are a couple of available lanes in the CSI of the Raspberry Pi, so it has a maximum perfomance of about 2 Gbit/s, although the camera only uses one lane.

DSI

The DSI is used to connect a screen to the Raspberry Pi. It runs at about the same speed as the CSI. Both are connected directly to the GPU, so they can't be used for generic data in the same way as a USB port. The DSI only uses one of its lanes when displaying data on a screen.

FROM THE RASPBERRY PI FAQ RASPBERRYPI.ORG/HELP

Why is there no real-time clock (RTC)?

The expectation is that non-network-connected units will have their clocks updated manually at startup. Adding an RTC is surprisingly expensive once you have factored in batteries, area, and components, and would have pushed us above our target price. You can add one yourself using the GPIO pins if you'd like an interesting electronics project.

Can I add extra memory?

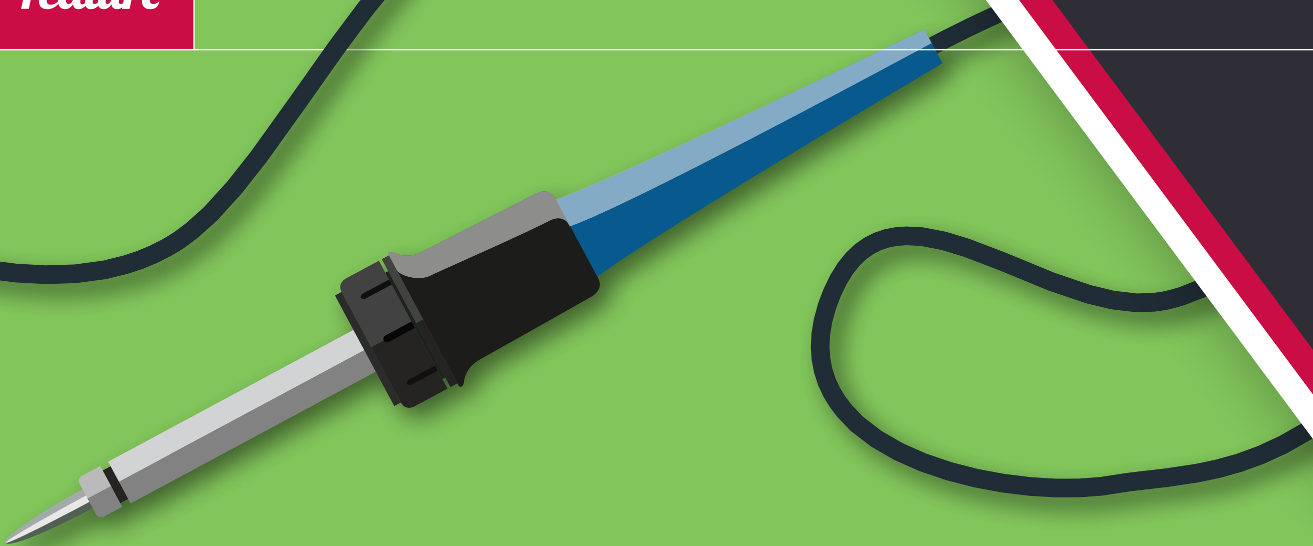
No. The RAM on the Model A, A+, B, B+, Zero, and Zero W is a package-on-package (PoP) on top of the SoC, so it is not removable or swappable. The 1 GB of RAM on the Model 2 and 3 is on a separate chip on the bottom of the PCB, but this is the maximum amount of RAM that the SoC in the more recent models can support.

Why doesn't the Raspberry Pi include <insert name> piece of hardware or <insert name> sort of port?

Our main aim is a charitable one: we are trying to build the cheapest possible computer that provides a certain basic level of functionality, and keeping the price low means we've had to make hard decisions about what hardware and interfaces to include.

What is its operating temperature?

The Raspberry Pi is built from commercial chips which are qualified to different temperature ranges. The LAN9512 is specified by the manufacturers as being qualified from 0°C to 70°C, while the AP is qualified from -40°C to 85°C. You may well find that the board will work outside those temperatures, but we're not qualifying the entire board to these extremes.



MAKERS' TOOLKIT

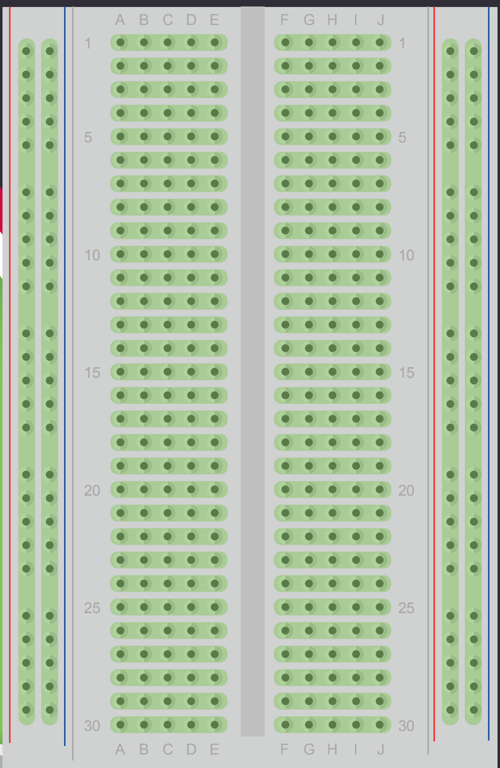
Everything you need to build awesome projects

We're into digital making here at *The MagPi*. Incredible projects using Raspberry Pis cross our news and social feeds every day, and we love to discuss them and how we'd go about making them. We tend to cover many programming and small projects in the magazine, so we don't often get to do the big impressive builds. If you have collected a few bits and bobs to build these small projects, you may have assembled the kit you need to attempt something bigger.

Over the next few pages we'll show you what you need to make your own amazing, Raspberry Pi-powered monster projects.

PROTOTYPING CIRCUITS

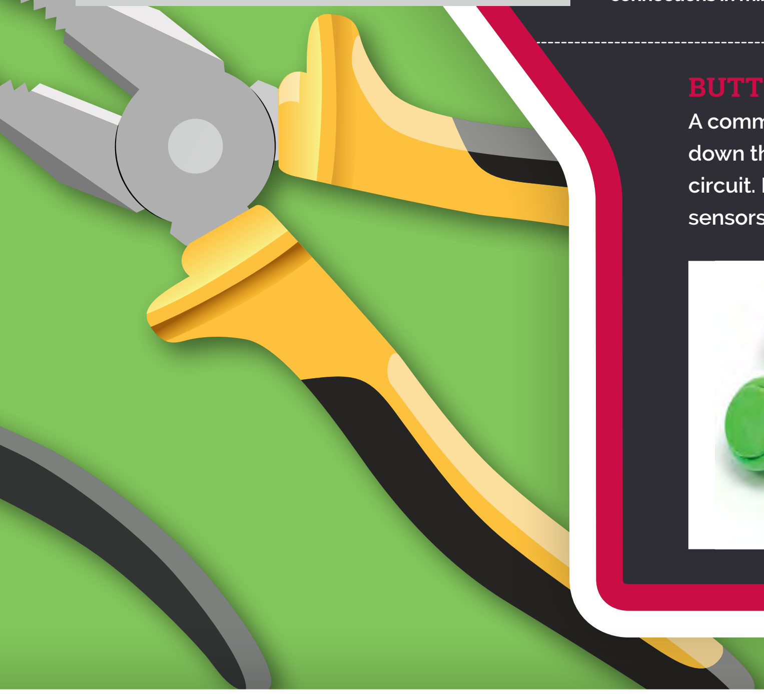
Taking your first steps into building circuits for a project



Left A diagram showing how the holes in a breadboard are connected. Keep these connections in mind as you build

The humble breadboard is an amazing device for prototyping circuits. We often use them in our tutorials, as they allow you to quickly wire up LEDs and other components to test concepts for your projects. Hopefully you already have one, but if not you can buy them from anywhere that sells electronics components, including Pimoroni, The Pi Hut, ModMyPi, etc.

Breadboards allow you to connect up components without the need for soldering. You can reuse the components and quickly rearrange them to test your ideas. You can wire them up to each other and to a Raspberry Pi to create a circuit. Some of the holes are connected by built-in strips of metal, allowing you to use fewer wires to complete your circuit.

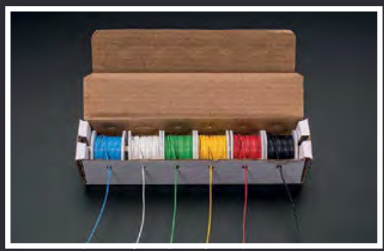


COMPONENTS

You can use these components in a breadboard. Most of them can be used in your final circuits as well.

PROTOTYPING WIRE

This wire uses a single core of metal throughout, which means that it bends easily. This makes it easy to use for prototyping circuits. While you can solder with this wire, it is not ideal for soldered circuits.



JUMPER WIRES

These make it very easy to hook up a breadboard to the Pi.

RESISTORS

Most electronic circuits will require a resistor. Different circuits will require resistors of different strengths.

BUTTON

A common input device; pushing down the button completes the circuit. Inputs can also include light sensors, temperature sensors, etc.

LED

The classic LED can be used to test your circuit, or to add lights to your project. Other outputs include buzzers, speakers, motors, actuators, and more.

BUILDING REAL CIRCUITS

Turn your prototype circuit into a permanent circuit

Breadboard circuits are great, and you could easily fit one into a project if you have space. Alternatively, use a soldering iron to make your circuit smaller and more permanent, and to protect against loose wires. Here's what you'll need to get started with soldering.

SOLDERING IRON

The most important piece of kit for soldering is the device that allows you to solder. It's basically a very hot pen tip that melts a metal alloy. After cooling, the solder keeps your wires and components in place, and conducts electricity.

SOLDER

This is the metal alloy that melts and cools to join components and wires together. It usually contains tin and lead, or tin and copper.

HELPING HANDS

When soldering tricky circuits, you can sometimes feel as if you need several sets of hands. 'Helping hands' can help! They usually include a heavy base and two crocodile clips, so you can hold two things together steadily. They often include a magnifying glass to help with very delicate tasks.

DESOLDER PUMP

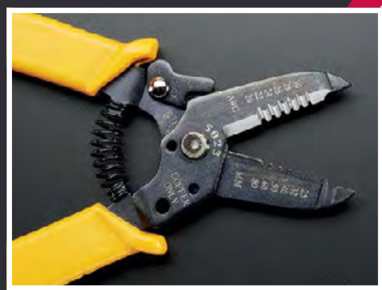
If you need to remove some solder from a circuit, heat it up with the soldering iron and then suck it up with this handy tool before it cools. Also known as a 'solder sucker'.

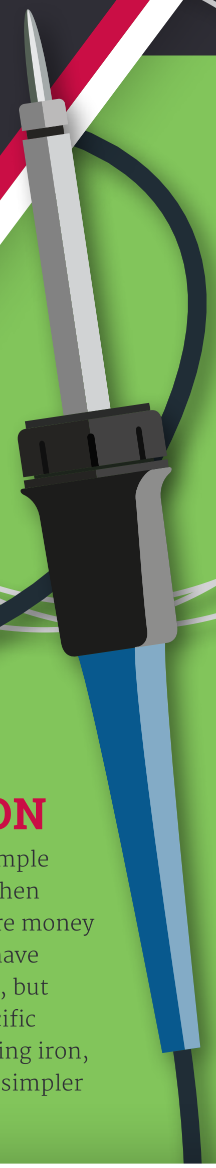
MULTICORE WIRE

This bendy, flexible wire is much better suited to soldering than the single-core wire used for prototyping.

WIRE CUTTER/STRIPPER

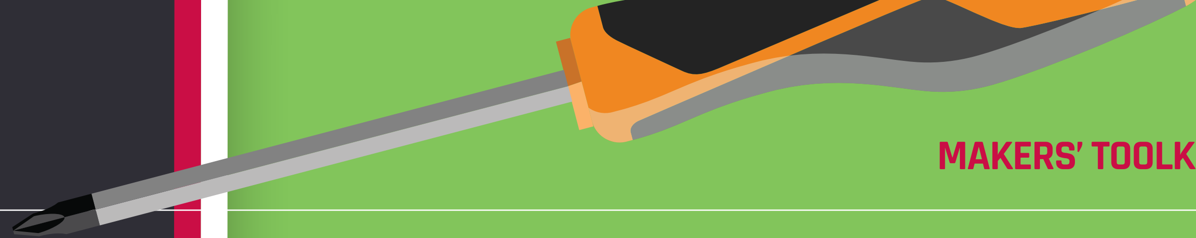
A special tool that not only helps you to cut a wire to the desired length, but also can strip the plastic coating away from the metal inside.





CHOOSING A SOLDERING IRON

Most soldering irons have a simple on/off switch; they'll be hot when they're on. You can spend more money and find soldering irons that have multiple temperature settings, but you'll only need these for specific applications. For a first soldering iron, you can easily get away with a simpler (and cheaper) option.



INTRODUCTION TO SOLDERING

Not everyone learns how to use a soldering iron in school

PREPARING YOUR SOLDERING IRON

>STEP-01
Not plug and play

A soldering iron is not ready to be used straight out of the box. You first need to put it through a process called 'tinning'. Turn your soldering iron on and allow it to heat up. Soak the sponge from the soldering iron stand with water and squeeze it out a bit before replacing it on the stand. Some stands have a brass wire cleaning ball instead of a sponge – these don't require water.

