

THE *Official*

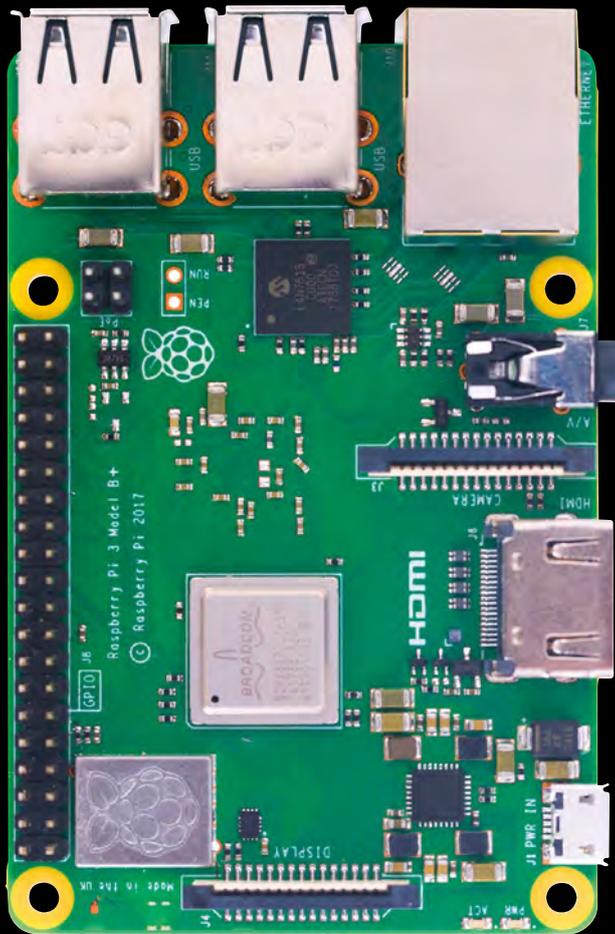
# RASPBERRY PI PROJECTS BOOK

VOLUME 4

200 PAGES OF  
IDEAS & INSPIRATION



DIY Games Console



Build a Robot



Set up a Spy Cam



Make a Magic Mirror

**55** PROJECTS & GUIDES

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

# SUBSCRIBE TODAY FROM ONLY £5

## SAVE UP TO 35%



**FREE**  
Raspberry  
Pi Zero W  
Starter Kit

WORTH £20

### Subscriber Benefits

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

### Rolling Monthly Subscription

- ▶ **Low monthly cost** (from £5)
- ▶ **Cancel at any time**
- ▶ **Free delivery to your door**
- ▶ **Available worldwide**

### Subscribe for 12 Months

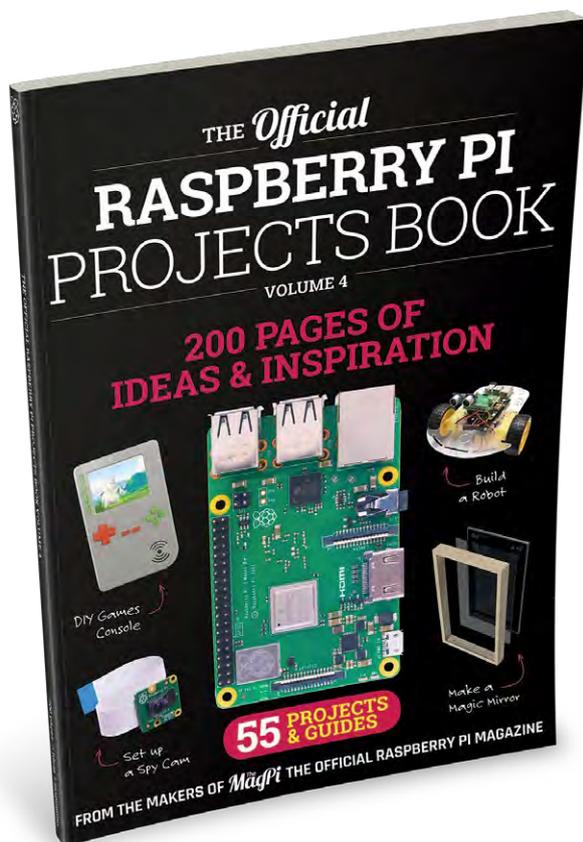
£55 (UK)    £90 (USA)  
£80 (EU)    £90 (Rest of World)

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

📞 Subscribe by phone: **01293 312193**

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

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



# WELCOME!

**E**very year it gets harder and harder to choose which projects to put into one of these Projects Books. With millions more Raspberry Pi boards out in the wild, that's millions more people getting into digital making and turning their dreams into a Pi-powered reality.

Being so spoilt for choice though means that we've managed to compile an incredible list of projects, guides, and reviews for you. I'm particularly proud that we managed to fit in our magic mirror big build so that you can make your own excellent piece of Pi furniture at home. The rest of this book's 200 pages are stuffed with articles that will help Pi newbies and digital making experts alike, so there should be plenty for everyone.

I hope you enjoy this book. Happy making!

**Rob Zwetsloot**

**FIND US ONLINE** [raspberrypi.org/magpi](http://raspberrypi.org/magpi)

**GET IN TOUCH** [magpi@raspberrypi.org](mailto:magpi@raspberrypi.org)

The **MagPi**

Available on the **App Store**

GET IT ON **Google Play**

**CC BY NC SA**

## EDITORIAL

Publishing Director: **Russell Barnes**  
Production Editors **Phil King, Rob Zwetsloot**

## DISTRIBUTION

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

## ONLINE STORE

[magpi.cc/store](http://magpi.cc/store)

## DESIGN

Critical Media: [criticalmedia.co.uk](http://criticalmedia.co.uk)  
Head of Design: **Dougal Matthews**  
Designers: **Lee Allen, Mike Kay**  
Illustrator: **Sam Alder**

## MAGAZINE SUBSCRIPTIONS

**Raspberry Pi Press**  
Mann Enterprises, Unit E, Brocks  
Business Centre, Haverhill, CB9 8QP  
[magpi.cc/subscribe](http://magpi.cc/subscribe)

## PUBLISHING

For advertising & licensing:  
[russell@raspberrypi.org](mailto:russell@raspberrypi.org)  
Comms Director: **Liz Upton**  
CEO: **Eben Upton**

## CONTRIBUTORS

**Bill Ballard, Johannes Bergs, Henry Budden, Sanjeet Chatterjee, John Cole, Mike Cook, David Crookes, Lucy Hattersley, Spencer Organ, KG Orphanides, Carrie Anne Philbin, Tim Richardson, Marc Scott, Lorraine Underwood, Clive Webster**



This bookazine 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.

This official product is published by Raspberry Pi (Trading) Ltd, Station Road, Cambridge, CB1 2JH. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISBN: 978-1-912047-70-3.

# Contents

## RASPBERRY PI FOR NEWBIES

Get to know the Raspberry Pi and everything the community has to offer

PAGE 06

### Projects

#### 20 PEGASUS LAND SPEED RECORD

Trying to break the land speed record with the help of a Raspberry Pi

#### 24 QBEE

This dress reacts to activity on social media by lighting up

#### 26 SELF-PLAYING ORGAN

A organ that plays itself thanks to a Raspberry Pi and some clever code

#### 28 MUSEUM IN A BOX

Bringing priceless museum artefacts to anywhere in the world

#### 32 ZERO PHONE

A Pi Zero-powered phone that you can make yourself

#### 34 TRACKING TELESCOPE

Make sure you keep an eye on the right part of the night sky

#### 36 PI MINI MINT

Squeezing a Raspberry Pi and screen into an Altoids mints tin is a bit tricky

#### 38 PI FILM CAPTURE

Digitally converting reel-to-reel film with an old projector and a Pi

#### 40 POLAPI-ZERO

A thermal camera project that uses the Pi Zero to take its photos

#### 42 WINDOWS 98 WATCH

Why have a fancy modern smartwatch when you can use a 20-year-old OS?

#### 44 PI ORDER

Improving the takeaway order system of a restaurant with Pi

#### 46 TOUGH PI-ANO

When your piano needs to take a beating, use arcade buttons for keys



#### 48 RASPBERRY TURK

Play chess against a (real) Raspberry Pi-powered robot

#### 50 DAILY PROPHET

A Harry Potter-inspired newspaper that plays video

#### 52 SATNAV ROBOT

Improve robot automation with GPS and a Raspberry Pi

#### 54 TELE2

Step into someone else's shoes thanks to this VR invention

#### 56 DOG TREAT MACHINE

Now you can give your dogs treats when you're not there

#### 58 PIOCKET TINY

A very small handheld games console that you can soon buy

#### 60 PHOTO BOOTH

Take better photos at your wedding reception with this homemade booth

#### 62 PI BASH

Roll up and test your strength on this classic carnival game

#### 64 12-FOOT GUITAR

This giant guitar actually works, thanks to a Raspberry Pi inside it

#### 66 GULLIVER'S GATE

Making a miniature of New York's Times Square is easier with a Pi

#### 68 GOOGLE PI INTERCOM

Upcycling an old intercom so that it becomes a personal assistant

#### 70 WAVES

Print out your own voice as a waveform on a piece of paper

#### 72 PI DECK

Scratching digital music is possible when you're using a Raspberry Pi

#### 74 ALONSO'S ROBOTS

A Disney Pixar animator makes cute, emotive robots in his spare time

#### 76 INSTAGIF

Print out GIFs using this camera and a special, tiny display

#### 78 MESMERIC MIRROR

A motion-sensing mirror that lights up as you walk past it

#### 80 FM TOUCH SYNTH

This touchscreen synthesizer is very powerful and customisable

#### 82 3D BODY SCANNER

Get a 3D scan of yourself to show off to your friends

#### 84 C-TURTLE

This Raspberry Pi robot looks for land-mines to help save lives

#### 86 FRED-209

A Nerf gun robot that gives you 20 seconds to comply

#### 88 RAPANUI

These T-shirts get printed thanks to a Raspberry Pi-based system

#### 90 NASA ARM

Even robots like to express themselves sometimes, this one with drawings

#### 92 TURNIN' TURBO DASHBOARD

Play arcade classic Out Run on this upcycled Tomy toy

### Tutorials

#### 96 MAKE A PI SPY CAM

This simple project lets you create a nanny cam in your own home

#### 98 NIGHT-VISION WILDLIFE TRAP

Create a camera that works in the dark to snap animal photos

#### 102 BUILD A WHOOPI CUSHION

This classic prank has been updated for modern times with a Pi

#### 106 SAVE YOUR WINE

A room temperature sensor lets you know if your wine cellar is OK

#### 108 GET UP WITH A SUNRISE ALARM

This sunrise alarm clock gets you up with the sun for a better morning



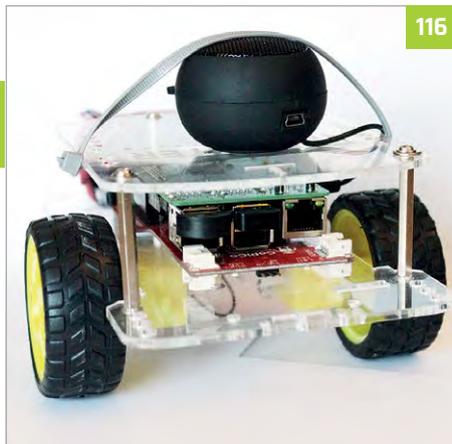
Reviews

- 172 RASPIO INSPIRING**  
Programmable LED shapes for impressive 2D and 3D displays
- 173 UNICORN HAT HD**  
256 LEDs that create a wonderful multicoloured matrix display
- 174 RAINBOW HAT**  
A multifunctional HAT that acts as an intro to Android Things on Pi
- 176 PI-TOP PULSE**  
A set of LEDs designed for pi-top laptops which are fully programmable

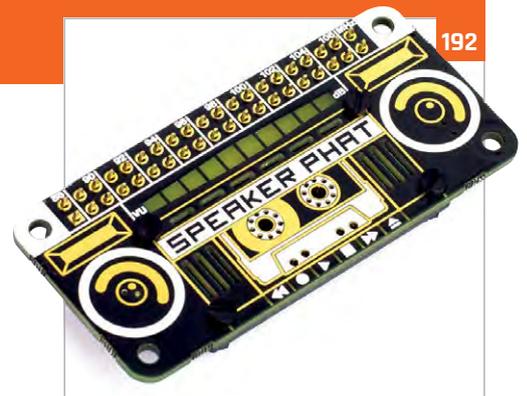


- 110 USING HUE LIGHTS WITH PI**  
Control your Philips Hue lights using this quick Raspberry Pi guide
- 112 TELL THE TEMPERATURE WITH STAIRS**  
This LED system shows you how hot it is as you go down the stairs
- 114 GIVE A ROBOT EMOTIONS**  
Not quite artificially intelligent, but it will recognise emotions and replicate them
- 116 ADD VOICE CONTROL TO A ROBOT**  
Use Alexa to add voice commands to operate a robot
- 120 BUILD A MAGIC MIRROR**  
Make the ultimate Raspberry Pi project with this thorough guide
- 132 TAKE DEEPCREAM PHOTOS**  
Use Google's DeepDream to make trippy photos out of what you see
- 134 MAKE ART WITH RECORD SCRATCHING**  
Beautiful art is at your fingertips – literally – with this DJ Hero mod
- 138 FIND FOLKS WITH MARAUDER'S CLOCK**  
Know where your loved ones are using this Harry Potter-inspired project
- 140 HACK AN AMAZON DASH BUTTON**  
You can use an Amazon Dash button to do a lot more than order loo rolls

- 177 MOTE PHAT**  
This mini HAT can control up to four programmable LED strips
- 178 PAPIRUS ZERO MEDIUM**  
A special E Ink display that fits neatly over a Pi Zero
- 179 PAN-TILT HAT**  
Perfect for robots, this allows you to have a moving camera
- 180 OLED BONNET**  
A tiny screen and buttons that might be handy as a mini menu display
- 181 TOUCH PHAT**  
Capacitive touch buttons that can be slotted above the Pi Zero
- 182 FLICK HAT**  
A HAT for adding motion-gesture controls to your Pi projects
- 183 INKY PHAT**  
This more advanced E Ink display has three colours

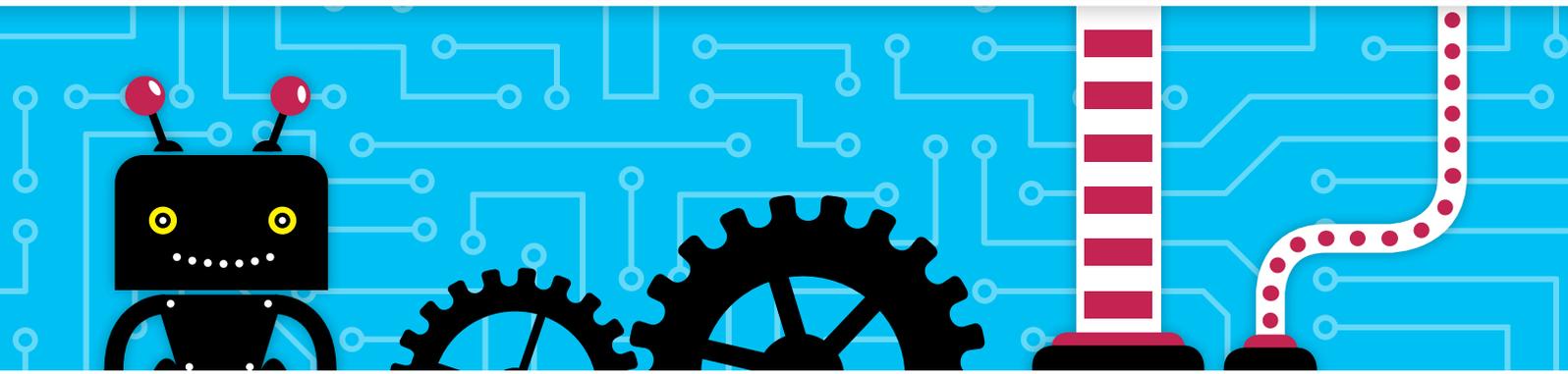


- 142 CREATE A CUT-OUT THEATRE**  
A digital cut-out theatre using JPGs and PNGs instead of paper
- 146 MAKE AN ULTRASONIC THEREMIN**  
Create space age music with this theremin using ultrasonic sensors
- 148 SAILING PI**  
Track your boating adventures with this GPS project
- 150 ADD SIRI VOICE CONTROL**  
Use Siri to add voice control options to any Pi project
- 152 AMIGA EMULATION GUIDE**  
Learn about emulating an Amiga the proper way on a Raspberry Pi
- 154 BUILD A RETRO GAMING HANDHELD**  
Make your very own Game Boy-like console with this big build



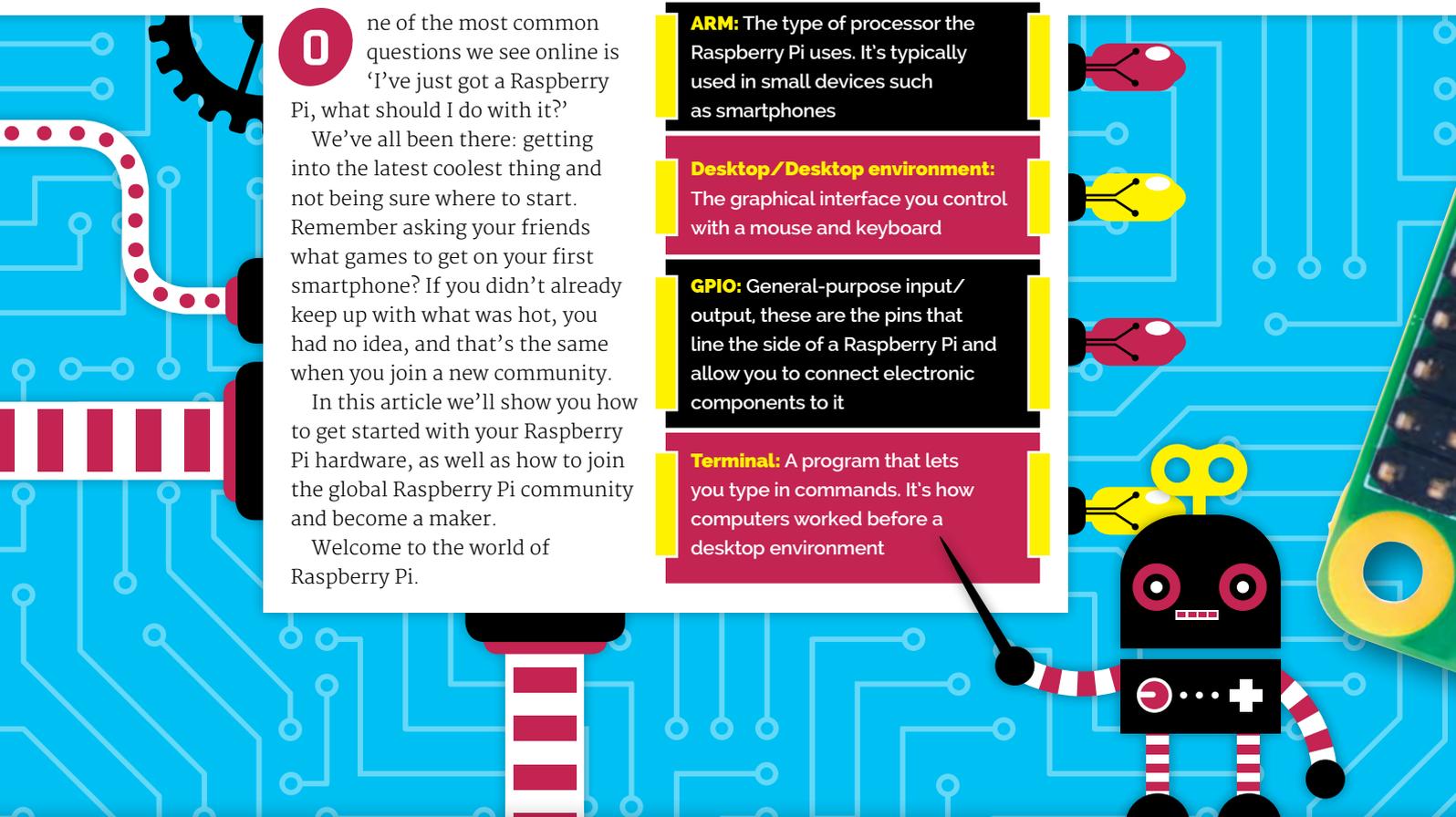
- 184 GOPIGO 3**  
A very simple robot kit for beginners, with plenty of learning resources
- 186 RASPIROVER KIT**  
This kit lets you create a powerful platform for robotics
- 188 TINY 4WD**  
A super-fast robot based on a feature written in The MagPi magazine
- 190 MEARM PI**  
Robot arm that's controlled with a Raspberry Pi
- 192 SPEAKER PHAT**  
Add music to your Pi project with this add-on that even has visualisations
- 193 JUSTBOOM DAC**  
A powerful DAC that can turn your humble Pi into a proper hi-fi
- 194 PIRATE RADIO**  
Create an internet radio with this fun Pi Zero project
- 196 JOY BONNET**  
This tiny controller fits onto the Pi Zero for compact gaming
- 198 PIJUICE**  
Keep your Raspberry Pi properly powered with this add-on
- 200 ARCADE CONTROLLER KIT**  
Make your own plug-and-play arcade controller with a Pi at its heart





# RASPBERRY PI FOR NEWBIES

Get to know your new Raspberry Pi computer and the community



**O**ne of the most common questions we see online is 'I've just got a Raspberry Pi, what should I do with it?'

We've all been there: getting into the latest coolest thing and not being sure where to start. Remember asking your friends what games to get on your first smartphone? If you didn't already keep up with what was hot, you had no idea, and that's the same when you join a new community.

In this article we'll show you how to get started with your Raspberry Pi hardware, as well as how to join the global Raspberry Pi community and become a maker.

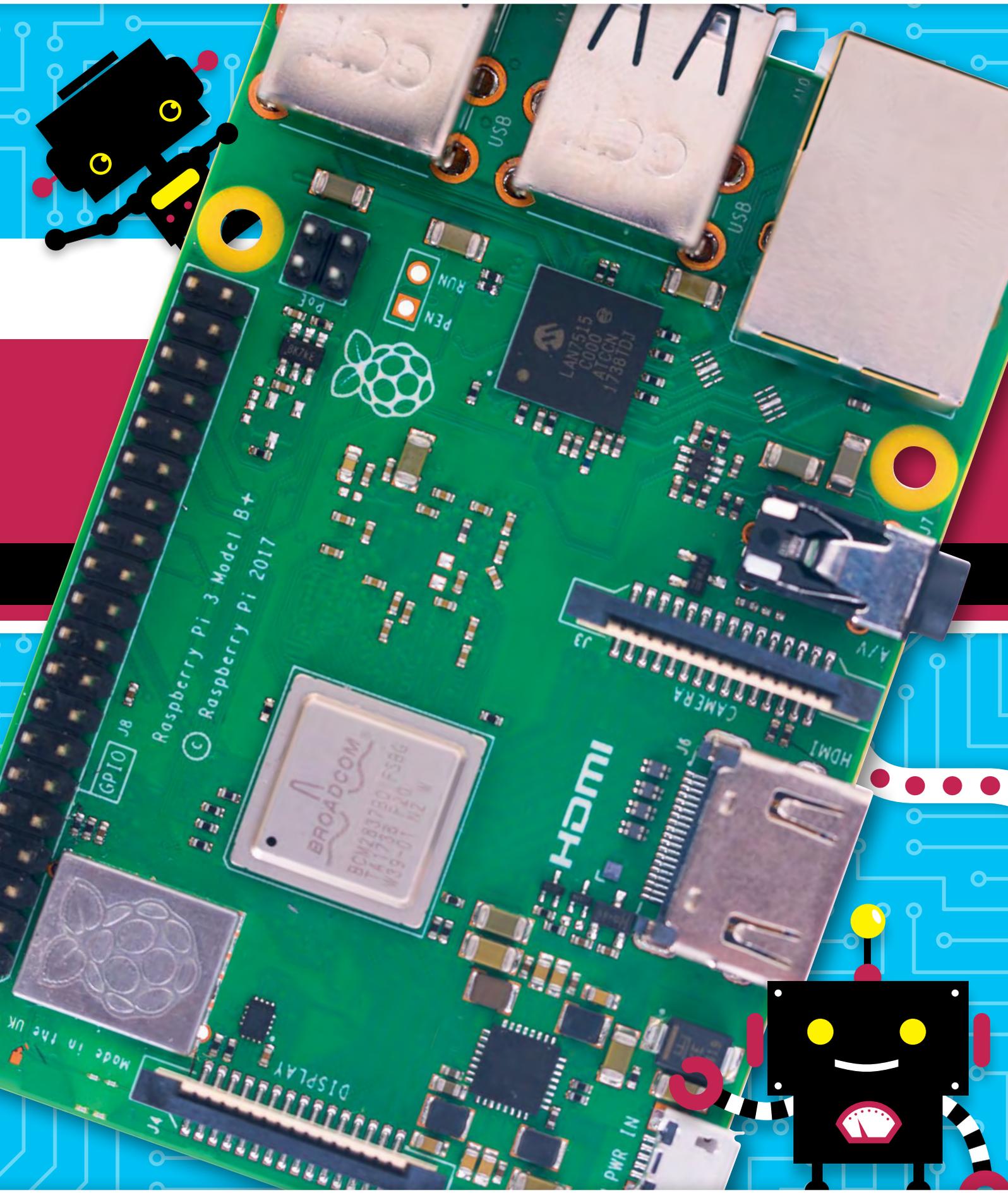
Welcome to the world of Raspberry Pi.

**ARM:** The type of processor the Raspberry Pi uses. It's typically used in small devices such as smartphones

**Desktop/Desktop environment:** The graphical interface you control with a mouse and keyboard

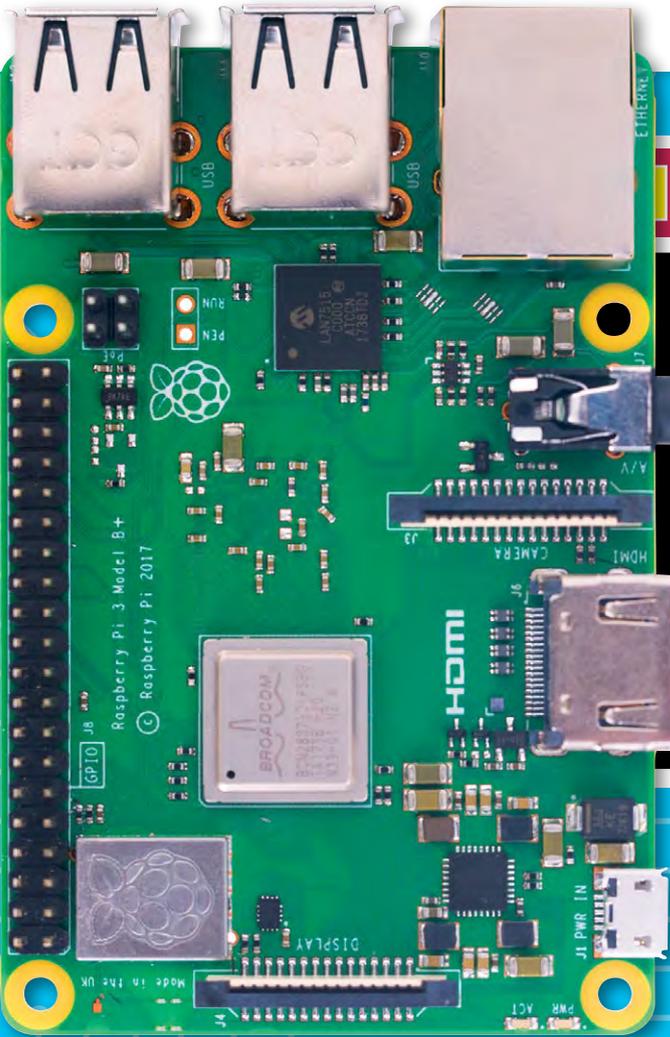
**GPIO:** General-purpose input/output, these are the pins that line the side of a Raspberry Pi and allow you to connect electronic components to it

**Terminal:** A program that lets you type in commands. It's how computers worked before a desktop environment



# GET TO **KNOW** YOUR PI

What actually is the Raspberry Pi?

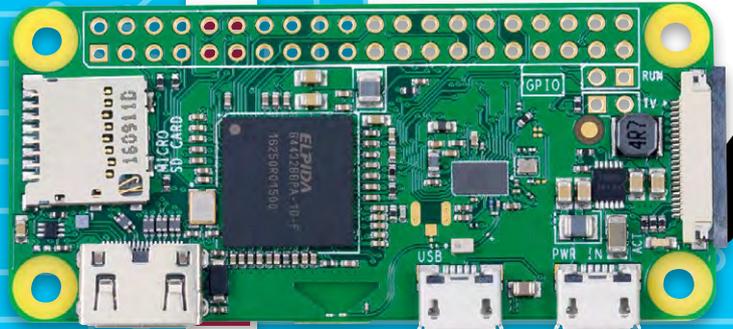


## RASPBERRY PI 3B+

- Processor:** 1.4GHz quad-core ARM processor
- Memory:** 1GB
- Networking:** Ethernet, wireless
- Connectivity:** HDMI out, analogue audio/video out (3.5mm headphone jack), Bluetooth 4.1, 4x USB 2.0 in, micro USB power, 40-pin GPIO, Camera Module port (CSI), Display Module port (DSI), microSD card slot
- Dimensions:** 86 x 56 x 19.5mm
- Weight:** 50g

## PI ZERO W

- Processor:** 1GHz single-core ARM processor
- Memory:** 512MB
- Networking:** Wireless
- Connectivity:** Mini HDMI out, Bluetooth 4.1, micro USB in, micro USB power, 40-pin GPIO, Camera Module port (CSI), microSD card slot
- Dimensions:** 65 x 30 x 5mm
- Weight:** 9g



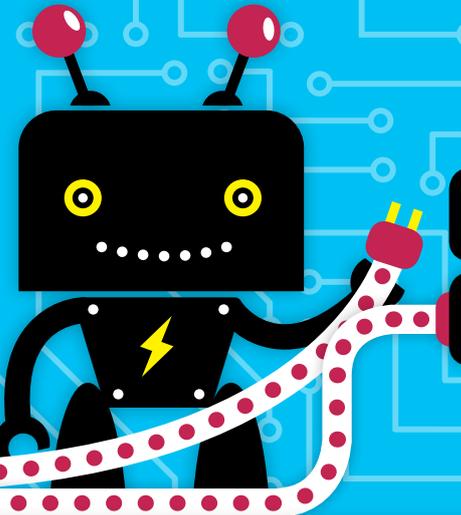
## WHAT CAN YOUR **RASPBERRY PI** DO?

The Raspberry Pi is a full computer. It has a processor, graphics processor, and memory, just like a normal computer or laptop – it's just a lot smaller.

The Raspberry Pi can be used in any situation a computer could be used. This can be as simple as being used as your desktop computer, or plugged into a

special laptop shell. Other people use them as mini servers in their house, as media computers for their TV, or as tiny computers to power their fun projects. It could be the brains of a robot, control a vegetable garden, or even just blink a light.

The possibilities with Raspberry Pi are limited only by your imagination.



# SET UP YOUR RASPBERRY PI

First time using a Raspberry Pi? Here's how to get it ready

## ESSENTIAL EQUIPMENT



MICRO SD CARD



MONITOR



KEYBOARD



MOUSE



HDMI CABLE



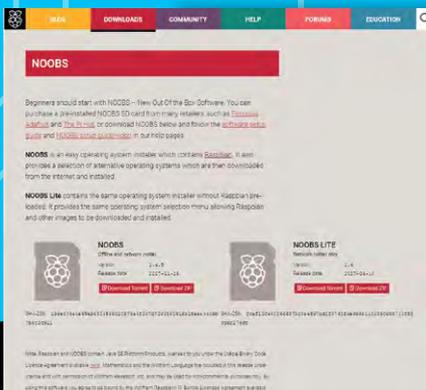
POWER SUPPLY (2.5A)



CASE

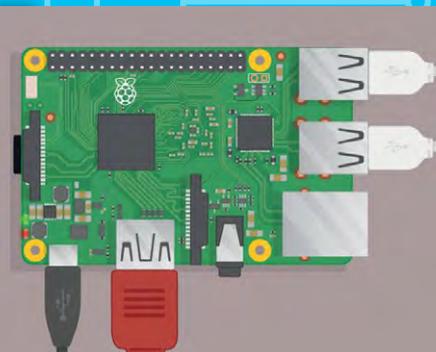
### >STEP 01 SET UP YOUR MICRO SD CARD

The first thing you need to do is make sure your microSD card has an operating system on it so that the Raspberry Pi can run. For newcomers, NOOBS is the recommended way of installing Raspbian, the Raspberry Pi operating system. Download it from [magpi.cc/2bnf5XF](http://magpi.cc/2bnf5XF) and then unzip the files directly onto a freshly formatted microSD card.



### >STEP 02 HOOK IT UP

Like any computer, you need to make sure everything is plugged in properly. Connect the monitor with the HDMI cable to the HDMI port on the Pi, firmly push the microSD card into the slot under the board, plug in your mouse and keyboard, and then finally insert the power so it turns on.



### >STEP 03 INSTALL RASPBIAN

NOOBS will give you the option to use one of many operating systems on your Raspberry Pi. Select 'Raspbian with Raspberry Pi Desktop' and it will boot into it. From here you can set up your wireless internet and any user preferences. It's also a good idea to update the files by opening the Terminal (via the little black box in the top bar) and typing the command:

```
sudo apt-get install update
```

...and then press **RETURN**, followed by the command:

```
sudo apt-get install upgrade
```



# OFFICIAL WAYS TO LEARN ABOUT RASPBERRY PI

Free resources from the Raspberry Pi Foundation to help you on your way

**W**hen most people think about the Raspberry Pi, they think of the computer. What a lot of people don't know is that the Raspberry Pi Foundation is a charity that also provides free resources for people wanting to learn about computing and making. This doesn't just apply to making stuff with the Raspberry Pi either.

## RASPBERRY PI RESOURCES

 [raspberrypi.org/resources](http://raspberrypi.org/resources)

The Raspberry Pi Foundation has a rich history of providing free resources for both makers and teachers alike, including offering free teacher training courses with Picademy ([rpf.io/picademy](http://rpf.io/picademy)). These range from simple programming lessons to full-on physical projects you can build in your own home. Here are some of our favourites.



### WHOOPI CUSHION

 [magpi.cc/2AgN6IW](http://magpi.cc/2AgN6IW)

This is an excellent project that melds physical making and computer programming. Also, it's an electronic whoopee cushion. This extremely fun project is pretty simple to make with just some foil, paper plates, and other bits you should be able to find around the house.



### SCRATCH OLYMPICS WEIGHTLIFTER

 [magpi.cc/2B9f0f1](http://magpi.cc/2B9f0f1)

There are plenty of Raspberry Pi resources that make use of Scratch, a beginner's programming language that makes use of blocks to create code. This weightlifter game uses fun retro sprites and a classic game mechanic that you learn to make yourself. You can then challenge your friends on it!

### GET STARTED WITH WEARABLES

 [magpi.cc/2B9o661](http://magpi.cc/2B9o661)

The Raspberry Pi's diminutive size means it's great for projects where space is at a premium. Wearables are a great example of this, and this excellent project teaches you how to upgrade your clothes with some Raspberry Pi magic and a few excellent programmable lights.



### BIG MINECRAFT PIANO

 [magpi.cc/2B9KGvY](http://magpi.cc/2B9KGvY)

There's a special version of Minecraft on Raspberry Pi that you can modify by programming in Python. This particular project shows you how to recreate the big piano from the movie *Big* in Minecraft, allowing Steve to jump around and make his own music.

# CODE CLUB RESOURCES

Code Club is a network of after-school and extracurricular computing clubs that provides free resources for kids aged 9–13 to learn how to code using Scratch and Python. There are thousands of clubs around the globe, so check to see if there’s one in your local area. You can even start a Code Club if you want, and there’s more info for that on the site. Here are some of our favourite projects.

[codeclub.org.uk](http://codeclub.org.uk)

## CHATBOT

[magpi.cc/EZAJwY](http://magpi.cc/EZAJwY)

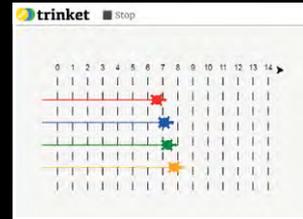
Want to talk to your computer? This fun Scratch project lets you create a ‘chatbot’, a program that tries to simulate the experience of talking to someone. Asking it specific questions will elicit specific responses.



## TURTLE RACE

[magpi.cc/LCUPKW](http://magpi.cc/LCUPKW)

This simple Python game has you racing four turtles against each other. The outcome is random each time, so you’ll be cheering for your chosen turtle throughout the entire race. This project also teaches some animation tricks with simple graphics.



## LIVE DJ

[magpi.cc/ViSgRb](http://magpi.cc/ViSgRb)

Sonic Pi is a cool program that lets you write music using code! There’s a series of Sonic Pi tutorials from Code Club which teach you the basics of creating your ‘loop’ for music before finishing off by teaching you how to put it all together in a live show.



# THE MAGPI RESOURCES

*The MagPi* is the official Raspberry Pi magazine, made for the community. As well as the regular monthly magazine, there is a selection of pocket books on single subjects in the Essentials range, as well as annual Projects books – such as the one you’re reading! If you can’t afford or find older issues, every issue of *The MagPi* is available as a free PDF online from the website. Here are some of our favourite releases for beginners – so far!

[magpi.cc](http://magpi.cc)

## SIMPLE ELECTRONICS WITH GPIO ZERO

[magpi.cc/2bA3ZP7](http://magpi.cc/2bA3ZP7)

Want to learn how to program physical objects on your Pi? The GPIO Zero library for programming language Python makes this very easy, and this book takes you from the basics all the way to controlling a robot with it.