>STEP-02
Clean and tin

When the iron is hot enough to steam when it touches the moist sponge, wipe it on the sponge or the cleaning ball on both sides to clean it a little. Now put a bit of solder all over the tip of the soldering iron.

>STEP-03
Repeat

Repeat this a few times before starting to solder a circuit, and do it a few more times while soldering. It's best to do a bit of tinning whenever you turn on your soldering iron.

SOLDERING WIRES

>STEP-01
Measure twice

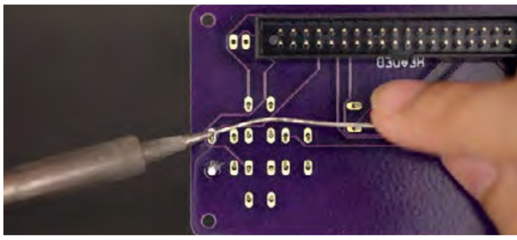
Make sure your wire is the correct length for your circuit. You can always strip one end and leave the rest attached to the spool to cut later. As the wire is very flexible, it is best to cut it a little longer than you need it to be.

>STEP-02
Tin the wire

Once you've stripped the plastic, you should tin the wire. Hold one end in a 'helping hand' and apply some solder to the exposed wire so that it coats and covers it. This makes it easier to solder onto something else.

>STEP-03
Attach the wire

Helping hands come in very handy here, holding both the wire and the component you want to solder. It is good practice to tin the leg of the component and the wire you want to solder to it, and then heat up the solder on both in order to fuse them together.



SOLDERING TO A BOARD

>STEP-01
Fit the component

Usually you'll be soldering a component to a PCB, so make sure you've got the right part and that it is orientated correctly. Slot it through the board and trim the legs of the component if you need to.

>STEP-02
Solder it on

Add a little solder so that the leg and the metal on the board are fused together. This makes sure that current can flow properly through the board and the component.

>STEP-03
Soldering considerations

If you're soldering on something like a header with several pins, make sure you have a plan of attack. Solder your components in a sensible order to avoid interesting wrist gymnastics.

FINISHING OFF

Physical circuits can be finished off by tying up trailing wires and using wire wrap to cover exposed connections. Wire wrap shrinks as it is heated up (with a heat gun or hair dryer) and keeps your connections safe and secure. To protect a PCB solder joint, we recommend covering it in a little hot glue using the low temperature setting on a glue gun.

TOOLS FOR BUILDING

Whether it's a kit or a custom design, here's what you'll need to put it together

A good traditional toolbox is the maker's best friend. While a well-stocked garage may provide the tools you need for most making tasks, here's what you should keep in your toolbox to complement your trusty hammer.

SCREWDRIVERS

Essential tools for building kits and your own custom projects. As well as the usual range of Phillips and flat-head screwdrivers you might find in a set, you should also buy some very small screwdrivers (the kind you use for glasses, or find in Christmas crackers) for the more fiddly screws.

RATCHET/SOCKET SET

Need some nuts tightening? We could probably phrase that better, but we're very serious about getting a ratchet set. It will make your life so much easier.

VARIABLE SPANNER

When a ratchet fails and you need a trusty spanner, we always like to have a wrench or variable spanner to hand to make sure everything gets tightly fastened.

PLIERS

Standard pliers and long-nose pliers can get you out of a jam when you can't quite brute-force something yourself. Don't get stuck without them.

UTILITY KNIVES

A good Stanley-style utility knife and a craft knife will help you cut and trim pieces for your project. Parental guidance advised.

FILES AND SANDPAPER

Want to give your project a nice finish or clean up a 3D print? Make sure you have a couple of good files and a selection of various grades of sandpaper. For plastic, you will need very fine, high-grade sandpaper.

GLUE

Sometimes you just need to glue something. Superglue, wood glue, hot glue, epoxy resin – it's good to have a selection to hand to cover any eventuality.

UNDERSTANDING SCREWS

You'll find different screw measurements used online, depending on whether the guide you're following uses imperial or metric scales. Here's a handy table for converting screw diameters between imperial (UNC or UNF) and metric.

UNC	UNF	METRIC
1-64	1-72	M2 × 0.4
3-48	3-56	M2.5 × 0.45
4-40	4-48	M3 × 0.5
6-32	6-40	M3.5 × 0.6
8-32	8-36	M4 × 0.7
10-24	10-32	M5 × 0.8
1/4-20	1/4-28	M6 × 1
5/16-18	5/16-24	M8 × 1.25
3/8-16	3/8-24	M10 × 1.5

3D PRINTING

A new way to make cases and parts, 3D printing can make makers' dreams a reality

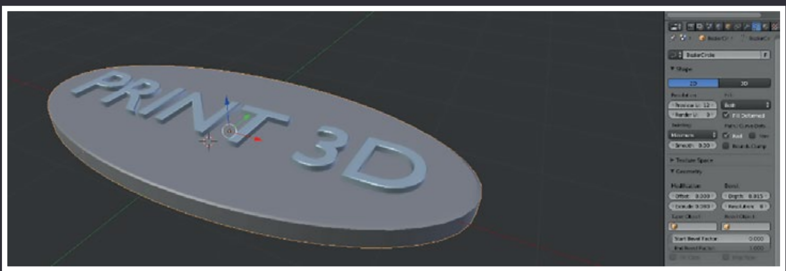
SORTING OUT YOUR FILES

Think of a 3D printer as being like a 2D printer: for 3D printing, you need to prepare the files you want to print. If you're printing files that someone else has designed, the process should be very easy. Download and sort through the files recommended for your project, and make sure you are using the right ones. 3D-printable files will usually have file names ending in .stl or .vrml.

Making your own files involves diving into the world of 3D modelling software. You can choose from many paid professional programs, but there's also the free and open source Blender 3D modelling software. There is a helpful guide to using Blender here: **magpi.cc/2oxxWFA**.

Remember, standard 3D printers print from the ground up in slices or layers. Keep this in mind when designing pieces to be assembled after printing.

The more simple the design, the easier it will be to print

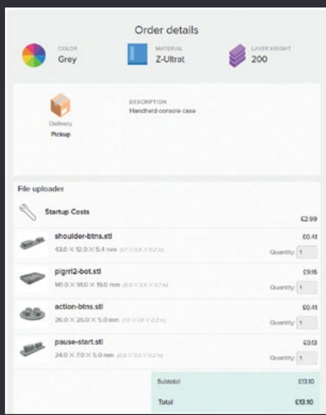


PRINTING YOUR FILES

Once you have your files, you need to print them. You could get yourself a 3D printer (we've always liked the MakerBot series), which then allows you to print as much as you want (plastic permitting), making tweaks to allow for the perfect print.

Printers are very expensive, not to mention noisy, so a popular alternative is to find a local 3D printer on **3DHubs.com**, and send the files to them. They'll be able to give you recommendations on materials to print with, and they will usually mail the items back to you, or allow you to pick up the print yourself in some cases. There's plenty of choice, and they are a lot cheaper than the setup cost for a 3D printer.

Of course, if you plan to do a lot of 3D printing, you might be better off paying the extra and investing in a 3D printer of your own.



Printing on 3D Hubs can be a lot cheaper than you think

FINISHING YOUR PRINT

Typically, 3D printed objects have a rough surface texture, due to the layering nature of the printing process. You can change the way you print to cut down on this effect, either by using different materials or by employing a different 3D printing technique.

Mostly, you'll find yourself sanding and filling and sanding and filling again to produce a smooth finish. Another solution is to add layers of PVA glue, which can be easily painted over.

You can also try using acetone vapour baths to smooth out your surface finish. These take very little work, but they have their own disadvantages: you'll need to build your own DIY kit and deal with some nasty chemicals.

Don't forget to buy a small round file to widen screw holes if you need to!

Right An amazing idea for a 3D print project we covered in issue 56.

Below Before and after images of an acetone vapour-smoothed 3D print from PeterA8's great guide: **magpi.cc/2oxLj8R**

WOODWORKING

Take your first steps into carpentry and make some truly wonderful projects

A fully stocked woodworking workshop can be vast, includes a lot more than just a saw, and can get quite expensive. Along with the tools in your makers' toolbox, here are some of the basics you'll need for projects involving woodworking.

TAPE MEASURE AND PENCIL

It seems obvious but it's worth repeating. You don't want to go about measuring with the 15 cm ruler from your old school pencil kit. Remember the golden rule: measure twice, cut once.

POWER DRILL

Need to make some holes? This is what you need. Pick up a variety of drill heads for wood and you'll be on your way. Be careful, though, as drills can be dangerous.

VICE

Keeping your wood in place with a vice makes it easier to cut. We also really like the workbenches that grip your piece of wood, and even create a gap underneath so you can safely drill holes without damaging the bench.

CLAMP

Clamps are very useful for holding sections of wood in place, to drill uniform holes or make precise cuts. You'll need several clamps for the best results.

POWER TOOLS

Want to improve your carpentry skills? Then you may want to get some power tools. Of course, these should only be used with adult supervision. We recommend a power sander (sanding large bits of wood by hand is very boring) and a jigsaw.

A jigsaw is a handheld tool with a saw blade jutting downwards. It is very useful for cutting complex shapes into, or out of, a piece of wood. To use it, you rest the metal plate on the wood and push the jigsaw along. This pushes the blade through the wood, usually following a pre-marked path.

SAFETY FIRST!

Woodworking is the most dangerous making method we've covered here! Be sure to buy some safety goggles (sawdust is nasty) and watch your fingers with everything you do!

WOODWORKING TIPS

01. MEASURE AND SKETCH

You can't just pick up some MDF from B&Q and cut it blindly; you need to know what you're doing. Design your project on paper, then break it down into the individual parts you'll need to make. Take measurements and create a mock-up with paper or cardboard if you can to make sure it's what you want.

02. GLUE AND FASTEN

If you want your build to be extra secure, and you're not too bothered about dismantling it afterwards, try using some wood glue on any joints you're also fastening with a screw. This will make the joint stronger than using either method on its own.

03. FILLING HOLES AND DENTS

Screw holes and any nicks or other minor deformities in the wood can easily be smoothed out by filling them with some wood filler. Use as little as possible and use a filler knife to scrape off any excess. Let the filler dry and sand it down until it is smooth. You may need to do this in several layers.

04. FINISHING THE PROJECT

You may want to paint or seal the wood if it is going outdoors, or has a specific function. Varnish and wood paint should be used in well-ventilated areas, and you should varnish or paint your structure before fitting any electronics.

MAKERSPACES

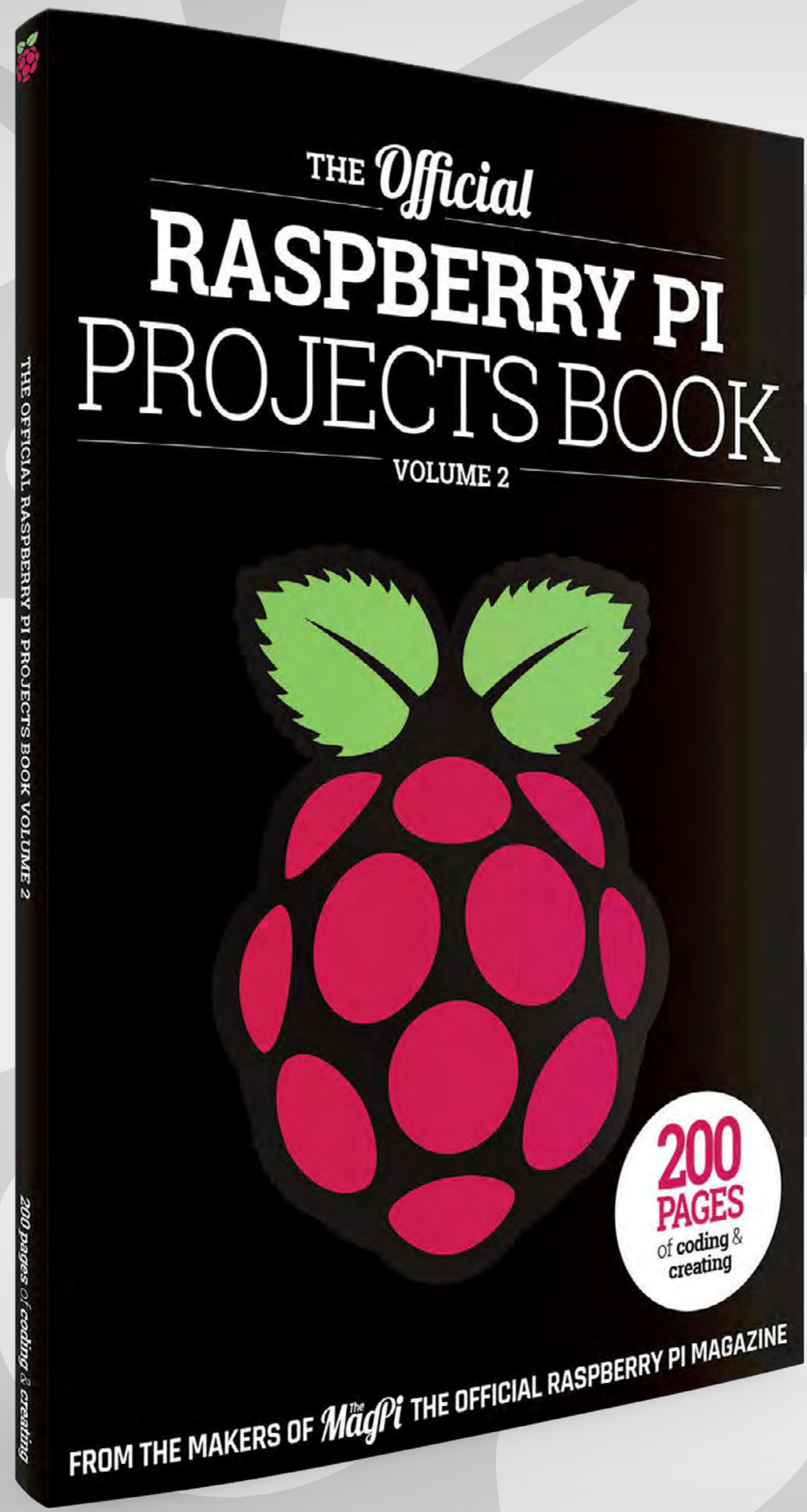
No space for a workshop? Find a local makerspace

Makerspaces (also known as hackerspaces) are amazing places where people come together to build things. From software to circuits and 3D prints to metalwork, a good makerspace gives you access to loads of tools for building and making. Buying tools can be expensive and finding space can be difficult. Many makerspaces require only a small monthly membership fee to use all their equipment. They're also usually filled with knowledgeable people who can help you with any tricky parts of your projects.

To find a makerspace, check out **hackspace.org.uk** and look for locations near you. Alternatively, a bit of Google-fu should help you find somewhere in your area.

Being a maker is also about being part of a wider community, and joining a makerspace can open the door to a community of makers, and a wealth of support, on your doorstep.

Right The Makespace in Cambridge is one of our favourite places in the world (and it's usually a bit quieter than this)


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 2
200 PAGES of coding & creating
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 2
200 pages of coding & creating

Maker Says

"Build your very own internet-connected radio with the Pirate Radio Kit!

Pimoroni

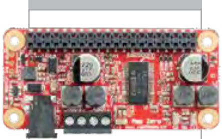
PIRATE RADIO
PI ZERO W PROJECT KIT

This musical pirate booty is something to treasure

Related

AMP ZERO PHAT

JustBoom's Zero-sized board delivers up to 20W RMS of high-resolution audio to a pair of passive speakers using an optional external power supply.



£24 / $37

magpi.cc/2lNJMKO

The most expensive of the four new Raspberry Pi Zero W project kits from Pimoroni, the Pirate Radio comprises a case full of quality components – everything you need to build your own internet radio. As with the other three kits, the packaging is top-notch and the hinged plastic case can be reused to store other components after the build.

The key electronic items featured in the Pirate Radio kit are a Pi Zero W, with built-in wireless LAN and Bluetooth, and one of Pimoroni's new pHAT BEATs. Also available separately, the latter is a neat bit of kit that crams dual I2S DAC/amplifiers onto a Pi Zero-sized board, and can pump out 3W per channel.

The body of the radio is fairly easy to assemble – from acrylic pieces, legs, retainers, nuts and bolts – using the illustrated step-by-step online guide (**magpi.cc/2o1i7d7**). As long as you follow the guide carefully to get the orientation right, everything slots together neatly. The acrylic pieces are all laser cut, including a neat speaker grille, so there are no nasty rough edges.

The supplied 5W speaker simply slots onto four bolts holding the translucent front acrylic layer in place. Fortunately, the speaker comes with a length of dual wire already connected, so there's no need to solder it. We're not sure why the wire is much longer than required, though; while you could always cut it to size, we just wound it round the bolts at the rear of the radio.

Unless you opt to buy a couple of Pimoroni's ingenious hammer headers separately, you will need to break out the soldering iron to attach the supplied standard male and female headers to the Pi Zero W and pHAT BEAT. The latter then slots onto more bolts at the rear of the radio, with the Zero W mounted on top. The speaker wires are inserted into a couple of the terminal blocks on the pHAT BEAT, with the latter's dip switch set to mono to combine its stereo channels. With that, your internet radio is built!

Streaming software

The Pi Zero W's built-in wireless connectivity means there's

magpi.cc/2ot01eJ

£40 / $50

Left The kit comes packed in this neat reusable case

no need to use a WiFi dongle plugged into a USB to micro-USB adapter, which makes for a more streamlined look to the radio. Even so, such an adapter is included in the kit, along with an HDMI adapter. This is presumably to enable you to hook the Zero W up to the monitor to install the software in Raspbian and set up WiFi, although we went the instant headless route by adding **ssh** and **wpa_supplicant.conf** (with our router details) files to the microSD card before first boot.

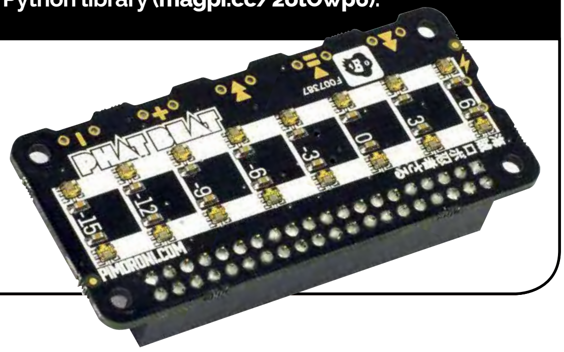
On the software side, Pimoroni has put together guides for three project examples. The first is for an internet radio based on the VLC daemon. As with the other examples, a single command is used to install all the required packages. You can then edit the playlist file to add URLs for your favourite radio stations. With this particular project, everything can be controlled via the pHAT BEAT's five side-mounted buttons: forward/back to select stations, pause/play audio, and volume up/down. The only slight downside is that the buttons are tiny, and a little difficult to locate on the side of the radio at times. The sound quality is good, however, with a decent amount of volume. Its real-time volume level is shown dynamically by the pHAT BEAT's super-bright LED VU meter.

> "Quality components – everything you need to build your own internet radio"

In addition, we followed Pimoroni's tutorial to turn the Pirate Radio into an AirPlay speaker for streaming audio from an iPhone and iPad. Both this and the VLC radio work alongside each other happily, so you can switch from one use to the other. Highlighting the radio's versatility, Pimoroni has also put together a Spotify streaming project using Modipy, controllable from a remote computer or device.

pHAT BEAT

Pimoroni's new Pi Zero-sized audio board packs twin MAX98357A DAC/amplifiers for stereo output. While the Pirate Radio uses mono mode, you could always add an extra speaker. Alternatively, you can buy the pHAT BEAT separately (£16.50/$21) and build your own custom stereo radio, or even repurpose an old ghetto blaster. The pHAT BEAT also features six push-buttons and two rows of super-bright APA102 RGB LEDs, for use as a VU meter or custom-controlled using the board's Python library (magpi.cc/2otOwp6).



Last word

While it's a slight shame that it doesn't make use of the pHAT BEAT's stereo capabilities, this is an excellent kit that is easy to assemble and results in a genuinely useful audio device with good sound quality. As well as internet radio and music streaming, potential uses include an Alexa-style voice assistant (with the addition of a USB mic), a speaker for musical HATs, and a speaking clock.

★★★★★

magpi.cc/2oqkumh

£25 / $31

Maker Says

A powerful, sleek and intelligent smart device that is perfect for controlling your high-definition audio setup

JustBoom

JUSTBOOM SMART REMOTE

A minimalist AV remote with a six-axis gyro installed. Should this replace your current media remote?

There were always jokes about the Wii Remote looking like a TV remote control, and it looks as if the JustBoom team has taken these jokes as inspiration for a slightly different kind of media remote. While looking simplistic, with only a few buttons, the Smart Remote's secret is that it contains a six-axis gyroscope that allows for a degree of motion control.

Luckily, this feature isn't on all the time. It is activated and deactivated by pressing the button in the middle of the volume control strip, and emulates a mouse pointer moving around the screen as the user dictates, by flicking and tilting the remote.

It works pretty well, making an instant transition between modes, and can easily be recentred by turning the motion controls off and on again. We found on Kodi that it tracked across the screen fairly slowly, forcing you to bend your wrist to extreme angles to get to the corners. On Windows PCs, and indeed on the Raspberry Pi, it required less work to move around the screen.

As for being a pure media remote, it does the job well. JustBoom claims that it's designed to have the bare minimum of useful buttons for media viewing, and for Kodi this works well. The thing we missed most was a Play/Pause button, but you can definitely get used to tapping OK twice to pause, so it's quite a minor complaint.

The build quality is pretty decent as well, with the buttons making satisfying clicks as you push them. They're also nice and large, so you're unlikely to push the wrong one.

As mentioned earlier, they do work on the Raspberry Pi desktop, which is pretty neat. The OK button works as a normal mouse click, and Back and Home work on a browser. It could well be useful in a classroom environment, as well as in the home, for media viewing.

Related

RII MINI I7

A similar idea to the JustBoom remote, this one has a few more features to allow use as a wireless PC mouse as well.

£15 / $33

magpi.cc/2mLP3qc

Last word

It's a decent bit of kit with a unique feature. We'd like a Play/Pause button, but otherwise it's perfectly serviceable.

★★★★☆

magpi.cc/2ou5KTh

£26 /$35

Maker Says

"Add low-power display technology to your Pi Zero projects!

Pi Supply

PAPIRUS ZERO MEDIUM

An ePaper screen for the Raspberry Pi Zero

Related

PAPIRUS HAT

The original full-size PaPiRus features a taller display, up to 2.77 inches in size. The same ePaper technology is used to render text and images.

£35 / $50

magpi.cc/2ouxPtz

Many moons ago, we took a look at the original, full-size PaPiRus HAT. Now it's the turn of its smaller sibling, designed specifically for use with the Pi Zero. Unlike conventional displays, it uses electronic paper (ePaper) technology to render text and images, which can remain on screen without any power connection for many days before slowly fading. Obviously, this is ideal for saving battery power in a portable project, such as a smart conference badge. The display is high contrast and readable even in bright sunlight.

Like its big brother, the PaPiRus Zero is supplied in two main parts: the pHAT board with pre-soldered female header, and the screen itself, which is very thin. Assembly involves inserting the screen's mini ribbon cable into the connector at the side of the pHAT, then securing the screen to the board with the double-sided sticky pads supplied. We tested out the 2.0-inch Medium display with 200 × 96 pixels, but a Small 1.44-inch screen (128 × 96) is also available, along with a multi-screen pack containing both sizes.

With the unit assembled and mounted on a Pi Zero's GPIO equipped with male header (not supplied), it's time to install the software via a single Terminal command. A second command is required to set the correct screen size, then you're ready to roll.

One of the installed folders contains a few Python code examples to get you started, including a temperature readout that makes use of the pHAT's built-in LM75 temperature sensor. The Buttons example demonstrates the use of the pHAT's five GPIO-connected buttons. Located along the top edge, they're tiny but could prove useful for switching the info displayed. Other code examples include Conway's game of life, which works well, and a digital clock. While some minor latency is noticeable as the numbers change on the latter, a 'clear' command can be used to wipe the display clean.

The Papirus Python library includes a write function to print text strings, which can be positioned and sized, although it's not clear how to change the font. A draw function displays a one-bit (black and white) bitmap image – *The MagPi* logo looked really cool! Handily, the software will convert most image types automatically.

Last word

While not suitable for applications requiring a fast screen refresh, the PaPiRus Zero is ideal for saving battery power in portable projects. The ePaper display is very easy to read in all but the lowest light conditions, from any viewing angle, and features a decent 110 ppi pixel density.

★★★★☆

IDE GROUP TEST

Which program should you be using to code on the Raspberry Pi?

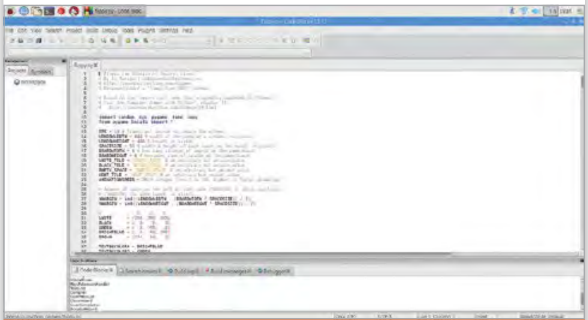
The humble integrated development environment (IDE). Used the world over to play with code and make programs on computers of all kinds. You're probably familiar with Scratch or IDLE, IDEs that serve a specific purpose, in this case coding in Scratch or Python respectively. Some IDEs let you program in several languages. Here are some of the best on the Raspberry Pi.

CODE::BLOCKS

A fairly traditional IDE, Code::Blocks is well known as an alternative for C coding, but it can handle many different languages as well. It's not included in Raspbian PIXEL by default but it is in the package manager, so it's just a case of installing it from the Terminal or in the software installer.

Code::Blocks lets you organise your code and projects very easily, displaying the different functions and classes and whatever is important in the sidebar. It also contains a lot more debugging, testing, and building options than main rival Geany; however, we had some issues getting it to preview some Python code.

It's a very solid choice if you need a little more data and debugging in your projects, and again you can run compiled software straight on your Raspberry Pi for very efficient testing.