## ISSUE 50: 50 GREATEST RASPBERRY PI PROJECTS

[magpi.cc/2dcswel](http://magpi.cc/2dcswel)

Want to be inspired by all the amazing stuff people have made in the community? We count down the 50 best projects as voted for by the community. We could easily do 50 more amazing projects as well – there are so many talented people in the community!



## ISSUE 53: BEGINNER’S GUIDE TO CODING

[magpi.cc/NfTNMt](http://magpi.cc/NfTNMt)

Discover the joy and art of computer programming with your Raspberry Pi. The beginner’s guide in this issue of the *The MagPi* helps you get started with coding using the popular Python language. Discover how to use loops, conditionals, variables, and more.



## ISSUE 64: ELECTRONICS STARTER GUIDE

[magpi.cc/lndLOj](http://magpi.cc/lndLOj)

In this issue of *The MagPi*, we show you how to read electronic circuits, and how to construct them so you can make your own amazing projects. Raspberry Pi makes it relatively easy, but you still need a stepping-off point.



# JOINING THE COMMUNITY

The Raspberry Pi community is huge and welcoming to everyone

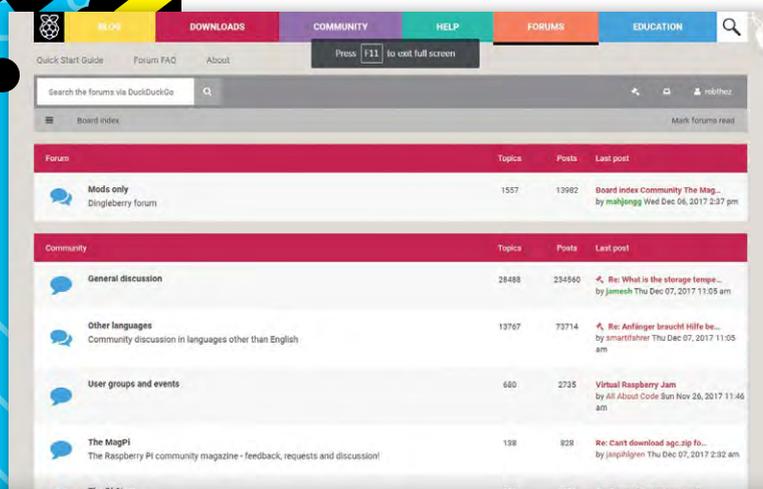
**W**e've talked about how the Raspberry Pi is both a computer and a charity, but it's also a huge community of like-minded makers and coders. Don't be intimidated, though! The community is very welcome to newcomers, as the Raspberry Pi is all about getting people excited to use computers. Here's how you can dip your toes into the community at large.

## THE OFFICIAL FORUMS

 [raspberrypi.org/forums](http://raspberrypi.org/forums)

The official Raspberry Pi forums are a thing of wonder. Hundreds of people come together to help people with their projects and problems – from simple things to incredibly advanced Linux tweaking. If you ever have an issue with the Raspberry Pi or Raspbian, head to the forums and use the search function to see if anyone else has ever had a similar problem and if not, start a new thread.

There are sub-forums dedicated to many specific parts of Raspberry Pi, from beginner's guides and troubleshooting to education chat and info on programming in specific languages.



## OTHER FORUMS

Some Raspberry Pi help and discussion can be found outside the official community forums, sometimes dedicated to more specific software or hardware. Here are some good places to take a look for more Raspberry Pi.

### RASPBERRY PI SUBREDDIT

 [magpi.cc/2AhIO42](http://magpi.cc/2AhIO42)

The Raspberry Pi subreddit is a great place to see some of the coolest projects from the community get highlighted, as well as get news from the community itself about products and updates! They'll even answer some Pi-related queries, although you should check out the official forums for that first.

It's also a great place to show off your project and get feedback for it from the community at large – it's always fun to have a little following, after all!



### RETROPIE FORUMS

 [magpi.cc/2B9laHb](http://magpi.cc/2B9laHb)

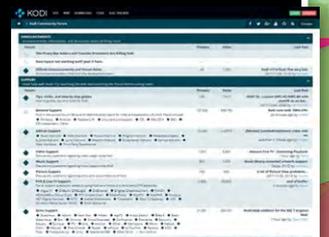
A lot of people like to use the Raspberry Pi to play retro games on it, and RetroPie is one of the premier bits of software for the Raspberry Pi that lets you do this. The forums cover almost everything about the project, including basic help and support, ideas for modding and improving your RetroPie, and even a bit of game discussion as well!



### KODI FORUMS

 [forum.kodi.tv](http://forum.kodi.tv)

Equally as popular as retro gaming, the Pi is great as a media PC hooked up to your living room TV and most home-theatre PC software solutions will use Kodi for this. While you may be better off looking at the documentation for your particular software, if that doesn't help then you can always check out the Kodi forum. It's a big and busy forum, though, so remember to make good use of the search function!



# CODE HELP

It's probably no surprise to hear that there are a lot of people in the world who code. Whether they're professional coders for big companies or teens playing with Python, they all have one thing in common: sometimes they might need a little bit of help. There are a couple of great places to head to when you find yourself up against a proverbial brick wall.



## GOOGLE

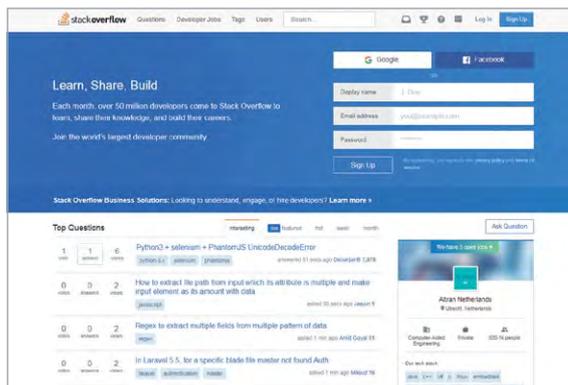
[google.com](http://google.com)

We're not being patronising here – a good Google search will genuinely help you out with many code issues. There's nothing new under the sun as they say, and there's usually someone who has had the exact same issue as you, or you might trip over the wording for what you want to do in a very helpful bit of official documentation. Always give any issue a quick Google and you may be surprised at how easy you can find an answer, and how simple the solution may be.

## STACK OVERFLOW

[stackoverflow.com](http://stackoverflow.com)

This website is one of the best places to ask about coding problems as not only is it incredibly popular, rewards are given to people who help out with an answer. We've seen people ask questions on a huge number of topics at varying skill levels, such as how to correctly call an item from a Python list or use complex database commands for a website. There's also a robust search feature included, and Google will generally point you towards pages on Stack Overflow if you've looked there for answers first.



## SOCIAL MEDIA

Here are some great social media accounts that you should follow



### TWITTER

- @Raspberry\_Pi**  
The official Raspberry Pi Twitter account
- @TheMagPi**  
The MagPi's official Twitter account

**@pimoroni**  
Maker of fun Pi projects and add-ons

**@FormulaPi\_**  
Raspberry Pi-powered racing robots



### FACEBOOK

**raspberrypi**  
The official Raspberry Pi Facebook account

**MagPiMagazine**  
The MagPi's official Facebook page

**RPISpy**  
Raspberry Pi Spy is a great way to keep up on Pi news and see awesome tutorials

**codeclubuk**  
The official Code Club UK Facebook account, keeping you up-to-date on all the cool things they do



# PROJECTS FROM THE COMMUNITY

Want to be inspired? Here are some amazing projects that we've seen from the community

**E**very day we keep an eye on Twitter and Reddit to see what amazing stuff people are coming up with to do with their Raspberry Pis. Here's just a taster of what we see online and what you can see in the magazine.

## AIY PROJECTS

### MARTIN MANDER'S INTERCOM PROJECT IS RETRO AND LOVELY

[magpi.cc/2vPQK6E](https://magpi.cc/2vPQK6E)

An amazing project that uses the Google AIY Projects Voice Kit to hack an old-school intercom to become a digital voice assistant. It's an ingenious bit of repurposing, and we always love seeing what Martin Mander is going to upcycle next. For some reason, though, everyone else has been having the same idea and we saw loads of toy phones and intercoms and other voice-powered products upgraded with an AIY Voice Kit. Here are a couple of our favourites...



### OPERATOR!

[magpi.cc/2vN4rEo](https://magpi.cc/2vN4rEo)

More retro upcycling with this old-school phone upgraded with an AIY Voice Kit. Just dial 0 and you can talk to the assistant to get answers to your everyday queries. We have visions of someone just picking it up and putting it under their chin to ask for ounce-to-gram conversions while they mix some batter, Nineties style.



### KIDS SMARTPHONE

[magpi.cc/2vMuSty](https://magpi.cc/2vMuSty)

This one is a little mean but honestly, we laughed. The Fisher-Price toy phone is a true classic and we love the idea of it getting a very hefty upgrade with AIY, complete with massive glowing button in the centre. Would it be creepy if you had it sing while the eyes moved? Very probably.



# WEEKEND PROJECTS

# HALLOWEEN PROJECTS

**CIRCUITBEARD**  
@circuitbeard

Follow

Blogged: Tomy Turnin' Turbo Dashboard Outrun Arcade - The build log for my mini #outrun arcade  
circuitbeard.co.uk/2017/08/28/tom ...  
#raspberrypi



## OUT RUN TOMY TOY

[magpi.cc/2xckwUo](http://magpi.cc/2xckwUo)

While we have seen many console or arcade hacks over the past few years, this inventive hack of one of those sports car dashboard kids toys wowed us when we saw it. It doesn't hurt that it's themed around Out Run, one of the best arcade series of all time.

## FRED-209

[magpi.cc/Pkvvuc](http://magpi.cc/Pkvvuc)

A custom Nerf-firing robo-tank that gives you only 20 seconds to comply. David Pride 3D-printed lots of the parts to make it fire darts from a Nerf gun magazine, tilting the mechanism to aim. Follow the link to learn more and see it in action!

**David Pride**  
@davejavpride

Following

@Raspberry\_Pi @TheMagP1 Meet FRED-209. A #RaspberryPi powered, #3DPrinted Nerf Tank! Details and video here: [bit.ly/2hc9Y03](http://bit.ly/2hc9Y03)




## POSSESSED PORTRAIT

[magpi.cc/2yMfQrI](http://magpi.cc/2yMfQrI)

The picture for this project doesn't really do it justice, so take a quick look at the video: [magpi.cc/2yMgAgu](http://magpi.cc/2yMgAgu). Needless to say, it's a very effective and scary project that uses a little illusion and a motion sensor to make you think the painting is moving. And attacking.

## HAUNTED JACK IN THE BOX

## HAUNTED JACK-IN-THE-BOX

[magpi.cc/2yMcoGZ](http://magpi.cc/2yMcoGZ)

This automated jack-in-the-box uses a camera to detect if someone is around. If you turn up in front of it, surprise! Pop goes the weasel and also about three years off your life. Put it in an inconspicuous part of your house to scare the bejesus out of friends and children.



## THE POPLAWSKIS' HOLIDAY FRIGHTS

[magpi.cc/2yMfIse](http://magpi.cc/2yMfIse)

The theme changes depending on the season, but the original setup allowed you to control Halloween decorations on the Poplawkis' lawn. There's a camera recording the whole thing, and you can control various decorations – with the option to keep your selection active for one minute by spending a credit (costing 10 cents).

# COMMUNITY EVENTS

Meet Raspberry Pi users in real life and see talks by computing boffins that might help you

**T**he Raspberry Pi community online is so big that it regularly leaks into the real world, usually through the power of a Raspberry Jam. However, the Pi makes itself into many other events that involve technology, such as Maker Faires.

## CODERDOJOS

CoderDojo is part of the Raspberry Pi Foundation and, much like Code Club, provides free resources for people wanting to create programming clubs for kids. There are also CoderDojo events such as the Coolest Projects showcase, allowing young people to show off the awesome stuff they've been making.

 [coderdojo.com](http://coderdojo.com)

Pi-monitored bees make them a bit safer to keep

You can find robots galore at the Coolest Projects showcase

Glowing ping-pong balls are a favourite for LED projects

## RASPBERRY JAMS

A Raspberry Jam can be a lot of things, but usually it's a social event people can attend to learn about the Raspberry Pi. Typically you'll see people showing off their projects, as well as stalls where you can buy Raspberry Pi add-ons and electronics kits.

These Jams are hosted by community members; in fact, anyone can host a Jam if they wish! Head over to the info page to find out more on Raspberry Jams, including an events calendar so you can try to attend one!

 [rpf.io/jam](http://rpf.io/jam)

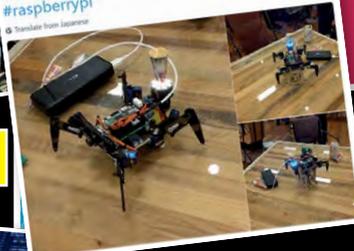
The Raspberry Pi birthday parties are huge, weekend-long Raspberry Jams!

The yearly robot challenge Pi Wars spun off from the popular CamJam



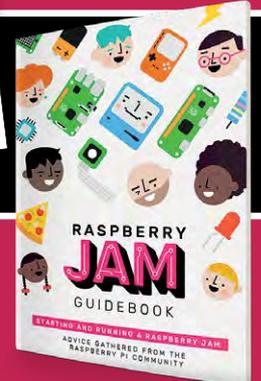
Be sure to check out the #raspberrypijam hashtag on Twitter to see what awesome things people are showing off

鈴木光宗 @s\_mitsu  
カメラで顔を認識すると、クラッカーを鳴らすロボ  
#raspberrypijam  
#raspberrypi  
@ Translate from Japanese



### GET INVOLVED

Get all the info you need to set up your own Raspberry Jam: [magpi.cc/2q9DHfQ](http://magpi.cc/2q9DHfQ)



# RASPBERRY PI RETAILERS

Want to buy more Pi goodies? Take a look at these places...

## PIMORONI

[shop.pimoroni.com](http://shop.pimoroni.com)

Purveyor of Raspberry Pi goods and general maker ware, Pimoroni is one of our favourite places for fun and beginner-level projects, while also offering plenty of components for almost any project you can think of making.



## ADAFRUIT

[adafruit.com](http://adafruit.com)

Adafruit not only creates incredible electronics for makers, but also posts amazing tutorials that make use of some of its electronics. There's plenty of Pi-compatible stuff and it's perfect for getting Pi paraphernalia if you live in the US.



## MODMYPI

[modmypi.com](http://modmypi.com)

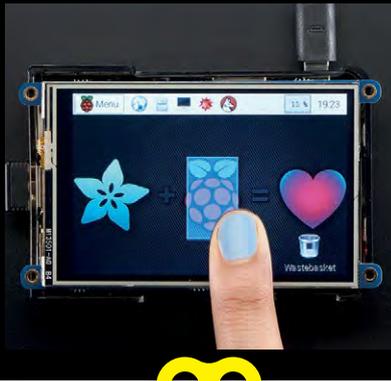
While it does carry a lot of Raspberry Pi-specific hardware and add-ons, we love how much maker gear you can also get at ModMyPi, including obscure components you might have trouble finding elsewhere.



## THE PI HUT

[thepihut.com](http://thepihut.com)

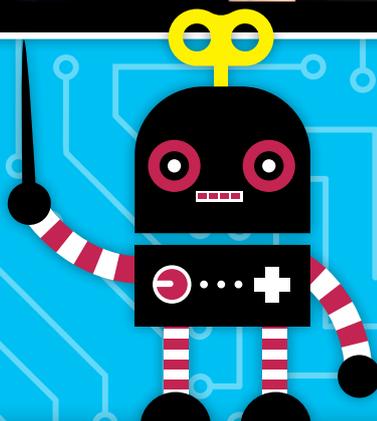
The Pi Hut sells a lot of Pi add-ons and gear, along with robot parts from the excellent PiBorg, as well as a host of simple kits like the cute 3D Xmas Tree. It also has a robust maker store with plenty of extra parts you'll need for many projects.



## PI SUPPLY

[pi-supply.com](http://pi-supply.com)

A great online store for everything Raspberry Pi, Pi Supply usually has some very interesting Pi add-ons, and they accompany projects as well, including the Flick 3D gesture boards and the PiJuice mobile power HAT.

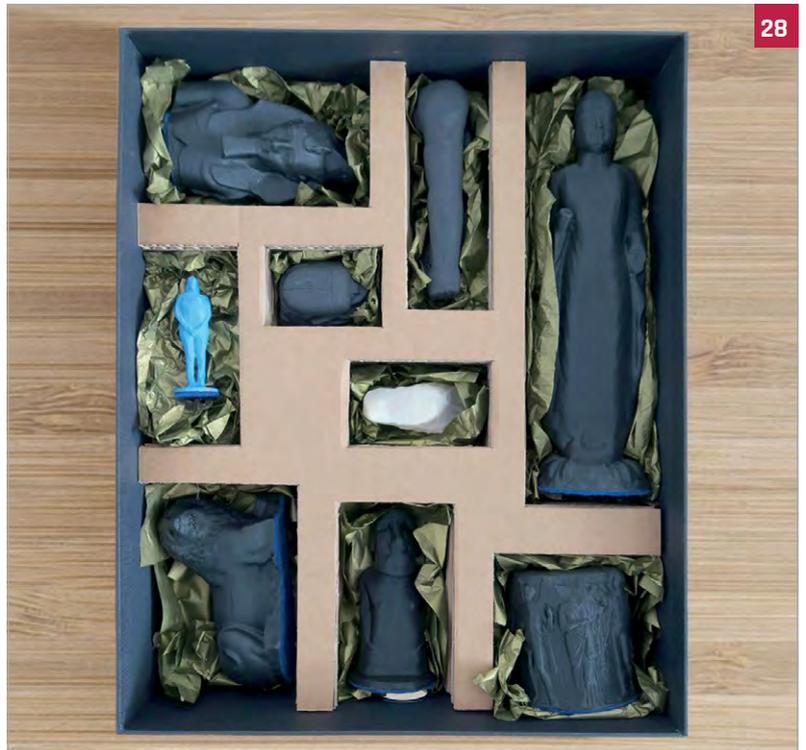


# PROJECTS SHOWCASE

Here are just some of the amazing projects that the Raspberry Pi community make every day. Hopefully they'll give you some amazing ideas of your own



20



28



24



48



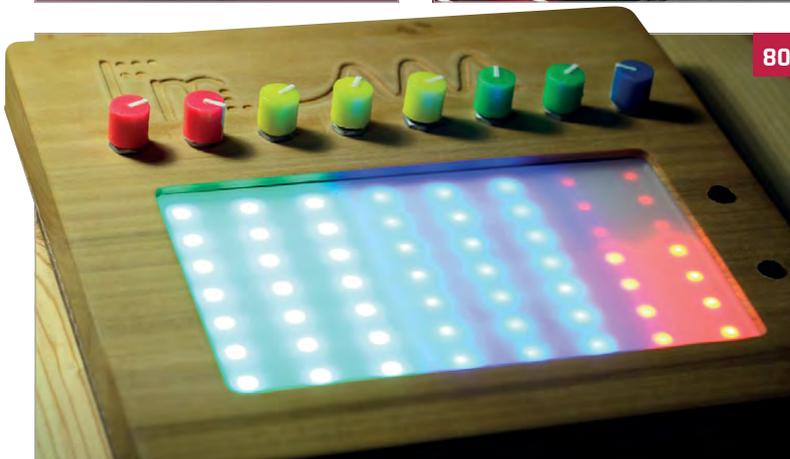
54



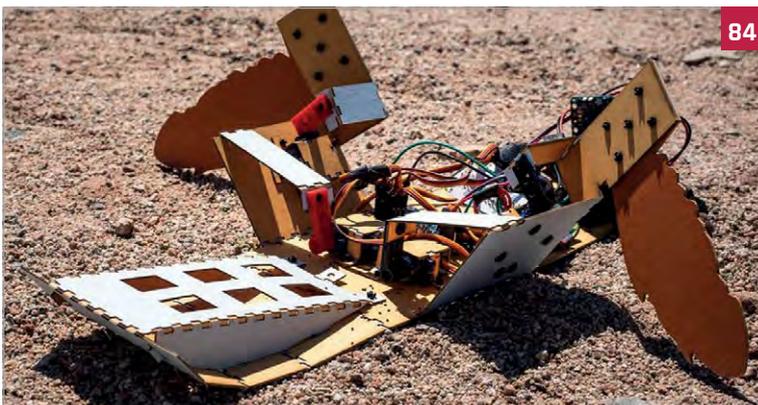
58



64



80



84

# Projects

- 20 PEGASUS**  
Trying to break the land speed record with the help of a Raspberry Pi
- 24 QBEE**  
This dress reacts to activity on social media by lighting up
- 26 SELF-PLAYING ORGAN**  
A organ that plays itself thanks to a Raspberry Pi
- 28 MUSEUM IN A BOX**  
Bring priceless museum artefacts to anywhere in the world
- 32 ZERO PHONE**  
A Pi Zero-powered phone to make yourself
- 34 TRACKING TELESCOPE**  
Find the right part of the night sky
- 36 PI MINI MINT**  
Squeezing a Pi and screen into a mints tin
- 38 PI FILM CAPTURE**  
Digitally converting reel-to-reel film
- 40 POLAPI-ZERO**  
A thermal camera project using the Pi Zero
- 42 WINDOWS 98 WATCH**  
A smartwatch with a 20-year-old OS
- 44 PI ORDER**  
Improving the takeaway order system of a restaurant with Pi
- 46 TOUGH PI-ANO**  
When your piano needs to take a beating, use arcade buttons for keys
- 48 RASPBERRY TURK**  
Play chess against a (real) Pi-powered robot
- 50 DAILY PROPHET**  
A Harry Potter-inspired newspaper with video
- 52 SATNAV ROBOT**  
Improve robot automation with GPS and a Raspberry Pi
- 54 TELE2**  
Step into someone else's shoes thanks to this VR invention
- 56 DOG TREAT MACHINE**  
Give your dogs treats when you're not there
- 58 PPOCKET TINY**  
A very small handheld games console
- 60 PHOTO BOOTH**  
Take better photos at your wedding reception with this booth
- 62 PI BASH**  
Roll up an test your strength on this classic carnival game
- 64 12-FOOT GUITAR**  
This giant electric guitar actually works
- 66 GULLIVER'S GATE**  
A miniature of New York's Times Square made using Pi
- 68 GOOGLE PI INTERCOM**  
Upcycle an intercom into a personal assistant
- 70 WAVES**  
Print out your own voice as a waveform
- 72 PI DECK**  
Scratch DJing on a digital vinyl system
- 74 ALONSO'S ROBOTS**  
A Disney Pixar animator makes cute robots
- 76 INSTAGIF**  
Print out GIFs using this camera and tiny display
- 78 MESMERIC MIRROR**  
A mirror that lights up as you walk past it
- 80 FM TOUCH SYNTH**  
This touchscreen synthesizer is very powerful
- 82 3D BODY SCANNER**  
Get a 3D scan of yourself to show off to your friends
- 84 C-TURTLE**  
This Raspberry Pi robot looks for land-mines
- 86 FRED-209**  
Nerf gun robot gives you 20 seconds to comply
- 88 RAPANUI**  
T-shirts get printed using a Pi-based system
- 90 NASA ARM**  
This robot expresses itself with drawings
- 92 TURNIN' TURBO DASHBOARD**  
Play arcade classic Out Run on this upcycled Tomy toy

This article is dedicated to the memory of Edward Joseph Shadle, 1941-2018



**MATT LONG & MARK NICHOLS**

Matt Long and Mark Nichols (pictured) are Microsoft cloud solution architects who work on the Pegasus Mission. [pegasusmission.com](http://pegasusmission.com)

# PEGASUS AND THE NORTH AMERICAN EAGLE

## Quick Facts

- ▶ The car is 17 metres long
- ▶ It has 42500 horsepower in full afterburner
- ▶ The wheels are milled from solid aluminium
- ▶ The wheels can endure speeds of over 800mph
- ▶ It consumes up to 90 gallons per minute during a run

This land speed record-attempting jet car has a Raspberry Pi sitting right behind the driver

**I**nside the cockpit of one of the most remarkable machines on Earth sits a Raspberry Pi.

The North American Eagle is an ongoing attempt by an American team to reclaim the land speed record.

The current world record stands at 763.035 mph. This was set in October 1997 by British Royal Air Force pilot Andy Green, driving his jet-powered Thrust SSC car.

“We have a mission to break the World Land Speed Record of 763 mph (1227.93 km/h),” says

Brandyn Bayes, team member. “Starting with the battered hulk of a Cold War-era jet plane, we have developed one of the most sophisticated racing machines ever.”

The North American Eagle started life as a Lockheed F-104 Starfighter. “It was used as a chase plane,” reveals Brandyn. These are planes used to follow experimental aircraft and measure engineering data. It followed some of the greatest experimental aircraft in history, among them the X-15 and the Northrop-Grumman B-2 Stealth Bomber.

The particular F-104 that has been transformed into a superfast car was designated 56-0763. By 1998 it was “beyond a shambles,” says Brandyn. “The North American Eagle project gave it new life, so that it could once again be one of the fastest machines on Earth.”

**Right** Sensor data from inside the cockpit is gathered by a Raspberry Pi





The Raspberry Pi is placed inside the cockpit, right behind the driver's head

Aluminium wheels are attached to the jet, turning it into a car

The North American Eagle is a Lockheed F-104 jet with its wings removed

### High-speed Pi

While North American Eagle is gunning for the speed record, the Pegasus Mission behind it usually has its eyes to the sky.

“Pegasus Mission uses a high-altitude balloon as a delivery

a Microsoft cloud solutions architect) to the North American Eagle project. “We asked to join the North American Eagle team,” Mark tells us.

The Pegasus Mission balloon “provides a rich and interesting

Mark. That device was a Raspberry Pi 3 running Windows 10 IoT Core. The software was written in C# as a Universal Windows Platform application.

“Microsoft has been a partner supporting North American Eagle for quite a while,” says Mark. “Volunteers got together to build this hardware, software, cloud, mobile, and web app capability.”

The main aim of the Raspberry Pi was to analyse the car during its test run. “It gathered real-time sensor data,” says Mark. GPS, atmospheric, acceleration, and sound level data was gathered by the Raspberry Pi and sent to the Microsoft cloud.

The Pegasus team could “evaluate the benefits of real-time data reporting and analysis in combination with the goals of the North American Eagle team,” he explains.

## “ The North American Eagle started life as a Lockheed F-104 Starfighter ”

system to carry a payload,” reveals Matt Long, a Microsoft cloud solution architect. The balloon is “packed with meteorological sensors to stream telemetry and control flight operations in real time.”

It’s this expertise in real-time communication that attracted Matt and Mark Nichols (also

experience for users viewing the flight as it happens,” Mark explains. “We started with the idea that the Pegasus team could provide a real-time view of what was happening with the vehicle to a global audience.”

“For this project I built a custom device that was installed in the cockpit of the vehicle,” reveals

# HIGH-SPEED TELEMETRY

## >STEP-01

### The device

A Raspberry Pi forms the heart of the telemetry device. It's packed with sensors and placed inside the cockpit of the North American Eagle.



## >STEP-02

### Azure cloud

The Raspberry Pi connects to Microsoft Azure Cloud during the speed run. It runs Windows IoT Core and uses software written in C# as a Universal Windows Platform application.



## >STEP-03

### Doing a run

Users can join in on the speed run and send messages to the team, which are displayed during the run. Telemetric data from the run is sent in real-time back to the team on the sidelines.

The more entertaining aspect was the real-time interaction with observers during the test run. “People using [our mobile] apps could also send goodwill messages through the cloud and out to the car, where they were shown on an LCD display,” says Mark. The messages were displayed in a screen behind the driver.

Mark also placed a GoPro camera behind the driver to record the user messages on the screen. “The driver could not see them while driving,” he explains, “as that would be a distraction, but the camera could see the display and the cockpit.”

Naturally, the team needed to filter out inappropriate messages. “The system is configured to check a user message for profanity by calling a third-party SaaS service, Web Purify,” says Matt. “If the message passes, then it’s sent back into the system and onto the device in the cockpit for display.

“The extra benefit [of the camera] is that it also gave us a front view out of the cockpit,” continues Matt. This video was used during a debug session with Jessi Combs (one of the test drivers) after her first run.

“We also invested in two drones to record video, which gave us a great field of view during runs, as well as recording other aspects of the event and its remote location,” adds Matt.

## Technology matters

“The technology that makes this real-time and scalable is Azure,” Matt reveals.

“Azure provides a delivery mechanism for global communications in real time,” he explains. “We can capture the telemetry from the device, messages for users, and analyse or even replay data exactly as it occurred.”

Matt tells us that North American Eagle’s comprehensive



on-board telemetry can be uploaded to Azure and viewed as a time-series event. “A team of around 20 Microsoft volunteers built mobile applications on Xamarin for iOS, Android, and Windows mobile, and a website in Azure App Services.”

“Since day one, people have said this project was just a little crazy,” comments Brandyn. “Perhaps that’s a bit of an understatement, yet we couldn’t agree more. It’s that very adventuring spirit that has pushed us to continue to go further, to do something few would ever dare.”

Microsoft Research enabled the team to stream live telemetry to fans as the vehicle raced across the empty desert. “We were able to share over 500 million points of telemetry to over 4500 people on Jessi’s first run alone,” reveals Brandyn.

“We received messages from users in Australia, New Zealand, UK, France, Norway, and the USA.



One of the users watching Jessi Combs’s first run sent a message to the device with her approximate top speed before she stopped.

“While we didn’t break the records we had hoped for,” says Brandyn, “what we found was a reminder of why we love racing: the undeniable rush of adrenaline

when things go right, despite the setbacks along the way.”

“The North American Eagle team, Ed Shadle, and Jessi Combs have pushed themselves to the limit in the race for a land speed record,” adds Matt. “In the process, they have made us all proud members of the human race.”

**Above** The North American Eagle uses its jet engine to achieve astonishing speeds





**CLODAGH O'MAHONY**

Holding an MSc in Interaction Design, Clodagh is in charge of FABTABLAB in Utrecht, the Netherlands. She is also a master rower and archer. [clodaghmahony.com](http://clodaghmahony.com)



All components are located at the back of the dress

Different colours represent touch to different body parts, recording this data to the Raspberry Pi

The aim was for a dress that would still be fashionable and comfortable, despite the additional tech

# Q BEE A SPECULATIVE SOCIAL MEDIA PLATFORM

**Quick Facts**

- Clodagh's Instagram documented the entire process
- The dress was redesigned to improve comfort and functionality
- The Raspberry Pi records touch and keywords
- It then uploads this data to its SQL database
- Conductive thread was used to create 'sectors'

Clodagh O'Mahony's university thesis project records touch and voice data to award points for social interaction

**O**n her website, Clodagh O'Mahony describes herself as a "multi-disciplinary designer with experience in product, graphic, and UX/UI design, as well as illustration and media production."

Having completed her BSc in Product Design and Technology at the University of Limerick, Clodagh went on to study for her master's degree at the same establishment, this time in Interactive Media. This is where the Raspberry Pi comes in.

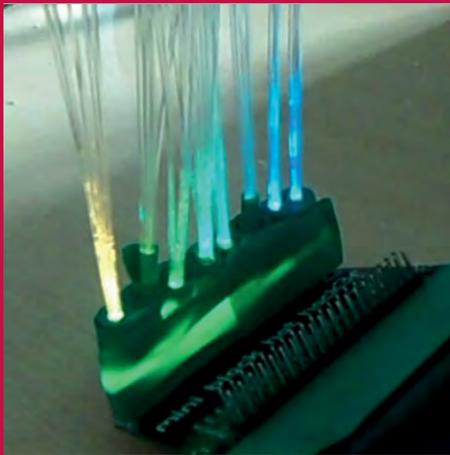
For her thesis project, Clodagh created a dress and an accompanying website to comment on the progression of social media interaction – the idea that it's

getting harder and harder to 'hide' on platforms such as Facebook and Twitter due to the sheer amount of personal information we pump into our timelines. Whereas a person could once create an entirely new persona through the predominantly text-based interaction of blogs and chat rooms, we now live a more visual existence online. Photo, video, and emojis have replaced textual communication, adding more 'face' to the name, and inevitably adding more reality. With this in mind, Clodagh set out to design "a wearable connected platform that introduces what is sold as a 'purer' form of social

media. The quantitative data means users would have to go to extraordinary lengths to misrepresent their lives, thereby making its information more reliable than that of its competitors."

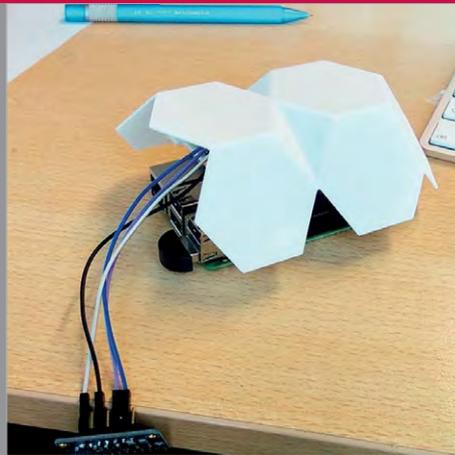
Clodagh created a corporation named 'Qbee', an abbreviation of Queen Bee, with the associated honeycomb theme playing a significant part in the look of both the dress and website. This corporation, if given true life, would provide a range of wearable tech – similar to her dress – that would allow for the recording of social interaction data, updating it to the wearer's online Qbee account.

## THE GREAT ELECTRONIC SEWING BEE



### >STEP-01 Lights

The fibre optics attach to the individual RGB LEDs of the Blinkt, allowing colour control of each cluster. This removes the need to wire multiple LED lights through the dress.



### >STEP-02 Housing

The 3D-printed casing replicates the hexagonal look of a honeycomb, a theme which is consistently represented across the dress and website.



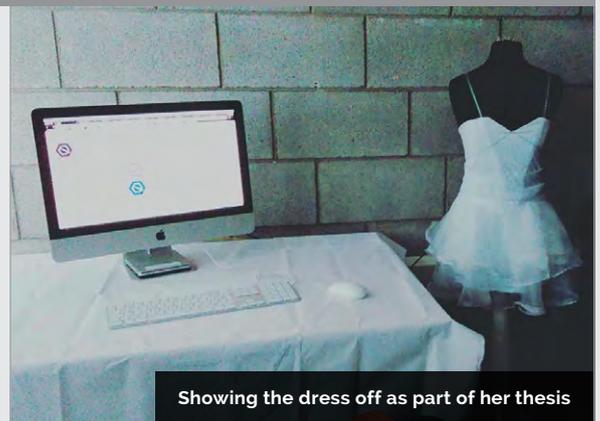
### >STEP-03 Wearing

The original dress design fits the Raspberry Pi and other components at the back of the wearer, stylishly incorporating form and function.

The aim of the build is to record physical interactions between the wearer and the people with whom they come into contact in the real world. A touch to the waist, for example, would be recorded with a certain set of points, whereas a touch to the back would record another. Alongside this physical interaction data, a microphone is used to listen out for any of a series of keywords that are listed

activates the dress to glow a warm purple, a touch to the hip turns it green, and so on.

The dress went through a couple of redesigns throughout the process of the build, all documented on Clodagh's Instagram account ([magpi.cc/2eJgHuZ](https://www.instagram.com/magpi.cc/2eJgHuZ)), allowing for improvements to cost, comfort, and usability factors. The original dress, though fitting exactly to the design plan of colour-related sectors,



Showing the dress off as part of her thesis

“ A wearable connected platform that introduces what is sold as a ‘purer’ form of social media ”

as either positive or negative, whereupon the relevant point data can be recorded.

The build incorporates an Adafruit 12-key capacitive touch sensor breakout board, Pimoroni Blinkt, fibre optics, and a Raspberry Pi, all fitted within a beautiful hexagonal 3D-printed casing.

Clodagh's aim was to use the Blinkt and fibre optics to add colour to the data recording: the touch of a hand to the waist

wasn't very comfortable. This led Clodagh to create another. Though the second dress doesn't offer exactly the same functionality, it does look the way she wanted, and still uses the Blinkt, though in a slightly different manner. Touch the new dress in any of the sectors and the Blinkt runs through a rainbow sequence until the touch is concluded: it is enough to demonstrate the idea of data recording and capacitive touch.



Above Clodagh experimented with multiple 3D-printed cases, finding the ideal location for the tech to be housed



**WENDELL KAPUSTIAK**

A retired investment banker from Venice, Florida, Wendell is an experienced woodworker. He uses a Raspberry Pi to log weather station data and monitor his electricity usage. [wakrec.blogspot.co.uk](http://wakrec.blogspot.co.uk)

# SELF-PLAYING PIPE ORGAN

This handcrafted wooden instrument can play any MIDI tune

**Quick Facts**

- > Wendell first tried making a MIDI wine glass player
- > The organ took about three years to build
- > He bought the blower on eBay for \$80
- > Pipes range from 9 to 40 inches in length
- > He's now rebuilding the organ in solid oak

**E**xperienced woodworker Wendell Kapustiak needed all his carpentry skills to create this impressive self-playing pipe organ ([magpi.cc/2fMacLy](http://magpi.cc/2fMacLy)). “The most fundamental [problem] was that I had no idea how a pipe organ actually worked,” admits Wendell, who eventually based its mechanical workings on a project by Matthias Wandel ([woodgears.ca](http://woodgears.ca)). The most difficult part was making the 42 wooden

pipes, which span three-and-half octaves, since each one has unique dimensions to produce the correct pitch. For this, Wendell used information provided by Raphi Giangiulio’s YouTube videos ([youtu.be/~mibK\\_Dp-ZY](http://youtu.be/~mibK_Dp-ZY)).

The pipes are linked via PVC plumbing to a wind chest powered by a Kooltronic KBR125 blower, as used in data centres. “I had originally tried a small shop vac as a blower,” Wendell tells us. “It had

two problems. The first was that it was very noisy; the second was that it ran hot. When I enclosed it in a box to control the noise, it got so hot that my first one burned out.”