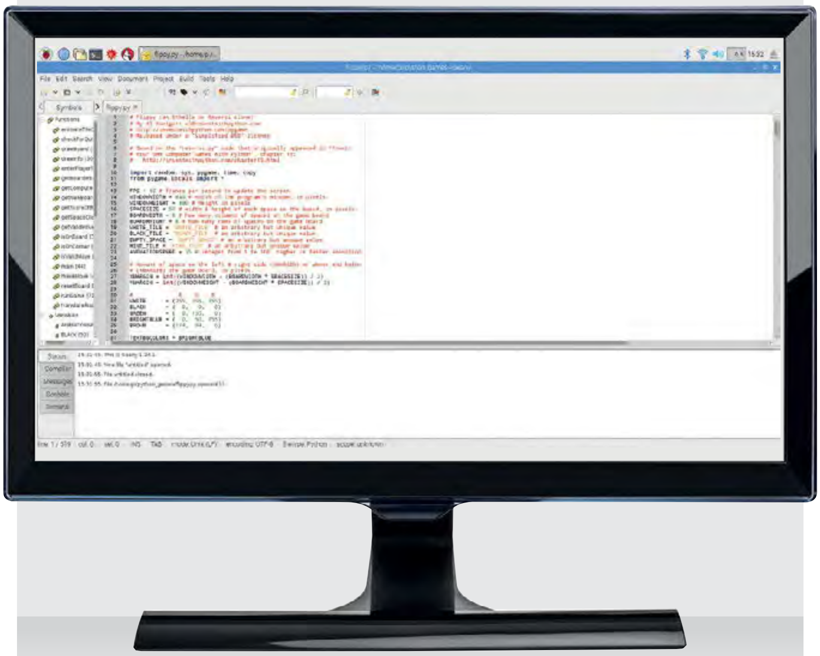
codeblocks.org

GEANY

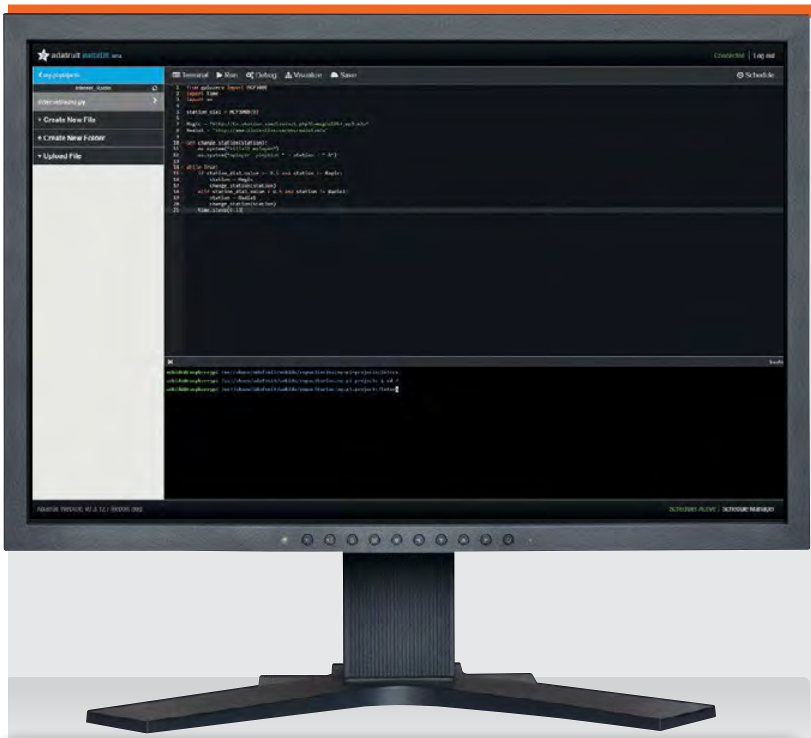
A very popular cross-platform IDE, Geany has been around for quite a while and has received a reputation as a nice lightweight alternative to the bigger IDEs like Eclipse. It's perfectly suited for the Raspberry Pi, aiding you in coding in all manner of programming languages, and it's actually pre-installed on Raspbian PIXEL by default.

As well as allowing for the all-important highlighting of code depending on the language, Geany also offers the ability to test code in a very similar way to the Run function in IDLE. Different variables and functions can be jumped to from the left-hand column, allowing for quick browsing of the entire code.

It's a great piece of software that allows you to easily manage files and entire projects that use many types of code, and as it's already on Raspbian, you can test your code on it as well.



geany.org



ADAFRUIT WEBIDE

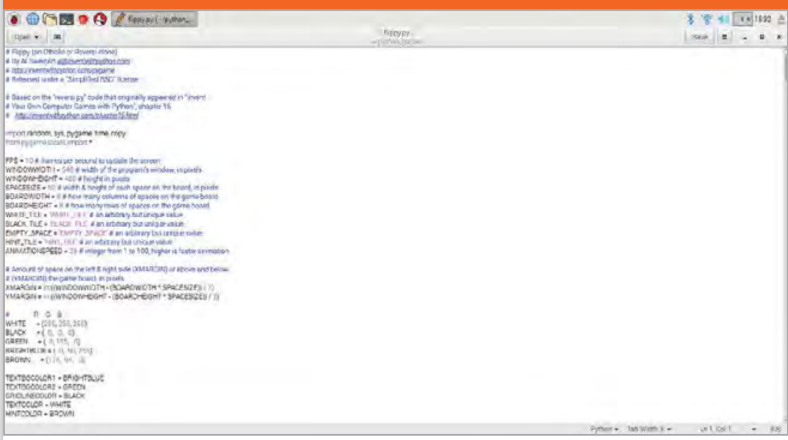
A quite different IDE, this one resides completely online. It's not like Scratch 2, though, where you work on software online. This is a hybrid system that you install on the Raspberry Pi and then access via a web browser on a networked computer.

The code and software on here is then stored in the cloud through Bitbucket, meaning you can access it from anywhere. Code can also be run directly to the Raspberry Pi via a Terminal so you can test it out as well. It allows you to use a whole host of different programming languages, and supports decent highlighting of them all.

It doesn't offer the same debugging and compiling suites of Geany or Code::Blocks, nor does it have a breakdown of functions and variables, instead favouring a project overview with your folders and files visible at all times. Still, it's a great idea, especially when using a headless Raspberry Pi.

magpi.cc/2nwsuod ★★★★☆

GEDIT



Running locally on the Raspberry Pi, gedit is the GNOME text editor. While there is already a text editor in Raspbian, gedit supports code highlighting for a variety of programming languages, which the default text editor does not do.

You'll need to install it from the package manager, and it does add a fair few other packages to get it running. The good thing is that as well as being in the program menus, you can access it via the Terminal if you want to edit a file in there. Instead of using something like **`nano python.py`** you can use **`gedit python.py`**.

It's quite simple, though, offering just a pure view of the code with some highlighting and no project view, debugging, or compiling options. It's pretty good if you need just a simple IDE that has few distractions.

gedit.org ★★★☆☆

AWARDS

WINNER
GEANY

RUNNER-UP
ADAFRUIT WEBIDE

DISTRACTION-FREE CODING AWARD
GEDIT

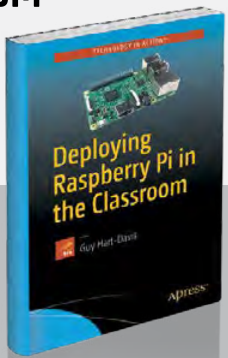
RASPBERRY PI BESTSELLERS

CLASSROOM CODING

The new computing curriculum is a big step for many teachers. These useful guides will ease the transition.

DEPLOYING RASPBERRY PI IN THE CLASSROOM

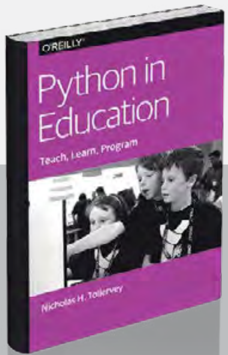
Author: Guy Hart-Davis
Publisher: Apress
Price: £19.99
ISBN: 978-1484223031
magpi.cc/2ndLCo8



A manual of everything a teacher, classroom assistant or school IT admin needs to deploy and run a network of Raspberry Pis in the classroom. Liberate your school from Excel boredom!

PYTHON IN EDUCATION
TEACH, LEARN, PROGRAM

Author: Nicholas Tollervey
Publisher: O'Reilly
Price: Free download
ISBN: N/A
magpi.cc/2ndCi3U



Thoughtful introduction to Python's strngths and suitability for education, its readability, simplicity, cross-platform, and the secret weapon – a wonderful community. Pass it on to a teacher you know.

HELLO WORLD

Authors: Various, contributing editor Miles Berry
Publisher: Raspberry Pi Foundation
Price: Free PDF download
ISBN: N/A
helloworld.cc

Written "by educators, for educators," a magazine from the Raspberry Pi Foundation that takes a cross-curricular look at tech, and educating young digital makers. A fascinating and diverse range of useful content.

THE HARDWARE HACKER

Author: Andrew Huang
Publisher: No Starch
Price: £23.99
ISBN: 978-1593277581
magpi.cc/2ndLt44

Part technology tour, part cultural travelogue, wholly engaging – the first section of the book takes the reader into the electronics markets of Shenzhen, and the inside secrets of the factories where most makers who scale up to full manufacturing will end up looking for serious production runs. Memorable statistics like "Foxconn workers consume 3,000 pigs a day," rub shoulders with fascinating asides on everything from the use of chopsticks in assembling Tux USB sticks, to the hidden problems of gloss-finish plastics.

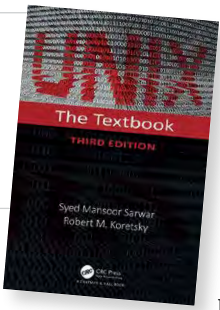
As if that were not enough, a short section on 'intellectual property' – a weasel term that Huang dismisses rapidly, before looking at shanzai innovations rarely seen here – leads on to Huang's investigations into fake SD cards. Part 3 starts with a brief, personal history of open hardware leading on to three open hardware platforms built by the author: chumby, the Novena laptop, and chibitronics. Each a fascinating technical and manufacturing story.

Rounding off this absorbing personal guide to the world of hardware is a 'hacker's perspective' section, with insights into reverse engineering, then bioinformatics – and some surprising comparisons between H1N1 and a computer virus. Lastly, extracts from interviews giving further insights into a remarkable career. Essential reading.

Score ★★★★★

UNIX: THE TEXTBOOK

Authors: Syed Mansoor Sarwar, Robert M Koretsky
Publisher: CRC
Price: £89.00
ISBN: 978-1482233582
magpi.cc/2ndNhtU



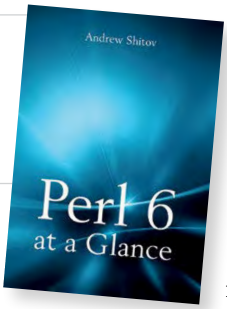
Raspbian is one of many GNU/Linux distributions that can run on the Pi, but GNU/Linux itself is one of more than 200 members of the UNIX family. An operating system fast approaching its half century, thanks to a number of sensible design decisions which, amongst other things, have permitted gradual evolution, while retaining a set of commands which would work on a release of any vintage.

Changes can be most clearly seen when a classic old manual like this is updated after more than a decade. Sarwar and Koretsky focus on Solaris and PC BSD, but the coverage here will be more than 90% relevant to your Pi, or the VPS that hosts your website. The telltale changes include a new, nearly 100-page chapter on Python, and coverage of Git and virtualisation.

You'll need a lot of shelf space for this 1300-page hardback, but it'll pay you back its price over many years if you're looking for a text book which combines introductory tutorials on everything from shell scripts and systems programming, to file encryption and remote logins. While not the only good, comprehensive 'Linux reference', its wider remit, and many exercises for the reader, make it one of the more educational.

Score ★★★★☆

PERL 6 AT A GLANCE



Author: Andrew Shitov
Publisher: Deep Text
Price: £26.00
ISBN: 978-9082156836
magpi.cc/2ndJnkS

As the dust settles on the long-awaited Perl 6, 2017 should see the publication of several – no doubt weighty – tomes on the language. First off the mark, with a nicely concise introduction (150 pages), Andrew Shitov's guide is based on a series of introductory articles originally written in Russian.

Perl 6 is not an incremental set of improvements over Perl 5. It's a different language, beautifully declarative, with useful improvements including: good concurrency support; a better, Smalltalk-influenced object model; lazy evaluation; Lisp-like macros; and gradual typing – giving you the ability, effectively, to 'turn on' static typing when needed. Floating point numbers can be stored as ratios for accuracy, and the 'approximately equal to' operator deals with minor imprecisions with e.g. complex numbers.

Perl 6 At a Glance lives up to its title, giving excellent language coverage with no padding or fluff. Short paragraphs and code snippets take you through everything from building a web server, to creating new operators. Promises, and Perl 6's Channel class, fit the language for the concurrent world, and Grammars build on Perl 5's powerful way with regular expressions. This book will open your eyes to Perl 6's capabilities and uses: it is as concise and expressive as the language.

Score ★★★★☆

THE PHOTOGRAPHER'S GUIDE TO DRONES

Author: Colin Smith
Publisher: Rocky Nook
Price: £26.99
ISBN: 978-1681981147
magpi.cc/2ndyFeh

Lots of makers have bought or built drones – they're cool and have plenty of potential applications – but most people aren't planning to set up delivery businesses with them. Photography from places you can't otherwise get to is a great example of technology assisting and enabling new creative endeavours. Indeed, the first few sumptuous pictures in Colin Smith's book are of coastal scenery and cities taken from several hundred yards out at sea, as are many of the spreads which open each chapter.

Despite this and the landscape format of the publication, it is more than a coffee table book. It is also a practical guide for any photographer wanting to take to the air without actually flying themselves. After suitable safety warnings – "spinning propellers and falling objects have the ability to injure people," and most countries have regulations that you need to be aware of (this book is written from a US point of view) – it's time to look at the gear, from the best available drones, to camera accessories.

Smith has plenty of sound advice on flying a drone, and using it for photography and shooting video. Workflow, post-production, advanced techniques (panoramic images, special effects), and editing video round off an interesting and helpful guide.

Score ★★★★☆

ESSENTIAL READING: MUSIC & AUDIO

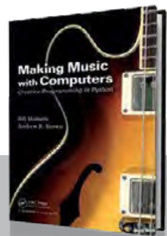
We love Sonic Pi (read our Essentials guide), but look what you can also do musically...

Mastering MuseScore

Author: Marc Sabatella
Publisher: MuseScore
Price: £25.99
ISBN: 978-1508621683
magpi.cc/2ndHZid

Make and play back beautiful scores, and join a community of users sharing their music. Superb tutorial and comprehensive reference.

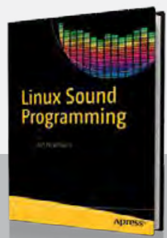
Making Music with Computers



Authors: Bill Manaris, Andrew R Brown
Publisher: CRC Press
Price: £33.99
ISBN: 978-1439867914
magpi.cc/2ndCBvf

Well-written, beginner-friendly guide to creating music programmatically – with algorithms and data – in Python (via Jython).

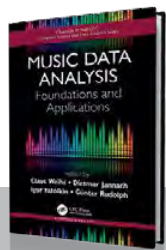
Linux Sound Programming



Author: Jan Newmarch
Publisher: Apress
Price: £27.99
ISBN: 978-1484224953
magpi.cc/2ndOOAr

The missing manual on programming all things audio on Linux, from sound fonts to controlling latency.

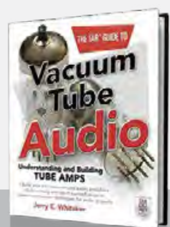
Music Data Analysis



Authors: Claus Weihs, Dietmar Jannach, Igor Vatolkin, Günter Rudolph
Publisher: CRC
Price: £44.99
ISBN: 9781498719568
magpi.cc/2ndoSF6

Nicely varied collection of academic essays, from instrument recognition through machine learning to automated composition.

The TAB Guide to Vacuum Tube Audio



Author: Jerry C Whitaker
Publisher: McGraw-Hill Education TAB
Price: £34.99
ISBN: 978-0071753210
magpi.cc/2ndAaZF

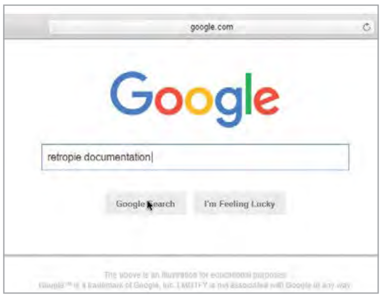
Get the most from your DAC HAT with a home-built valve (tube) amplifier. Fascinating and idiosyncratic reworking of classic circuits.

THE MONTH IN RASPBERRY PI

Everything else that happened this month in the world of Raspberry Pi

FOOL ME ONCE...

For better or worse, every year on 1 April people and companies come together to lie to – sorry, fool – their audience in the great tradition of April Fools' Day. The Raspberry Pi community is not one to pass up on such an opportunity and there were a few jokes going around on the day, centred around the humble Blueberry Delta.



RETROPIE FORUM

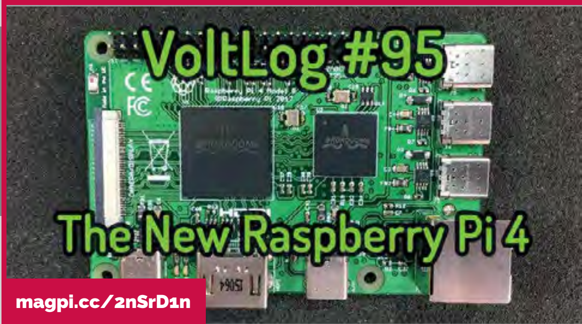
The RetroPie forum brought back a bit of a classic joke, redirecting everyone trying to get to the RetroPie forum for support to a LMGTFY (let me Google that for you) page, searching for 'retropie documentation'.

KODI SHUTDOWN

In a stealth move to upgrade the website and forums, Kodi pretended they'd been shutdown. Apparently a real threat they get regularly (people don't understand that Kodi doesn't provide content), it fooled many people until the site reverted.

APRIL FOUR-LS

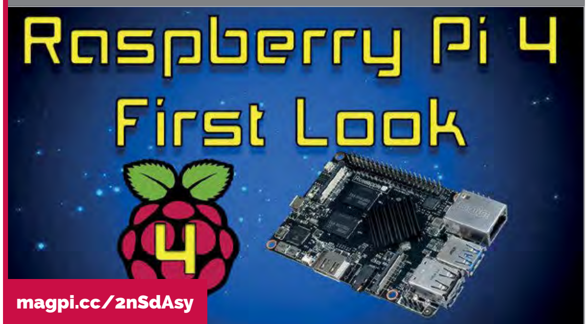
The Raspberry Pi 4 isn't out, but some YouTubers had their own fun with fake ones



magpi.cc/2nSrD1n

VOLTLOG

We're very, very impressed with the working PCB mock-up of an imaginary Raspberry Pi 4 design that was made for this video. It looks very real and even lights up! A lot of work went into making this board just for a little joke. There's a follow-up video showing how he made it, which is fascinating.



magpi.cc/2nSdAsy

ETA PRIME

We can't quite identify the board used in this video, although we'd like to point out the boot-up screen for Raspbian is now a bit different, which instantly gives it away. A great effort, though.

Here’s a Code Club that’s inspired us this month

CODE CLUB OF THE MONTH

FIRFIELD PRIMARY CODE CLUB

Operated by: Amanda Coffey
Occupation: Teacher and parent

Amanda Coffey was talking with another mum at her school when she heard about Code Club for the first time.

“It sounded really interesting,” Amanda tells us. “I am not a coder, but I had dabbled in the past. I remember typing code into my Spectrum when I was six or seven… but I had never been taught to code. Coding wasn’t a scary thing, it was just something I hadn’t learnt. I knew that most of my peers didn’t feel the same way: they were scared of coding. I wanted my children to grow up seeing coding as just another tool, not something scary.”

> “I wanted my children to grow up seeing coding as just another tool, not something scary”

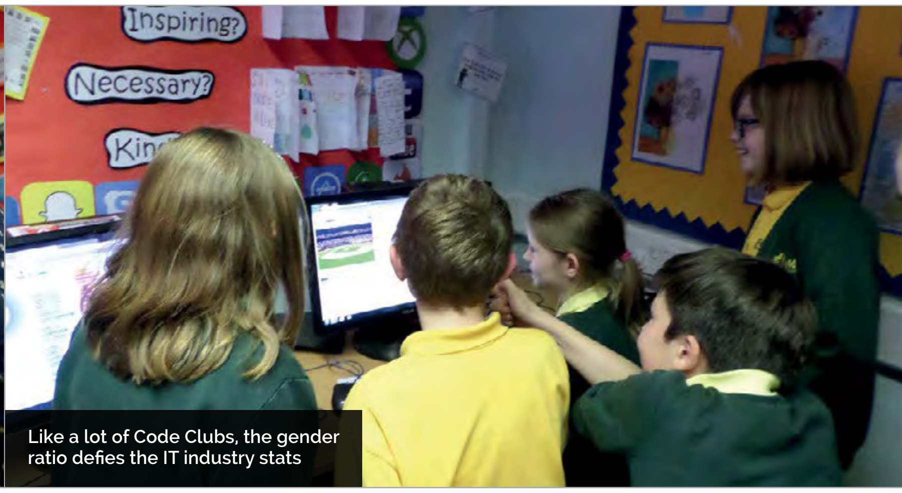
After learning about Code Clubs from fellow mum Jasjit, Amanda decided to start one with her running club coach, Ian. Their Code Club has been running for nearly two years now and is always oversubscribed with students wanting to learn how to code. It’s also resulted in some more formal lessons outside the Code Club to try to catch the kids who hadn’t been able to get in.

“I was standing in the playground when a child who had recently gone up to secondary school walked in to meet her family,” mentions Amanda. “She wasn’t a Code Clubber, but had been taught by us in the lessons we delivered. She saw me and ran over, so excited, and told me that in computing lessons at their new school they were coding in Scratch, that the teacher had told them what game to make but only given them a few blocks. She was quite smug when she said it was easy and that her game worked perfectly and was great.

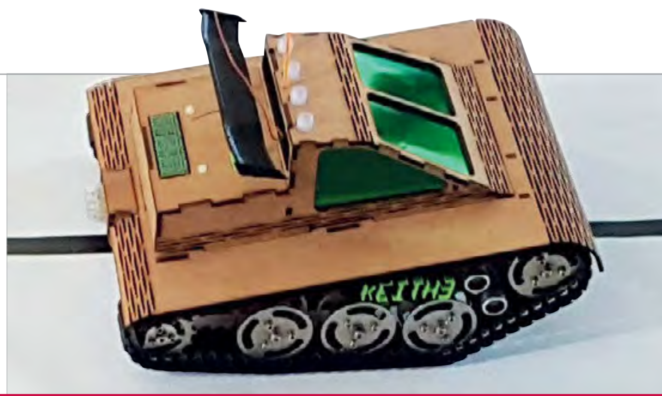
“I have seen first hand how even a small amount of coding knowledge can help in the workplace,” Amanda continues. “However, I also grew up with peers who thought it was worse than Latin, impossible to learn, too geeky, etc. This new generation will need coding to succeed, more than we ever have. If we can let them learn through playing whilst they are still young enough not to have the barriers society gives us in their heads, then they will get to secondary school and just see coding and computing as the tools they are.”

If you want to start your own CodeClub, you can find out more details at **codeclub.org**.

Amanda mostly teaches her students Scratch, but they do dabble in Python and HTML



Like a lot of Code Clubs, the gender ratio defies the IT industry stats



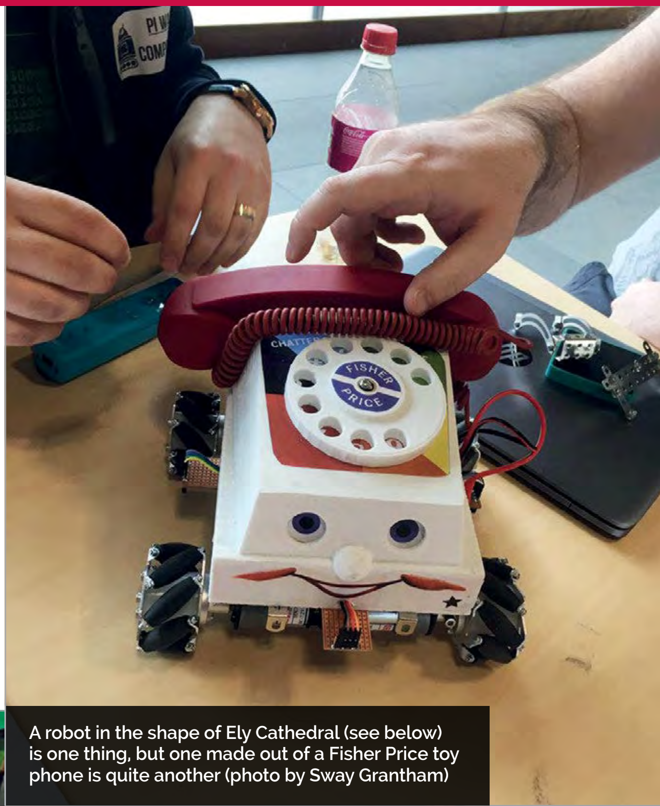
PI WARS 2017

The robot challenge finally returns

It's weird to think that since we last had a Pi Wars, the original *Robot Wars* TV show has seen a revival and is already onto its second season. While the robots in that show fight to the very end, the only combat the Pi Wars robots face involves popping balloons. Although that doesn't mean the robots don't break down during other challenges…

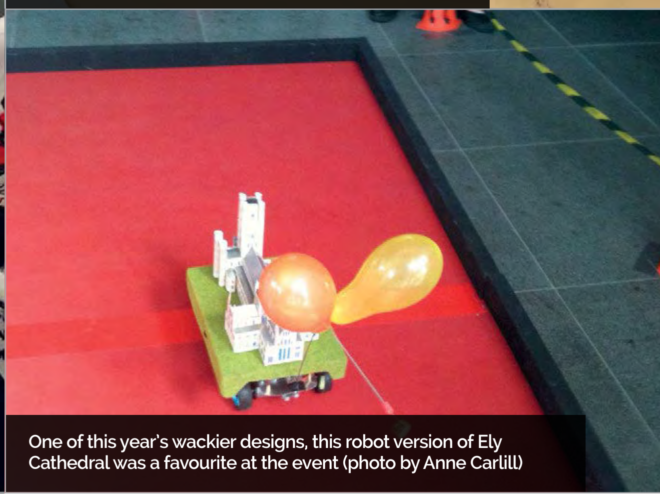
> "The event was a huge success and included special guest host Dr Lucy Rogers"

The event was a huge success and included special guest host Dr Lucy Rogers who, since appearing in the magazine last year talking about robot dinosaurs, has gone on to become a judge on the revived *Robot Wars*. Here are some pictures from Pi Wars 2017.