To make it play, each wooden pipe has a valve opened and closed by a solenoid, triggered from an Arduino Due via a power-boosting driver board. The ‘brains’ of the operation is a Raspberry Pi, which performs three main functions. As well as a graphical



Each of 42 handcrafted wooden pipes produces a different musical note

When triggered, solenoids open their respective pipe valves to play notes

The solenoids are wired indirectly to an Arduino Due receiving data from a Pi

## MAKING SWEET PIPED MUSIC

### >STEP-01

#### Wooden pipes

Each of 42 wooden pipes produces a different note. While larger usually means lower, some of the bottom notes are produced by shorter pipes containing an airtight stopper to reduce the pitch.



### >STEP-02

#### Solenoid valves

Solenoids are used to open and close the pipe valves to play and stop notes. O-rings prevent them from sticking in the on position, while also stopping the plungers making a clacking noise.



### >STEP-03

#### Wind chest

The wind chest features a weighted hinged door which shuts automatically when many pipes are playing, in order to boost the air pressure. The wind is provided by a Kooltronic KBR125 blower.



user interface for selecting music to play, it converts the MIDI binary files into delay/note-on/note-off commands, plus musical directions: “The Raspberry Pi takes into consideration tempo changes and any other subtleties in note timing, and converts them into a number of microseconds,”

**Above** A home-built driver board with transistors is used to boost the power from the Arduino output to trigger the solenoids

a sample in about 20 minutes and it worked a treat.”

Another issue involved the spacing of the pipes, which Wendell discovered could interact with each other if placed too

“ The Raspberry Pi takes into consideration tempo changes and any other subtleties ”

explains Wendell. “In this way, the Pi has done all the heavy lifting as far as calculations go.” Another Python program is then used to send this data to the Arduino via USB.

While the high-tech side of the project proved fairly hassle-free, the physical engineering was more problematic. One difficulty was the wind pressure regulator, which originally had a bellows-type mechanism. “I was trying out a few different designs, trying to develop one that was mechanically simple and dependable while producing a very stable pressure.” Fortunately, Wendell’s friend Jim, a retired HVAC engineer, stopped by and suggested a new approach, using a weight-controlled door as used in ventilation systems. “I whipped up

closely together. “I suddenly understood why organs were designed with the pipes in a few rows but spread out, not bunched together.” He also reversed the overall layout to put the solenoids at the front, so listeners can see them operating the valves. He has since added some LEDs: “It makes it much easier to follow what’s going on than just watching the solenoids.”

The finished pipe organ generated a lot of interest from attendees at the Orlando Maker Faire. “I think most people were able to walk away with some level of insight as to how the combination of the old technology of the pipe organ and the new technology of the computer and microcontroller fit together.”



**MUSEUM IN A BOX**

The core team consists of George Oates, Tom Flynn, Adrian McEwen, and Charlie Cattel-Killick.  
[museuminabox.org](http://museuminabox.org)



- The Raspberry Pi, along with the RFID reader, acts as the 'brain' of the box
- Each object is fitted with an RFID tag, preset to play back relevant content
- 3D-printed objects allow us to bridge the 'do not touch' gap with ease

**Quick Facts**

- ▶ Started as an R&D project by Good, Form & Spectacle
- ▶ The team are based in London and Liverpool
- ▶ The first box was built at Somerset House
- ▶ All the original pieces were scans from the British Museum
- ▶ The company incorporated in October 2015

# MUSEUM IN A BOX

Museum in a Box gives us the chance to experience incredible pieces of art, artefacts, music, and more at our fingertips, anywhere in the world

**Y**ou can visit Hoa Hakananai'a at the British Museum, London. A Moai, you'd likely recognise him as one of the Easter Island statues. He's 2.4 metres in height, nearly a metre wide, and is estimated to weigh around four tonnes. He sits on a high plinth surrounded by text

regarding both his own history and that of his fellow Moai and it's fair to say that, unless you can go to the British Museum in person, you're unlikely to see him visiting your local museum, school, or library on loan.

Now imagine holding a smaller version of Hoa Hakananai'a in your hand. He fits perfectly on

your palm and allows you to feel the texture of his surface and the shape of his features. You can pass him around, reposition him, and even drop him if you lose your grip. And as you 'boop' him on the top of the Museum in a Box's Raspberry Pi-powered 'brain', he starts to tell you the story of his

Shrunken models of famous 'giants' allow for greater access to pieces across the globe



sea voyage from Easter Island, of the history of his creators, and his first encounter with the explorer Captain Cook in 1774.

### Bonding with history

This idea of forming stronger connections with objects through touch and sound is the heart of

get to grips with a history they may otherwise miss out on.

On the technical side of the project, the Museum in a Box consists of a wooden box 'brain' that houses a Raspberry Pi and RFID reader. There's also a volume control for playback and lights to indicate when the unit

“ The team aim to break the disappointing, yet often necessary, Do Not Touch stigma ”

the Museum in a Box objective. Through 3D-printed models and wooden sculptures, 2D images such as postcards and photographs, and 'do it yourself' feedback cards, the team aim to break the disappointing, yet often necessary, Do Not Touch stigma of museums, allowing everyone the chance to

is powered and functional. Each object, whether it be 3D or flat, is kitted out with its own unique RFID tag. When placed upon the box, the tag is read by the reader beneath the surface and informs the Raspberry Pi to play back the appropriate sound file via a built-in speaker system.

## LETTING OBJECTS SPEAK FOR THEMSELVES



### >BOX-01

#### The Planets

[magpi.cc/2iLlwqQ](https://magpi.cc/2iLlwqQ)

Box Prototype No. 13 – The Planets. Seven identical wooden balls sit within a plain black box. When 'booped', each plays a track from the USAF Heritage of America Band's rendition of *The Planets* suite by composer Gustav Holst.



### >BOX-02

#### Frogs in a Box

[magpi.cc/2iLrd88](https://magpi.cc/2iLrd88)

With a somewhat 'flatter' approach, this box uses postcards to play the various calls of the illustrated amphibians, all recorded by "a mid-20th-century herpetologist called Charles". Frogs in a Box is part of a larger pilot programme in conjunction with Smithsonian Libraries.



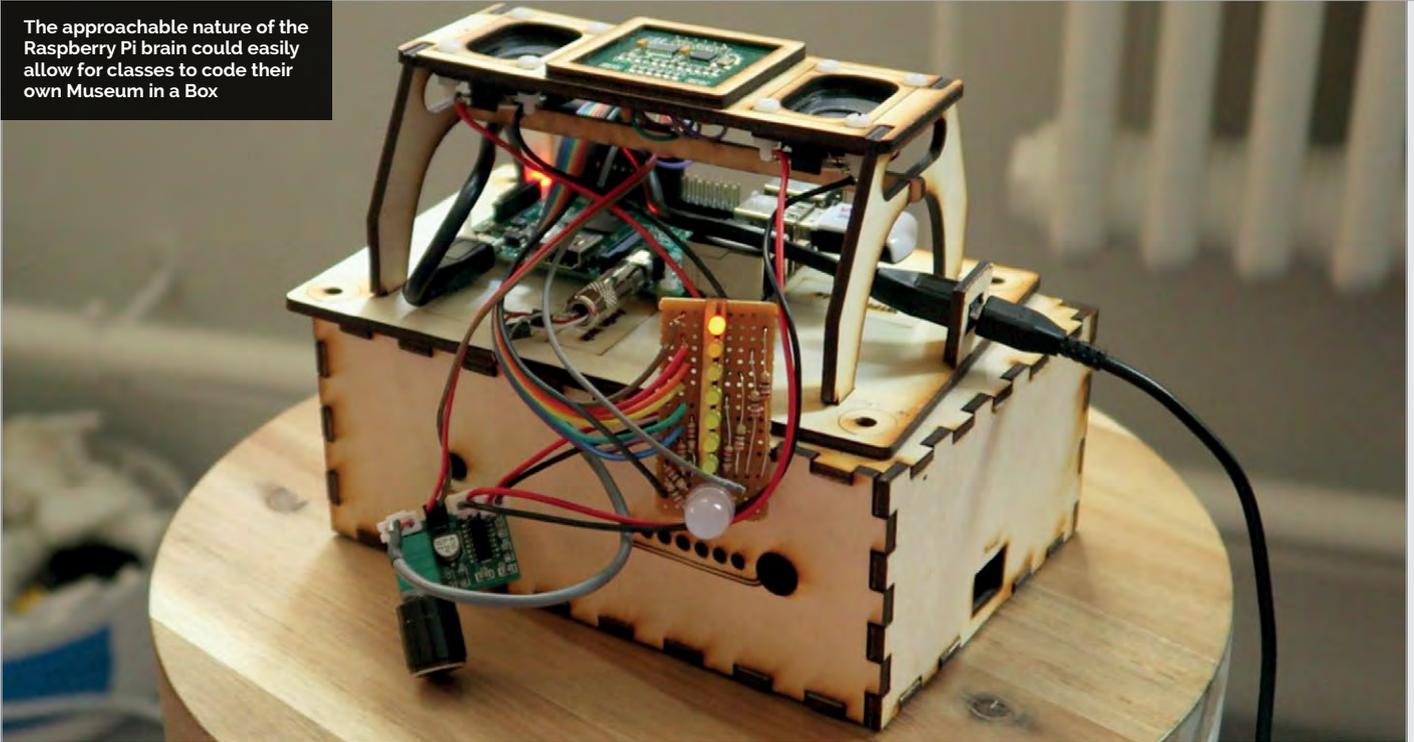
### >BOX-03

#### Ancient Egypt: Daily Lives

[magpi.cc/2iLmaES](https://magpi.cc/2iLmaES)

The 3D-printed items within the Ancient Egypt box depict objects used in the daily lives of those living over 2000 years ago. After it was shown to families at the British Museum, the team upgraded the box to include cards that offer visitor feedback when booped.

The approachable nature of the Raspberry Pi brain could easily allow for classes to code their own Museum in a Box



The sound files vary from object to object. Some play music, such as the Planets prototype box; home to seven identical wooden balls, it is set to play one of the seven tracks of Gustav Holst's *The Planets* suite. While the balls lack any differentiation on the outside, their insides are unique. Boop a ball and the experience is overpowering, thanks in part to the beautiful effect of the USAF

Heritage of America Band as the sound of their instruments swells into life. Now imagine if these balls were 3D prints of the instruments that played famous solos. Or imagine if the ball was the 3D-printed bust of a famous composer who narrates the inspiration behind their work as their music plays in the background, captivating young and old musicians alike.

### Inspired education

It's difficult to experience this project without finding yourself coming up with hundreds of your own ideas for its use.

Given this inevitable excitement and enthusiasm, the second intention of the project comes to life. Imagine if a classroom had its own Museum in a Box, and, alongside their brain box, the students had a second box allowing

## RFID TAGS

A radio-frequency identification (RFID) tag is an inexpensive way of giving an item its own unique, readable code. Often confused with the NFC technology that allows you to touch your mobile phone to pay at your local supermarket or coffee shop, an RFID tag is a fairly simple piece of technology that can either be active, maintaining its own power source to allow you to use it to find lost keys, or passive, taking power from an RFID reader to let you into a locked building with the touch of a plastic card.



As the identical planets play their own themes, imagine the fun of guessing which is which



them to record content onto their own RFID tags. Maybe the class collects objects from their local town and records the items' history and their own thoughts directly to them. Once complete, they're able to send the objects to a different class in a different part of the world and share their experiences with others. Perhaps a museum records narration to a postcard and sends it out to teachers for them to share with their students. The nature of the Raspberry Pi allows for multiple data files, so a first boop could ask a question and further boops could provide more information to continue class discussion.

The team – whose core members include CEO and co-founder George Oates, an interactive designer and project manager; co-founder and designer Tom Flynn, an expert 3D creative; technical lead Alan McEwan, and junior

designer Charlie Cattel-Killick – use their combined expertise to build constantly upon the core concept of the project. A great multi-platform use of the box is a recent integration with augmented reality. Cards depicting the fire-damaged ruins of pieces from South London's Cuming Museum can be scanned with the 'Augment' mobile app and brought back to life before your eyes. Pair this with an RFID tag and these lost pieces of history suddenly tell their story. Download the app and try it for yourself with the image here: [magpi.cc/2iLuCUA](http://magpi.cc/2iLuCUA).

At last count, the team had 13 prototype boxes, with others already commissioned for use in programmes such as the Smithsonian Libraries' 'I See Wonder' pilot; and with such incredible scope for use in education, it's not hard to see Museum in a Box thriving.

The British Museum displays a mere 1% of its collection, leaving so much kept from sight



>BOX-04

[magpi.cc/2iL8V6W](http://magpi.cc/2iL8V6W)

Statues of Women in London

Only one audio recording survives of British writer Virginia Woolf and as her 3D replica is booped against the brain of the box, she begins to speak to you. "Words, English words, are full of echoes, of memories, of associations..."



>BOX-05

[magpi.cc/2iLkM4Z](http://magpi.cc/2iLkM4Z)

The Museum Fire

Alongside the box, the team also work on rather impressive content for an augmented reality app called Augment. When South London's Cuming Museum burnt down, this technology allowed for destroyed pieces to be revived with fascinating results.



>BOX-06

[magpi.cc/2i8WMwf](http://magpi.cc/2i8WMwf)

The Brain Extension

The extension permits the recording of new content to tags, giving students, professors, museum visitors, and others the chance to personalise objects with their own unique viewpoints and information.



**ARSENIJS PIČUGINS**

Arsenijs is an 18-year-old student from Riga, Latvia. He's a programmer and hardware repair person who works closely with Linux.  
[crimier.wordpress.com](http://crimier.wordpress.com)

# ZEROPHONE

**Quick Facts**

- The display resolution is 128x64
- Currently, it costs around \$50 in parts
- It connects to a 2G network (a 3G model is planned)
- The OS is based on Raspbian Linux
- There are plans to crowdfund the project as a kit

A SIM800 GPS module and antenna are used to connect to the mobile phone network

A small LCD screen displays the operating system (based on Raspbian Linux)

Buttons are connected to the PCB, and an ATmega328P module is used to convert button pushes into phone commands

**WANT TO SEE HOW FAR IT'S COME?**

Visit: [magpi.cc/rXgkZf](http://magpi.cc/rXgkZf)



## Turning a Pi Zero into a mobile phone

**Z**eroPhone is an open-source project to create a mobile phone kit for the Pi Zero.

Owners of ZeroPhone will be able to make calls and send text messages. It will also have basic apps such as an alarm clock, calendar, calculator, web browser, and music player.

Behind the project is Arsenijs Pičugins, an 18-year-old student from Riga, Latvia. He now has a team of people helping him.

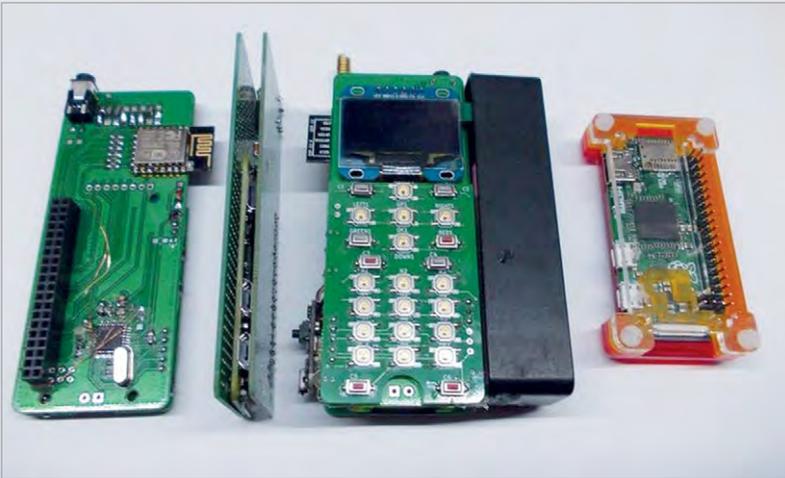
“It’s open-source, Linux-powered, and you can make one yourself for \$50,” he says.

Arsenijs has been studying mobile phones for a couple of years. He hopes to sell a commercial kit, while keeping it all open-source for people to build themselves.

“The phone hardware is at ‘alpha’ stage,” he tells us. Arsenijs has designed the first revision of PCBs (printed circuit boards) and is developing the software.

“The next revision will be ready-to-assemble,” reveals Arsenijs. Raspbian is being customised to run on the ZeroPhone. “It can be tailored to suit our purpose very easily,” explains Arsenijs.

“I’ve currently got a framework I’m using for headless Pi configuration,” he continues. Python is being used to create a user interface (UI). An



**Above** The components and software are under development, and a kit will be made available when it's ready

earlier project called pyLCI ([magpi.cc/2kqv2oA](http://magpi.cc/2kqv2oA)) forms the base of the interface.

Developed by Arsenijs in 2016, pyLCI is a Linux control interface. It enables you to control and configure your Pi with SSH, networking, or using an HDMI monitor.

"I developed it because that's what I needed to improve my Raspberry Pi tinkering and development workflow," recalls Arsenijs.

"Developing a good UI is one of the primary goals of this project," Arsenijs adds.

"That's what I'm focused on, and that means extensively testing the hardware I've got now, to find everything that has to be fixed."

Once v1.0 is ready, he will publish the design files and instructions, so people will be able to start assembling their phones if they wish. He'll then prepare a crowdfunding campaign to fund a manufacturing run and software development.

"It's the only phone you can assemble by getting all the parts yourself," claims Arsenijs. It doesn't use any rare parts or fine-pitch soldering. "You can even

" I developed it because that's what I needed to improve my Raspberry Pi tinkering "

Because Linux is the base, it can support a range of programming languages. Arsenijs expects "original ideas" for apps. He will be working on productivity and healthcare apps.

The open-source software approach is "likely to be secure and not privacy-invading," says Arsenijs. You can disable features you don't like, and it doesn't depend on any cloud services. You will be able to install firewall, I2P, and TOR nodes, as well as secure messenger services.

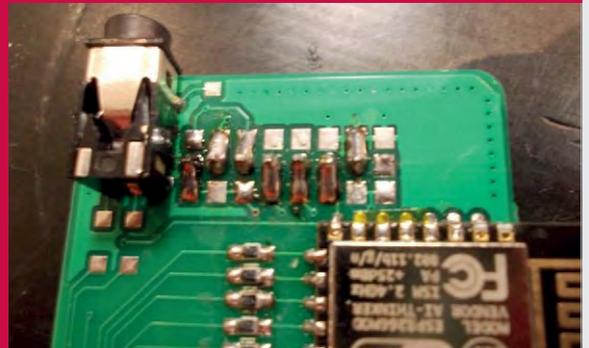
Arsenijs's current goal is to make the v1.0 PCB revision.

breadboard this phone if you're dedicated enough.

"For me, this project is a huge undertaking that, if done right, will influence the way we perceive mobile phones," remarks Arsenijs. He explains that it will help solve privacy and security concerns, as well as allow us to develop better open-source GSM technology. It will also help power mobile phone hacks.

"Besides, this is an open-source Linux-powered phone you can independently assemble for less than \$50," observes Arsenijs. "That's pretty cool."

## DESIGNING A PI ZERO PHONE KIT



### >STEP-01 SIM module

A SIM800 module is used to connect to the GSM network. AT (Attention) commands, from the Hayes command set ([magpi.cc/2jvCYyO](http://magpi.cc/2jvCYyO)), are used to control the modem connection to the mobile network.



### >STEP-02 LCD and battery

A small LCD screen is used to display the user interface. A battery with 18650 cells is used to power the Pi Zero, display, and SIM800 module.



### >STEP-03 Controls

Two-layer PCBs (two 4x10 cm and one 4x6 cm) are used. Regular buttons are placed on the PCB to form a standard keypad. It's all hooked up to an ATmega328P that's used to read the button pushes.



## MIKE HAMENDE

Mike Hamende hails from Manteno, Illinois. With a passion for making, his projects continuously gain him online recognition.  
[imgur.com/a/IOkCI](https://imgur.com/a/IOkCI)

# TRACKING TELESCOPE MOUNT

For Mike Hamende, making something is far more exciting and rewarding than logging into an online store, as his Pi-powered tracking telescope mount proves

## Quick Facts

- Mike has no previous experience of coding Python
- The tracking mount is set to find 800 preset locations
- The mount runs on a battery, making it portable
- It cost \$350–\$400 to build
- See a video of the mount working here: [magpi.cc/2lGLgWf](https://magpi.cc/2lGLgWf)

If you search for Mike Hamende’s user name ‘8PumpkinDonuts’ online, you’ll soon realise that he’s quite the craftsman. From a suspended lounge for his fiancée, to a wristwatch built completely from scratch, Mike is a true maker. So it’s really no surprise that he’d eventually build a Raspberry Pi project that we could sink our teeth into. Despite his lack of previous Python knowledge, and ignoring the fact he could easily purchase what he’d decided to build, Mike produced a tracking telescope mount for his 8” Orion reflector. “This is definitely the most involved project I’ve ever completed,” Mike tells us. “I’m aware there are mounts I could have bought to accomplish the same thing, but, while astronomy is a hobby of mine, building things is what I’m really into.”

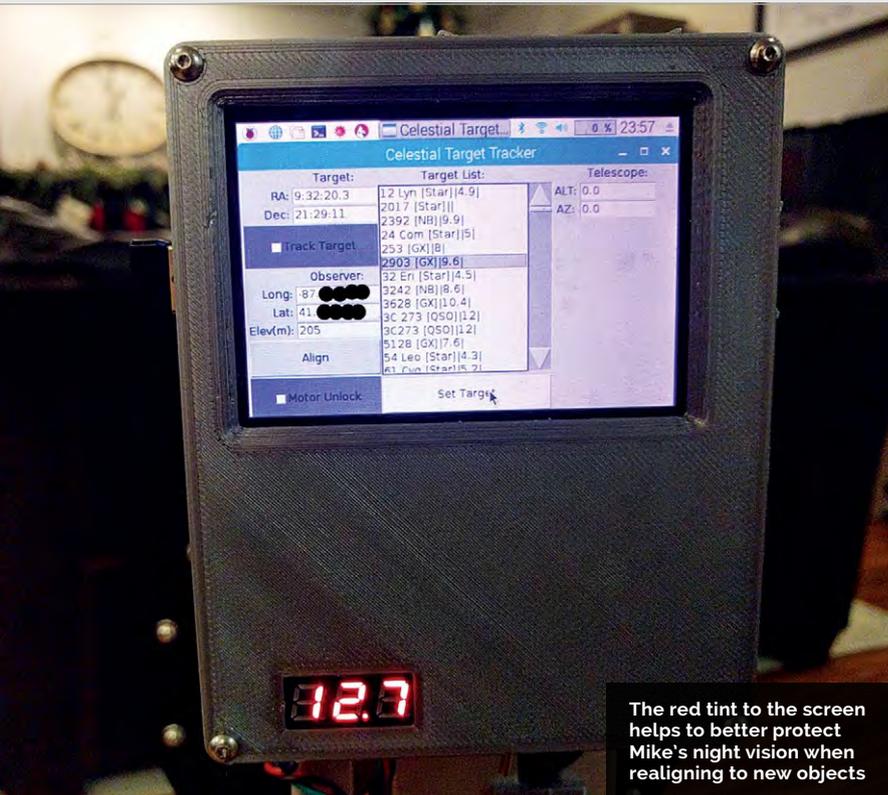
Mike used two Nema 17 stepper motors, powered by a Raspberry Pi 3, to angle the mount. For the frame, he used 3×3” T-slot aluminium, utilising his CNC mill, CNC router, and CNC lathe to cut further pieces of aluminium for the mount.



• Mike 3D-printed several parts of the mount when he was unable to find the correct pieces

• The Raspberry Pi sits beneath a touchscreen, housed in a 3D-printed case

• The mount runs on a 12V/12Ah SLA battery hooked to the back of the frame



The red tint to the screen helps to better protect Mike's night vision when realigning to new objects

Two turntables with 60 mm thrust ball bearings act as the basis for the motion mechanics, cut from  $\frac{1}{2}$ " thick aluminium. Mike then had to build his own drive pulleys using 3D-printed parts and fibreglass flanges. He amended technical documents he'd found online, removing teeth, until the pulley fitted perfectly within the build. From here, he also went on to print, rout, and fit all the pieces needed for the mount and frame

to control the stepper motors and linear acceleration, so he was still faced with the challenge of quickly learning enough Python to see him through.

Once he'd cleared up his prototype breadboard and soldered the boards into place, Mike fitted the Pi with a touchscreen. The screen enables him to select one of 800 preset targets for the telescope to focus on. If he wants to view any other

“ My hobby is making things and I've been doing it since I was old enough to hold a screwdriver ”

before turning to a less-familiar subject: the Raspberry Pi.

Mike originally created a script that would position the telescope to align with celestial objects by way of information from the US Naval Observatory. However, upon completing the code, he discovered PyEphem, a Python package that offered a much larger scope for celestial exploration. Though he had the PyEphem code, he couldn't find anything

object, he simply uses a small wireless keyboard to enter the chosen coordinates manually.

Mike went to school to study mechanical engineering, though he attributes much of his skill to being a hobbyist. “My hobby is making things and I've been doing it since I was old enough to hold a screwdriver.” He even converted his home lathe and router to be CNC, and built his own 3D printer too.

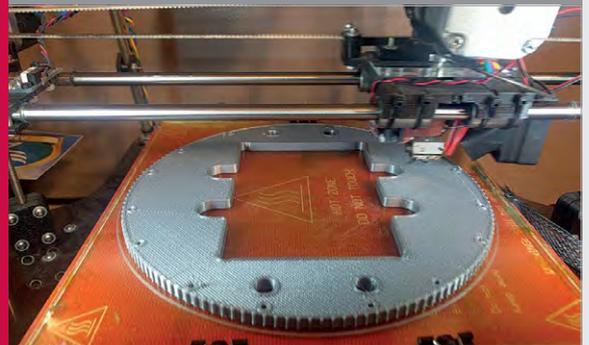
## SCOPING THE SOLAR SYSTEM



### >STEP-01

#### Creating the parts

Mike used his own tools to design and cut the appropriate parts for the mount out of  $\frac{1}{2}$ " aluminium. This allowed for greater control of the final look.



### >STEP-02

#### 3D-printing pulleys

Mike redesigned pre-existing technical documents to make them fit within the mount. With 288 teeth, this pulley was the largest he could print.



### >STEP-03

#### Getting on the level

This large bullseye level allowed for less complication when aligning the telescope to the desired object, ensuring he was always on a flat surface.



**MATT WAGNER**

Matt spends his time developing new Raspberry Pi projects, for a more connected world. In the future he hopes to spread this passion for a full-time profession. [hackmypi.com](http://hackmypi.com)

**Quick Facts**

- The original version featured a Pi NoIR camera
- Sugru stops the charger port moving around
- Hot glue is added to secure solder points
- Matt's new model will feature a phone SIM
- Planned Pi projects include smart luggage tracking

# PI MINIMINT

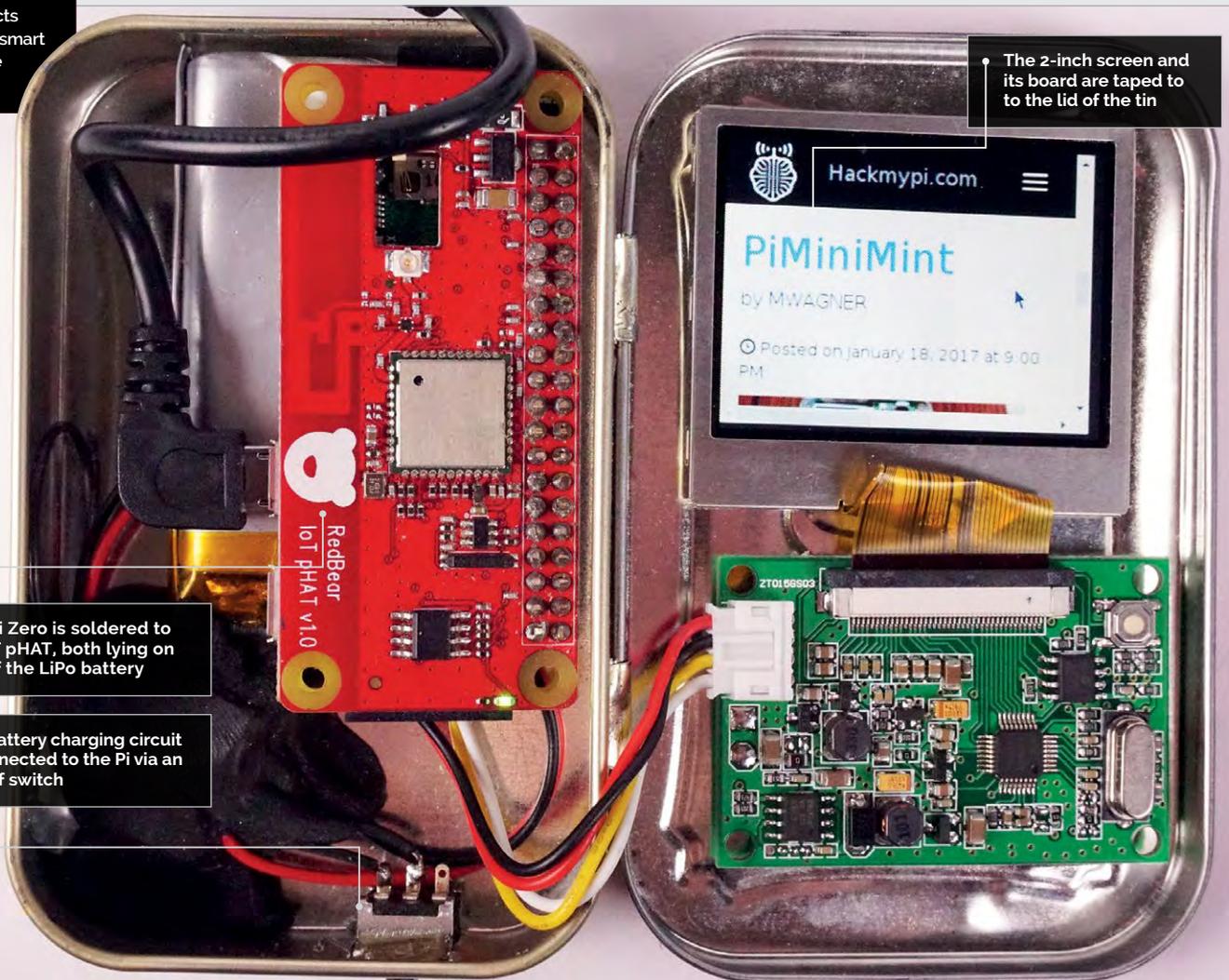
A fully functioning computer and display crammed into a mints tin

**M**att Wagner first thought about fitting a computer into an Altoids tin in 2012, shortly after the launch of the original Raspberry Pi. While it wasn't quite possible back then, the release of the considerably smaller Pi Zero changed everything, although not immediately for Matt. "I was deep into another project,

and the low availability of Zeros made it challenging to find and use for anything at the time," he recalls. "However, I now work in an environment [Micro Center store] surrounded with DIY materials, so the idea just came back to me one day."

The result is the PiMiniMint, containing a Pi Zero connected

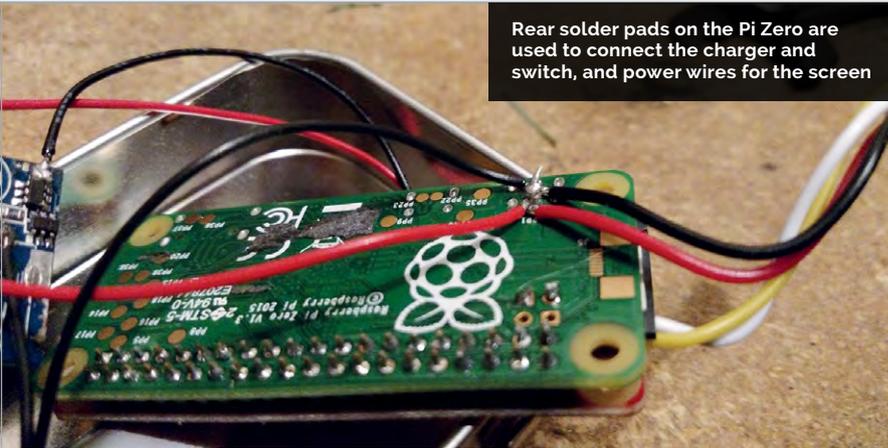
to a 2-inch LCD display and – thanks to a RedBear IoT pHAT – equipped with WiFi and Bluetooth. "The original design took a few weeks, between prototyping and sourcing parts," Matt tells us. "The concept of it is simple: strap together already available components. In actuality, though, sourcing parts was a bit of an issue



The Pi Zero is soldered to an IoT pHAT, both lying on top of the LiPo battery

The battery charging circuit is connected to the Pi via an on/off switch

The 2-inch screen and its board are taped to the lid of the tin



Rear solder pads on the Pi Zero are used to connect the charger and switch, and power wires for the screen

(shipping costs, and buying pre-made parts gets very expensive), so I had to find creative ways to solve problems.”

This included a solution for rechargeable battery power. “In my first iteration, I had no way to charge the battery,” explains Matt. “An Adafruit PowerBoost would have been ideal, as it puts out enough amperage, but the cost would have been tight for me. The [Pimoroni] Zero LiPo is amazing, and I am currently using it in an ongoing project of mine; however, it sits too high on the GPIO and wouldn’t allow for the lid to close with the screen inside.”

In the end, at the suggestion of a co-worker, Matt opted to modify a standard USB phone charger, extracting its charging circuitry to use with a slimline LiPo battery. He also added an on/off switch. “I had some issues regarding wiring and power needs, but those were related to where in the circuitry I put my switch, and the kind of switch I used,” he reveals.

“Changing to a better switch did the trick.” Matt’s 2500mAh LiPo provides impressive battery life: “I haven’t done a full 100%-0% test and timed it, but I have had it on for six-plus hours at a time.”

One drawback of using a built-in battery was that there was no longer room in the tin for the infrared camera and full-size exterior USB port featured in the original design. “However, I do still have an interior full-size USB [via an adapter] that is available when opened.”

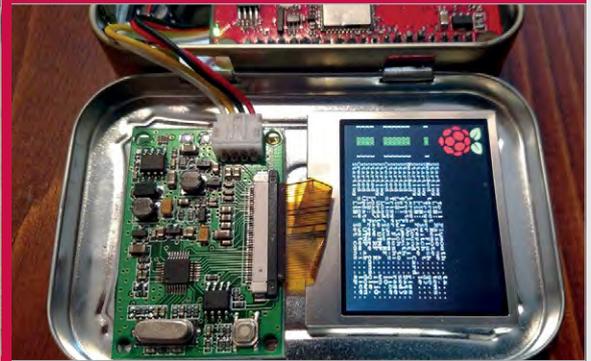
Matt is currently making a second PiMiniMint for a friend, featuring a few modifications: “His is going to have two on-board WiFi antennas and a dedicated SIM for transmitting data over 3G/4G.”

As well as it being a cool pocket PC, Matt thinks the PiMiniMint is ideal for portable retro gaming. “Additionally, I have used it as a WiFi access point for saving files to, a WiFi repeater by plugging a second WiFi antenna into the USB port, and I’m sure there are more ideas out there!”



Amazingly, everything fits into the 60x95 mm Altoids tin; there’s even room for a USB adapter

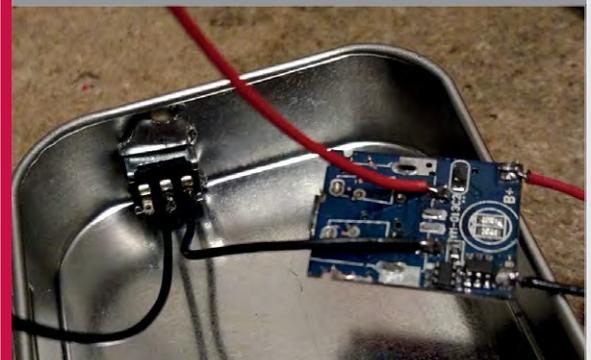
## HOW TO TIN A PI



### >STEP-01

#### Mount the screen

The 2-inch screen is attached to the lid of the Altoids tin using double-sided tape. The display board’s white charge port needs to be near the hinge to avoid stretching the wires.



### >STEP-02

#### Make a charger

Matt stripped the mini circuit board from a standard phone charger and removed its USB port. This board is then soldered to a slimline LiPo battery, and connected via a switch to the Pi Zero.



### >STEP-03

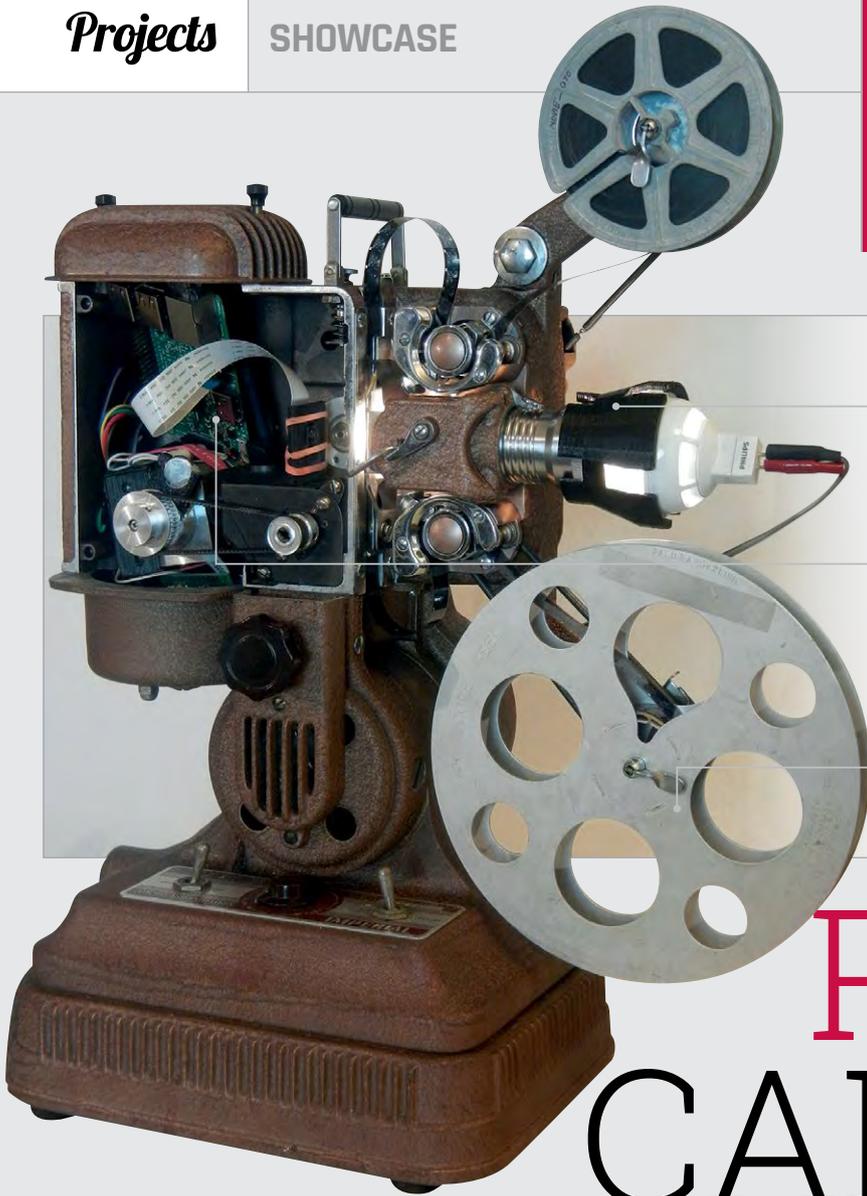
#### Add connectivity

A headerless RedBear IoT pHAT is soldered directly via a male 40-pin header to the Pi Zero to provide WiFi and Bluetooth connectivity. If using the Pi Zero W, this could be omitted.



**JOE HERMAN**

Joe is a tinkerer and programmer with an interest in film preservation. [vimeo.com/jphfilm](https://vimeo.com/jphfilm)



Placed inside the projector, the Raspberry Pi Camera Module records each image, one shot at a time

The Raspberry Pi is used to capture the images from the projector and sends them to a client computer for processing

A stepper motor (controlled by the Raspberry Pi) is used to advance the film reel one frame at a time

# PI FILM CAPTURE

Expertly converting old film reels to digital video one frame at a time

**Quick Facts**

- ▶ A five-minute film reel can produce 4000 still images
- ▶ 8mm captures need a 25mm focal length magnifying lens
- ▶ The projector uses a NEMA 17 motor and a DRV8825 chip
- ▶ Joe 3D-printed mounts for the lamp and motor
- ▶ It uses the original 5MP Pi Camera Module

**R**estoring home movies captured on film can be a real challenge. Film conversion is a time-consuming and often expensive task.

One maker, Joe Herman, took matters into his own hands by hooking up a Raspberry Pi Camera Module to an old projector.

Joe’s cousin and uncle uncovered a box of 130 reels recorded by his grandfather. Leo’s moves “dated back to 1938,” says Joe.

The preservation of these memories was important for Joe’s large extended family. But to have them professionally restored would have cost “many thousands of dollars.”

The obvious approach with a project like this would be to

run the movie and record it using a camcorder. The results from that approach are very poor, though.

Professionals photograph one still image at a time and then stitch it all together, which is why the process is so expensive.

Pi Film Capture is a smart film transfer system that uses a Raspberry Pi and Pi Camera Module with Bell & Howell 8mm and Super 8 projectors, or an Ampro Imperial 16mm.

The setup employs a frame-by-frame approach. The Raspberry Pi takes a still photo of a film frame and sends it to a remote client computer for processing.

The Raspberry Pi is connected to a stepper motor that moves the frame to the next cell. It then

repeats the process until the entire reel is captured. “Most old projectors would work,” reckons Joe. “They were built to last.”

Once the process is completed, the result is a folder of images. “I can combine these quickly into a viewable movie using FFmpeg ([ffmpeg.org](https://ffmpeg.org)),” says Joe. If a more sophisticated clean-up is required, he uses a program called AviSynth ([avisynth.nl](https://avisynth.nl)).

“A key design goal for me was to use easily obtainable parts,” reveals Joe. “Mostly because when I began the project, I was unsure whether it would go anywhere and didn’t want to invest too much in expensive equipment.”

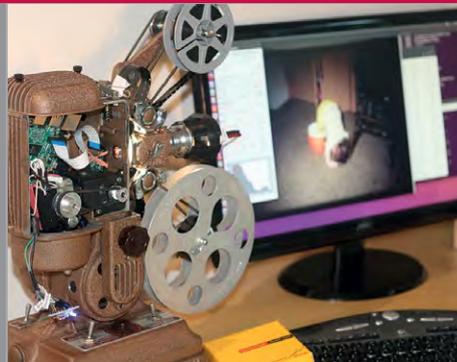
Joe solved “endless” challenges while designing Pi Film Capture.

# BUILDING A CINE-CONVERTER



## >STEP-01 Direct capture

A Pi Camera Module is stripped back and placed directly inside the lens housing. This approach removes any interference from outside light.



## >STEP-02 Client computing

Each still frame of the reel is captured, one frame at a time. The Raspberry Pi sends the images to a client computer for storage and editing.



## >STEP-03 Stepper motor

After capturing the frame and sending it to the client, the Raspberry Pi controls a stepper motor that moves the film to the next frame.

“Most were due to my inexperience and ineptitude,” he tells us. “Or bad programming techniques.”

The first proof-of-concept device drove the projector and triggered a capture as each frame advanced. “It fooled me into thinking this would be an easy project,” recalls Joe. But frames were often under- or over-exposed. Sometimes it would miss frames altogether.

As Joe iterated through different versions of the project, he added a separate client computer: a PC running Ubuntu Linux.

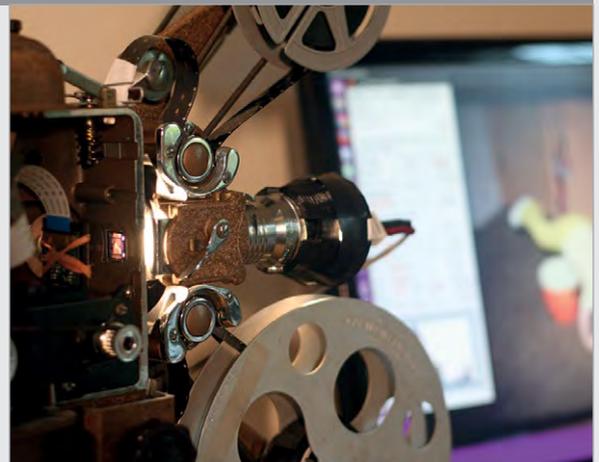
The client computer runs a control panel that lets the user adjust the camera, and displays

the captured frames as they come off the Raspberry Pi. It also sends some simple projector control commands to the Pi.

The latest version of the Pi Film Projector uses OpenCV ([opencv.org](http://opencv.org)). This software adds computer vision and multi-core processing to the project.

“OpenCV provides functions similar to the HDR mode on cell phone cameras,” explains Joe. Given several photos at different exposures, it can blend them into a single image that preserves detail in both bright and dark areas.

“For some films, this can make an enormous difference,” says Joe.



“The downside is it slows down my captures by nearly half as I take more photos, so I try only to use it when necessary.”

Joe advises prospective makers to have patience with the build. “It has a lot of pieces,” he says. The captures take a lot of time, and getting good results can take a lot of practice.

The results are worth it. Joe has converted over 100 of his grandfather’s reels. “Bringing some family history to life again and sharing it has made this the most rewarding tech project I’ve ever worked on,” he tells us. “I hope it inspires other people to bring that old box of film cans out of the closet and reconnect with their history.”

**Above** As the project has progressed, the image editing software has become more advanced, creating higher-quality recordings



Joe has captured over 100 reels of film recorded by his grandfather

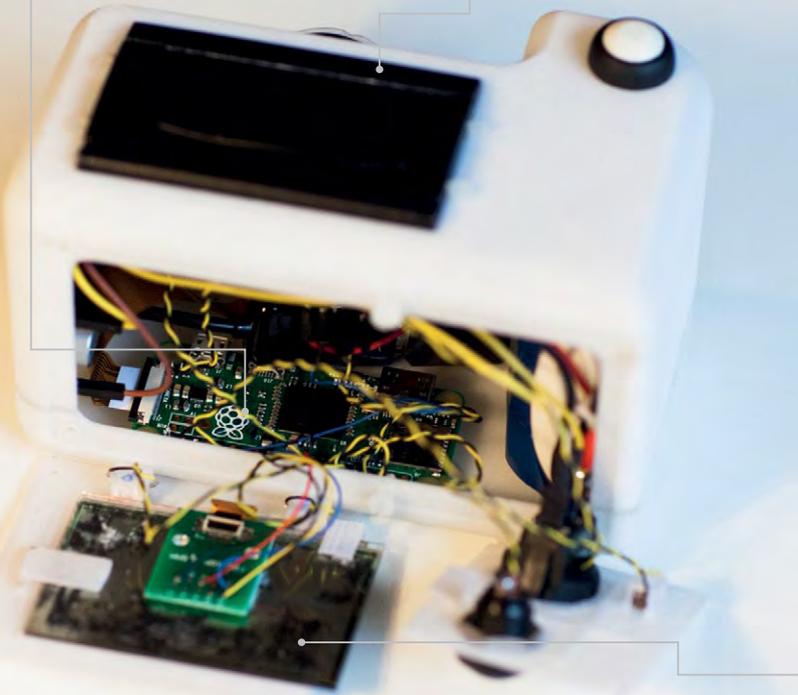
PIERRE MUTH



Pierre Muth is an electronic technician with a love for making things, thermal printers, and the Adafruit Big Red Button. [magpi.cc/2ndIKg8](http://magpi.cc/2ndIKg8)

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

# POLAPI-ZERO

The tiny Pi 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](http://magpi.cc/2ndslE3)
- ▶ You can navigate past images via button press
- ▶ Watch the camera in action here [youtu.be/8D-sL3GoFZM](https://youtu.be/8D-sL3GoFZM)

**T**he 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.

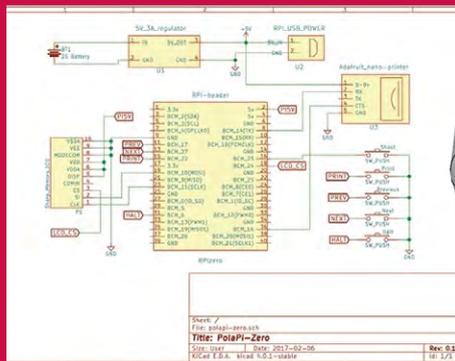
We've seen many digital camera builds using the Raspberry Pi and Camera Module. From 3D-printed

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 Vit 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/2nds1E3](https://magpi.cc/2nds1E3)) 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](http://314reactor.com)

**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 has also created a Robo Skull

# WINDOWS 98 WATCH

Wind back the clock with this chunky wristwatch

**I**f 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.

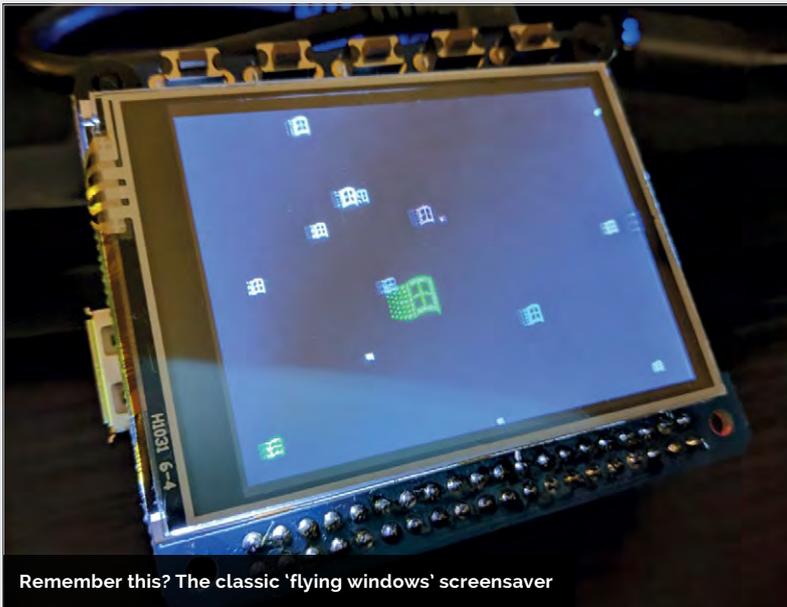
While he admits the ancient OS used to drive him mad back in the day, he has an odd nostalgia for it. “Many years later you look back on it and want to relive it. I think time has a funny way of keeping more of the good than the bad within memory.”

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](http://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. “It’s 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,

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

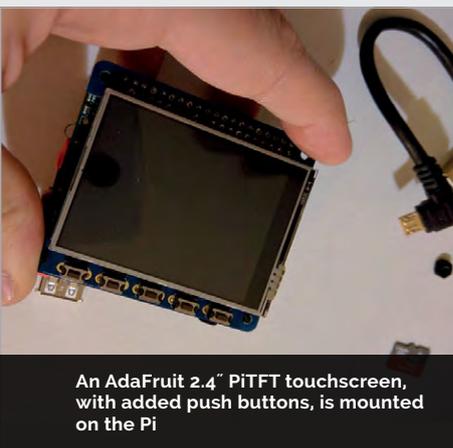
“ I can potentially put a bigger screen on and have multiple operating systems selectable ”

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

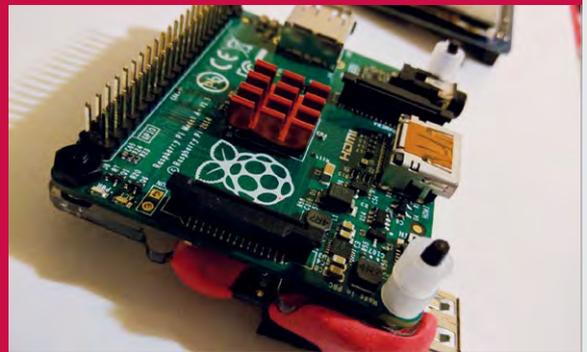
Pi Zero and it would give a bit of a speed boost... I am thinking of going the other way, though, and using a Pi 3 in future.” Although this would be even bulkier, it would run a lot faster as the Pi 3 handles emulation far better. “I can potentially put a bigger screen on and have multiple operating systems selectable from the buttons, such as 95, XP, and even something out there like ReactOS or some random Linux distro.”

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 1000 mAh 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**

Formerly a waiter at his father's restaurant, he is now a Platform Reliability Engineer. [magpi.cc/2mFCdZB](https://magpi.cc/2mFCdZB)



A Camera Module scans the printer order as it comes through. This scan is used to provide a backup

The Raspberry Pi, housed inside the box, hosts an Apache web server to provide web access

The Pipsta printer is used to print out the order. The kitchen staff tear off the order and cook the food

**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

# PIORDER

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

“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](https://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](https://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.

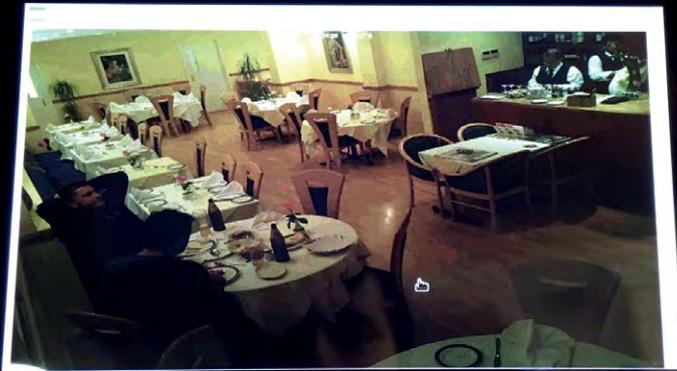
In the kitchen is a Raspberry Pi board hosting the Apache website. A program written in PHP and HTML is used to provide the webpages.

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

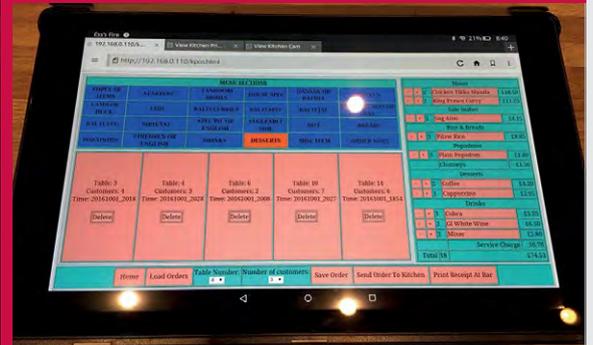
Chefs are alerted to new orders via a speaker attached to the Raspberry Pi.

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



# 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.

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 their 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



## >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.

“ 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.

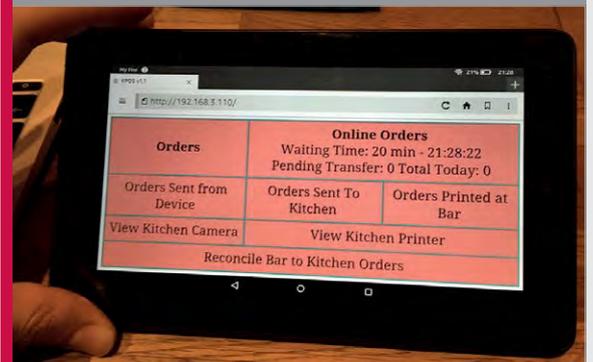
The other waiting staff and Ehsan’s father have completely stopped using pen and paper.

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.



## >STEP-03

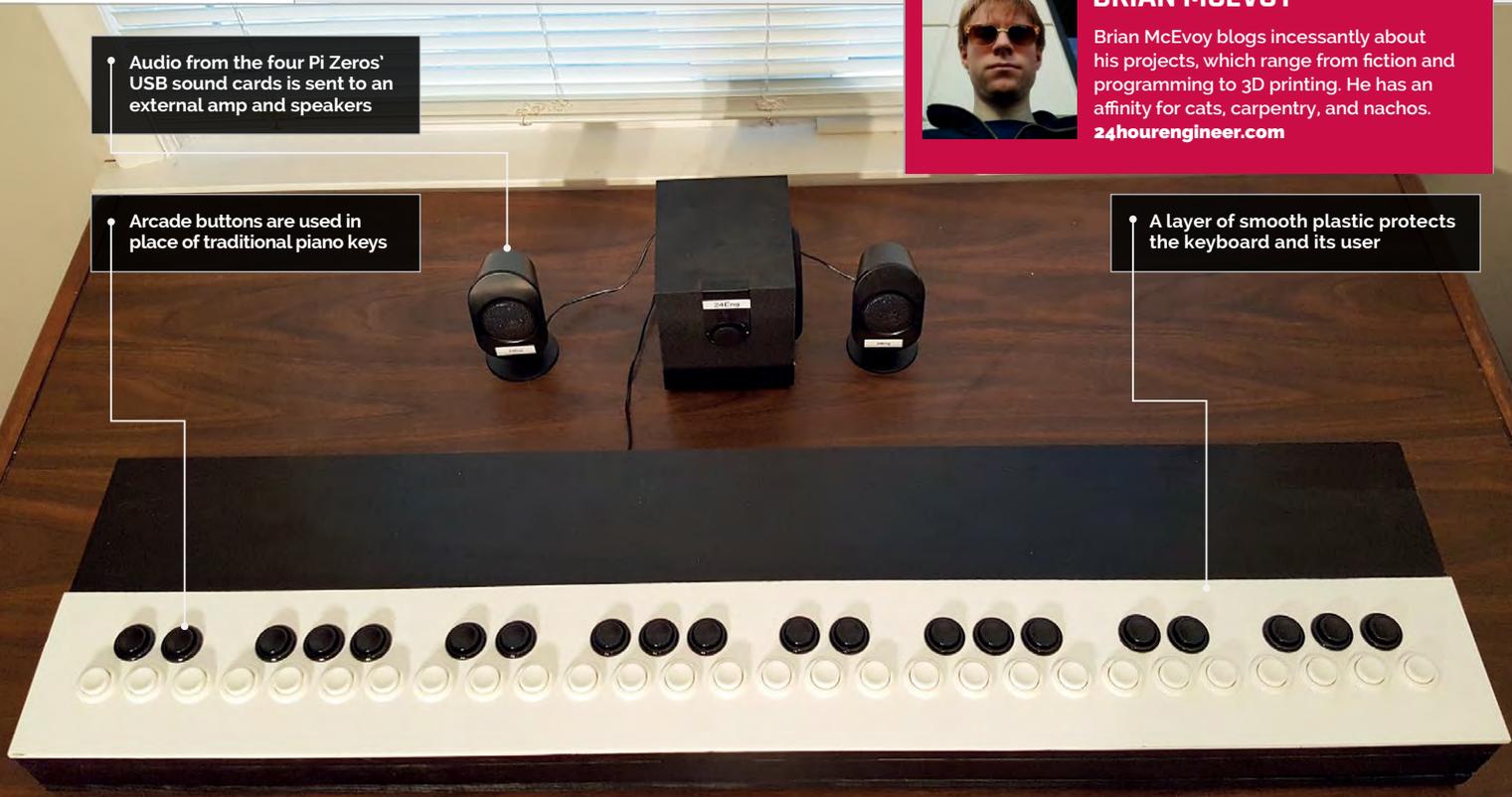
### Remote management

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](http://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

**B**rian 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



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.

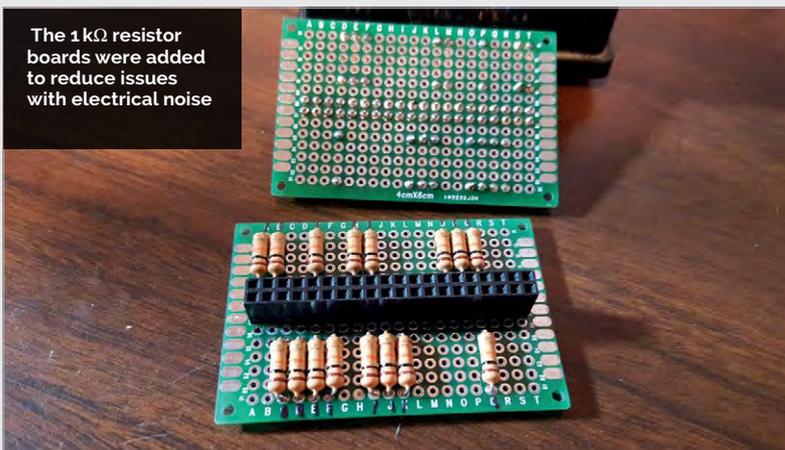
Brian wrote a Python program using the Pygame library to read the arcade button presses and play WAV piano samples. While he considered adding a 3.5mm jack to each Pi Zero,

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

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

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.”

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



## MAKE A TOUGH PIANO



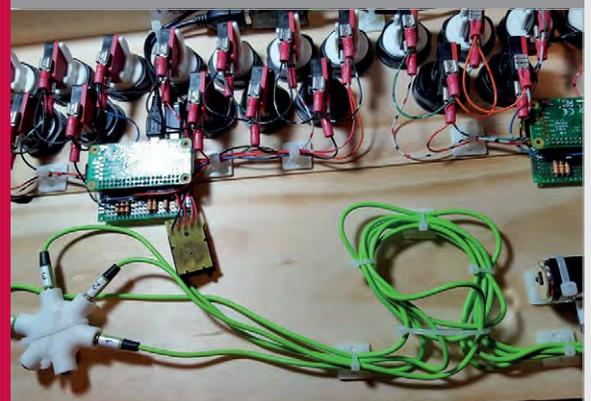
### >STEP-01 Wooden chassis

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.



**JOEY MEYER**

Joey is an experienced software engineer with an extensive background building iOS apps, and a deep understanding of the fundamental concepts in machine learning. [raspberryturk.com](http://raspberryturk.com)



The arm lowers a beam with an electromagnet to pick up pieces by their metallic tops

Made from Acrobotics, the robotic arm can move freely horizontally while maintaining a fixed height

The pieces are brightly coloured to aid identification in the camera view using OpenCV algorithms

**Quick Facts**

- ▶ The Turk takes 20–40 seconds to make a move
- ▶ It makes use of the Stockfish chess engine
- ▶ The chessboard is spray-painted onto the table
- ▶ Three 100 W LED floodlights illuminate the board
- ▶ A daemon process on the Pi handles all the software

# RASPBERRY TURK

The amazing chess-playing robot with a Raspberry Pi hidden inside

The chess player ponders the next move. Suddenly, a mechanical arm whirs into action, moves over the board, lowers an electromagnet, and picks up a piece... Checkmate! Joey Meyer’s Raspberry Turk ([raspberryturk.com](http://raspberryturk.com)) is an ingenious chess-playing robot that was inspired by the eighteenth-century ‘Mechanical Turk’. While the latter machine had a human player concealed inside to determine its moves, the Raspberry Turk uses a Raspberry Pi 3 as its brain.

“My co-worker introduced me to the 18th century Turk years ago

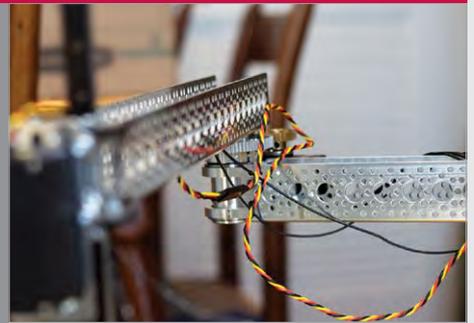
and I was always fascinated by it,” Joey tells us. “A couple years ago I read *The Turk: The Life and Times of the Famous Eighteenth-Century Chess-Playing Machine* by Tom Standage, and loved it. After spending time learning computer vision and machine learning last year, I began looking for a project that would allow me to use what I had learnt. I made the connection and decided it would be a fun and challenging project.”

Joey says the hardware was the hardest part of the project. “I am a software engineer, and building



The gripper mechanism works by activating an electromagnet at the end of a vertically moving beam

# BUILD A CHESS ROBOT



## >STEP-01 Make a table

The Turk is built into a small 3×3 ft (91×91 cm) table. A box on one side houses all the electronics, while the robotic arm is mounted on top.

## >STEP-02 View the board

To evaluate the positions of the pieces, a top-mounted Pi Camera Module captures a view of the board which is then perspective-transformed using OpenCV.

## >STEP-03 Move the arm

The arm's motion is controlled by the rotation of two servos attached to gears at the base of each link. Another servo controls the gripper mechanism.

hardware is a very different process.” One difficulty encountered was in interfacing Dynamixel servos with the Acrobotics components that make up the robot arm. “This gave me the opportunity to use 3D printing to build components to solve this problem.”

I started this so I could use material I had learnt in a real project, but documenting the build process gave me the opportunity to help others learn, too.”

Asked how difficult would it be for other makers to replicate, Joey replies: “If you’re comfortable with

The code I wrote for mine is freely available. The website explains how everything works in detail and I am happy to answer questions for anyone who wants to take on the challenge. Several people have already reached out, telling me that they are working on building their own!”

While the Turk hasn’t been showcased in public at the time of writing, Joey says the response of those who have seen it has been interesting. “Reactions are usually positive, but then they quickly change to shock when they realise the robot isn’t just playing them, it’s beating them – badly!”

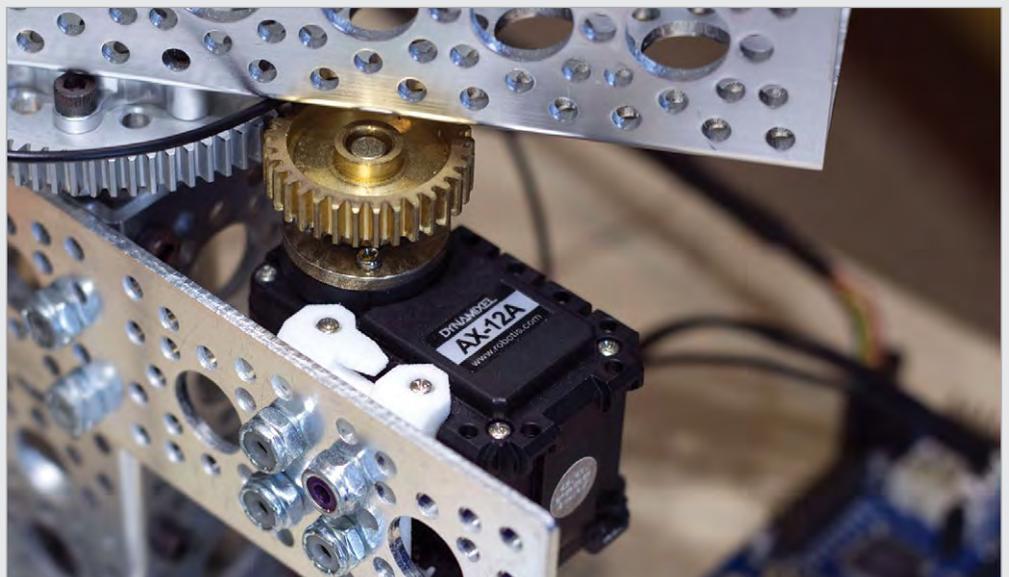
## “ Another challenge was making the arm movements precise ”

Another challenge was making the arm movements precise. “In a perfect system, the movement of the arm could be modelled by a simple math equation, but due to inaccuracies in measurements, and unexpected real-world effects, this simple math equation model broke down. It did well, but not well enough to consistently grab the piece every time.” Joey solved the issue by collecting a dataset of arm movements to see where the model was having problems. “The results worked well and the arm can move very accurately now.”

The Raspberry Turk took Joey around five months to build and the process has been fully documented on his site. “I knew I wanted to open-source the robot and describe the build process on a website from the beginning.

some electronics, programming, math, and some simple handiwork, this project would be a big challenge, but is definitely doable.

**Below** The arm’s Dynamixel AX-12A servos are controlled by the Pi via an ArbotiX-M Robocontroller





**PIET RULLENS JR**

Engineer and business owner by day; Daily Prophet reporter Rita Skeeter by night. [magpi.cc/2qvNA8i](http://magpi.cc/2qvNA8i)

# HARRY POTTER & THE DAILY PROPHET

When muggle newspapers simply weren't engaging enough, **Piet Rullens Jr** decided to create an animated fan build of the iconic Harry Potter tabloid

**Quick Facts**

- ▶ The Wizarding World of Harry Potter is in Orlando, Florida
- ▶ Piet took the idea from the moving tabloid in the books and movies
- ▶ A hidden lead keeps the frame powered and ready for action
- ▶ A sepia effect gives the video an added layer of authenticity
- ▶ Despite popular belief, Neville was the chosen one, not Harry

**W**hen Piet and Linda Rullens took a trip to The Wizarding World of Harry Potter in Orlando, Florida, they made sure to bring back a memory card's worth of holiday footage. But what do you do with holiday video footage once you're home? Taking his inspiration from the fictional world that forged the destination for their vacation, Piet

decided to create his own Daily Prophet newspaper, complete with moving images.

For those unfamiliar with Harry Potter, the Daily Prophet is the main newspaper publication of the wizarding world – the only rival being the often fantastical Quibbler peddled by Luna Lovegood and her father. Similar in function to that of a 'muggle', non-magical newspaper,

the Daily Prophet shares the headlines of the world – with one major difference... the images move. Imagine using an animated GIF in a news blog, but on paper.

With his videos to hand, Piet set about creating the newspaper by designing the front cover in Adobe Photoshop. Not only did this enable him to include personal references in the copy, such as mention of himself and his wife being spotted at the theme park, but it also allowed him to create the perfect-sized window for the Raspberry Pi 7-inch display that he was to fit within the frame. "First, I designed the whole poster in Photoshop. Within the design, I marked an area with the exact size of the Raspberry Pi screen. Next, I plotted the poster on normal paper at 100 percent, so the marked area still matched the Raspberry Pi screen."

From there, Piet marked out the measurements of the screen onto the hardboard of a poster frame, giving him a guide to cut through for the additional electronics.

Luckily for Piet, on the other side of the wall from where he would be hanging the frame was a small cupboard. He was able to drill directly through the wall, hiding any wires from view, and adding to the magical illusion of the piece.

With on-board wireless connectivity on the Pi, the only wire needed was the USB power cable.



Piet designed the paper in Photoshop before printing it to his exact specifications

The Raspberry Pi 7-inch screen was the perfect size to act as the lead headline photograph

The wiring is kept out of view to add to the illusion of the magic newspaper

CODING MAGIC



**Above** Both the glass of the frame and the bevel of the screen keep it fitted snugly within the newspaper, with no need for glue or screws

With this firmly in place inside the cupboard, Piet was able to remotely access the Pi and create the code to run his holiday footage.

Piet created a simple Python script with two functions. The first detects the presence of someone

photo frame. The motion sensor then triggers Omxplayer to play five minutes of footage before turning the screen off again.

To add to the look and feel of the newspaper, Piet edited the footage to give it a grainier, sepia

“ Within the design, I marked an area with the exact size of the Raspberry Pi screen ”

passing by, and the second then runs the holiday footage on the screen. To complete the first task, he used an IR distance sensor from Adafruit. This would detect motion within a set range around the

tone in line with the movie prop. He converted the footage to H.264 so that it played through the Raspberry Pi, creating a beautifully executed and impressive magical holiday souvenir.



Added texture and colour effects give the video the appearance of a Daily Prophet image



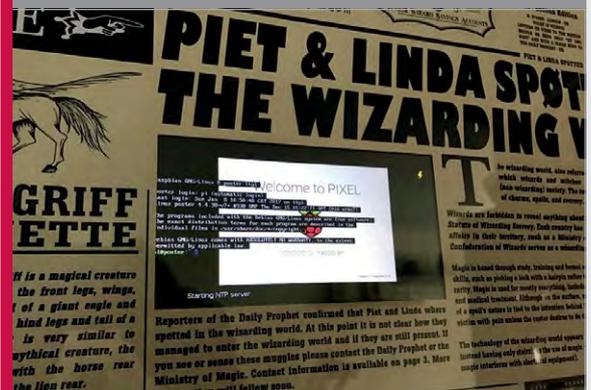
>STEP-01 Behind the scenes

The Raspberry Pi, screen, and wiring fit perfectly inside the recess of the frame, with a hole in the wall leading to a power supply on the other side.



>STEP-02 Look, no wires!

By cleverly diverting the power cable through the wall, Piet adds an extra level to the magical illusion of the piece.



>STEP-03 Spotted!

The Python script and an IR sensor control the screen and the duration of the video: approach the frame and the system begins to play.



**INGMAR STAPEL**

Ingmar has been building Pi-powered robot cars since 2012. He's also working on a particulate matter sensor.  
[custom-build-robots.com](http://custom-build-robots.com)



# SATNAV FOR ROBOTS

Guide robots with incredible accuracy using this differential GPS system

**Quick Facts**

- ▶ The differential GPS system is accurate to 20 cm
- ▶ The robot uses three Python programs
- ▶ €6 IKEA lunch boxes protect the base stations
- ▶ Big Rob has a 12V 30Ah battery
- ▶ See videos of Big Rob at [magpi.cc/](http://magpi.cc/) [zrPFenz](https://www.youtube.com/channel/UCzrPFenz)

**W**hile building a rugged robot, named Big Rob, for outdoor use, Ingmar Stapel wanted to create a system for precise navigation. Disappointed with the accuracy of the standard USB GPS modules used on his previous robots, he decided to build a navigation system based on differential GPS.

Differential GPS uses the RTK (Real Time Kinematic) method for carrier-based ranging between a base station and the robot. Data sent from the base station

enables the robot to correct the inaccuracies in its own GPS signal, mainly caused by atmospheric effects, in order to calculate its precise position. This technique produces an accuracy of 20 cm, compared with 4–5 m for standard GPS.

Both units are equipped with a RasPiGNSS Aldebaran module from [drfasching.com](http://drfasching.com): “It is the only one I know which is especially built for the Raspberry Pi,” explains Ingmar. They are also fitted with a Tallysman GNSS antenna that

can receive not just GPS signals, but also those from the Russian GLONASS system, as well as SBAS systems such as Europe’s EGNOS.

**XBee or WiFi?**

To minimise GPS inaccuracies, Ingmar is using the RTKLIB Python library ([rtklib.com](http://rtklib.com)) to perform the RTK calculations. “The most complex part of this project was to configure the RTKLIB library and to set up the XBee communication between the base station and the robot.” The main advantage to using XBee Pro rather than standard WiFi is its superior range of up to 1.6 km. However, Ingmar has also tested the system using WiFi, which offers a higher bandwidth. Both versions of the RTKRCV program are described on his blog.



The base station sends its own position to the robot to eliminate GPS inaccuracies caused by atmospheric effects

The two units are in constant communication via XBee modules or standard WiFi

The robot calculates its precise position using differential GPS, and can direct itself to various waypoints

# BUILD A GPS SYSTEM



## >STEP-01 GPS modules

Both the robot and the base station feature a Raspberry Pi equipped with a RasPiGNSS Aldebaran module and Tallysman antenna to receive satellite signals.



## >STEP-02 XBee communication

XBee Pro modules, connected to Raspberry Pis via XBee Explorer boards, are used for long-range data communication between the robot and base station units.



## >STEP-03 Big Rob

Ingmar has equipped his Big Rob robot with the differential GPS system, along with a Sense HAT at the rear to aid orientation when navigating between waypoints.

## Robotic navigation

So, how does it all work in practice? Ingmar admits that it can take a while to obtain a GPS fix, depending on the surroundings and weather. "If I use the setup in open country with no clouds, the calculation of a fix position takes between 10 and 15 minutes. If I use it in the yard with high buildings around the base station and robot, it could take up to an hour to calculate only a floating GPS position."