A robot in the shape of Ely Cathedral (see below) is one thing, but one made out of a Fisher Price toy phone is quite another (photo by Sway Grantham)

The Micro Pi Noon course for very small robots, built by Brian Corteil who was responsible for the robot build in *The MagPi* #51



One of this year's wackier designs, this robot version of Ely Cathedral was a favourite at the event (photo by Anne Carlill)

BEST OF THE REST

Here are some other great things we saw this month

magpi.cc/2nR03zS

MARIO KART CAR

This is utterly stupendous. Gordon Hlavenka hacked his actual car to control Mario Kart 64. If you watch the video, you can see him flash his lights as he jumps to drift around corners. He's even able to use the real accelerator and brake pedals. Hopefully it get disconnected when he goes for an actual drive.

magpi.cc/2nQVIg9

FALLOUT TERMINAL

We've had Pip-Boys from Fallout in the mag before, but we really love the design considerations put into building this Fallout terminal. It's 3D printable, so you can make your own if you want to. Just make sure to copy that retro green-and-black monitor style.

A VERY SECURE RASPBERRY PI

magpi.cc/2nQIUGc

Want to keep your Raspberry Pi as secure as possible? Then lock it inside a lock box with a digital security code. At least with this one you can turn it on and off via a switch on the outside of the box. Does that perhaps ruin some of the security measures?

RETROPIE ARCADE

magpi.cc/2nQUL7n

A good arcade build is always lovely to look at, and this wooden creation is no different. We especially like the glowing RetroPie text embedded into the top of the wood. This is designed to sit on a tabletop, but we'd love to see some more old-school full-size cabinet builds.

ZERO TERMINAL

magpi.cc/2nUJGUD

Using an old iPhone 5 slide-out keyboard, this maker has managed to create a hybrid Pi Zero W case that has a screen and a fully functioning keyboard input. It's one of the smallest mobile PC applications we've seen for the Raspberry Pi and it looks cool as heck.

magpi.cc/2nUQJN6

OCARINA CONTROLLED HOME AUTOMATION

This Raspberry Pi recognises five notes on an ocarina, so you can play the songs from the Legend of Zelda: Ocarina of Time to activate stuff around the house. It's so accurate that it lights specific LEDs for the specific notes being played. It comes with appropriate sound effects and recognises the best sound, Song of Storms, to turn on a humidifier.

COMMUNITY **PROFILE**

MATTHEW TIMMONS-BROWN

Boasting the most successful Raspberry Pi-related channel on YouTube, meet Matt, the electric skateboard-riding Raspberry Pi Guy

Matthew

Category: Advocate

Day job: Student and Raspberry Pi YouTuber

Web: theraspberrypiguy.com @RaspberryPiGuy1

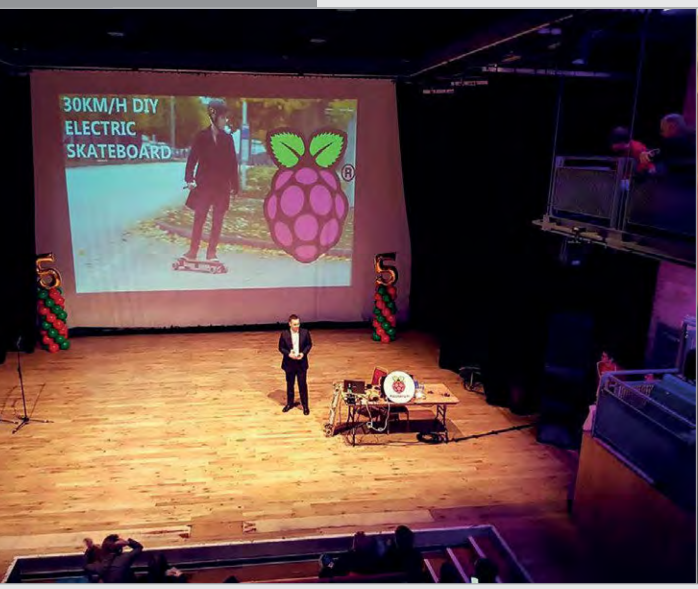
"I first set up my YouTube channel because I noticed a massive lack of video tutorials for the Raspberry Pi," explains Matthew Timmons-Brown, known to many as The Raspberry Pi Guy. At 17 years old, the Cambridge-based student has more than 60,000 subscribers to his channel, making his account the most successful Raspberry Pi-specific YouTube account to date.

If you've attended a Raspberry Pi event, there's a good chance you've already met Matt. And if not, you'll have no doubt come across one or more of his tutorials and builds online. On more than one occasion, his work has featured on the Raspberry Pi blog, with his yearly Raspberry Pi roundup videos being a staple of the birthday celebrations.

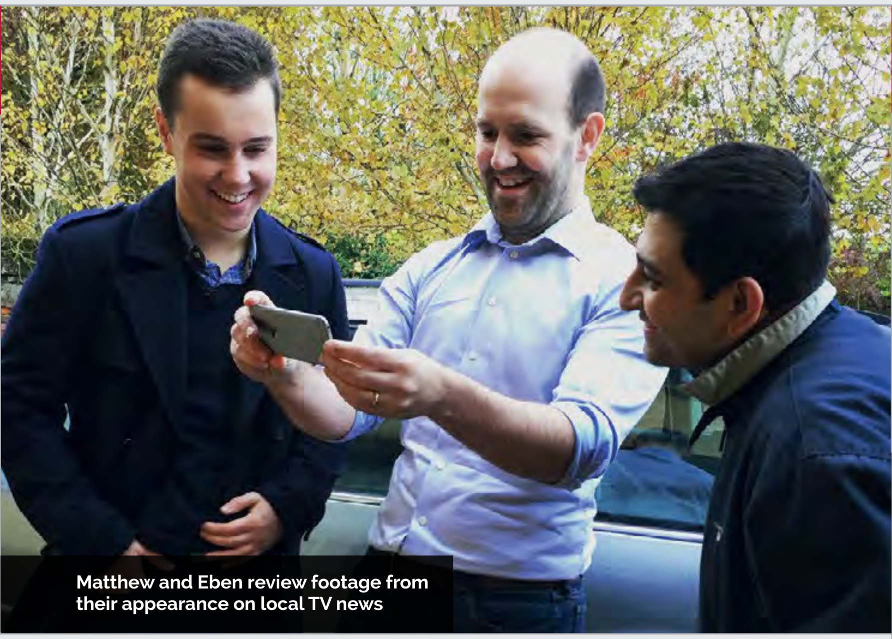
With his website, Matt aimed to collect together "the many strands of The Raspberry Pi Guy" into one, neat, cohesive resource, and it works. From newcomers to the credit card-sized computer to hardened Pi veterans, The Raspberry Pi Guy offers aid and inspiration for many. Looking for a review of the Raspberry Pi Zero? He's filmed one. Looking for a step-by-step guide to building a Pi-powered Amazon Alexa? No problem, there's one of those too.

Last summer, Matt introduced the world to his Raspberry Pi-controlled electric skateboard, soon finding himself plastered over local

Below From speaking at events to creating videos and resources, Matt Timmons-Brown continues to build upon a wonderful reputation in the Pi community as The Raspberry Pi Guy



Matt was invited to St James's Palace and the Houses of Parliament as part of the Raspberry Pi community celebrations in 2016


Matthew and Eben review footage from their appearance on local TV news

press, as well as the BBC and tech sites like Adafruit and **geek.com**. And there's no question as to why the build was so popular. With YouTubers such as Casey Neistat increasing the demand for electric skateboards on a near-daily basis, the call for a cheaper, home-brew version has quickly grown. Using a Raspberry Pi Zero, a Nintendo Wii Remote, and a little help from members of the Cambridge Makespace community, Matt built a board capable of reaching 30km/h, with a battery range of 10km per charge. Alongside Neistat, Matt attributes the project inspiration to Australian student Tim Maier, whose build we previously covered in *The MagPi*.

Despite the success and the fun of the electric skateboard (including convincing Raspberry Pi Trading CEO Eben Upton to have a go for local television news coverage), the project Matt is most proud of is his wireless LiDAR system for theoretical use on the Mars rovers. Built for his A Level computer science coursework, the build demonstrates Matt's passion for space and physics. Used as a means of surveying terrain, LiDAR uses laser light to measure distance, allowing users to create 3D-scanned, high-resolution maps of a specific area. It is a perfect technology for exploring unknown worlds.

In a recent interview at Hills Road Sixth Form College, where he is studying mathematics, further mathematics, physics, and computer science, Matt revealed where his love of electronics and computer science started. "I originally became interested in computer science in 2012, when I read a tiny magazine article about a computer that I would be able to buy with pocket money. This was a pretty exciting thing for a 12-year-old! Your own computer… for less than £30?!" He went on to explain how it became his mission to learn all he could on the subject and how, months later, his YouTube channel came to life, cementing him firmly into the Raspberry Pi community.

> "If you've attended a Raspberry Pi event, there's a good chance you've already met Matt"

HIGHLIGHTS

THE RASPBERRY PI GUY
theraspberrypiguy.com

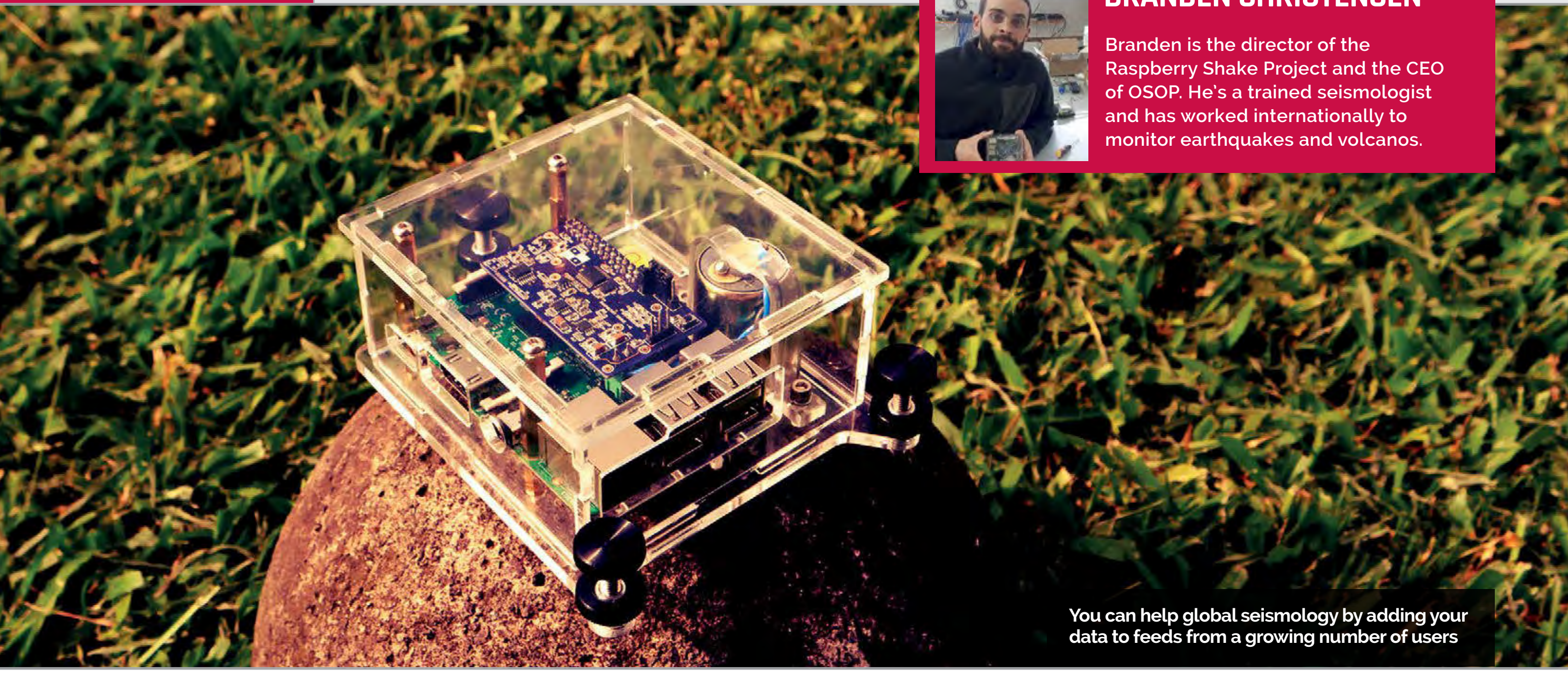
With more than 60,000 subscribers and nearly five million views, Matt's Raspberry Pi YouTube channel and accompanying website continue to provide makers and curious visitors with reviews, tutorials, and resources about the credit card-sized computer.

RASPBERRY PI ZERO SKATEBOARD
magpi.cc/2oyuWrU

While filming the video for his electric skateboard, Matt received more than one "Hey, Casey Neistat!" catcall as he tore through the Backs of Cambridge. Despite still being too young to officially join, Matt was able to call upon the help of local makespace members to complete the build.

LIDAR SYSTEM

Using a tablet app to define the angles, Matt's A Level coursework LiDAR build scans the surrounding area, returning the results to the touchscreen, where they can be manipulated by the user. With his passion for the cosmos and the International Space Station, it's no wonder that this is Matt's proudest build.