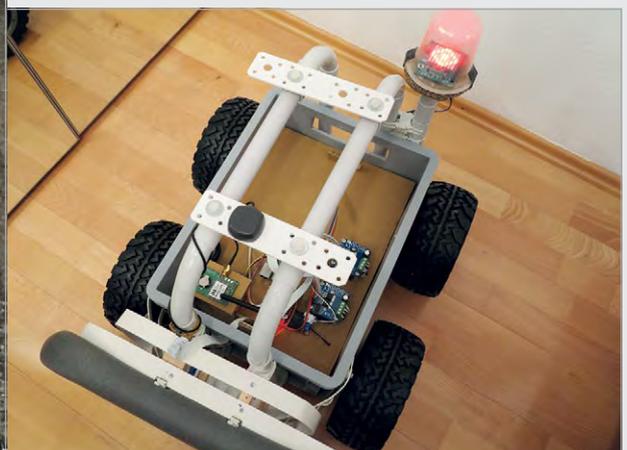
Once it obtains a fix, the robot moves continuously from one waypoint to the next, checking its orientation with the Sense HAT magnetometer: "If a deviation of five degrees is detected to the next waypoint, the Python program corrects the speed of the DC motors to face the waypoint. Only if the GPS signal gets lost will the robot stop and wait for the next calculated fix position." Waypoints can be entered in a Terminal window or imported via a Google Earth KML file.

Ingmar has a few ideas for putting his differential GPS system to practical use. "I will mount the metal detector (as used on his Discoverer robot) on the front of Big Rob, and together with the differential GPS setup, the robot will be able to search very precisely for metal in the ground. The next idea is to mount, for example, a lawnmower on the front or to build a fertilizer spreader. The differential GPS setup is a key feature for many ideas and setups."



**Left** The path of a test drive, showing the precision of the system with a floating solution from the RTK calculation

**Below** An early version of Big Rob showing the GPS system components, along with an illuminated Sense HAT at the rear





**ALVIN GROEN**

Alvin is a director, designer, and animator from The Netherlands. Currently, he lives in New York. [alvingroen.com](http://alvingroen.com)

# TELE2

How a Raspberry Pi kit helps people to live the life of a complete stranger

**Quick Facts**

- ▶ The Tele2 kit was used in five different countries
- ▶ The idea came from the movie *Being John Malkovich*
- ▶ 36 million people were made aware of the project
- ▶ A custom iOS app was used to control the data stream
- ▶ The equipment was in use for more than two months

**W**e have all imagined what it would be like to be somebody else, even if just for a short time.

Tele2 is one of the more intriguing projects we've come across. It was a custom Raspberry Pi electronics kit worn by various interesting people from around the world. Each person wore a GoPro Hero 5 and Røde Video Microphone to capture a first-person view of what they were seeing and hearing.

More intriguingly, the system collected data on each personality's heart rate and sweat level.

All of this data was used to measure the emotional state of the person, which was relayed to the viewers using visual effects layered on top of the audio and video. Brain sensor (EEG) data was translated into warmer or cooler video filters to represent the person's emotions.

We caught up with Alvin Groen, the director and designer of the Tele2 project, who dreamt of building a piece of equipment that "changes our perspective."

The Tele2 project is named after the Swedish telecom company that sponsored the project. "They were launching a new 100GB data plan and wanted a project that required high data bandwidth," recalls Alvin. With funding underway, the Tele2 project was born.

The Teradek VidiU enables the kit to stream live video to the web via the Tele2 modem

The Galvanic Skin Response Sensor, Polar Bluetooth HR monitor, and Muse headband all gather data on the emotional state of the wearer

An e-Health v2.0 board is mounted above a Raspberry Pi 3. The Pi collects the video and audio data, as well as the health measurements





“In Sweden, a lot of discussion around connectivity tends to be negative,” says Alvin, “especially when it comes to controlling our exposure to media.”

Alvin wanted to alter the outlook people had on other lives. “Each person was chosen because they have a big following online, or they have an interesting life,” explains Alvin. They included Simone Giertz, a robot maker from Stockholm, Joel Kinnaman, an

**Above** The kit recorded everything a person saw and heard while measuring their mood levels

processed all the video, audio, and sensor data.

Camera footage went to a colour grading box in the backpack. This was “fine-tuned for each location”, and the video and audio was streamed live via a Teradek VidiU wireless HDMI video encoder.

The built-in wireless features of the Raspberry Pi 3 made it a good

“ They were launching a new 100GB data plan and wanted a project that required high data bandwidth ”

actor from Vancouver (and star of the *RoboCop* reboot), and Noor Daoud, a female drift racer from Dubai. An impressive collection of folk.

Each of the personalities was hooked up to the Tele2 equipment. The team broadcasted nine live streams in five countries during a two-month world tour. “We measured the personalities’ heart rate, emotions, and sweat level,” using a Skin Response Sensor and Muse headband ([choosemuse.com](http://choosemuse.com)). Then, using WebGL and web audio technology, they translated the data live into audiovisual effects.

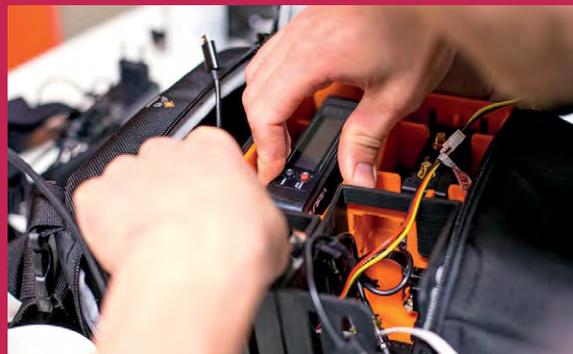
The team built a custom backpack using a 3D printer for each personality. The rig included a Raspberry Pi hooked up to an e-Health v2.0 board ([magpi.cc/2qnMyry](http://magpi.cc/2qnMyry)). The Raspberry Pi collected and

fit for the project. “The built-in Bluetooth chipset, wireless networking, and the fact that you can use a pretty standard flavour of Linux right out of the box made it very easy to work with,” says Alvin.

“Since all the other equipment was rather bulky, size wasn’t an issue either,” he adds. “The toughest challenge was trying to figure out if we could get the Muse headband to communicate directly with the Raspberry Pi, since the official Muse SDK wasn’t released for the ARM architecture.”

The project was a huge success, with hundreds of thousands of people tuning in to see what it’s like to be somebody else. Viewers could click on cheers and give feedback to the wearers. Overall, Tele2 was a very interesting combination of health sensors and live video streaming.

## WEARABLE



### >STEP-01

#### Build a backpack

The Raspberry Pi, e-Health shield, video adapter, wireless modem, and battery pack were carried by the wearers in a custom-built backpack.



### >STEP-02

#### Sight and sound

The GoPro Hero 5 was mounted on the wearer’s head, along with the various health sensors. Everything they looked at and heard was streamed live.



### >STEP-03

#### World in colour

A filter was laid over the recorded video based on four mood characteristics: Angry, Happy, Sad, and Calm. This filtered video was streamed live to the web.



**ERIC PAGE**

Eric enjoys building things for his fiancée, Shirin, and their dog, Pickles. When he's not DIYing, you'll probably find him backpacking or running an accounting business.  
[magpi.cc/zsUurv1](http://magpi.cc/zsUurv1)



- The cylinder is divided into sections for storage and dispensing
- A stepper motor rotates the inner cylinder to dispense the treats
- The Camera Module is used to detect motion and capture photos and video of the dog

# DOG TREAT MACHINE

With motion detection and video capture, is this the ultimate dog treat dispenser?

**Quick Facts**

- ▶ Build details are on Instructables [magpi.cc/zsTf5qU](http://magpi.cc/zsTf5qU)
- ▶ There are four ways to trigger the device
- ▶ It can dispense treats of different sizes
- ▶ Eric has added a ball launcher
- ▶ He intends to extend it further

**W**hile pondering what to get Shirin, his fiancée, for her birthday, Eric Page realised how much she misses her dog Pickles while she's at work. He decided to build a device to allow her to serve treats to Pickles remotely. "She uses it almost every day! Usually once a day when she is at her office. When she's travelling, she probably sends him three or four treats every day. I think it helps her feel closer to home."

There are four ways to trigger the Pi-powered device: email, MQTT, IFTTT, or Alexa. It then plays a sample of Shirin's voice, dispenses some dog treats, and checks for motion using OpenCV

analysis of the view from a Raspberry Pi Camera Module. If motion is detected, it captures a photo and video of Pickles, which are sent to Shirin in an email with a randomised cute comment. If no motion is detected, it sends an 'I'm not home but will enjoy the treats later' message, along with a couple of photos of Pickles playing outside.

Eric says motion detection was an important addition. "I didn't want to send my girlfriend a video of a floor full of treats. Even if Pickles was home, if he walked over to the treats more quickly or slowly than expected, the video would be blank for a period of time. So I built a motion detection

system using really good code from Adrian at PyImageSearch. This ensures that the video always starts with Pickles entering the area, and Shirin can see him trotting up with excitement."

Far from being a seasoned maker, Eric reveals "this was my first attempt at anything DIY in probably 25-plus years since junior high, so it took some trial and error." After starting the project with an Arduino, he soon realised that it lacked the processing power and storage required for video and other complexities. "After I switched to the Pi, it took 30-45 days to get to a version that would reliably deliver treats via email. After that, I periodically tinkered with

# MAKE A DOG TREAT DISPENSER

Eric reckons Pickles thinks of the device as "some sort of God of Treats. Sometimes, we'll see him just sitting in front of it, looking up at it with a mixture of awe and delight"

**Below** The ingredients for the perfect dog treat dispenser include a Pi, Camera Module laser-cut tubes and stand, and a stepper motor



## >STEP-01

### Cutting the parts

Eric used a laser cutter at a local makerspace to create the plastic stand and cylinders. The inner cylinder is divided into two compartments; one for storage and one for dispensing.



## >STEP-02

### Dispensing mechanism

To make sure the desired number of treats falls into the dispenser compartment, acrylic semicircles reduce the size of the opening. One of these is attached by Velcro and can be adjusted.

it for four to five months, adding additional delivery mechanisms and improving the reliability of the code, e.g. reconnecting everything if the WiFi went down."

Eric says the most difficult part of the project was getting the

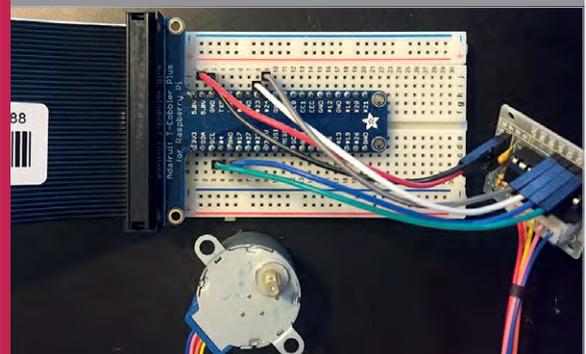
opening to a smaller section that holds and dispenses one serving."

Following the success of the treat dispenser, Eric has extended the project by building a ball launcher using RFID-tagged balls, which he's trained Pickles to

“ Now Pickles can get his own treats by putting those balls in the receptacle. He loves it ”

device to deliver a small number of treats (not zero or too many) by gravity from its plastic cylinder, when rotated by a stepper motor. "The design went through several iterations before I hit on the concept of a cylinder with a central storage area, with an adjustable

retrieve and place in a receptacle equipped with an RFID reader. "Now Pickles can get his own treats by putting those balls in the receptacle. He loves it. Now, he even sits by the ball launcher making funny noises – his way of saying 'Dad, launch a ball!'"



## >STEP-03

### Stepper motor

A stepper motor with driver board is used to rotate the cylinder, causing the treats to fall out of the opening, while allowing more treats to drop into the dispenser compartment.



**PETER BARKER**

A full-time software developer, part-time hacker, with a passion for making fun little projects to get a Pi in his pocket.

@PioCKET / github.com/moosepr

# PIOCKET-TINY PI

Possibly the world's smallest Pi-based gaming device

**Quick Facts**

- > Tiny Pi took two months to create
- > It runs for 3.5 hours on a 1000 mAh battery
- > Peter is now selling Tiny Pi kits...
- > ...so you can build your own tiny console
- > He'd like to create a Pi pocket computer

**P**eter Barker was inspired to start making miniature gaming devices following the release of the Pi Zero, and seeing other makers cramming it inside an old Game Boy. "After ordering all the parts, I initially wanted to do the same, but couldn't bear to take a working Game Boy apart."

Instead, he created his first standalone Pi Zero-based gaming device using a cheap 2.2-inch SPI

screen flanked by two stripboard button arrays. "Things progressed from there, really," recalls Peter. "I didn't like how big the circuit board was for the screen, and realised that there weren't all that many bits to it, so I started designing my own."

This resulted in his diminutive Game Boy Zero device featuring a 2.2-inch display on a custom PCB, "which was pretty much as

small as I thought I could go." However, Peter then found a smaller, 1.44-inch TFT screen that perfectly matched the 128x128 resolution of the Pico-8 fantasy games console ([magpi.cc/2rCswv1](http://magpi.cc/2rCswv1)) that he'd stumbled across after looking at the PocketCHIP handheld computer. "It took me back to the good old days of the [Sinclair] Spectrum, where the system was small enough that



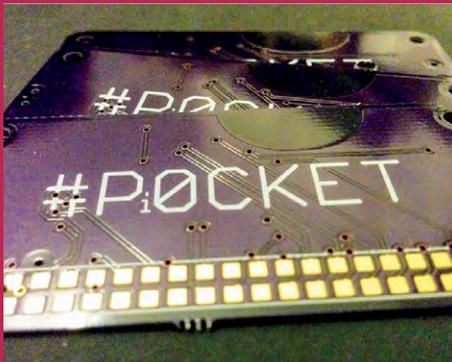
A three-way switch can be mapped to extra functions

Tiny Pi's custom-designed PCB is just 0.8mm thick

The 1.44 inch TFT screen's 128x128 resolution is a perfect match for Pico-8 games



# BUILD A TINY GAMES MACHINE



## >STEP-01 Make a custom PCB

To keep the size down, Peter designed his own slimline printed circuit board and had it made by OSH Park, a US-based custom PCB supplier.



## >STEP-02 Add a Pi Zero

The Pi Zero is mounted directly to the rear of the PCB, its GPIO header soldered to the connections. A LiPo battery and TP4056 charger are strapped in place.



## >STEP-03 Make a case

Peter designed his own slimline case using OnShape. It comprises a top, bottom, and backplate, and covers for the joystick and buttons. Find the files at [magpi.cc/zqu0P6f](http://magpi.cc/zqu0P6f).

kids in their bedrooms could make brilliant games. I tried it on my GBZ project and it was good fun to have a portable system that could play all these games, almost like having a part of my childhood in my pocket.”

The PiOCKET-Tiny Pi ([magpi.cc/2sdhmf8](http://magpi.cc/2sdhmf8)) took him two months to put together. “The biggest hold up was getting my hands on some screens. There

by two piezo transducers situated behind the screen.

As well as Pico-8 software, Peter has programmed the Tiny Pi to run classic console games via the RetroPie emulation system. “You are restricted on the games you can play. The 128×128 resolution makes things like SNES or Mega Drive games hard to see, although the Pi is actually capable of playing them.”

“ As well as Pico-8 software, Peter has programmed the Tiny Pi to run classic console games ”

are many suppliers who sell them attached to a PCB, but I didn’t want that extra bulk.” Instead, he designed his own super-skinny PCB, a mere 0.8 mm thick, which is mounted directly to the rear of a Pi Zero W.

On the front of the PCB are a five-way joystick (with push-in function) and two push buttons. There’s also a three-way navigation switch at the top, which can be mapped to extra functions. The device is powered by a slimline LiPo battery, strapped to the Pi Zero along with a TP4056 charger. Stereo sound is supplied

The original version of the Tiny Pi was ‘naked’, but Peter has since clothed it in a 3D-printed custom case. Designed using OnShape ([onshape.com](http://onshape.com)), he’s made it as slimline as possible, to match the rest of the project. “It needs a spot of refinement, but the main change I would like to make would be to the software side of things. My real goal is to have it like a Game Boy, where it starts up quickly, you can play a few games, then just switch it off, no worries about shutting down and waiting for it to finish things.”



**Above** You can get an idea of just how tiny the Tiny Pi is when it’s placed next to a Nintendo Game Boy  
**Above top** With RetroPie installed, the Tiny Pi can emulate classic games machines such as the Sega Game Gear



**JACK BARKER**

Jack is based in Melbourne, Australia. By day he works as a business analyst; by night he writes code, blogs, and muses on all things technology. [jackbarker.com.au](http://jackbarker.com.au)

**Quick Facts**

- Build details are on Jack's blog ([magpi.cc/2s07JZo](http://magpi.cc/2s07JZo))
- The booth took 20 hours to make
- By default it takes four photos per session
- Live internet backup of photos is possible with a WiFi connection
- External lights would improve night-time shots

# PHOTO BOOTH

Need a wedding photo booth? You could always build your own...

**A**s first Raspberry Pi projects go, building a photo booth for your own wedding is an ambitious and potentially stressful one, but that's just what Jack Barker did for his and fiancée Pam's big day. "I've always been fascinated with tinkering and programming but, in truth, the primary motivation came when I saw how much it cost to hire a photo booth!" After

getting quotes upwards of A\$500 (£300), and struggling to stay within their wedding budget, Jack realised he could build his own booth for considerably less money.

"When I first mentioned the idea to Pam, she was a bit nervous that it might be another project that wouldn't eventuate!" admits Jack. The booth took him 20 hours to complete, spread across a number of weekends.

absolutely essential, including a printer to produce instant copies of the photos.

Since the venue had no WiFi connection, Jack's plan to run a live backup of photos to the internet also had to be shelved. "My low-tech solution was to simply write the images to the Pi's SD card (and hope for the best). The following day I was very glad to see that they had all been saved without issue,

"As the wedding date drew closer, Jack had to de-scope some ideas that weren't absolutely essential"

This included writing all the code from scratch – available on his GitHub page ([magpi.cc/2sNsRic](http://magpi.cc/2sNsRic)). "Certainly a lot of that time was spent trying out different ideas, and deciding on which features to include." As the wedding date drew closer, Jack had to de-scope some ideas that weren't

and I was then able to post them online for the guests to view."

Building the main booth structure from plywood, Jack cut out a section for the LCD screen and drilled holes for a large arcade button below and the Raspberry Pi Camera Module above. Jack used a 3D-printed part to align the



Jack tested the screen and camera setup thoroughly before inserting it into the wooden cabinet





Image credit: Jonathan Lang Photography

The happy couple, Jack and Pam, trying out the homemade photo booth

camera perfectly with the hole, but he says this is optional. The Pi itself was also taped to the rear of the front panel. For improved lighting of indoor shots, Jack placed a strip of white LEDs across the top, later adding a simple light diffuser using an angled piece of wood.

Jack set up the Python software so that when a guest presses the button, a seven-second countdown is started. “During the first four seconds of the countdown, an instructional slide is shown (‘Get ready for photo 1’). For the final moments of the countdown, the participants

can see themselves live on the screen and they can also see the timer counting down. This helps the participants to position themselves in front of the shot, before the photo image is taken.” Once the first photo is taken, the cycle repeats to capture a total of four images.

So, how did it work out on the big day? “The guests were impressed that I’d been able to build it myself,” says Jack. “It served as a great topic of conversation, and helped get some of the guests mingling. Oh, and yes, there has been interest from people asking to borrow it.”

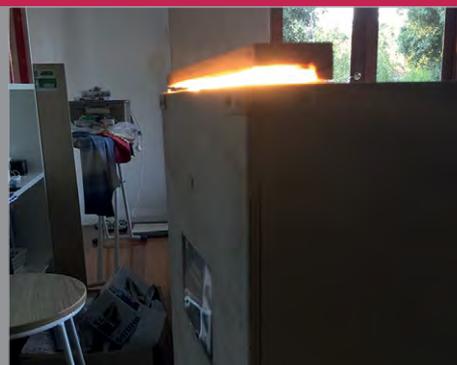


## BUILD A PHOTO BOOTH



### >STEP-01 Wooden cabinet

Jack used pieces of plywood joined with brass brackets to make the photo booth’s cabinet, sawing out a carefully measured section of the front panel to fit the LCD screen.



### >STEP-02 LED light diffuser

An angled piece of wood is used to diffuse the light from a strip of white LEDs. Rather than acting as a flash, Jack opted to keep the LEDs turned on, which avoided tricky timing issues.



### >STEP-03 Wire it up

Currently, Jack is using three separate power sources – one each for the Pi, LCD screen, and LEDs. However, he’s looking to switch over to a dual rail power supply (5V plus 12V) soon.



**STEVE UPTON**

His parents would say he was breaker, but today he's more of a maker. Founder of Makespace (Cambridge) and creator of lots of amazing projects. [ste5eu.com](http://ste5eu.com) / [@ste5eu](https://twitter.com/ste5eu)

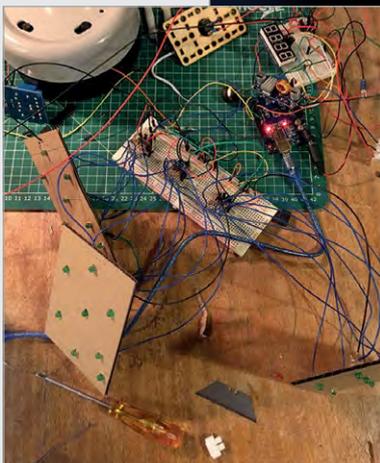
# PI BASH

When Steve Upton was challenged to have a bash at creating a Hi Striker machine, he hit the ground running and ended up developing two

**Quick Facts**

- ▶ It took Steve 24 hours to make
- ▶ It cost £174 in parts from RS, Pi Hut, and Pimoroni
- ▶ The current highest scorer is a blacksmith
- ▶ The frame was designed using Autodesk's Fusion 360
- ▶ A refined version was produced for a Maker Faire

**Below** Steve built a full working version, adding an element at a time. This image is an early prototype



**I**f you've ever been to a fairground and been asked to 'step right up' and 'test your strength', then you may have found yourself smashing a hammer against a puck in an attempt to ring a bell. These so-called Hi Striker attractions have been popular

for close to 100 years, but when Steve Upton based one around an Arduino, he didn't realise he'd soon have a Pi-based hit on his hands.

The idea for building the gaming machine came from a company called AJW Distribution, which had approached the members

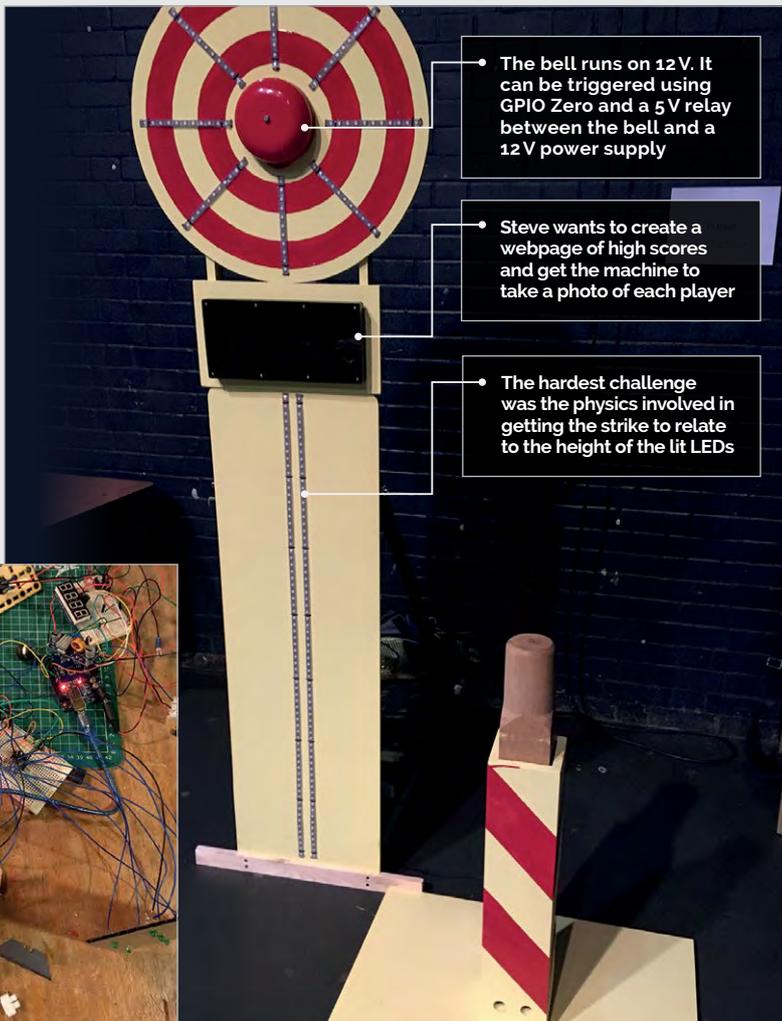
of Cambridge's inventing shed, Makespace. The firm wanted something to entice people to its stand at a forthcoming trade show. "They wanted it to look like the side of a building, and use their cladding and roofing material," says Steve, who agreed to the challenge.

**New and improved**

Having created the machine, which made lots of cash for charity, Steve decided to play around with his design. "I knew it would be great for Raspberry Pi parties, Jams, and Maker Faires," he says. "So I just had to create another improved version based on what I'd learnt." He was going to call it Pi Striker, "but I couldn't write an 'r' with the displays I intended to use for the scoreboard." Instead, he named the new project Pi Bash and, grabbing a wad of paper, began to draw up his plans.

Steve knew what he needed to create the machine based on his previous experience. He decided to use the small, inexpensive plug-on RasPiO Duino board for sensing and the focused monitoring of speed. He also used a pressure sensor which could detect a hit, a vertical LED strip that would light up according to how hard the impact was, some star-shaped LED strips, and a bell with a relay controller to produce a satisfying ding if the player was successful.

He also wanted a mode selector so that the game could be



- The bell runs on 12 V. It can be triggered using GPIO Zero and a 5V relay between the bell and a 12 V power supply
- Steve wants to create a webpage of high scores and get the machine to take a photo of each player
- The hardest challenge was the physics involved in getting the strike to relate to the height of the lit LEDs

# FAIRGROUND ATTRACTION



## >STEP-01 Give it a whack

When you go for a strike, a hit triggers the vibration sensor. The Arduino calculates the power of the strike based on the amount of vibration, and converts it to a number between one and 100.



## >STEP-02 Lighting the way

The information is sent to the Raspberry Pi, which loops and counts up to this value. It triggers updates to the LED strip and the scoreboard via a GPIO port.



## >STEP-03 Ring that bell

If the measured power is enough to reach the top, the relay is triggered, again via a GPIO port. It connects the 12 volts to the bell, which then rings. Success!

adjusted to suit the player; a reset button; and the aforementioned scoreboard. “Once I had all of these working individually, I built the wooden support structure and started to add them together,” recalls Steve. “I could then tune the sensing calculations to make it feel as realistic as possible.” There wasn’t much programming involved, although the sensing code, written in C, was fairly complex.

### What are the scores?

The scoreboard was tricky to make. Rather than using a screen, Steve wanted to use three large seven-segment LEDs to make the score clear, “but I couldn’t find anything

appropriate for sale, so I built one myself.” The seven-segment displays were connected to a laser-cut Perspex mount which Steve built at Makerspace.

and sensors also proved tricky to control, so they remain connected to the Arduino at the moment.

More difficult, though, is actually achieving a high

“ I’ve got a couple of sensitivity settings so parents and youngsters can compete on a more equal footing ”

“I bought the LEDs online, along with the components to build controllers for them,” he adds, referring to the resistors, shift registers, optocouplers, and connecting wire. “This was one of the more challenging parts of the project.” The lights

score, but the reaction to the Pi Bash has been great. “I’ve got a couple of sensitivity settings so parents and youngsters can compete on a more equal footing. We had a lot of laughs at the Pi Birthday Party earlier this year.”



**CHRIS RIEBSCHLAGER**

Chris Riebschlager works as Lead Software Developer at Dimensional Innovations, Kansas City, USA. [magpi.cc/zu887BS](http://magpi.cc/zu887BS)

# THE PLAYABLE 12-FOOT ELECTRIC GUITAR

Asked to create ‘something amazing’ for the entrance of Boulevardia, a two-day music festival, the team at Dimensional Innovations didn’t disappoint

## Quick Facts

- The build took five weeks in total
- The body is constructed using layers of MDF
- Stickers were printed on 3M Controltac
- It uses HDMI for sound
- Watch the build process at [magpi.cc/zudZooz](http://magpi.cc/zudZooz)

**B**oulevardia is Kansas City’s two-day music, food and beer event, drawing crowds to the historic Stockyards District. When tasked with making something ‘amazing’ to be situated just inside the entrance to the event, the team at design firm Dimensional Innovations “wanted to create something iconic, interactive, and engaging.”

Although the idea of a 12 ft (3.6 m) tall electric guitar wasn’t the first concept to grace their ideas board, it was the one that

stuck. And soon afterwards, the team got to work designing both the internal electronics and external aesthetic of the build.

“Honestly, I didn’t know if we would be able to make this happen,” explains Chris Riebschlager, Lead Software Developer at Dimensional Innovations, “but I was smitten with the idea, so I pitched it to the team.” Once the idea was green-lighted, Chris finally had an excuse to use the Bare Conductive capacitive touch board he’d been coveting as a maker, and they got to work on a prototype.

## Prototyping

The plan was to use 16-gauge galvanised wire as the conductive material connected to the board, allowing the team to assign a WAV sound file to each wire. ‘Strum’ a wire and the board would register the connection and task the Raspberry Pi with playing back a specific note. A simple enough idea, and one we’ve seen countless times using fruit and tin foil – but for a 12-foot guitar, there was a little more work to be done.

Chris originally attempted to source his guitar note WAV files

Arcade buttons connected to GPIO pins are used to select chords, and choose a clean or dirty sound

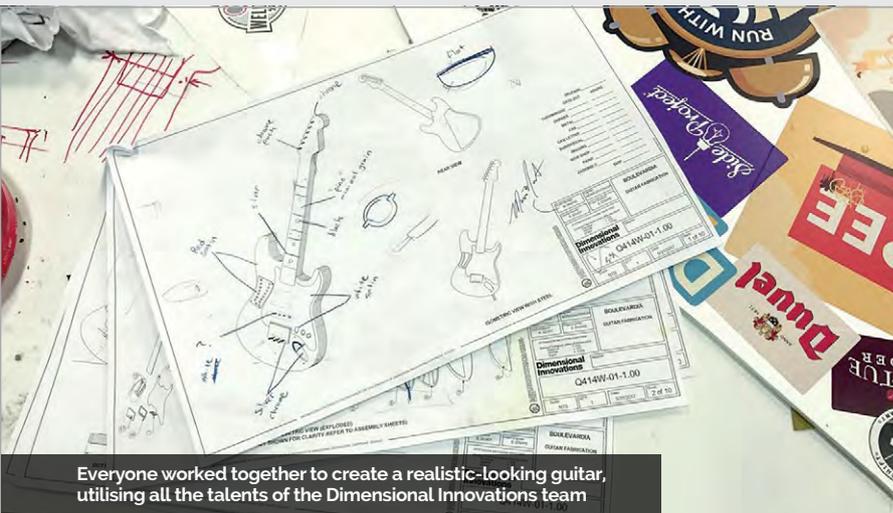
Stickers designed and printed in-house represent the breweries and bands taking part in the Boulevardia event

16-gauge galvanised strings connect to a Raspberry Pi via a Bare Conductive capacitive touch board



The final build was hugely successful, flooding social media accounts during the two-day event





Everyone worked together to create a realistic-looking guitar, utilising all the talents of the Dimensional Innovations team

from an electric guitar, but finding the sound to be “inconsistent and weird”, he decided to instead take the pre-existing sound files from GarageBand. Still not content with the prototype, the team moved on

### Building the body

Turning to the design team for help with the guitar body, the team created an authentic-looking set of stickers, each depicting breweries and bands taking part in

“ One of the most exciting, challenging, and rewarding projects I’ve been a part of ”

to incorporating arcade buttons linked to the GPIO pins, allowing the user to select a specific chord before strumming the strings. The Python code would then determine which sounds to play, and for how long, based on the user interaction with the strings and buttons.

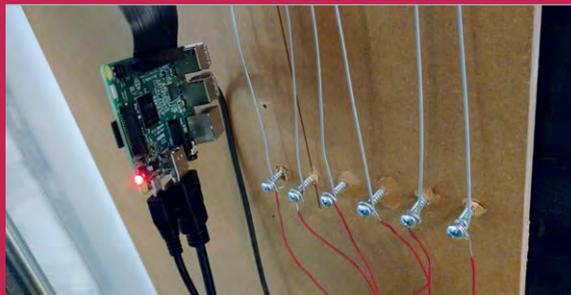


the event. These were then printed for inclusion on the MDF and steel body of the instrument, all put together in-house by the awesome engineering team.

The final piece was installed inside the entrance to Boulevardia, and received overwhelming reactions from all attendees to the event. “This project was without a doubt one of the most exciting, challenging, and rewarding projects I’ve been a part of,” continues Chris on his blog entry for the build. “From concept to execution, the entire project took only five weeks. The fact that we pulled this off in so little time really speaks to the talent and dedication of the people I get to work with.”

If you happen to be in Kansas City, you can visit the guitar at the Dimensional Innovations shop, where it sits proudly on display for visitors to play – a constant reminder of what can be achieved with a little hard work and a great team.

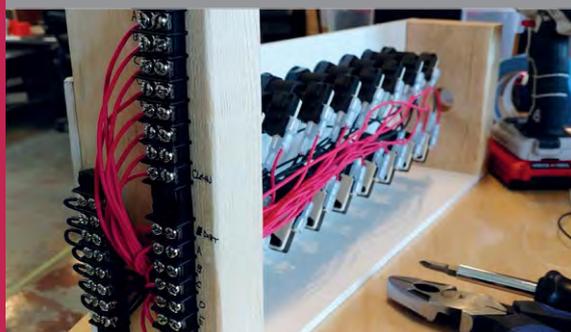
## A GIANT IN THE MUSIC INDUSTRY



### >STEP-01

#### Strings

A Bare Conductive board recognises capacitive touch via 16-gauge galvanised wires running down the body of the guitar. Each touch triggers the playback of a specific WAV file.



### >STEP-02

#### Sound

Arcade buttons connected to the GPIO pins allow users to select different chords as they play. They also allow the guitar chords to sound ‘dirty’ or ‘clean’, depending on how rock ‘n’ roll the user wishes to sound.



### >STEP-03

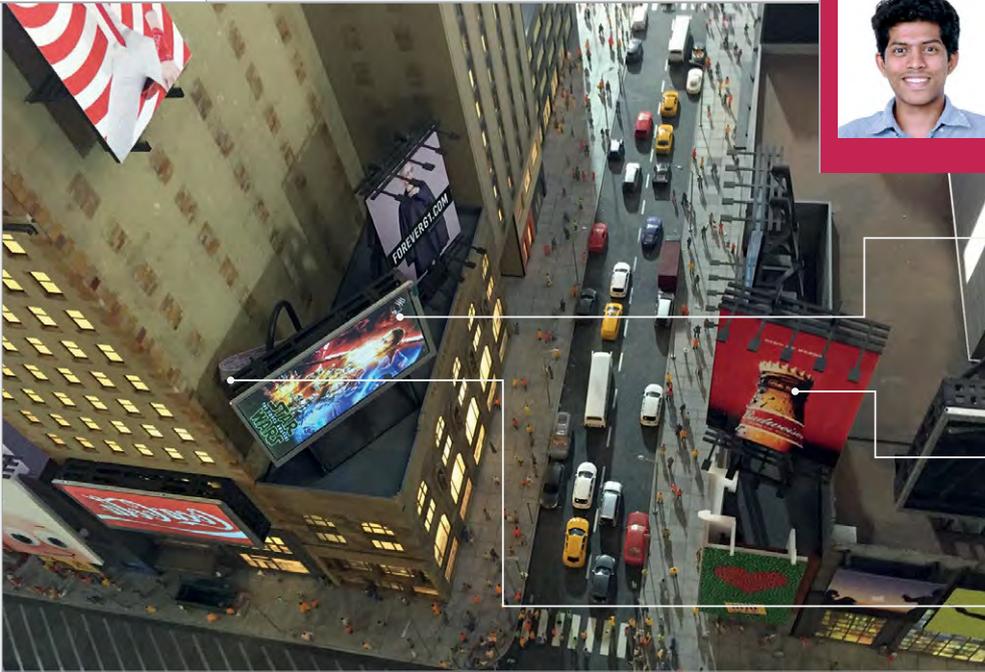
#### Body

The body of the guitar is built from steel and MDF, the latter shaped by the engineering team at Dimensional Innovations using a CNC router. Stickers representing performers and other participants in the event adorn the completed orange body.



**UTTAM GRANDHI**

An Elastic Mind enthusiast based in Brooklyn, NY. His interests span electronics, sculpture, biology, maths, animation, and origami. [utt.am/about](http://utt.am/about)



It was important to get the viewing angle, screen brightness, and orientation right for each screen

By mixing the Pi-powered screens with static mini-billboards, a busy, realistic-looking Times Square emerged

The cabling is tucked away and hidden behind the screens to make for a tidy installation

# GULLIVER'S GATE

When recreating Times Square in miniature, Uttam Grandhi and his team knew the bright lights of its billboards would have giant appeal

**Quick Facts**

- Gulliver's Gate is inspired by the satire *Gulliver's Travels*
- It features landmarks from more than 100 cities
- Visitors use telescopes to see the detail close up
- The Pi boards are connected to a 24-port Netgear switch
- The team also bought a Raspberry Pi starter kit

**V**isit New York and it's hard not to be overwhelmed by the glittering scale of Times Square. Spread over millions of square feet, flanked by skyscrapers and assaulting the eyes with brightly lit advertisement hoardings, it is one of the world's most visited tourist attractions.

While the owner of the globe's largest and most expensive digital billboard once declared, "size matters in Times Square," one company, Gulliver's Gate, is proving that small can be equally beautiful. It has spent \$40 million bringing together 50 nations in a 49 000 sqft (4552m<sup>2</sup>) miniature

world, situated in the former New York Times building on 44th Street. What's more, it has made great use of 23 Raspberry Pi Zeros.

**Pi-powered models**

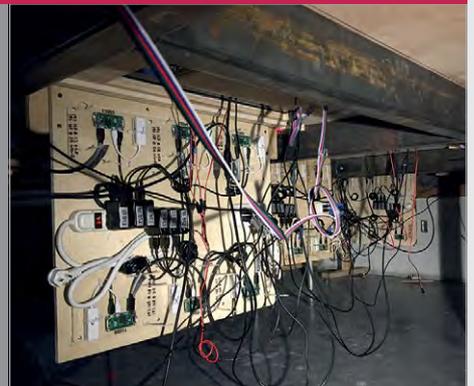
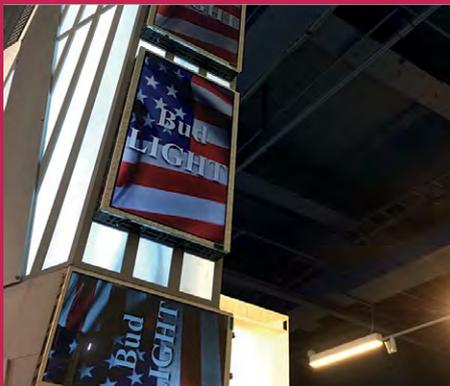
By connecting the Pi boards to 23 ten-inch and seven-inch screens, artist and design technologist Uttam Grandhi, together with a talented team from Brooklyn Model Works, has brought a 1:87 scale recreation of Times Square to life. They have used the displays (which were bought, complete with LCD control boards, from the electrical retailer **Banggood.com**) as small-factor digital billboards, fixing them to both the façades and terraces of the recreated towers.

"We've situated them on different miniature buildings and in different orientations," Uttam says, having used them in both landscape and portrait mode. "Our designer, Martin Eisler, had to design a modular



Laser-cut hooks are used to attach the screens to the buildings – the team used the Speedy 300 laser cutter

## LIGHTING UP A MINIATURE TIMES SQUARE



### >STEP-01

#### Managing the screens

The screens were used in portrait and landscape modes. Based on their power draw, the displays were distributed between different power supplies situated at the back of the hub plates.

### >STEP-02

#### Connecting the Pi boards

The 23 Pi boards are installed across four hub plates, with varying lengths of power and HDMI cables running between the Pi devices and the screens. The displays are configured to use HDMI by default.

### >STEP-03

#### Shining a light

Testing the screens is difficult because the Pi boards are under the platform – one person troubleshoots the Pi devices while another checks the displays. The adverts run via Adafruit's Pi Video Looper.

mounting mechanism for the frames based on the Banggood screen dimensions. The frame parts were cut in 1/8-inch black cast acrylic and they were sanded and glued together with Acrifix solvent cement.”

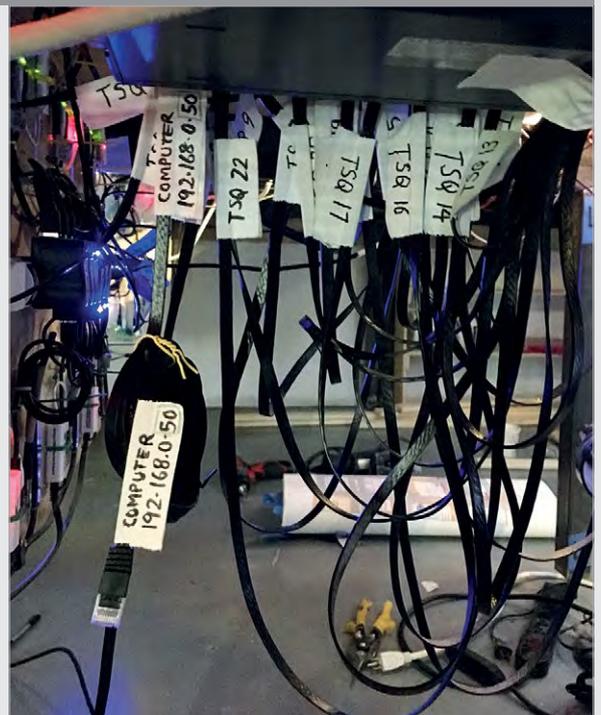
### Construction and power

To fix the screen controllers to the displays, the team marked the mounting holes before drilling and taping them. “We only used three holes to mount the controller board because its base is not flat, and tightening all four screws would

also provided a measure of safety. “The board wouldn't be damaged if someone accidentally yanked the cables,” Uttam tells us. The cables were then secured with ties to keep them contained.

### Coding real adverts

With construction complete, they could prepare the Pi Zeros. Uttam burned the Raspbian OS on to a microSD card using the Etcher app, and installed Adafruit's Raspberry Pi Video Looper to display the ads. “We used a library, so most of the programming task was already



**Above** Every cable has been carefully labelled to make troubleshooting easier

“ Targeting visitors by running real ads on the miniature screens is a real stroke of genius ”

have tilted it to one side,” explains Uttam. “The screens were attached to the frames using clear VHB tape.”

With all of that in place, they could attach the HDMI and power cables to the screen controllers. There wasn't enough room for standard cables, so they used HDMI flat angle adapters, which

complete,” he continues. “But to save some time in configuring all of the Pis, I wrote scripts to set static IPs and change host names.”

The Pi boards were connected to a 24-port Ethernet switch and placed under the platform on which the models sit. The team could then begin to gather

adverts, which have come from the advertisers themselves, shining brightly from the buildings they adorn. “They really added depth and dynamism to the otherwise still buildings,” Uttam says of the final result. “Targeting visitors by running real ads on the miniature screens is a real stroke of genius.”



MARTIN MANDER

Martin has 13 Pi boards, many earning their keep around the home he shares with his wife and two children. He works as an analyst in Norwich. [magpi.cc/zvPQK6E](http://magpi.cc/zvPQK6E)

Quick Facts

- A set of three intercoms originally cost £99.95 in 1986
- Martin bought his set for a bargain £4
- He put the Pi behind the case's grille
- The case was coated with matt-finish craft paint
- He'd like to include a manual rotary volume control

# GOOGLE PI INTERCOM

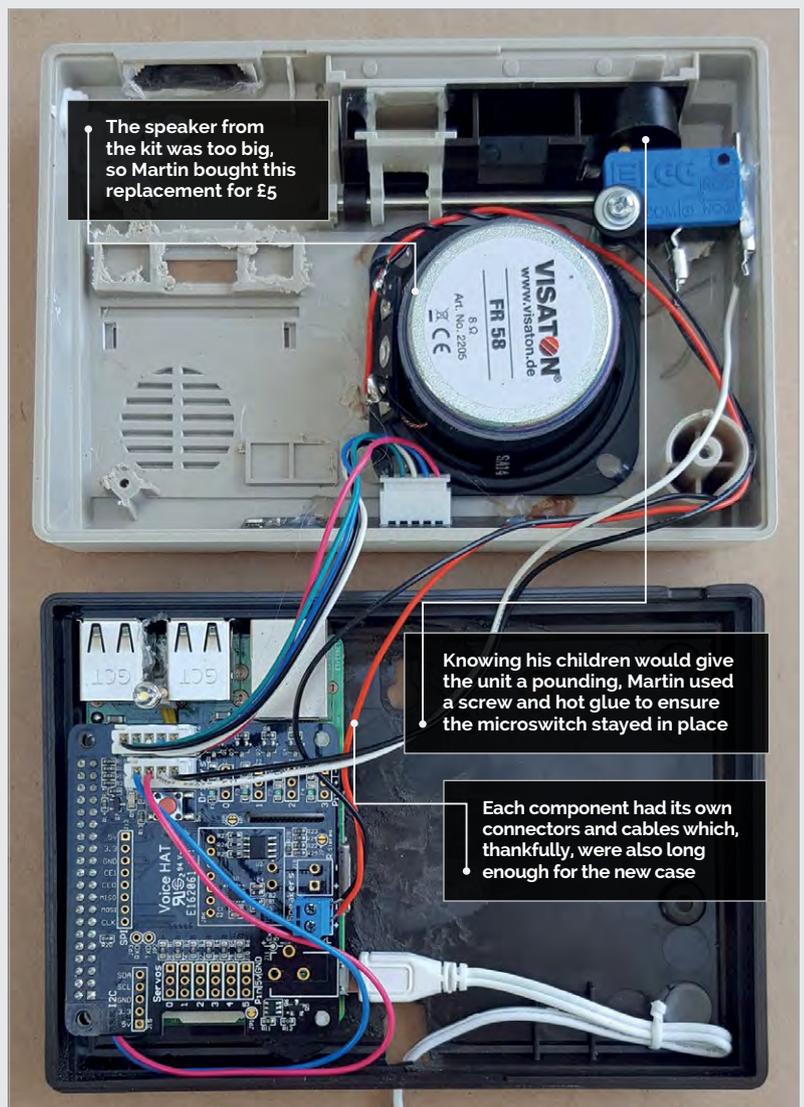
Eager to evoke a sense of nostalgia, retro nut **Martin Mander** decided to place his Google AIY kit inside an old FM wireless intercom

**W**hen Martin Mander saw *The MagPi* magazine was giving away a free hardware voice kit from Google in issue 57, he did the sensible thing and rushed off to buy a copy. But after he assembled the components, placed them in the cardboard housing, and pressed the button to activate the Google Assistant to ask a question, he started to think about the future possibilities.

Ironic, then, that he promptly delved into the past. “The family and I enjoy the local car-boot sales and I’m always looking for old, obsolete or broken technology that I can use in projects,” he says, relishing the chance of picking up something from the 1970s and 1980s.

He noticed some intercoms which were in their battered original packaging, and he thought it’d be fun for the kids to be able to chat between their rooms. “They didn’t work, and so I shelved them in the workshop,” Martin continues. “Then the Voice HAT came along and I decided it would be nice to preserve the spirit of these intercoms in a small way.”

Martin’s idea was to fit the Google AIY kit into one of these old units. “I’d followed the instructions in *The*

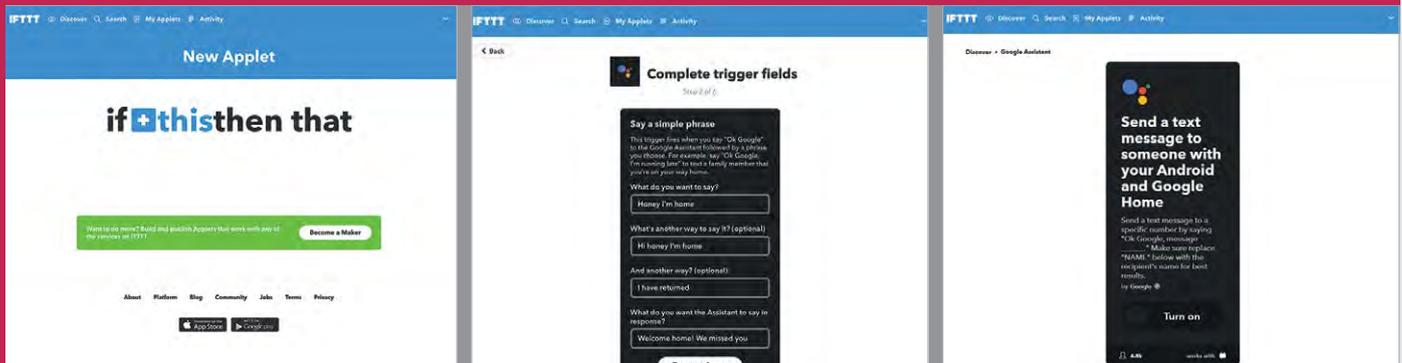


The speaker from the kit was too big, so Martin bought this replacement for £5

Knowing his children would give the unit a pounding, Martin used a screw and hot glue to ensure the microswitch stayed in place

Each component had its own connectors and cables which, thankfully, were also long enough for the new case

## A CONNECTED HOME



## &gt;STEP-01

## Using IFTTT

Martin wanted to expand the Google Pi Intercom's abilities, so he logged into [ifttt.com](http://ifttt.com), clicked My Applets, selected New Applet, and then clicked the '+' icon on the next screen.

## &gt;STEP-02

## Finding Google Assistant

Having searched for Google Assistant and clicked Connect, he signed into his Google account. He could then select a trigger and fill in the phrases that would activate it.

## &gt;STEP-03

## Organising the response

His idea was to get the intercom to send a text to his wife when he says he's home. He chose SMS as the action service and filled in his mobile number.

MagPi to assemble the HAT and get the software up and running on the Pi," he recalls. With the Pi and HAT running on his desk, he tested some voice commands before getting on with the task of cracking open the old intercom. "There's always a slight feeling of naughtiness in ignoring the 'do not open' warning and seeing what's inside," he says. "The great thing about older tech is things tend to be screwed or bolted together, allowing them to come apart nicely, leaving just the case."

Having made lots of tiny measurements, he was convinced the kit would fit, so he cleaned and



Left The project used a repainted and cleaned intercom bought from a car-boot sale

“There's always a slight feeling of naughtiness in ignoring the 'do not open' warning

spray-painted the front of the case, let it dry, and began bolting the Pi into the case, adding the other components around it. "I set aside the green push-button from the kit, but kept the microswitch and fixed it inside the case alongside the big hinged intercom button."

Getting the microswitch in exactly the right place was the greatest challenge and it involved lots of

trial and error: "Because it was being activated at an angle by the rear of the button, even a couple of millimetres made the difference between the microswitch sticking 'on' or there being too much play in the action."

Martin then removed the LED from its case and soldered in a two-hole component connector to secure it between the Pi's USB ports, under

the original microphone grille. "The kit's microphone board was just glued into place once I'd drilled holes in the top of the case. Once that was in place, I connected the cables and closed the case."

Martin is pleased with the result: "The kids have described it as fancy and futuristic, which I'll happily take." He has since linked the intercom to his IFTTT account (see 'A connected home' box). "This took some configuring," he concludes, "but thinking up responses for the intercom to read out when the triggers are activated was really good fun."



**MATT, BOMANI & EUNICE**

Matt Zheng, Bomani McClendon and Eunice Lee were students of design and computer science together at Northwestern University in Illinois. [magpi.cc/2vANCvc](http://magpi.cc/2vANCvc)

# WAVES

Turn the sound of your voice into a printed waveform

Image: Eunice Lee



**Above** Two printouts of different responses, both answering the question on the left

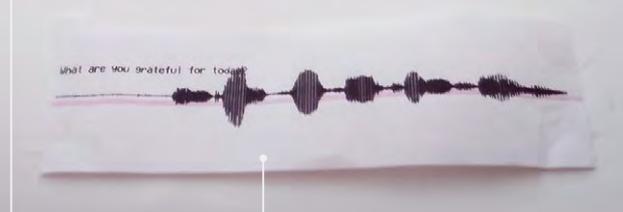
**Quick Facts**

- The project took 7–8 hours to make
- A controller script handles subprocesses
- There's no maximum recording length
- The same amount of paper is used each time
- Build details and code are on GitHub [magpi.cc/2vANCvc](http://magpi.cc/2vANCvc)

“I’m grateful for finishing this project,” says maker Eunice Lee as she holds a button while speaking into a large silver microphone. A printout of the sound waves produced by her voice emerges from the device’s thermal printer.

This is a demonstration of Waves, a side project created by Matt Zhang with Eunice Lee and Bomani McClendon, who were students of design and computer science together at Northwestern University in Illinois.

“During our first meeting, we shared some inspiring hardware projects we’d seen, and had a brainstorm session where we drew out lots of wild, funny, and weird project ideas on Post-it notes,” recalls Bomani. “Afterwards, we voted on the ideas and filtered them by feasibility, materials cost, and (most importantly) excitement. We chose Waves,



**Below** Matt Zheng looks pleased with the thermal printout of his answer – we wonder what he’s grateful for!



Image: Matt Zheng

- Holding the relevant colour-coded button, the user records their answer
- Users choose to answer one of four open-ended questions
- The thermal printer plots a waveform of the recorded response

Image: Eunice Lee



Image: Matt Zheng

and later learned that Raspberry Pi would work well for the project.”

Meeting for weekly sessions, it took the trio two to three months to create the finished project. “We encountered a fair number of problems during every work session, but we managed to overcome them by the end,” reveals Matt.

“Having a couple of ground rules set from the beginning really helped,” reckons Eunice. “For

of the sound waves would mean more to the participant.”

A Python script converts the recording of the user’s answer into a graphical waveform to print. “The programming wasn’t too tricky, since we borrowed a lot from open-source Python code,” says Matt. “The plots take up the same amount of paper, no matter how long the recording is, but we have to manually adjust the volume on the microphone, to

“ We all brought different skills to the table, but a mutual excitement for Waves and what it could be ”

instance, we decided to always meet once a week and to commit to a project that would take less than ten hours to build. This kept the project lightweight, fun and stress-free.”

“For me, the best part of this project was getting to work with Eunice and Matt,” says Bomani. “We had a lot of laughs during our sessions.”

“We all brought different skills to the table, but a mutual excitement for Waves and what it could be,” adds Eunice.

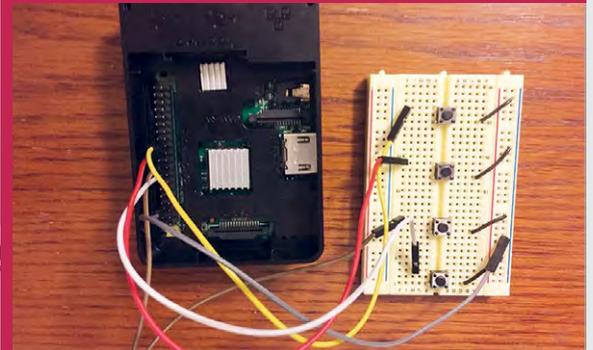
After brainstorming ideas for open-ended questions for users of Waves to answer, the team narrowed them down to the final four. “We wanted the questions to be personal and reflective,” explains Eunice. “That way, receiving a print

make sure that the audio doesn’t blow out or disappear entirely.”

The result can also be uploaded to a locally hosted website, which was displayed on a monitor during an annual Design Expo at the university. “The idea for the website was that people could see other people’s sound waves and wonder what was said,” says Eunice. “It was a really fun addition that turned some heads and made people interested in what our project was.”

Eunice was delighted with the overall reaction to Waves: “It was really great to see people’s faces light up when they finished recording and saw their sound waves being printed. [...] The best part was being able to give that to the user and say, ‘You can keep it!’.”

## PRINT OUT YOUR VOICE



### >STEP-01 Press a button

Four push-buttons – one for each question – are placed on a breadboard and connected to the Raspberry Pi’s GPIO pins. The user holds down a button while recording their response.



### >STEP-02 Box and printer

The Pi and buttons are placed inside a cardboard box, spray-painted black, along with a tiny thermal printer bought from Adafruit ([magpi.cc/2up57jH](http://magpi.cc/2up57jH)) – any similar model will do.



### >STEP-03 Colour coding

The four questions are printed on cards with coloured stickers matching the ones next to the push buttons on the device. A cool-looking Blue USB mic records the user’s voice.

Images: Matt Zheng



**DANIEL JAMES**

Daniel James is the director of 64 Studio Ltd, which produces, integrates, and documents free and open source software for media. [64studio.com](http://64studio.com)

**Quick Facts**

- Daniel estimates a Pi Deck setup costs \$182
- There is no soldering or modification to the turntable
- Audio Injector sound cards cost around £20
- A Serato CV02 control vinyl pressing is recommended
- The FLAC music format gives the best results

# PI DECK

Digital music can't be scratched – or can it? **Daniel James** and **Chris Obbard** go 'old school' with a Pi-based digital vinyl system

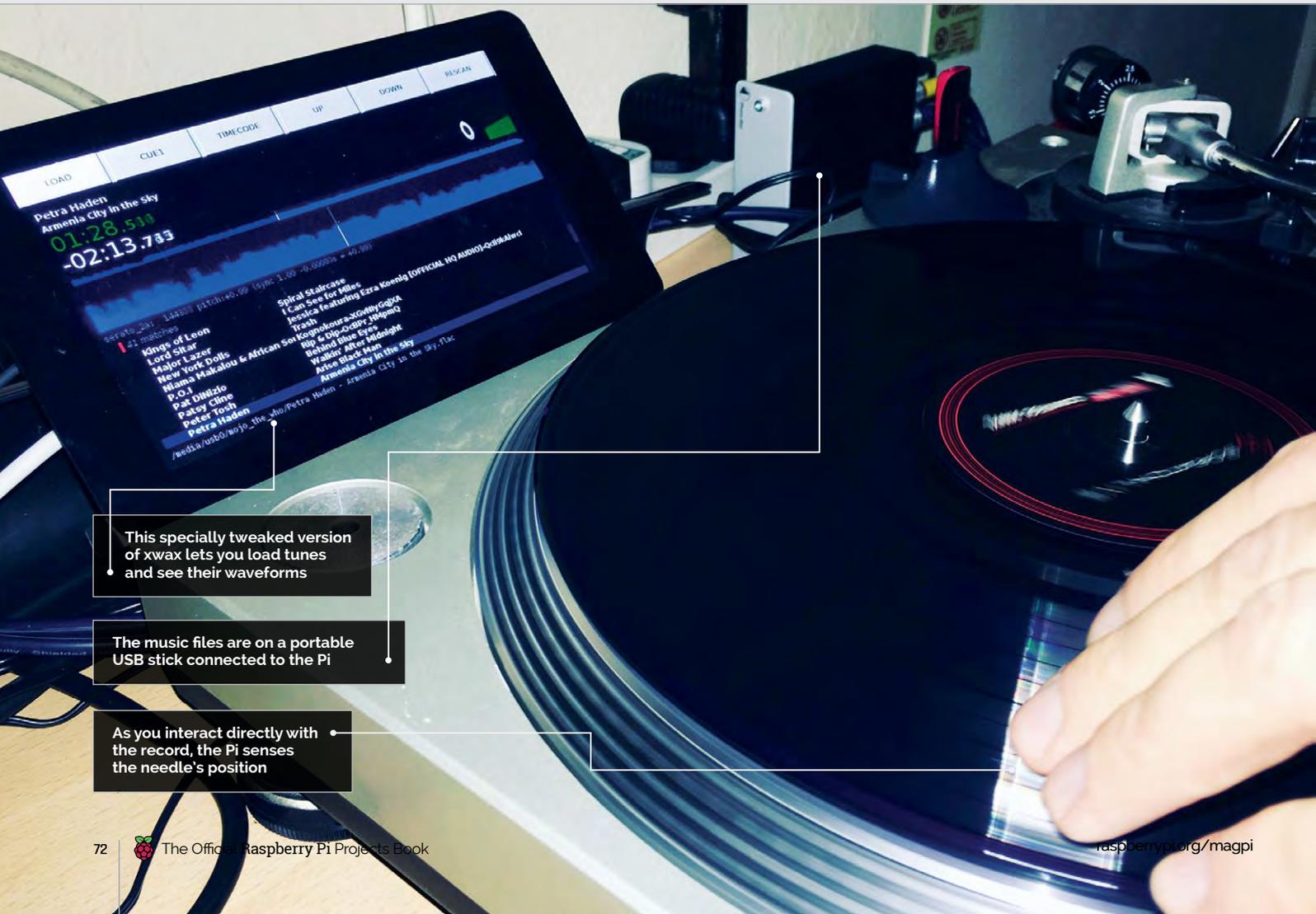
**E**ver since he saw *Breakdance: The Movie* in 1984, Daniel James wanted to be a DJ. But he soon realised scratching expensive vinyl or carting it from venue to venue is far from ideal. “Unless you are famous enough to have a roadie, large boxes of vinyl are heavy and impractical,” he tells us. There’s also the problem that a lot of music

is not being released on analogue vinyl, despite the huge resurgence of interest in the format.

Digital Vinyl Systems (DVS) have long provided an alternative. They allow you to DJ using digital music files and software while letting you benefit from the feel of a turntable. You simply place a special vinyl record on a standard turntable and the system will read

the position of the stylus from a timecode. This is used to play back a digital music file at the same point in time – but these systems can be hugely expensive.

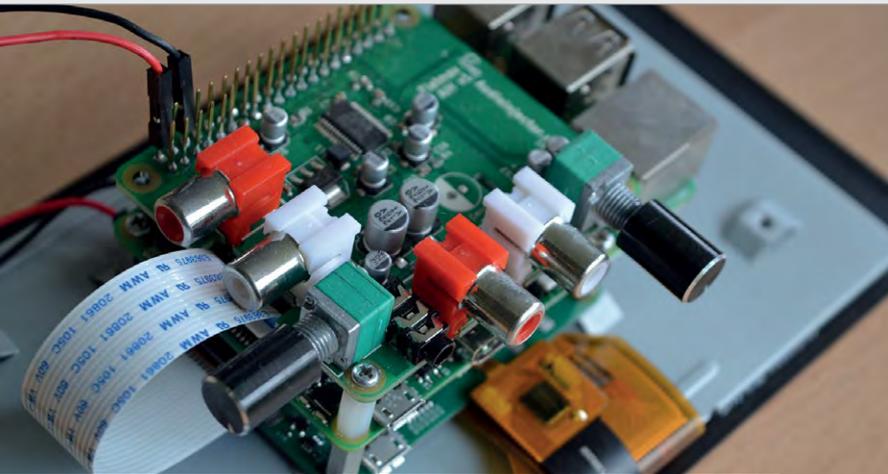
Not now. Daniel and his colleague Chris Obbard at 64 Studio put a Raspberry Pi 3 alongside a standard vinyl deck, eliminating the hassle and expense of hooking digital decks to a laptop. They figured that



This specially tweaked version of *xvax* lets you load tunes and see their waveforms

The music files are on a portable USB stick connected to the Pi

As you interact directly with the record, the Pi senses the needle's position



The Audio Injector stereo sound card adds line-level audio inputs to the Pi

all you need is a USB stick packed with music files. “As long as there is low latency in both directions of travel, you can make effects such as rubbing the kick drum beats, scratching and backspins sound, feel and look realistic,” Daniel says.

Motivated by a desire to show that low latency applications could work reliably on small and inexpensive ARM devices such

The greatest challenge was getting the audio interfaces to work with low latency since many of the known techniques are for PC architecture. “Interrupt tweaking doesn’t work the same way on the ARM architecture, for instance,” Daniel explains, “but switching to newer Linux kernels helped a lot.” The majority of the work involved configuring

“ Most of the setup is done with shell script, which is easy to hack ”

as the Pi, Daniel and Chris got to work. They used *xwax* ([xwax.org](http://xwax.org)), an open-source DVS for Linux capable of playing MP3, FLAC, and AAC files, among others. “But because *xwax* isn’t easy for people new to GNU/Linux to set up the system optimally, we created a ready-made Debian image for the Raspberry Pi that is specialised for this one application,” reveals Daniel. “It means DJs don’t have to go through as many steps to get it running.”

The pair had tested Raspbian images and various kernels with hand-soldered audio hardware to create the inputs and outputs that they needed. “Then we tried a HAT sound card from [audioinjector.net](http://audioinjector.net) and some USB audio interfaces which include phono cartridge preamplifiers. These are needed because the stylus on the record outputs a signal at a very low level, compared to other audio devices.”

and scripting various programs to work smoothly together with minimal user input. “Most of the setup is done with shell script, which is easy to hack.”

A screen completed the package. Daniel and Chris decided to use the official Raspberry Pi seven-inch touchscreen. “We wanted something that would be easy to set up,” Daniel continues, “but the Pi Deck works with any compatible screen. If you’re not using a touchscreen, an ordinary mouse can be used to click the buttons. A QWERTY keyboard is useful for searching music titles.”

Since creating the Pi Deck last October, the pair have been excited at the level of positive feedback. Hearing about people using it across the world has been music to their ears. “People appreciate the low latency and stability,” Daniel concludes. “These are the crucial factors for any performance.”

## HEY, MR DJ



### >STEP-01

#### Put the record on

Once Pi Deck is set up, place a control vinyl on the turntable, power up the Pi, and insert the USB stick. Waxed paper under a felt slipmat will reduce friction.



### >STEP-02

#### Position the stylus

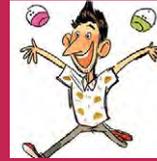
Put the stylus on the control vinyl and use your hand movements to find the first beat of the record. Cue this via the mixer headphones or the waveform display in *xwax*.



### >STEP-03

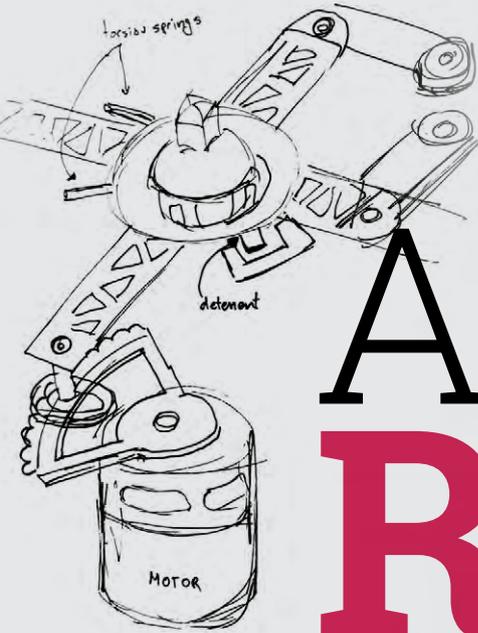
#### Scratch away

From this point on, the control works just like an analogue vinyl record. You can back-cue and scratch the vinyl, for instance, pushing the record forward or in reverse.



**ALONSO MARTINEZ**

Working as a Technical Director for Pixar, taking his experience in character creation into his exploration of interactive robots. @alonsorobots



# ALONSO'S ROBOTS

**Quick Facts**

- > Alonso learnt the majority of his maker skills from internet videos
- > Mira went through multiple designs before her final egg shape
- > She charges via a small port on her base
- > Mira's small on-board camera can utilise face recognition
- > Alonso's next build, Lumens, doubles as a night light

**Alonso Martinez** explores the ideas of interaction and individuality in robots, creating adorable Raspberry Pi-based characters that play and learn

**A**n animator by trade, Alonso Martinez wanted to take his maker skills to the next level. He decided to explore the idea of interactivity with robot characters by creating Gertie, Mira and Lumens.

With Gertie, he was able to explore the idea of the ball bounce – a staple in animation school, where students are asked to animate a ball, and the reaction of its surface when it connects with another object. Where Gertie excelled at bouncing, there was still a lack of the emotion and personality that Alonso hoped to achieve.

**Mesmerising Mira**

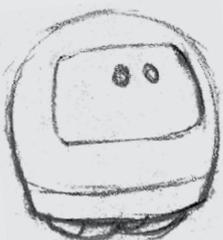
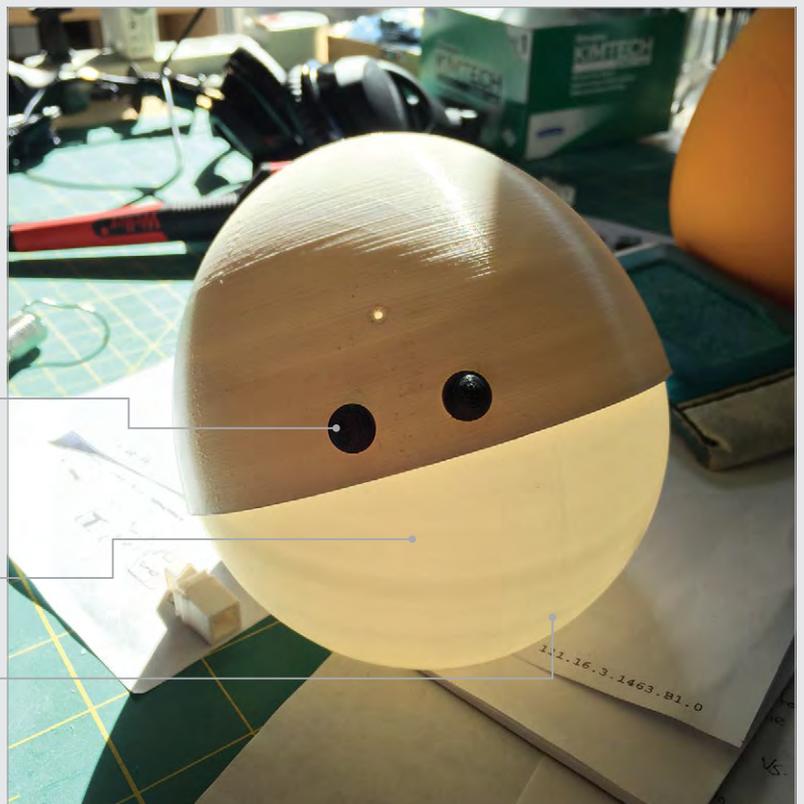
Enter Mira. Alonso wanted to create a character that was not only aesthetically pleasing and highly interactive, but also able to explore the science behind how we perceive a character based on its shape and features. Mira's round form is the

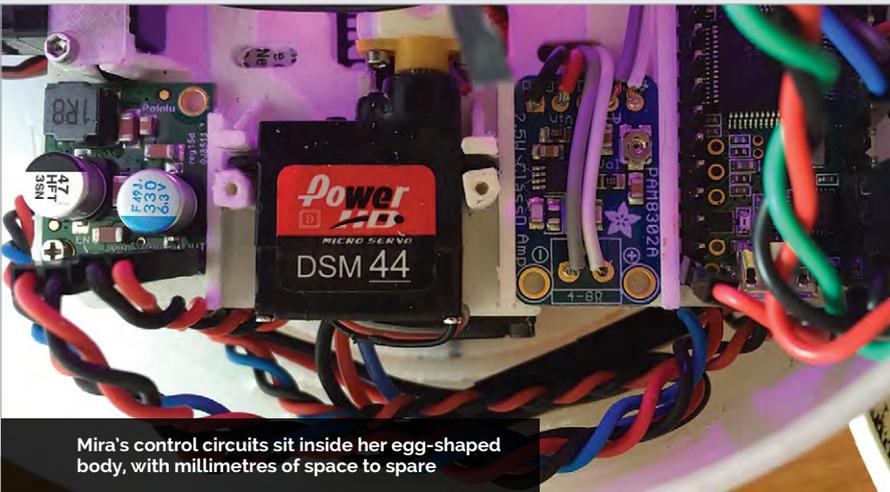
result of a study into shape science. Her lack of limbs and sharp edges creates an instantly accessible and pleasing character. And her eyes, with their mini LED screens that blink with their deep, somewhat galactic blue orbs, draw us to her that little bit more.

Mira's blinking eyes are powered by two small LCD screens that run constant animations

Her shape is the result of an exploration into shape science, and how we perceive certain features

Mira's recent appearance on the YouTube Tested channel attracted a storm of people desperate to buy their own Mira





Mira's control circuits sit inside her egg-shaped body, with millimetres of space to spare

But Mira is more than just a pretty face. Inside her body lies a Raspberry Pi, laden with code. Cover your face, for example, and Mira will recognise the game of peekaboo via her on-board camera, offering up a shocked squeal of excitement when you say “Boo!” and reveal yourself. Sit Mira beside you at a piano and she'll sing back to you any

Mira and Lumens build their own code and truly create their own unique identities?

### Millimetre-perfect

Stepping back from the mind-blowing concepts of interactive, intelligent robots, the physical build behind Mira and Lumens is equally impressive. As an animator, Alonso is highly skilled in 3D design

“ Her lack of limbs and sharp edges creates an instantly accessible and pleasing character ”

note that you play, shaking her head like a bird from a classic Disney movie. And while all this interaction is fun and pleasing, Alonso is planning to take the build further.

### Robotic evolution

Alonso wants to explore the idea of code evolution through interaction and play. While it's cute to play with Mira – or her successor Lumens, a mushroom-shaped light that hides under its cap – what if she could do more with the information received? What if, through the unique individuality of the user, the robot created its own, similarly unique personality? While explaining the concept, Alonso touches on Conway's Game of Life and the concept of cellular automation with code. Could

using the computer animation software, Maya. And it was through Maya that Alonso sculpted his robots, enabling him to recreate the Raspberry Pi and components, and measure his 3D prints to the millimetre. Those blinking eyes that give Mira so much personality come from two ultra-thin screens that slip between the two layers of her body. The space was so tight that Alonso found himself having to sand down solder just to make them fit. And as for the way her ‘head’ moves over several axes? He took the idea of the joystick he'd used as a child to play games and recreated the function using 3D-designed and printed parts.

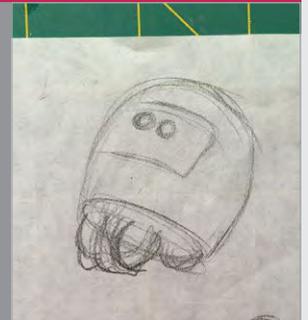
Mira and her growing family are constantly evolving, and we look forward to seeing how Lumens takes shape moving forward. Hands up who wants their own?

## BRINGING MIRA TO LIFE

### >STEP-01

#### Design

Mira went through multiple designs during the process of her conception, each variation collected as doodles by Alonso.



### >STEP-02

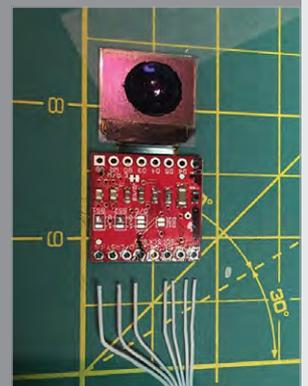
#### The bare bones

Each wire and component has its place inside Mira, with no space to spare. Given what's under the hood, Mira's small size is an incredible feat of design.

### >STEP-03

#### Those eyes

Mira's blinking eyes sit within the few millimetres of space between her head and body. The small LEDs are covered with acrylic domes that add a layer of depth to her gaze.