BRANDEN CHRISTENSEN

Branden is the director of the Raspberry Shake Project and the CEO of OSOP. He's a trained seismologist and has worked internationally to monitor earthquakes and volcanos.

You can help global seismology by adding your data to feeds from a growing number of users

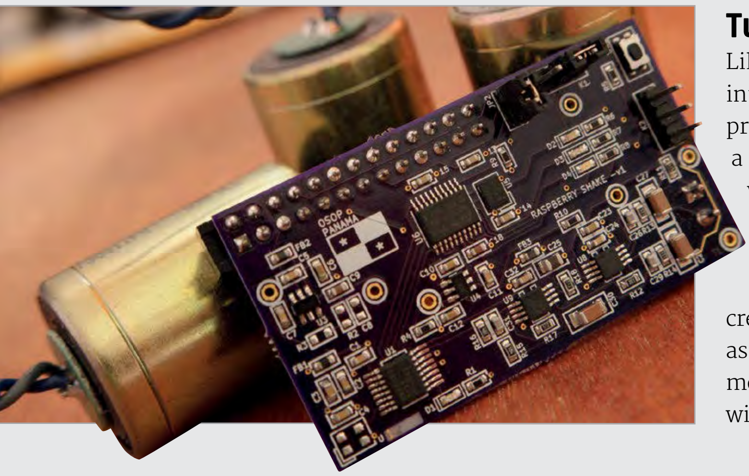
RASPBERRY **SHAKE**

OSOP brings seismography to everyone with the incredible Raspberry Shake kit

Seismic activity monitoring is important. It helps give people early warnings about possible natural disasters, as well as plotting how the earth beneath our feet is moving. It helps scientists to make better predictions and better models of our planet.

"Our company is called OSOP and it's existed for about ten years. We're dedicated to making high-density solutions for earthquake monitoring." Branden Christensen is the CEO of OSOP and he recently came to fame in the maker and Raspberry Pi communities when his company ran a very successful Kickstarter campaign for a consumer-grade seismograph.

"That's what we've been doing for years," Branden continues. "We had a product that cost an order of magnitude less money than anything else on the market and with that product we've helped countries through geophysical institutes. That solution, even though it was an order of magnitude cheaper than other things available on the market, was still $5,000 a unit."



Below The Raspberry Shake is very small. It only needs 26 GPIO pins, so it can work on any Pi model

Turning to crowdfunding

Like Raspberry Pi, they were interested in lowering the price even further and decided a network of hobbyist scientists with seismographs could have fun while still collecting useful data.

"We're always interested in creating as high-density a network as we can," explains Branden. "The motivations behind this are that with earthquake locations we always want to try and use automated processing systems, and those systems work best if you have a high density of stations. To have a really high level of confidence that seismic activity is of a certain magnitude in a certain place, you need to have high density. Raspberry Shake was born out of that philosophy, but it was also born out of the idea that we wanted to create a hobbyist seismograph. When we went into the Kickstarter campaign, we went into it thinking we just wanted to create a seismograph that will be cool for us and our friends. It was something we'd wanted to do as kind of a side project and we didn't know if there would be a market for it, so we thought why don't we throw it on Kickstarter and we'll have our answer.

"We thought if we can make the Kickstarter campaign successful and sell 20 or 30 of these units then we'll make it. We ended up hitting our funding level for 20 units in 18 hours. And then we went up to $50,000, and finally we almost made it up to $100,000."

The campaign raised $99,258 from 597 backers, and more people began pre-ordering after the funding period passed.

Upgrading the kit

"When we hit $50,000, we knew we had stumbled upon a new market," Branden tells us. "A market that hadn't been exploited before for doing citizen science seismology. So at that point we completely redesigned the product. We took it from being this gee-whizz seismograph to being something for geophysical institutes as well. We changed those specifications. We were able to get better prices on components so we ended up with a more robust solution that was competitive in the market."

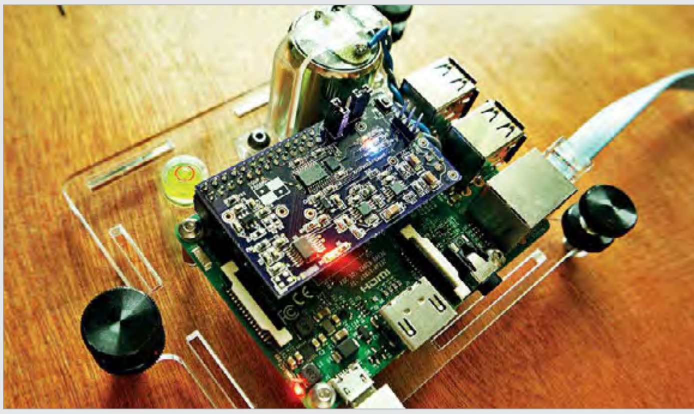
> We're always interested in creating as high-density a network as we can

Branden admitted that the Raspberry Pi wasn't the only small computer considered for what became the Raspberry Shake. However, to them it was the best choice for the project.

"The Raspberry Pi was central to the success of our campaign… We were sitting around a table and we thought, you know what, we're pretty sure that there are lots of geeks out there like us that have a bunch of Raspberry Pis just kicking around.

"So we said, let's do this differently. Let's create a product that will run on any Raspberry Pi, and let's make it so that people don't have to buy the Pi from us – let's make it like a kit." Of all the different options you can buy, the most successful sales we've had to date are not the turnkey solutions that include the Pi and everything and you just plug them in; it's the one that has everything but the Pi."

You can get yourself a Raspberry Shake, or find out more about it, from **raspberryshake.net**. Join the growing network of hobby seismologists!

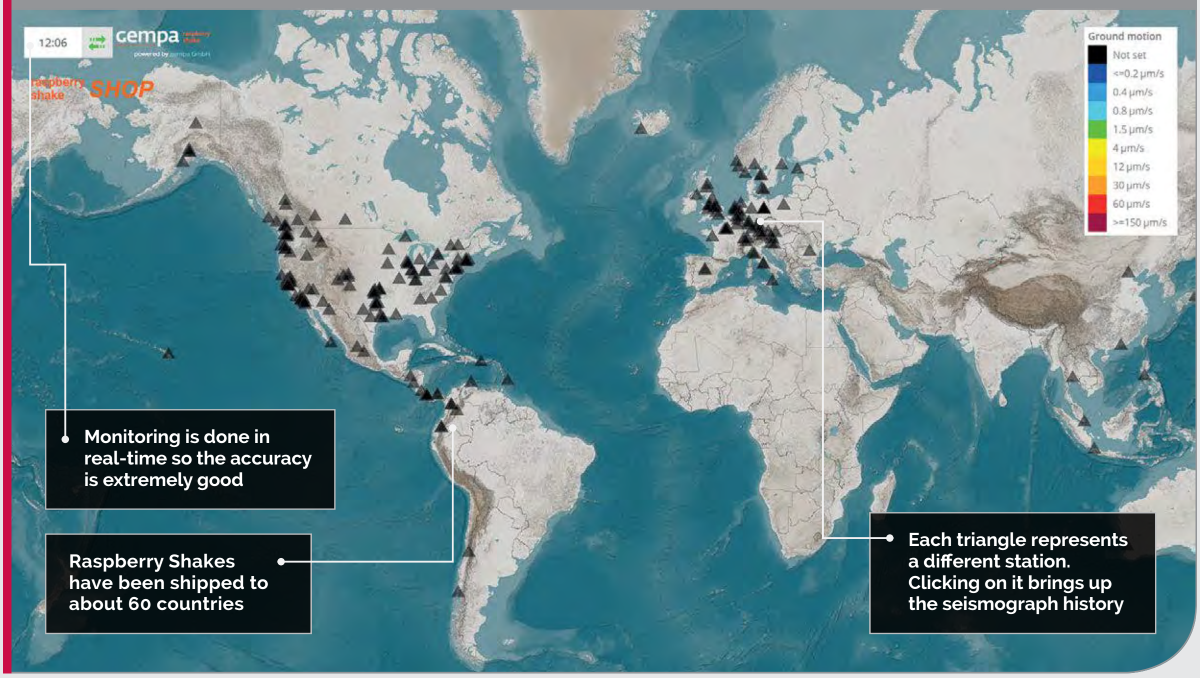


Above The assembled kit. It's very sensitive for such a low-cost device

STATIONVIEW

This is StationView: a readout of all the connected devices in the network. It shows the feedback from the high-density/high-fidelity system Branden is aiming to create.

"These are about 220 of the 700 or so units that we've shipped," Branden informed us, although the number has since gone up. "There are units coming online every day. Yesterday Aruba came online and Poland came online last week. Saudi Arabia came online but then disappeared. People at home plug their Raspberry Shake in, choose data sharing and boom, they automatically appear on the map."



RASPBERRY JAM EVENT CALENDAR

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

FIND OUT ABOUT JAMS

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

HIGHLIGHTED EVENTS

1 SKIPTON RASPBERRY JAM

When: Saturday 13 May
Where: Craven College, Skipton, UK
magpi.cc/2mY0syW
Interaction, lightning talks and project sharing with other Raspberry Pi enthusiasts.

2 ROANOKE RASPBERRY JAM

When: Saturday 20 May
Where: CoLab Roanoke, Roanoke, VA, USA
magpi.cc/2mXQZrB
Bring your favourite projects to share with other DIY makers. Attend a soldering workshop.

3 LEAMINGTON RASPBERRY JAM 2.0

When: Saturday 27 May
Where: Leamington Spa Library, Royal Leamington Spa, UK
magpi.cc/2mXSsOF
Create sweet music with Minecraft and Sonic Pi on the Raspberry Pi.

4 NORWICH RASPBERRY JAM

When: Saturday 27 May
Where: The Forum, Norwich, UK
magpi.cc/2oH1tQh
A separate part of the Norwich Gaming Festival focusing on learning with the Raspberry Pi.

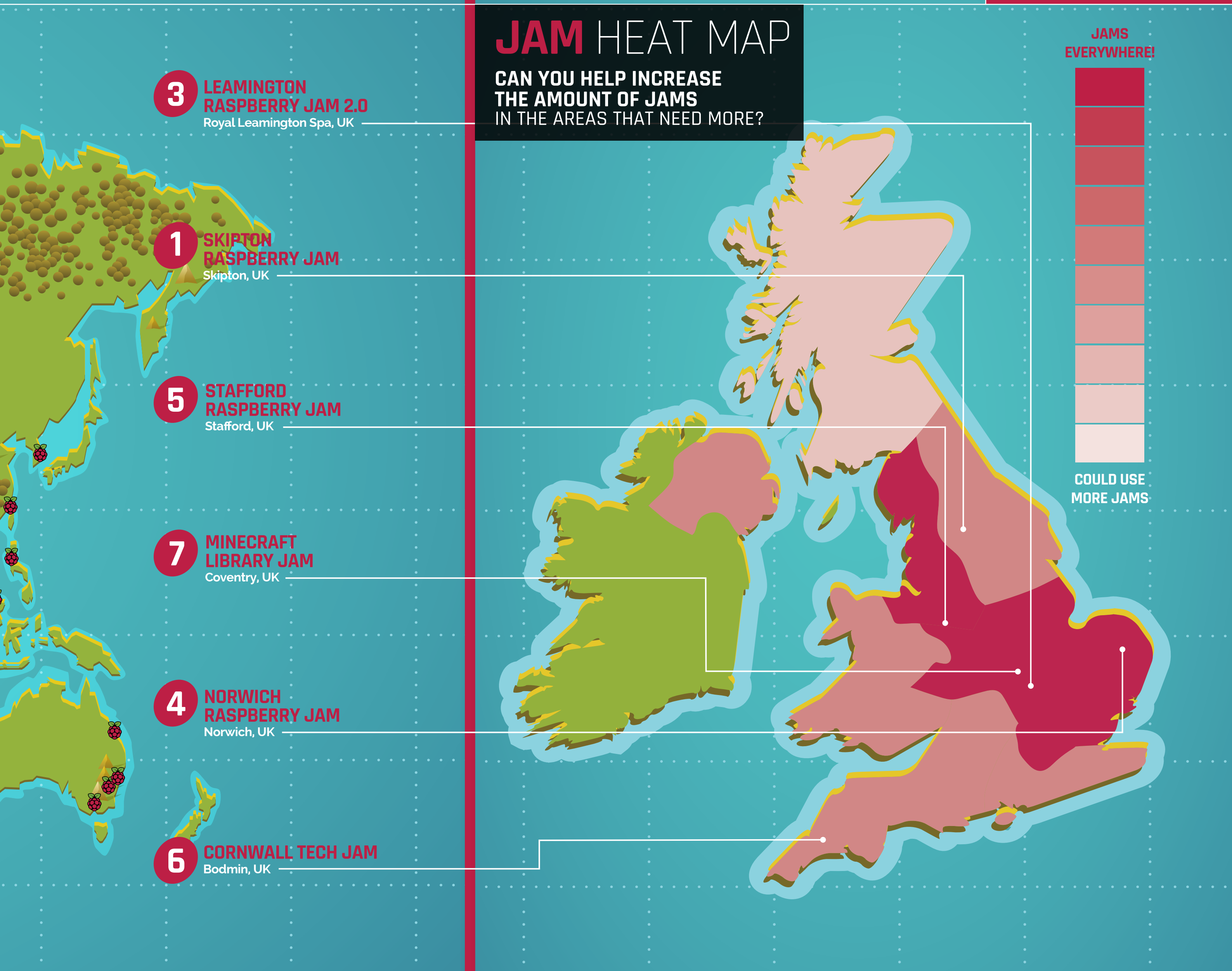
REGULAR EVENTS

5 STAFFORD RASPBERRY JAM

When: Tuesday 9 May
Where: Stafford Library, Stafford, UK
magpi.cc/2mvLiVJ
A big meetup of Pi enthusiasts where they all share ideas, help each other, and have fun!