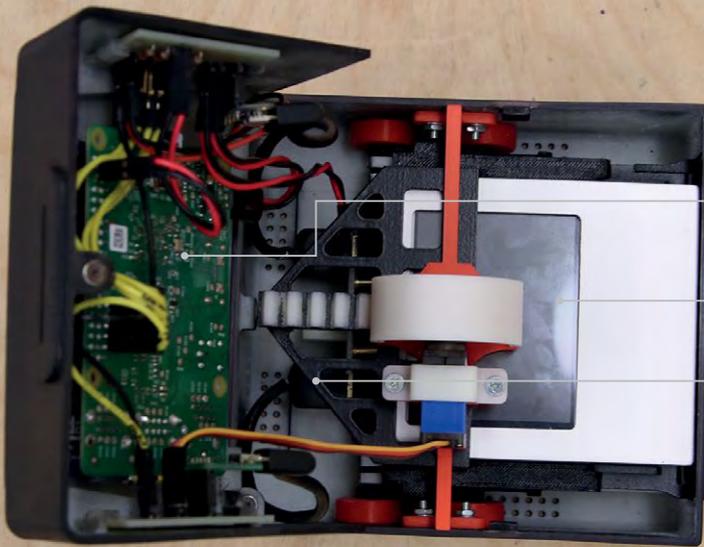


With his latest creation, Lumens, Alonso aims to incorporate the function of a night light into the robot, while maintaining the ideas behind Mira



**ABHISHEK SINGH**

A UX designer by trade, Abhishek enjoys making things that use augmented reality and 360-degree video. [magpi.cc/gyNiur](http://magpi.cc/gyNiur)



A Raspberry Pi 3 is connected to an Adafruit PiTFT screen on the back of the camera – Abhishek dubs this the CamPi

The cartridge (aka SnapPi) slots into the CamPi, and shows the resulting GIF. Each new GIF replaces the previous file

An ad hoc wireless network is created on bootup, which allows the two Pi boards to communicate without an internet connection

**Quick Facts**

- The Instagif is inspired by the Polaroid OneStep SX-70
- A Pi 3 and Pi Zero W are used
- Instagif takes three-second GIFs
- The GIF fades into view like a Polaroid image
- It took four weeks to design and build

# INSTAGIF NEXTSTEP

Ever wished your instant camera could cope with moving images? Abhishek Singh has the answer!

**B**efore digital cameras became popular, photographers would use film to capture their images. They'd take a set number of shots and send their snaps away to be processed, but then they'd have to wait days – or even weeks – to see the results.

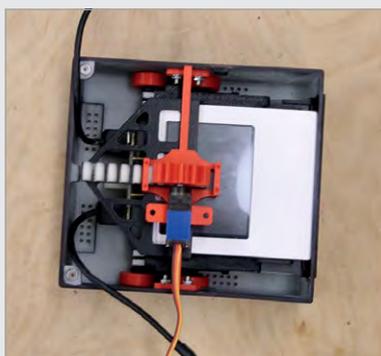
Instant cameras were invented to speed up this process. Polaroid became famous for making cameras that could print out a paper-based photo in seconds. A few shakes, and the image would appear as if by magic. While instant cameras have fallen out of favour with the advent of digital cameras, the appeal of the concept endures.

of the Raspberry Pi and moved the instant camera into the 21st century. By inserting a cartridge with a tiny screen into his cool retro-inspired invention, you can record a GIF, wait for the cartridge to pop out and, after a few seconds, watch and enjoy the mini-video. What's more, the device looks just like a vintage Polaroid OneStep camera.

**Instant update**

Abhishek Singh certainly likes it. Motivated by the desire to build something physical, and spurred on by his love of animated GIFs (“my earlier projects have revolved around them as well,” he tells us), he has harnessed the power

“Something about holding a moving image sounded intriguing,” Abhishek explains. “It’s instantly relatable, and it has the unique and exciting interaction of a physical image coming out immediately that you can hand to a person. I wanted to recreate the experience of a Polaroid in a new way.”



**Right** The cartridge looks like a Polaroid picture, and is ejected using a rack-and-pinion mechanism

## SNAP HAPPY

**>STEP-01****Inserting the components**

There wasn't much space for the components in this build. Abhishek fitted a 2.8-inch PiTFT screen into the cartridge, and he chose a 400 mAh LiPo battery to power the cartridge when it's ejected.

**>STEP-02****Using a Pi Zero**

A Raspberry Pi Zero W was soldered to the screen. Abhishek removed the header pins from the screen, sanded down the Pi's micro USB ports and PCB, and removed the top casing from the HDMI port.

**>STEP-03****Charging up**

The cartridge is charged by a PowerBoost 1000C charger in the camera. Python and Node scripts run in the camera, handling the connections and camera control. Clicking the button starts and stops the GIF recordings.

**Instant design**

To begin with, Abhishek sketched out his plan on paper, before turning to Autodesk Fusion 360. His design had to take two units into account: the camera (dubbed CamPi) and the cartridge (SnapPi). To produce the camera, he used a Raspberry Pi Camera Module, an iPhone lens, and an 8GB SD

The Pi boards were trimmed and sanded. The Ethernet port and a dual USB port were removed from the Raspberry Pi 3, so the mounting holes could be used for the camera chassis. Four circuits were produced to manage the wiring and distribute power, and Abhishek made his own USB cables. Major issues included finding a suitable

were ready, it was simply a case of assembling them into the printed plastic components that made up both the camera and the cartridge. Finally, Abhishek could point the camera, press the record button, and watch as the cartridge popped out and the GIF appeared.

Was it worth it? "The reaction has been super-positive," he says. "For me, it was not so much about the learning curve being steep as about finding creative solutions to the problems that arise every step of the way."

“ For me, it was not so much about the learning curve being steep as about finding creative solutions ”

card. The cartridge uses a 2.8-inch screen and another 8GB SD card. A Raspberry Pi Zero W was fitted into the cartridge, and a Raspberry Pi 3 was used in the camera.

"I had to think and visualise ways of assembling the thing, and look at how it would be disassembled if I had to troubleshoot something," explains Abhishek. "I needed to consider how everything would fit together and be placed to minimise the size. A lot had to be done to conserve space, and I mutilated both of the Raspberry Pis I used in the project quite a bit, but that's what made using them so great."

way of ejecting the cartridge from the camera (Abhishek chose a rack-and-pinion mechanism with wheels to eliminate friction), and solving a wide-angle problem. "But I like the community the Pi has," he says. "A solution to the problem you are facing is probably out there somewhere."

**Instant success**

It helped that he split the project into small, doable chunks. "It meant I could go through an individual process of iteration, testing and refinement," he adds. Indeed, once the working pieces



The parts were printed using a Project 7000 SLA printer, before being sanded and painted



## ROBERTO TYLEY

An open-source enthusiast, Roberto develops software for The Guardian, and authored the BFG Repo-Cleaner. [magpi.cc/2wVeHNE](http://magpi.cc/2wVeHNE)

# MESMERIC MIRROR

Looking to brighten a dark bedroom, **Roberto Tyley** decided to cast fresh light on the idea of a motion-sensing illuminated mirror

## Quick Facts

- It took nine months to build
- The laser was £139 and the project cost £250
- Roberto used an eyeball-safe, low-power Class I laser
- A Bluefruit EZ-Key acts as the on/off switch
- Find the Python code at [magpi.cc/2wURUBy](http://magpi.cc/2wURUBy)

**O**n reflection, he could have just bought a lamp. But after Roberto Tyley stood on one too many of his children's toys in the dark corners of his bedroom, and decided it was time for some additional light, he was hit by a flash of inspiration.

What better way of illuminating the room, he mused, than a full-length portrait mirror that would shine brightly whenever anyone wandered by? One with individually controllable LED lighting that would directly respond whenever someone posed in front of it, perhaps? As light-bulb moments go, he admits it was certainly high on the wacky scale.

But Roberto saw the potential for both practicality and fun. "The idea was to have a mirror that would know if my children were waving their hands around," he explains. "I wanted the lights to move and sparkle around their fingers as they did so."

## Pi-powered

Roberto's first decision was to put a Raspberry Pi at the heart of the build. He had only ever used the computer once before, to create a video demonstrating the speed of a tool he'd written to clean Git



This shelf of electronics effortlessly slides into the unit, connecting via a Multi-Beam XLE power connector

Roberto didn't want to use an obtrusive always-on video camera, so he used a laser rangefinder instead

The Adafruit DotStar LEDs use a simple wiring system, with four wires controlling the whole strip

## LET THERE BE LIGHT



## &gt;STEP-01

## The handy compartment

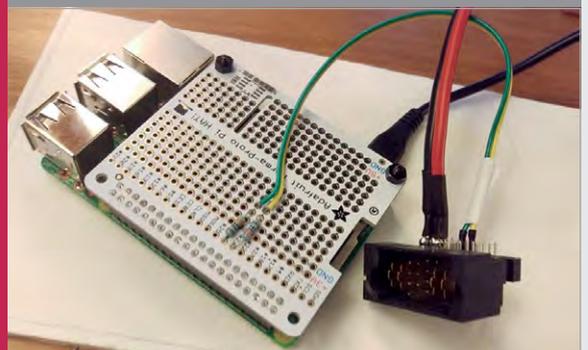
The electronic components sit inside this removable, easily accessible shelf. It includes a Raspberry Pi, backplane connector, speaker, laser rangefinder, AC power in, and power supply. The decorative brass grille provides ventilation.



## &gt;STEP-02

## Fixing the lights

The LEDs fit behind the mirror's frame. "Problem is, they have a tiny computer running in each pixel, so extra power taps are needed every metre or so to keep the voltage level high," says Roberto.



## &gt;STEP-03

## Connecting everything up

The key I/O components use four-wire connections: I<sup>2</sup>C for the LIDAR-Lite and SPI for the DotStar. Thankfully, the Pi can handle both. Roberto soldered the connections to an easily attachable Perma-Proto HAT.

“My children love it, and they show it off to their friends when they come round”

repositories. He liked the fact that the Pi was a fully specced Linux platform. “It meant I could use the tools and languages I was familiar with,” he says.

His next step was to research the best lighting, and he eventually chose Adafruit’s DotStar Warm White LED strips. “They are more expensive than simple strips, but they use a pretty straightforward wiring system,” Roberto explains. “Adafruit also supplies a nice Python library to control the strip from the Pi.”

## Motion sensing

From there, he looked for a way of detecting the presence of a person, before hitting on the idea of a laser rangefinder. “It occurred to me that if I had a laser rangefinder pointing straight down from the top to the bottom of the mirror, in the same axis as the LEDs, it could sense how high up anything was,” he continues.

This would allow the LEDs to light up to the height of a person, or work out if they were stretching their arm up, and illuminate further. “I decided to use the LIDAR-Lite rangefinder, which is a pretty cool but pricey sensor that encodes digital fingerprints into the bursts of laser light it sends out,” he tell us. “This means it

can uniquely identify the pulses of light that bounce back, so it’s not confused by secondary reflections.”

To tie everything together, Roberto created a simple program in Python: “The code polls the laser rangefinder to work out if there is an object closer to the top of the mirror than the mirror base. If there is, it increments the brightness in an array that tracks how bright all the LEDs should be.” But it wasn’t entirely straightforward. Creating the body of the mirror proved a daunting task.

## Learning by doing

“Before I started, I had zero woodwork skills – I’d been terrible at it at school,” Roberto admits. He watched several YouTube videos about making mirror frames (“the opposite sides really need to be the same length,” he laughs) and he also ended up buying all kinds of powerful tools to get really accurate mitre joints.

Still, it was worth it. During the grand unveiling, accompanied by the sounds of *2001: A Space Odyssey*, he watched the delight on his children’s faces. “It had become an almost embarrassing addiction,” he says of the process. “But my children love it, and they show it off to their friends when they come round.”



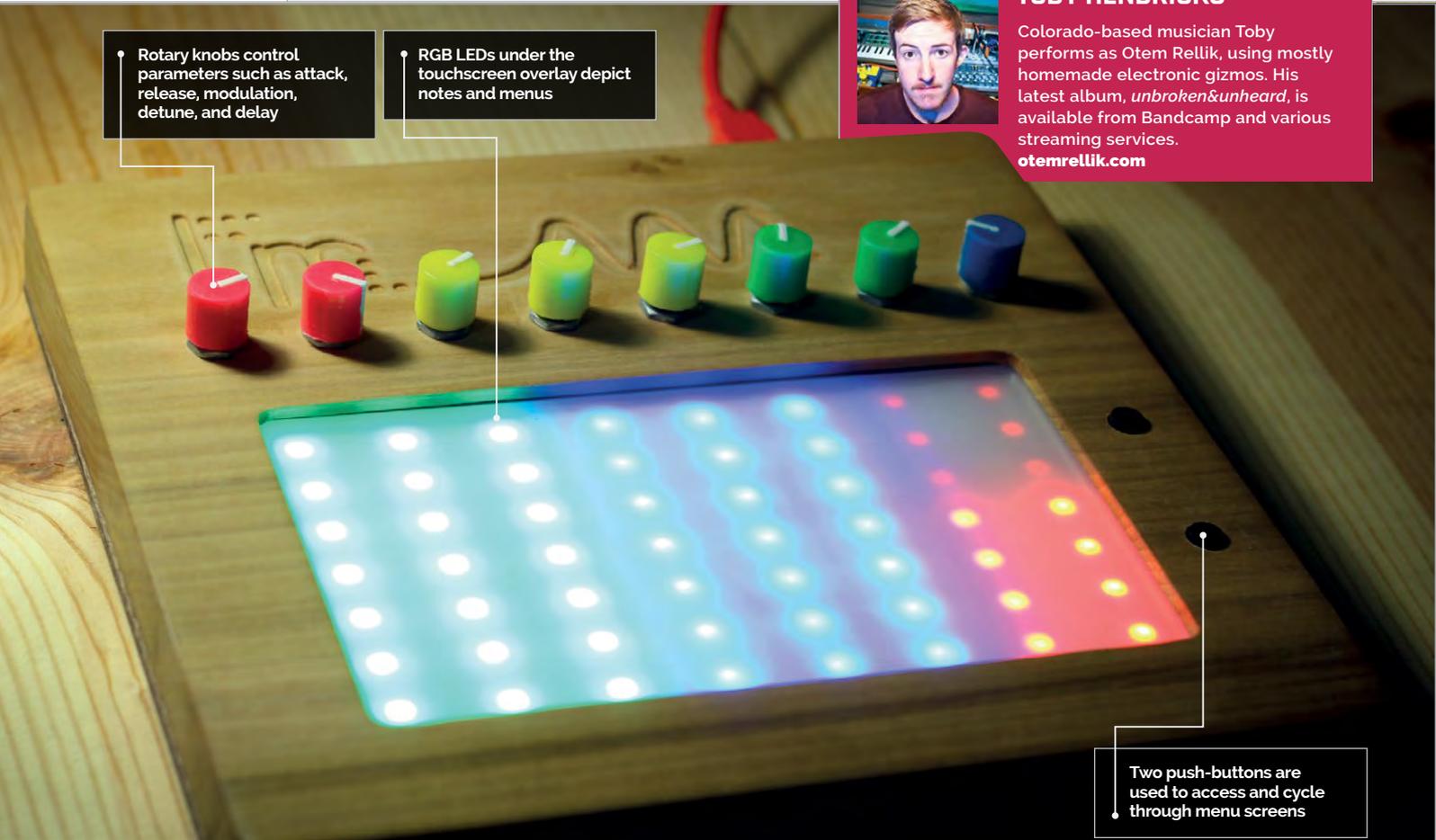
**TOBY HENDRICKS**

Colorado-based musician Toby performs as Otem Rellik, using mostly homemade electronic gizmos. His latest album, *unbroken&unheard*, is available from Bandcamp and various streaming services. [otemrellik.com](http://otemrellik.com)

Rotary knobs control parameters such as attack, release, modulation, detune, and delay

RGB LEDs under the touchscreen overlay depict notes and menus

Two push-buttons are used to access and cycle through menu screens



# FM TOUCH SYNTH

**Quick Facts**

- It took a month to build and program
- A resistive overlay is used ([magpi.cc/2gJSZlU](http://magpi.cc/2gJSZlU))
- Sounds can be stored in eight patch presets
- LEDs are used to display notes and menus
- A Teensy 3.6 handles controls and audio output

Frustrated by unreliable iPad music apps, **Toby Hendricks** built a homemade touchscreen synth based on a Raspberry Pi Zero

**N**ot long after impressing us with his Raspberry Pi Looper ‘synth drum thing’ ([magpi.cc/2xZTAXR](http://magpi.cc/2xZTAXR)), Toby Hendricks (also known as Otem Rellik) is back again with another Pi-based musical device. This time he has constructed a touchscreen synthesiser that offers a wide variety of stunning sounds, modified using a combination of rotary knobs and touchscreen control.

“It went through several iterations before I settled on the final design,” Toby tells us. “The original idea was to have two touchscreens for two-voice polyphony, but that ended up being a bit too large, so I went with one.”

**Zero debut**

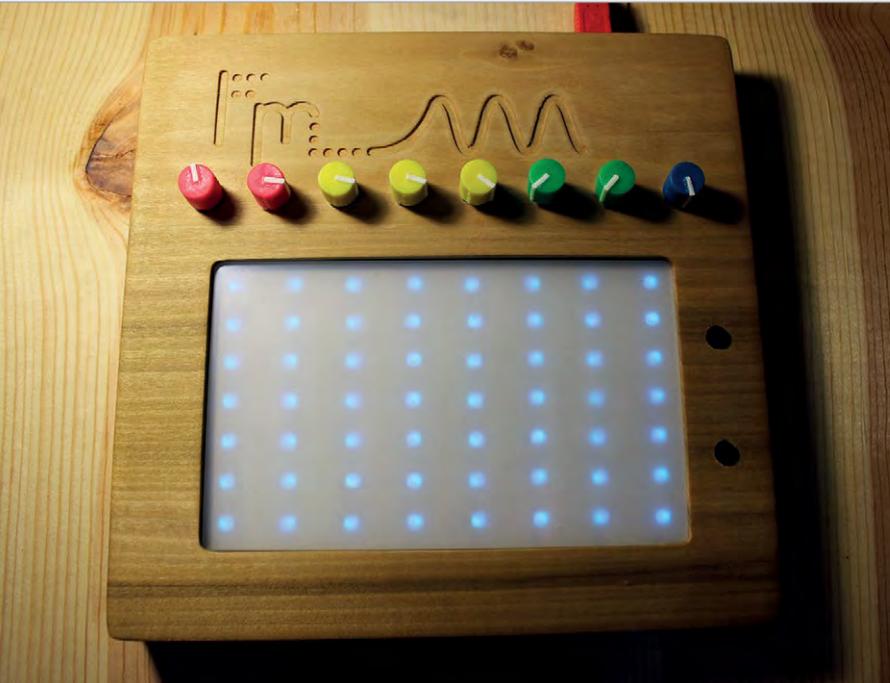
While a Teensy 3.6 board is used to handle the control data, all the audio – created using a combination of FM and subtractive synthesis – is processed in Pure Data on a Pi Zero. “This was my first project using the Zero,” says Toby. “I just wanted to see if I could run a somewhat simple Pure Data patch on a Zero without any audio glitches (and it turns out I could). Size was definitely a factor: I wanted to make a really slim, sleek instrument.”

Toby says the Pure Data programming language is easy to use. “Once you learn the basics, you can do so much with audio and MIDI. It’s a visual programming language, so instead of writing

code line by line, you create objects and start connecting patch cables. It’s great if you are coming from a non-technical perspective because everything is so easy to understand visually.”

**Touch the music**

While the synth’s rotary knobs are used to control aspects such as attack, release, frequency modulation, detune, and delay, the main input interface while playing and creating new sounds is the touchscreen overlay. Columns of RGB LEDs placed beneath it depict eight notes (the scale of A minor), while the position of the player’s finger modifies the sound produced.



Rather than displaying information on a regular touchscreen, Toby opted to use LEDs for everything, including menus. “It’s a bit more archaic and cryptic. I like to make instruments that you really have to learn to use, and aren’t immediately super-accessible. Sort of like the old hand-held video games I grew up with.”

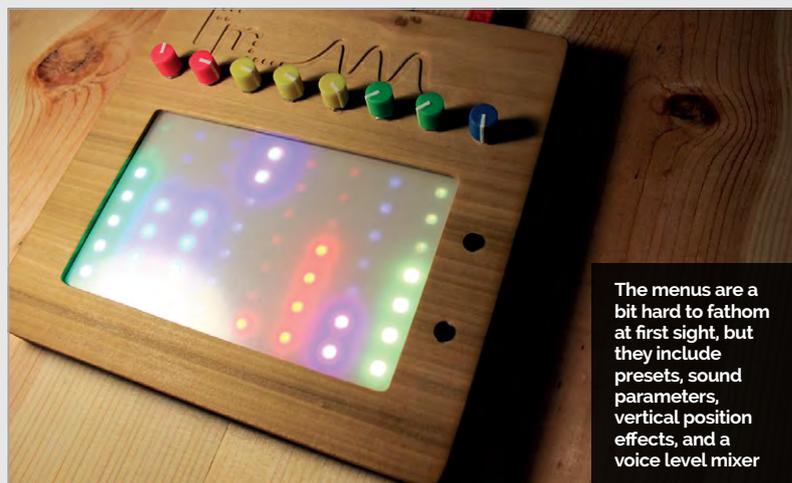
One drawback is that the resistive overlay can only handle one touch at a time, although a workaround involves sustaining notes. “I added the sustain feature close to the end of the project because I wanted polyphony. The synth just sounds so massive when you play it polyphonically. Perhaps in the future I’ll upgrade the screen to allow multitouch.”

**Above** Toby is pleased with the case, crafted using his new CNC machine: “I think all my future projects will have wooden cases”

### Live project

Toby has already used the FM Touch Synth in live performances. “It’s working out great! I’ve been doing a lot of ambient jam sessions with it, turning the delay way up and feeding into my Pi Looper. It’s been a lot of fun.”

Toby would really like to make the synth available commercially, which is why he hasn’t released the code yet, but he’s already working on another Pi Zero project: “a looper in which you can manipulate the pitch and speed of the loop in four different slots.”



The menus are a bit hard to fathom at first sight, but they include presets, sound parameters, vertical position effects, and a voice level mixer

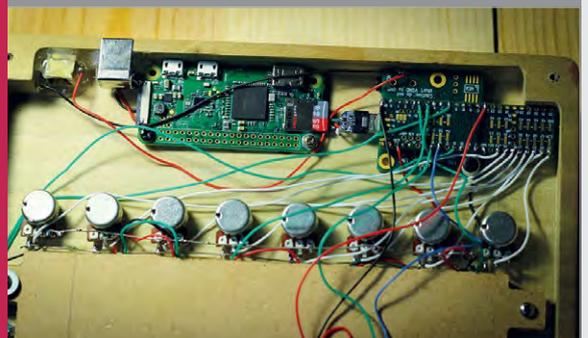
## BUILD A TOUCHSCREEN SYNTH



### >STEP-01

#### Making a case

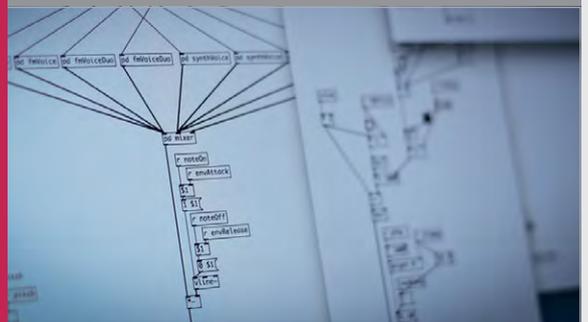
The wooden case was Toby’s first project using his new CNC machine. It houses the components, including the matrix of RGB LEDs that sit under the touchscreen overlay.



### >STEP-02

#### Teensy bit of help

Eight rotary knobs and the touchscreen overlay are wired up to a Teensy 3.6, which sends control data to the Pi Zero and also handles audio output.



### >STEP-03

#### Pure Data programming

All the synthesiser sounds are processed on the Pi Zero using Pure Data, a visual programming language in which objects can be created and linked together.



**POPPY MOSBACHER**

A FabAcademy student with an MA in Sustainable Design who won a Blue Peter robot design competition, aged eight. [magpi.cc/2xRmUJR](http://magpi.cc/2xRmUJR)



When someone stands inside the frame, Take Photo is pressed on the laptop software, and the images are taken and saved

Pi Zero Ws and cameras are housed in laser-cut 5 mm single-walled corrugated cardboard

Tubes, of varying lengths, are attached to each other using connectors and self-adhesive touch fasteners

# 3D BODY SCANNER

**Poppy Mosbacher** has created a relatively inexpensive full-body 3D scanner, and she hopes maker groups will enjoy replicating her project!

**Quick Facts**

- The project's Pi boards are named after Marvel characters
- Poppy wanted the build to cost less than £1000
- The images are processed using Autodesk ReMake
- Build Brighton members got fully involved in the project
- It uses mains electricity, so be careful!

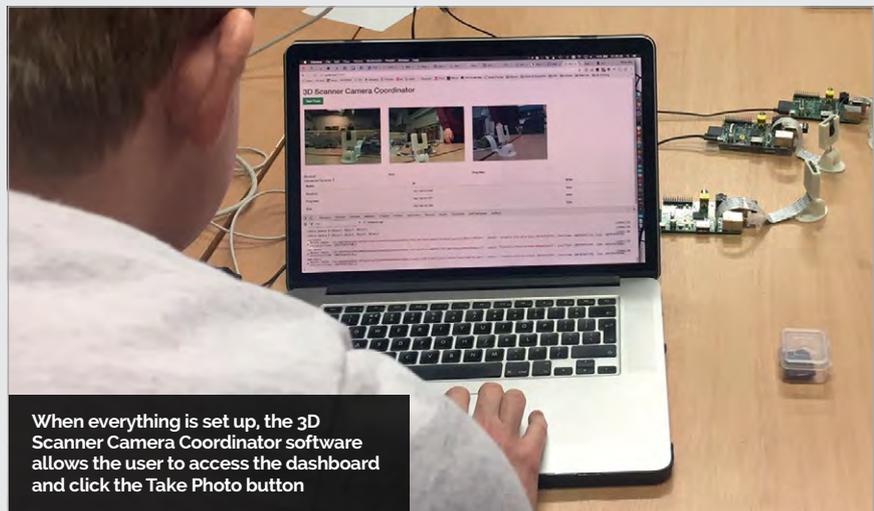
**W**hile learning to make her own clothes, Poppy Mosbacher wanted to visualise how they would look before she made them, and she began to think about how digital technology tools could help.

At first, she considered using a body scanner, but after talking to a friend, she learned that high-end 3D scanners using DSLR cameras can cost as much as £40 000. Just as bad, the cheapest alternative – of simply walking around an object or person and taking lots of photos with a single camera – proved slow and frustrating.

It was then that her friend and member of the not-for-profit makerspace Build Brighton, Paul Hayes, suggested that it might be easier to make a DIY version of a 3D scanner using Raspberry Pi boards.

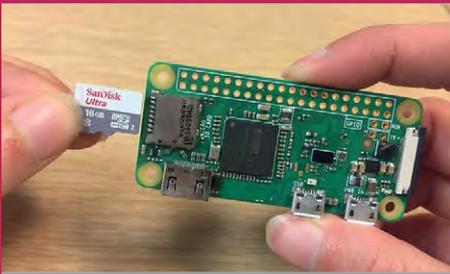
Before long, Poppy had secured a £1000 grant from Santander, which she used to buy 27 Pi Zero Ws (each snapped up by a different Build Brighton member to get around the one-per-customer rule). She

also bought 27 Camera Modules, 27 Pi Zero camera cables, and 27 USB to micro USB cables, as well as an assortment of battery packs, power regulators, wire connectors, and other electrical items.



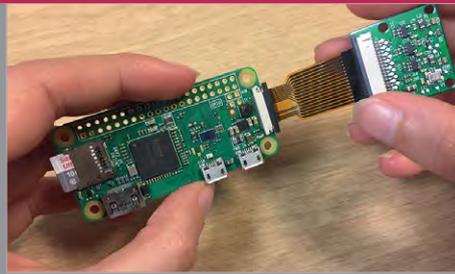
When everything is set up, the 3D Scanner Camera Coordinator software allows the user to access the dashboard and click the Take Photo button

## PREPARING THE PIS



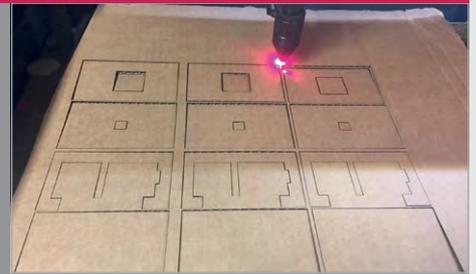
### >STEP-01 Code the Pi

Prior to her project, Poppy had only used the Raspberry Pi once, at a workshop at the Mozilla Festival in 2015. Arthur Guy wrote the actual code, which needs to run constantly.



### >STEP-02 Connect the camera

After setting up the server so that the cameras know which fixed IP address to send the photos to, a computer is connected. The cameras are then hooked up to the Pi Zero Ws.



### >STEP-03 House the Pis

The Pi Zero Ws and the cameras are housed in these cardboard cases, which are then placed around the structure. A 5V power regulator can be connected to up to three Pi boards.

### Cardboard engineering

Inspired by Richard Garsthagen ([magpi.cc/2xVr3Vr](http://magpi.cc/2xVr3Vr)), Poppy then looked at making the scanner affordable and portable. “The idea of having a portable rig that people could step into and take a picture in a few seconds was appealing,” she says. By using Zero Ws, she hoped

### Coding challenges

“The photos are sent wirelessly to the laptop and they are automatically saved in a new folder,” Poppy explains. Raspbian Jessie Lite is installed on the Pi Zero Ws, and the main server runs a node application. Another friend, Arthur Guy, wrote the code for the scanner

“Twelve 3 mm thick cardboard tubes were used, on which the Pis and their cameras were mounted”

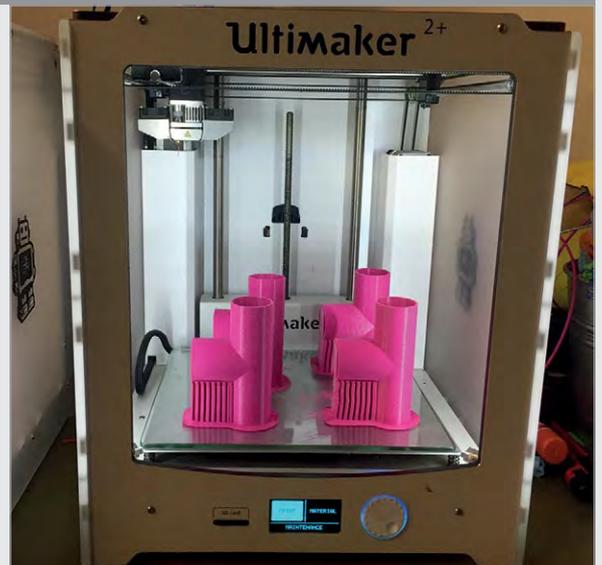
the scanner could be replicated in the future for less than £1000. To keep costs low, and to make the build easier for Poppy and the team, twelve 3 mm-thick cardboard tubes were used, on which the Pi Zero Ws and their cameras were mounted.

Poppy says the cardboard proved to be “a great material to cut and make holes in.” As well as the cardboard frame, a cardboard case was designed for the Pi Zero Ws. “It also helps hide the wires,” she adds. Created to work within the smallest possible diameter so that it can remain portable, the idea was to connect the Pi boards to a laptop to trigger the photos, so that all the cameras would take a snap simultaneously.

in JavaScript, building it up week by week, and adding features such as getting the Pis to look for updates on startup so that they all use the latest version of the software.

There were still some issues along the way, and a fair bit of trial and error, especially in positioning the cameras so that the photogrammetry software could digitally stitch the images together. There was also a problem with Poppy’s shiny long hair, which became apparent when she stood inside the structure. “I looked online, and it suggests putting powder on anything shiny, but I haven’t tried it yet.”

Some problems proved easy to overcome. Figuring out which



**Above** Eight cross joints and four T-junctions were 3D printed using an Ultimaker 2+

cameras weren’t working was solved by assigning them names, and Arthur also learned to change the white balance on the cameras to improve the image quality. Yet there are problems with time lag. “Some Pis take a photo instantly, and others take a few seconds, so we have to stay still until all the photos have been taken,” Poppy says.

Nevertheless, she is excited about future applications for the project. “It opens up new possibilities, such as of scans of children who won’t stay still long enough for the single camera method; building a personal database of scans taken at regular intervals to see the effects of aging; and making avatars for VR environments.”



**KEVIN S LUCK**

Kevin S Luck is a PhD student at Arizona State University working on robots and new machine learning methods with Joseph Campbell and Michael A Jansen. [magpi.cc/zyzh4pu](mailto:magpi.cc/zyzh4pu)

**Quick Facts**

- The total cost of a C-Turtle is about £50
- The goal is to create an autonomous fleet
- The experimental learning algorithm was written in MATLAB
- C-Turtle learned to move optimally in an hour
- It will blow up discovered land-mines – and itself

# C-TURTLE

Building a Pi-based robot only to blow it up may sound like a waste, but this mine detector could save lives

**I**t's a sad truth, but right now the world is littered with an estimated 110 million land-mines. Clearing them all could take as long as 1000 years and cost \$30 billion, but leaving them in situ is not an option. The number of people killed or injured by these hidden weapons recently reached a ten-year high – so how amazing would it be if the Raspberry Pi could help tackle this ever-present problem?

**Cardboard demining**

Scientists at Arizona State University have been putting their heads together to do just that. They have devised the C-Turtle, a cardboard robot with turtle flippers

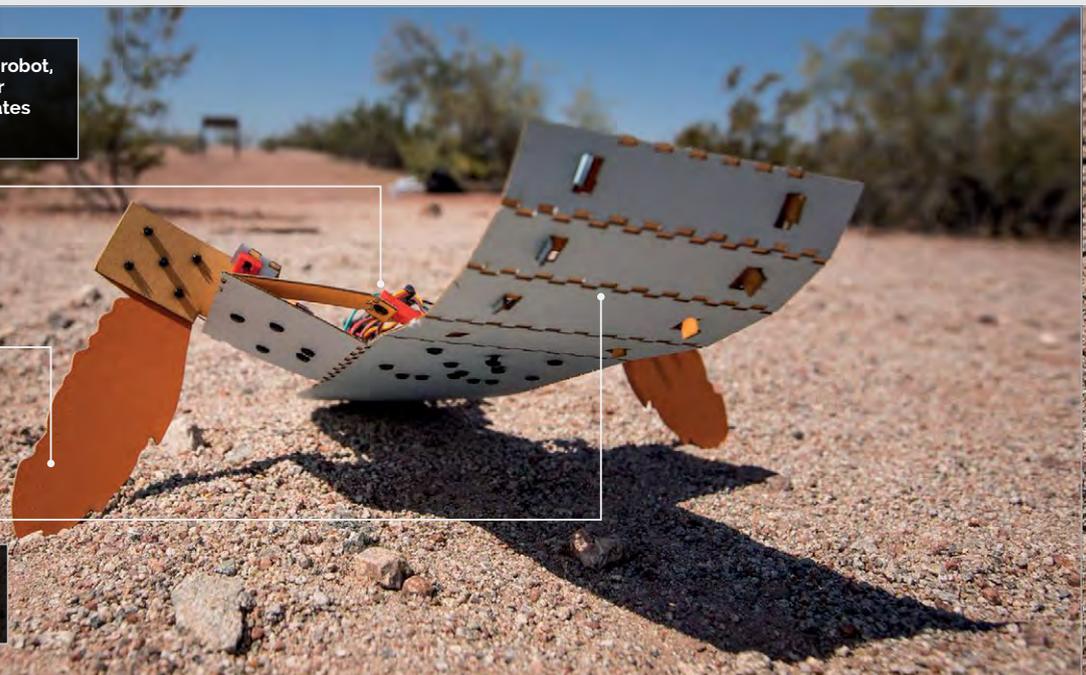
which has a Raspberry Pi at its heart. It uses machine learning to figure out how to walk across the most unusual and hazardous of terrain, constantly adapting to its surroundings. Modelled on a sea turtle (hence the name), it is not only inexpensive, but easy to transport.

“We were looking to develop a cheap and simple robot for the detection of land-mines,” says PhD student Kevin S Luck, who has worked on the project with Joseph Campbell and Michael A Jansen. “Undetected land-mines are a problem in many countries, and often these mines are particularly difficult to detect in sandy environments. The problem

is that sand in a desert moves over time and so the location and depth of the land-mines is constantly shifting.”

**Inspired by nature**

The C-Turtle is well equipped to cope with this issue. Housed within a single-sheet laminate comprised of layers of paper, foil and adhesive, it mimics the movement of a sea turtle. The scientific trio had noted how quickly sea turtle hatchlings can move over sand and how adults crawl while lifting their immense weight. This led to Michael developing a workable fin shape, and Kevin and Joseph figuring out how the Pi could best power the robot.



A Raspberry Pi Zero powers the robot, while a 16-channel, 12-bit motor driver from Adafruit communicates with the moving parts

This pair of laterally mounted fins moves with two degrees of freedom

The front end is curved to prevent it from digging into the ground as it moves

“We envisioned a system where each robot can carry sensors to detect and mark land-mines, but also where the loss of a single robot is relatively inconsequential for demining operations, thus reducing the risk for humans or bigger demining robots,” explains Kevin. During the design process, some key decisions were made. They ruled out using wheels – “they usually have issues with slippage on sand, and they would create a more complex manufacturing process,” says Kevin – and were unanimous in wanting to use a Raspberry Pi Zero.

### Lightweight connectivity

“The Pi felt perfect,” Kevin continues. “We not only wanted the ability to send commands to the robot via WLAN, but also to perform simple data processing and machine learning directly on the robot – a requirement for using multiple robots in a fully autonomous fleet. The Zero also requires relatively little power. Because of that, we’re exploring the possibility of using solar panels for recharging batteries during the daytime.”

**Below** Tests are ongoing to find out how well the robot, and the Pi, cope in extreme temperatures

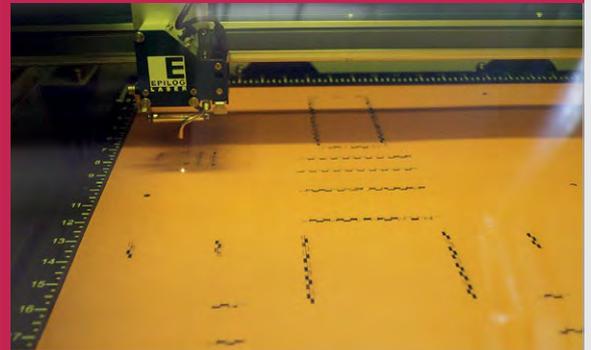
Kevin and Joseph have worked on an algorithm which allows the turtle bot to adapt its crawling technique. “The whole code infrastructure on the turtle robot, from motor control to the joint server and sensor collections, was written in Python,” Kevin reveals. “We used TCP/IP connections to send joint commands to the robot and also to collect data for evaluation.”