6 CORNWALL TECH JAM

When: Saturday 13 May
Where: Cornwall College, Camborne, UK
cornwalltechjam.uk
For anyone interested in technology, of all ages and abilities. Ask questions and learn about programming: Scratch, Python, Minecraft, and much more.



7 MINECRAFT LIBRARY JAM

When: Saturday 13 May
Where: Central Library, Coventry, UK
magpi.cc/2mvCGhO
Build the world of Minecraft in code, then code your way around it on a Raspberry Pi.

8 RASPBERRY JAM BOGOTÁ

When: Saturday 20 May
Where: Biblioteca Pública Virgilio Barco, Bogotá, Colombia
magpi.cc/28Kavk3
The premier Latin American Raspberry Jam returns for another session with a full day of events introducing people to the Raspberry Pi.

MORE WAYS TO LEARN

HOW TO START A RASPBERRY JAM

As you may have seen in the news, there's a brand new kit and guidebook from the Raspberry Pi Foundation community team that gives top tips and advice on how to start your own Raspberry Jam. It also includes resources for making promotional material, along with some stickers to give out! You can find out more information about the kit here: **rpf.io/jam**

YOUR LETTERS

Winners galore

Well done to all the winners of Pioneers and the latest Astro Pi competition! It was great to see so many people do cool and varied things over the last couple of months. I look forward to seeing more Pioneers stuff and the results of this Astro Pi competition. Will you be covering them like you did the first Astro Pi? I did like reading along and seeing how they worked (and the images from space in that one project were very cool!).

Terry

There were some amazing entries, Terry, and you can see some of them in the news in this issue! We're absolutely going to run more coverage of Astro Pi as the second round ramps up and gets sent to space. We're also looking forward to seeing the results we'll get from the new experiments. Keep an eye out in future issues of *The MagPi* for more on Astro Pi as it unfolds.

Below A retro favourite. Will the Raspberry Pi be remembered so fondly many years later? We hope so!

Forever third

Congratulations on the sale of so many Raspberry Pis! Beating the Commodore 64 seems like no mean feat. How come you'll forever be third, though? Are the numbers of PCs and Macs really that insurmountable? Also what is the debate surrounding the numbers of the Commodore 64 sold?

Jonny S

Well, as explained in the article, there have been about 1.25 billion PCs sold, x86 IBM/Windows PCs to be exact, and 80 million Intel Macs as well. Sales of both aren't slowing down, so it's not like competing with the C64 where a set number of machines was sold, and that number will never change.

As for that C64 number, the controversy comes from various claims about how many have been sold. A book by an executive at Commodore claimed that a certain number sold consistently throughout its life, which adds up to a total of around 23 million. Reports by Commodore itself give a much lower number, and a count of all the serial numbers actually sold gets us to the 12.5 million number. This one is considered to be the most accurate.

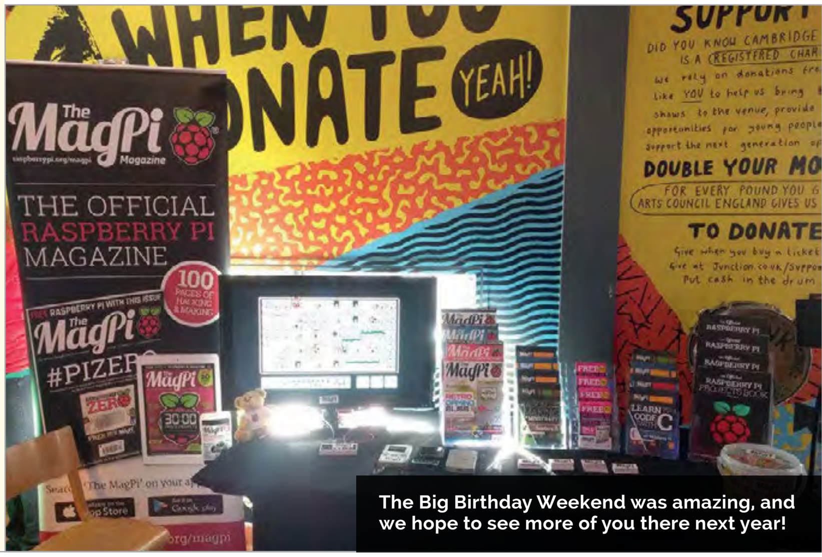
Party with Pi

I'm a big Raspberry Pi fan and I made my way over to Cambridge for the Raspberry Pi Birthday Weekend in March, and it was great! Unfortunately, I only got to go on the Saturday and missed the talk you did on the Sunday about writing for *The MagPi*! I have a few ideas that I'd love to write for the magazine. Is there a video online of the talk so I can learn how to send articles to you?

Lisa Nguyen

Shame you missed the talk, Lisa! We are possibly going to be posting our own version online (our excellent video guy filmed it). However, if you can't wait until then, a Raspberry Pi fan uploaded video of every talk to YouTube and you can see Rob give his talk there: **magpi.cc/2nn5tRj**.

We'll reiterate one of the key points about writing articles for *The MagPi*: just email Rob! You can get in touch with him directly at **rob.zwetsloot@raspberrypi.org** and he'll take it from there.



The Big Birthday Weekend was amazing, and we hope to see more of you there next year!

FROM THE FORUM: DIGITAL ERRATA

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community. Join in via **raspberrypi.org/forums**

I've just been having a read through issue 56, mainly the OS choice at the moment. Some OSs for me to have a look at, the camera one seems interesting.

Noticed one error, oddly for Raspbian. You say there is no graphical manager installing, but there is. However, I've just checked online and it's different. It also seems to have quite a few more OSs covered than my print version (unless I was sleeping?). It jumped from Raspbian to something like OSMC/RetroPie. Some three or four more pages? Is there a big error in the print version?
bensimmo

The error about the graphical package manager not being in Raspbian is our fault – we missed it when it went to print, but realised afterwards that it was wrong. As we hadn't actually launched the issue yet, we decided to do a little update to the PDF. We rarely do this with the magazine issues, and almost never after it's out, so we made the call to do it for the digital one.

As for the missing pages, they're in the print versions we've seen! Give the pages a little rub in case they need to unstick themselves.

Anyway, glad you found an OS or two to check out from the guide!

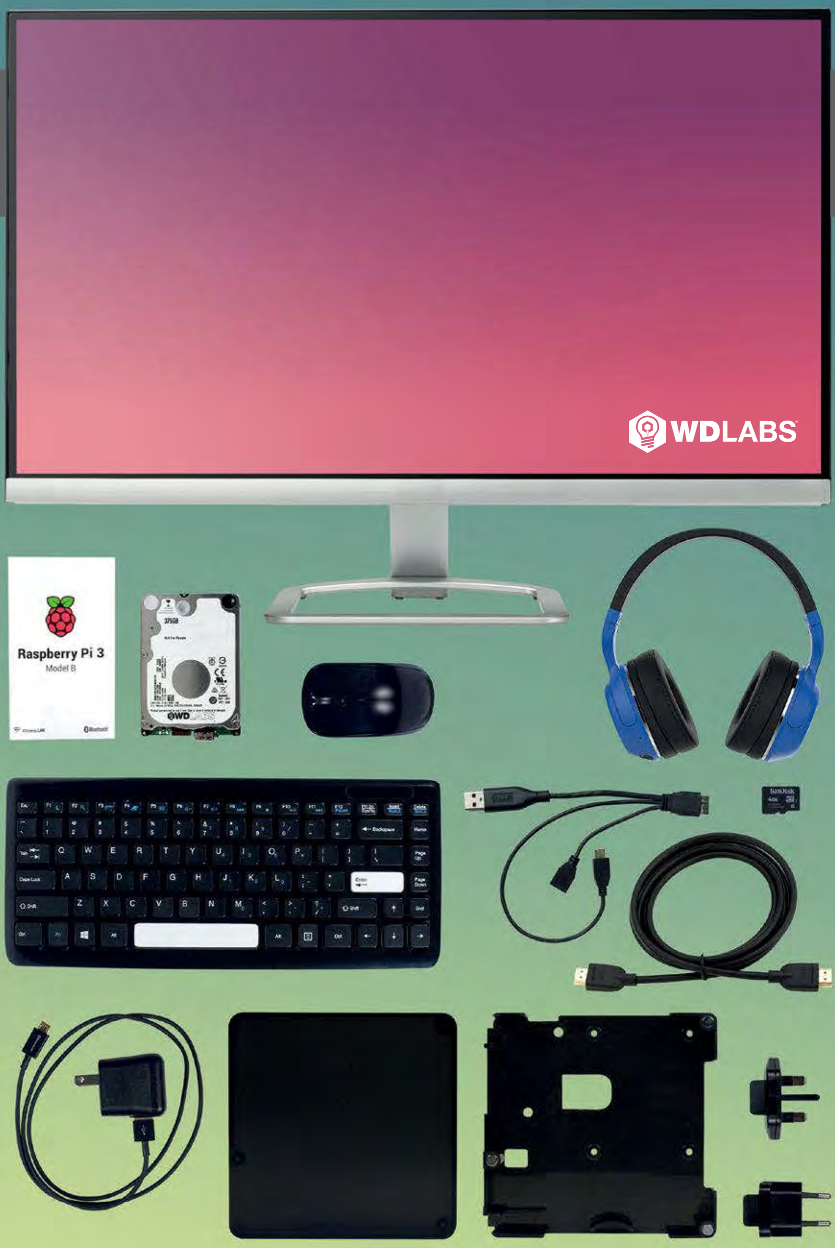
WRITE TO US

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**

$300+ OF WDLABS AND COMPUTER HARDWARE MUST BE WON!

In association with

WDLABS



WHAT WOULD YOU BUILD WITH THE **WD PIDRIVE COMPUTE CENTRE KIT?**

To enter visit:
wdlabs.io/mp57contest

Enter by 26 May 2017 for your chance to win!

Learn more:

Find out more about the WD PiDrive Compute Centre: **magpi.cc/ComputeCentre**

Terms & Conditions

Competition closes on 26 May 2017. Prize is offered in the UK and US 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 Richardson is the executive director of Raspberry Pi Foundation North America and author of *Getting Started with Raspberry Pi*. Contact him on Twitter **@MattRichardson**.

PRODUCT OR PROJECT?

Major product makers are opening their arms to hobbyists. **Matt Richardson** explores why

Taking inspiration from a widely known Mahatma Gandhi quotation, I like to tell people, "make the thing you wish to see in the world." In other words, you don't have to wait for a company to create the exact product you want. Prototyping with hardware has become easier and more affordable, empowering people to make products that suit their needs perfectly. And the people making these things aren't necessarily electrical engineers, computer scientists, or product designers. They're not even necessarily adults. They're often self-taught hobbyists who are empowered by maker-friendly technology.

It's a subject I've been very interested in, and I have written about it in this column before. Here's what I've noticed: the flow between maker project and consumer product moves in both directions. In other words, consumer products can start off as maker projects. Just take a look at the story behind many of the crowdfunded products on sites such as Kickstarter. Conversely, consumer products can evolve into maker products as well. The cover story for this issue of *The MagPi* is a perfect example of that. Google has given you the resources you need to build your own dedicated Google Assistant device. How cool is that?

But consumer products becoming hackable hardware isn't always an intentional move by the product's maker. In the 2000s, TiVo set-top DVRs were a hot product and their most enthusiastic fans figured out how to hack the product to customise it to meet their needs without any kind of support from TiVo.

Embracing change

But since then, things have changed. For example, when Microsoft's Kinect for the Xbox 360 was released in 2010, makers were immediately enticed by its capabilities. It not only acted as a camera, but it could also sense depth, a feature that would be useful for identifying the position of objects in a space. At first, there was no hacker support from Microsoft, so Adafruit Industries announced a $3,000 bounty to create open-source drivers so that anyone could access the features of Kinect for their own projects. Since then, Microsoft has embraced the use of Kinect for these purposes.

Consumer product companies even make versions of their products that are specifically meant for hacking, making, and learning. Belkin's WeMo home automation product line includes the WeMo Maker, a device that can act as a remote relay or sensor and hook into your home automation system. And iRobot offers Create 2, a hackable version of its Roomba floor-cleaning robot. It's aimed at STEM educators, but could be used for personal projects too. Electronic instrument maker Korg takes its maker-friendly approach to the next level by releasing the schematics for some of its analog synthesizer products.

Why would a company want to do this? I think there are a few possible reasons. For one, it's a way of encouraging consumers to create a community around a product. It could be a way for innovation with the product to continue, unchecked by the firm's own limits on resources. For certain, it's an awesome feel-good way for a company to empower their own users. Whatever the reason these products exist, it's the digital maker who comes out ahead. They have more affordable tools, materials, and resources to create their own customised products and possibly learn a thing or two along the way.

With maker-friendly, hackable products, being a creator and being a consumer aren't mutually exclusive. In fact, you're probably getting the best of both worlds: great products and great opportunities to make the thing you wish to see in the world.