### Real-world learning

This was put to the test when they drove out into the desert with their first prototypes. “We got a real-time feed of what was happening with our robot, and were able to test and debug different variations of the learning scenarios,” Kevin tells us. By using trial-and-error learning, the robot gets good and bad feedback which enables it to develop.

Through this process, the robot has managed to work out effective trajectories over poppy seeds as well as sand, but the scientists are continuing to refine the technology and their ambitions remain high. “We’d like to take the robots into space, too,” says Kevin. “It would be fantastic to use them to explore Mars.”

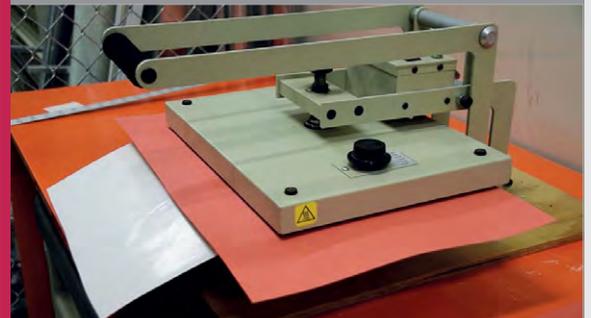
## CREATE THE C-TURTLE'S BODY



### >STEP-01

#### Laser-cut the layers

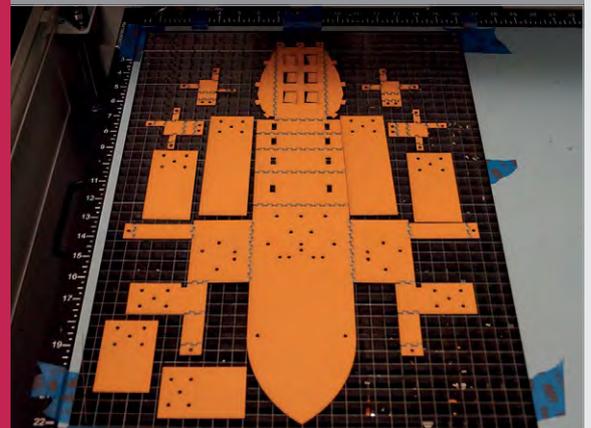
The cardboard layers are laser cut. For each of the five layers (two cardboard, two adhesive, and one foil), holes are cut in specific locations to allow hinges to be fitted later.



### >STEP-02

#### Begin the lamination

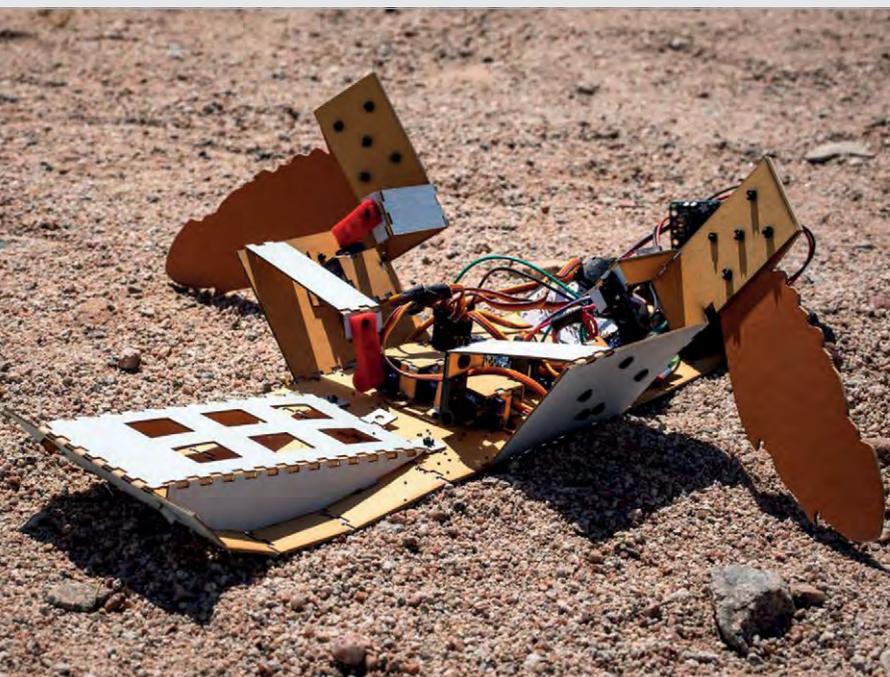
Once the layers are cut, they are laminated together to form a single layered sheet using a heating press.



### >STEP-03

#### Ready for assembly

The shapes of the individual parts are cut from the laminated sheet. The holes are mounting holes, designed to be used with rivets.





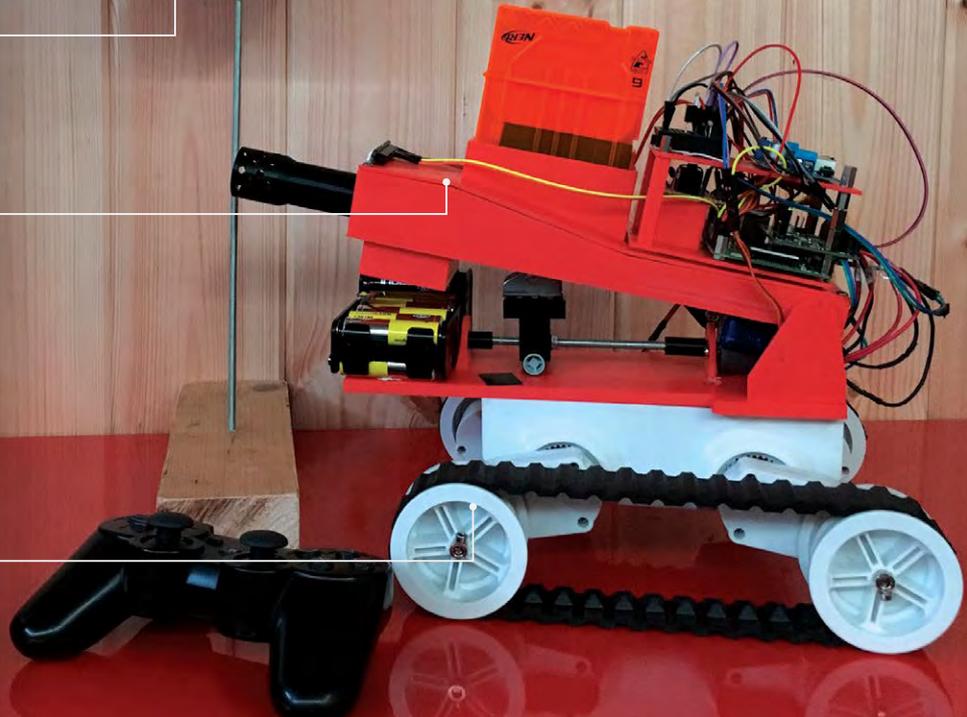
DAVID PRIDE

Currently studying for a PhD with the OU, David is a well-known member of the Pi community. Previous projects include a Lego-sorting robot arm and Connect 4-playing robot. [piandchips.co.uk](http://piandchips.co.uk)

Wall-mounted alien targets drop down when shot by FRED-209

The custom 3D-printed firing mechanism shoots the foam darts from a Nerf magazine

Built on a Rover 5 chassis, the robot has caterpillar tracks for extra grip



Quick Facts

- > It currently runs on 14 AA batteries
- > The firing mechanism is mainly 3D-printed
- > David plans to add a camera
- > He'll upload the STL files for 3D printing
- > FRED-209 is sadly too big for Pi Wars

# FRED-209

Say hello to **David Pride's** Nerf-toting robot! You have 20 seconds to comply...

**W**hile well-known Pi community member David Pride admits that Nerf guns hadn't been invented when he was a youngster, his interest was sparked when he saw two tables full of Nerf gear at a car boot sale. "I started wondering whether you could operate the trigger mechanism with a servo – turns out you can!"

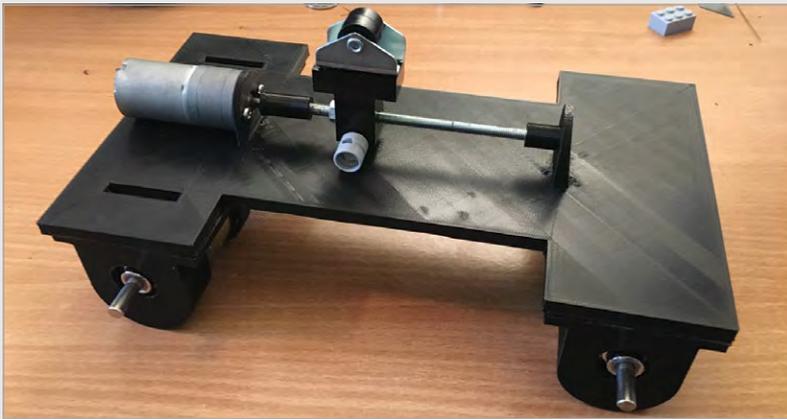
Following some successful experiments firing smaller, single-shot Nerf guns using a servo, David turned his attention to larger Nerf

models. "[I] realised that there are essentially two types: the pump action ones and those which use two flywheels to propel the dart. I didn't know exactly how the mechanism worked until I actually took one to bits!"

Initially, David simply strapped an upside-down Nerf gun to the top of his 2017 Pi Wars robot, X-Bot. "I realised that this wasn't really going to cut it, so set about designing and 3D-printing a complete setup that I could mount on top of a bot. It uses the original

Nerf flywheels, and the original Nerf magazine which can hold six darts. The rest is all 3D-printed. I also designed a simple mechanism to translate servo movement into lateral movement to push the dart into the launcher."

During the two months of evenings and weekends he spent working on the project over the summer, several design changes were made. "The biggest disappointment was that the huge chunky motors I had didn't have enough torque to turn the



**Above** A long threaded bar converts motor rotation into lateral movement to tilt the whole firing mechanism (not shown) up or down for aiming purposes

bot successfully. Getting the right motors is an area where I am definitely still learning and it's a critical factor to get right in building a successful bot."

Since he'd already bought a Dagu Rover 5 chassis, David opted to mount his Nerf mechanism on that until he obtained some stronger motors. He also dropped the

the Raspberry Pi. A tilt mechanism for aiming is controlled by the joystick's shoulder buttons.

David describes the robot's public debut at the Cotswold Jam as 'controlled chaos'. "It went down extremely well. I built some 'evil alien' targets to give the participants something to aim at – apart from each other!"

“ It uses the original Nerf flywheels, and the original Nerf magazine which can hold six darts ”

original chunky wheels in favour of smaller ones with caterpillar tracks.

Controlled manually using a wireless PS3 joystick, FRED-209 can fire multiple foam darts at the chosen target(s). Its twin motors are driven using a ZeroBorg board, while the firing servo is connected directly to the GPIO 18 PWM pin on

Continuing work on the project, David plans to power it with LiPo batteries – “It currently runs on 14 (!) AA batteries which really don't last very long as the drive motors and flywheel motors are pretty greedy.”

He also plans to add a camera to enable FRED-209 to find and fire at targets automatically. “I did some very simple vision processing for 4-Bot, my Raspberry Pi Connect 4 robot, but this takes it up several levels. I am currently learning OpenCV and SimpleCV for vision processing... The plan is the bot will recognise colour/shape to locate target. I can see it working well as a burglar detector... providing the burglar is wearing a black and white stripy shirt and carrying a bag marked 'swag'!”

**Left** David originally tried mounting a standard Nerf gun on the top of a robot, but soon realised that it wasn't really going to work that well



## BUILD A NERF DART-FIRING ROBOT



### >STEP-01

#### Servo pusher

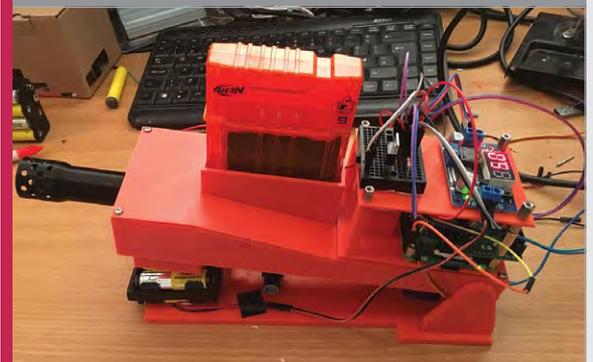
David 3D-printed most of the parts for his Nerf firing mechanism, including the servo pusher. An arm connected to the servo moves a rod forward to push the dart from the magazine.



### >STEP-02

#### Flywheel flinger

The dart is pushed into twin rotating flywheels which propel it rapidly down the barrel. These, along with the robot's drive motors, require a lot of battery power.



### >STEP-03

#### Completed mechanism

The 3D-printed lid has a slot to load the upside-down Nerf gun magazine. A roller underneath moves along to tilt the whole mechanism up and down to aim.

MART DRAKE-KNIGHT



Mart and his brother Rob want to make fashion more sustainable and they love organic cotton and renewable energy.  
[rapanui clothing.com](http://rapanui clothing.com)

# RAPANUI CLOTHING

## Quick Facts

- Rapanui's factory plugs into the Internet of Things
- All workstations and machinery are Pi-powered
- The UI has been made mistake-proof
- IBM's Node-RED is free and open source
- Fire destroyed the previous factory in 2016

**Mart Drake-Knight** and his brother **Rob** have put dozens of Raspberry Pi boards at the heart of their super-efficient clothing factory

**W**hen brothers Rob and Mart Drake-Knight struggled like mad to find a job in 2008, they could so easily have reached a point where they would have to sell the shirts off their backs. Luckily they decided to spend £200 setting up an environmental clothing company on the Isle of Wight instead, growing it from a garden shed to a large factory in a former Co-op supermarket.

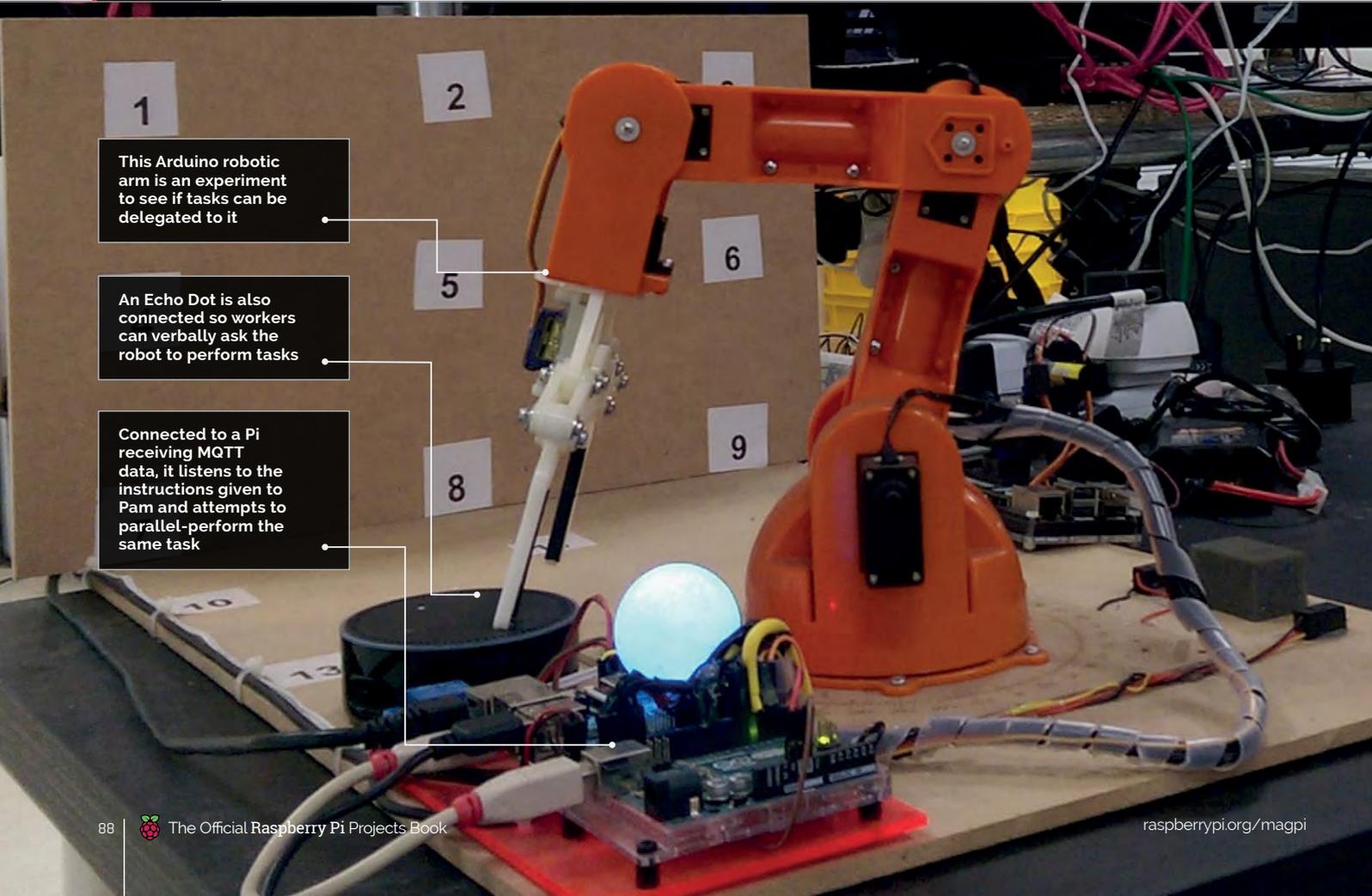
Today, the whole operation at Rapanui is powered by renewable energy and a creative, hackathon,

DIY spirit. More specifically, it also makes use of nearly 100 Raspberry Pi boards to perform many different tasks. "Some are connected to sensors or machinery that we've made in our machine shop," says Mart. "Some do simple stuff like open windows and help people solve problems."

Other Pi devices drive touchscreens and workstations and they enable staff to prioritise work. Put together, they have made Rapanui's manufacturing process very efficient while allowing the

firm's eco-friendly and sustainable clothing to be affordable. By automating the dull parts of the job and giving managers a solid overview of what's happening in the factory and where, Rapanui can go as far as offering tailored services: you could get the company to make a single T-shirt with your picture or logo or order 1000. Either way, you could have the garment the following day.

In order to bring the technology in the factory together, Rapanui uses a lightweight messaging protocol for



1 This Arduino robotic arm is an experiment to see if tasks can be delegated to it

5 An Echo Dot is also connected so workers can verbally ask the robot to perform tasks

8 Connected to a Pi receiving MQTT data, it listens to the instructions given to Pam and attempts to parallel-perform the same task

small sensors and mobile devices called MQTT and this allows the Pi boards to broadcast their behaviour to other machines or human interfaces. Using Node-RED, the hardware devices, APIs and online services are wired together.

“We use Node-RED in live production to broadcast order data between two specific machines that are time-critical,” Mart tells us. “More recently we used Node-RED for cool stuff like joining an Alexa API to our factory systems so that you can ask the factory questions such as how many jobs are pending, which lets us direct staff to the areas where they are needed. It also controls some robotic equipment we’re working on.”

### Creating tech jobs

The firm has since built on that system to produce **Teemill.com**, which lets users create an online store, sell T-shirts, and take the profits of each sale. “People can join and access the API directly via Teemill,” Mart says. But just as importantly, Rapanui’s approach to the development of its clothes has created desirable tech jobs on the Isle of Wight, with the company able to show its young



employees how to code through programming apprenticeships.

“The Pi has given us a great way to build confidence at the early

time production systems which allows products to be made in the seconds after they are ordered.

“The breakthrough has massively

**Above** Running Rapanui’s factory is a complex business, hence the mass of Pi-powered screens at this computer terminal

“ The whole operation is powered by renewable energy and a creative, hackathon, DIY spirit ”

stage in people who want to start a career with us in computing,” continues Mart. Indeed, today the company invests a lot of its time developing automated, real-

reduced waste and cost,” Mart tells us. “And since we’ve doubled in size every year and the Raspberry Pis are modular, they have scaled with us.”

## GETTING THE PROCESS DOWN TO A ‘T’



### >STEP-01 Creating an order

A customer can create any product they like at Teemill.com. When a T-shirt is ordered, Holly picks the order. The Pi runs the interfaces, which are connected via Node-RED/MQTT.



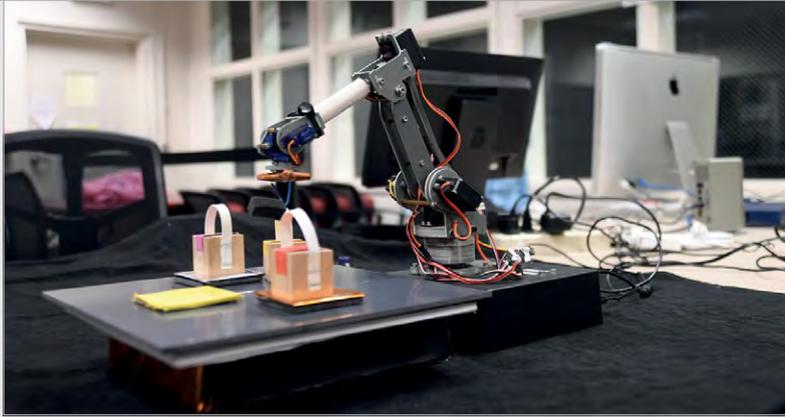
### >STEP-02 Printing the T-shirt

The lights change colour, alerting workers to the location of the next task. The workstation interfaces broadcast the machine’s status to the MQTT broker, which is run centrally on a Pi.



### >STEP-03 Sending it out

When the T-shirt is ready, Pam scans and packs it. The interface tracks her work, broadcasting the status back to the systems. The customer is notified that the order is on its way.



**DAN GRIBOK**

A robotics intern at NASA's Goddard Space Flight Center, Dan is currently pursuing a graduate degree in Aerospace Engineering at UMD focusing on Telerobotics and Exoskeleton technology.  
[sspd.gsfc.nasa.gov](http://sspd.gsfc.nasa.gov)

**Left** The manipulator arm tip can also be equipped with a hook to pick up blocks with loops attached

# NASA ARM

With the ability to draw any shape, this robot arm is out of this world

## Quick Facts

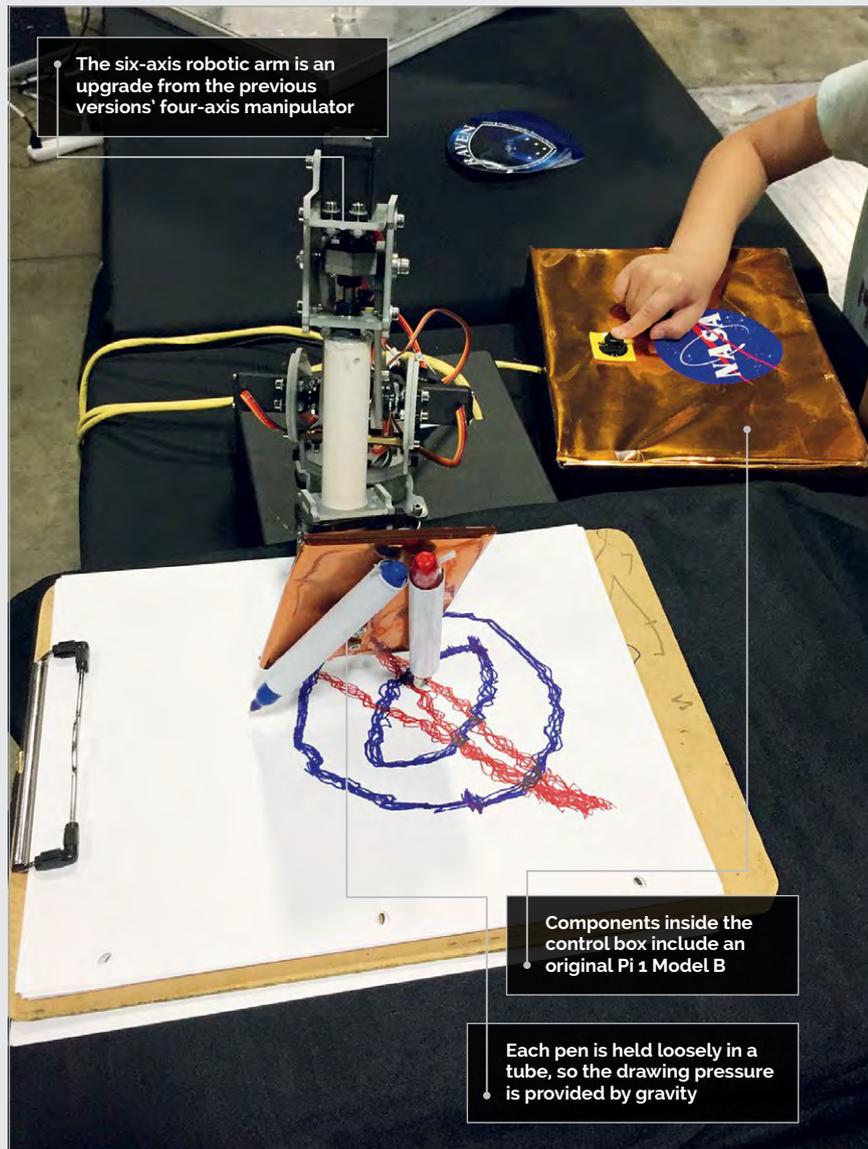
- ▶ It can draw any shape using x-y coordinates
- ▶ Dan took around a year to refine the software
- ▶ The arm can also be controlled manually
- ▶ It's powered by an original Pi 1 Model B
- ▶ A PCA9685 board drives the arm's servos

**A** hand reaches out to a shiny gold box emblazoned with the NASA logo. At the press of a button, the connected robotic arm springs into life, lowering a blue Sharpie pen onto a paper pad to draw the circular outline of the famous logo, then an inner 'spacecraft orbit' ellipse, before switching to a red pen to sketch the red chevron ([magpi.cc/2zHatI7](http://magpi.cc/2zHatI7)).

This is the Advanced Robotic Manipulator System Tools and Resources – or 'ARM' for short – built and programmed by Dan Gribok, a robotics intern at NASA's Goddard Space Flight Center within the Satellite Servicing Projects Division. Designed to be used at outreach events to educate the public in what NASA is doing with robots, the Pi-powered ARM is a versatile device that can also be controlled manually using an Xbox gamepad to pick up objects using a hook or grabber.

## Art for events

Dan tells us that the drawing ability came about from a need to have a constantly running demonstration at events, such as Awesome Con in Washington DC. "So we wanted a demo where you could just push a button, step back,



The six-axis robotic arm is an upgrade from the previous versions' four-axis manipulator

Components inside the control box include an original Pi 1 Model B

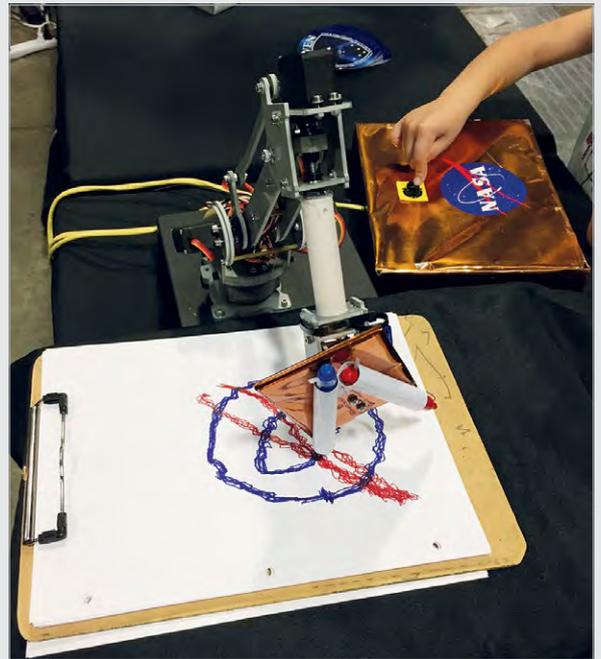
Each pen is held loosely in a tube, so the drawing pressure is provided by gravity

and everyone could watch all at once. And you can also give away a sheet of paper at the end and have them say, ‘Hey, a robot drew this – how cool is that?’”

Asked about what the ARM can draw, Dan says it depends on what he can program into it and what the hardware can actually replicate. He admits that the drawing action is a little jerky, which is due to the hardware itself: “There’s a lot of slop in the joints, a couple of

The force of gravity is used to get the pen pressure right, as each Sharpie pen is held loosely in a tube. “We previously tried some other spring-like mechanisms, but we just settled on gravity drawing because it’s so much easier and it works perfectly fine.”

So, what’s in the shiny control box? Somewhat surprisingly, along with a PCA9685 I<sup>2</sup>C servo controller and other components, the ARM uses an original Raspberry Pi 1 Model B – as did the two previous



**Above** Pressing the button causes the arm to start drawing a preprogrammed sketch, in this case the NASA logo

“ We wanted a demo where you could just push a button, step back, and everyone could watch ”

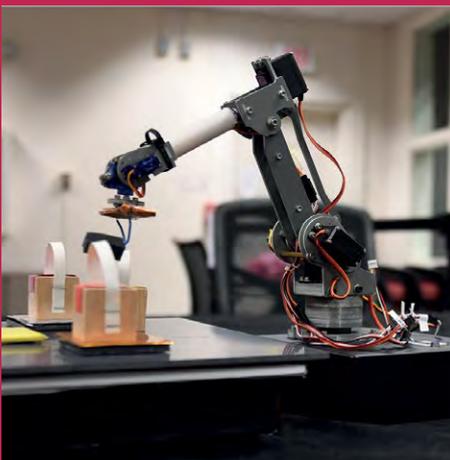
degrees, and some slop in the other mechanisms, which makes it really hard to draw [smooth] curved lines.” However, since it understands positions and coordinates, “Any line that you can express as x and y coordinates, so virtually anything, you can program into it and it’ll just follow that over and over.”

versions of the project, which were based on OWI Edge robot manipulators. In autumn 2016, Dan was granted permission to upgrade the project’s robots: “I kept about half the internals of the control box and got completely new robot manipulators.”

While the hardware setup was relatively straightforward, the

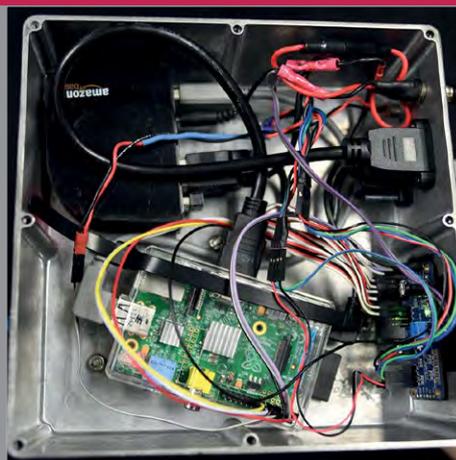
software took a lot longer. “The original robots were programmed in Python, but I noticed that we had performance issues, so we switched entirely to C++.” After getting basic robotic functionality working by spring 2017, Dan made improvements over the summer, including adding the drawing capability.

## BUILD A DRAWING ROBOT



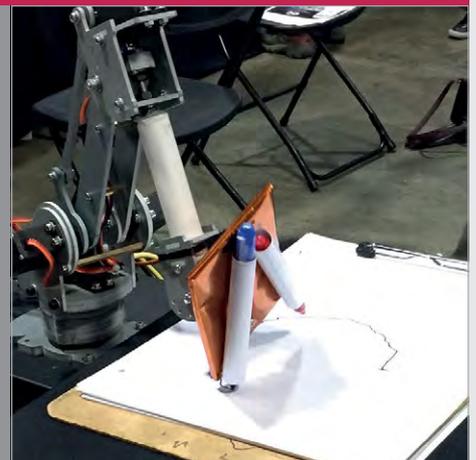
### >STEP-01 Manipulator arm

Dan replaced the four-axis OWI Edge arm of the earlier robots with a six-axis Sain Smart manipulator. It features four standard MG996 55g metal-gear servos and two SG90 servos.



### >STEP-02 Control box

Inside the control box, a Pi 1 Model B is hooked up to a PCA9685 breakout board to drive the servos. Along with status LEDs and wiring, there’s a USB hub to supply power to both the Pi and the arm.



### >STEP-03 Drawing mechanism

The tip of the arm rotates to lower one of two Sharpie pens for drawing. Each is held loosely in a tube so that the force of gravity provides enough pen pressure to mark the paper.



**MATT BRAILSFORD**

A web developer by day and hardware hacker by night, Matt (aka Circuitbeard) is a Barnsley-based maker who loves to hack old eighties tech. [circuitbeard.co.uk](http://circuitbeard.co.uk)

# TOMY TURNIN' TURBO DASHBOARD OUT RUN

## Quick Facts

- The fuel gauge represents the stage time remaining
- The turbo light comes on in high gear
- A small vibrating motor was added for haptic feedback
- Matt tried four different screens before finding one that fit
- Matt hacked the speed to change it to mph from kph

Combining two classics into one amazing racing project, **Matt Brailsford** used a Raspberry Pi to build this incredible 1980s mash-up of racing toys

**W**ay back in 1983, a Japanese company called Tomy created one of the most remarkable toys of its generation: Tomy Turnin' Turbo Dashboard was a driving simulator complete with gears, ignition, a working dashboard, a steering wheel, and even a looping display.

Three years later, Sega released arguably (well, we'd argue it) the

greatest, coolest racing game of all time: Out Run.

One maker, Matt Brailsford, picked up a Tomy Turnin' Turbo Dashboard on eBay and had a spark of genius: why not turn it into a fully working Out Run arcade machine?

Matt removed the original display from the Tomy Turnin' Turbo Dashboard and replaced

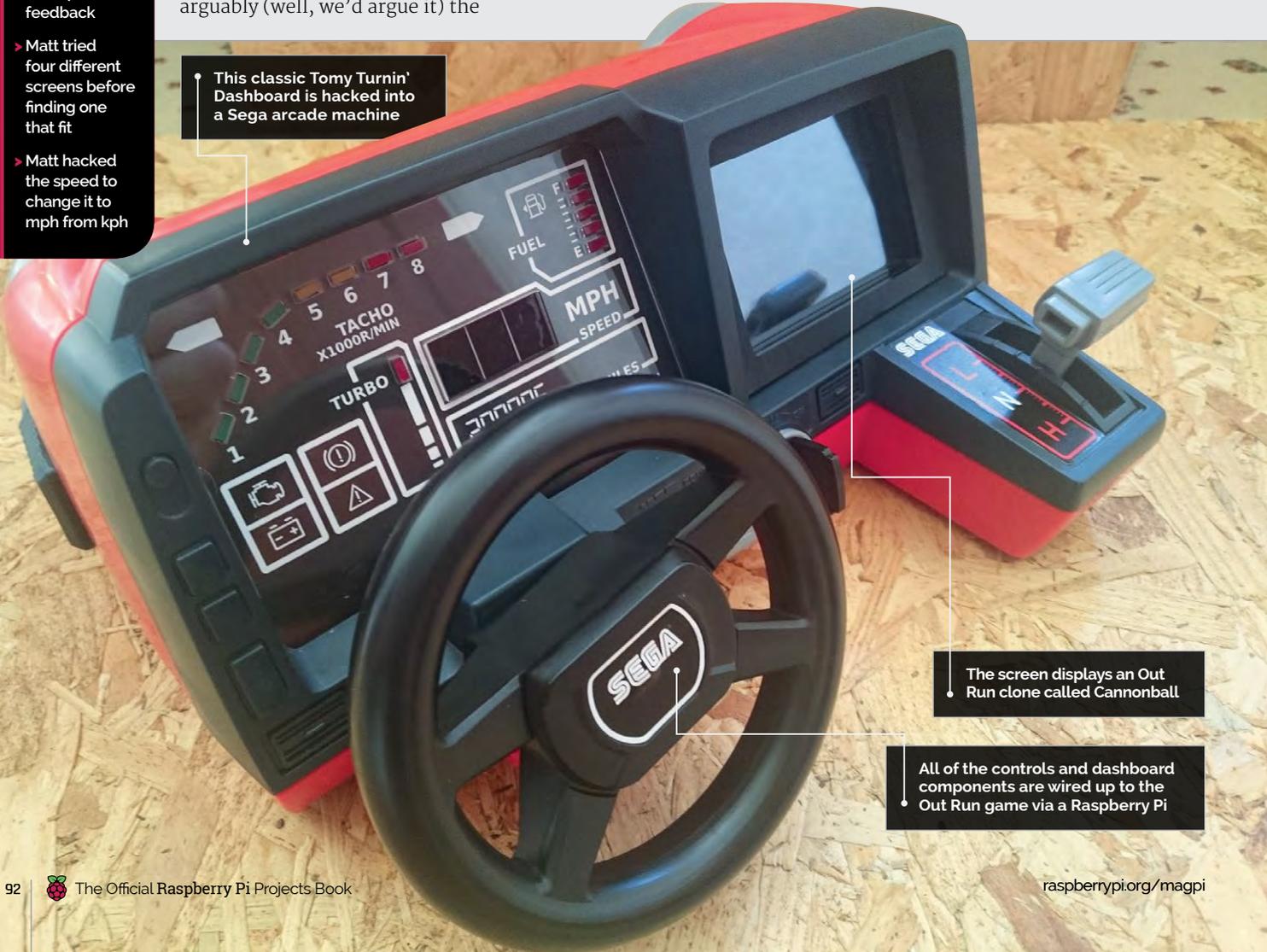
it with a modern screen. "I tried quite a few [screens] trying to get one that would fit," says Matt. His first attempt was a screen that worked from the GPIO, but this left few pins for all the other mods. "And the extra processing slowed the game down."

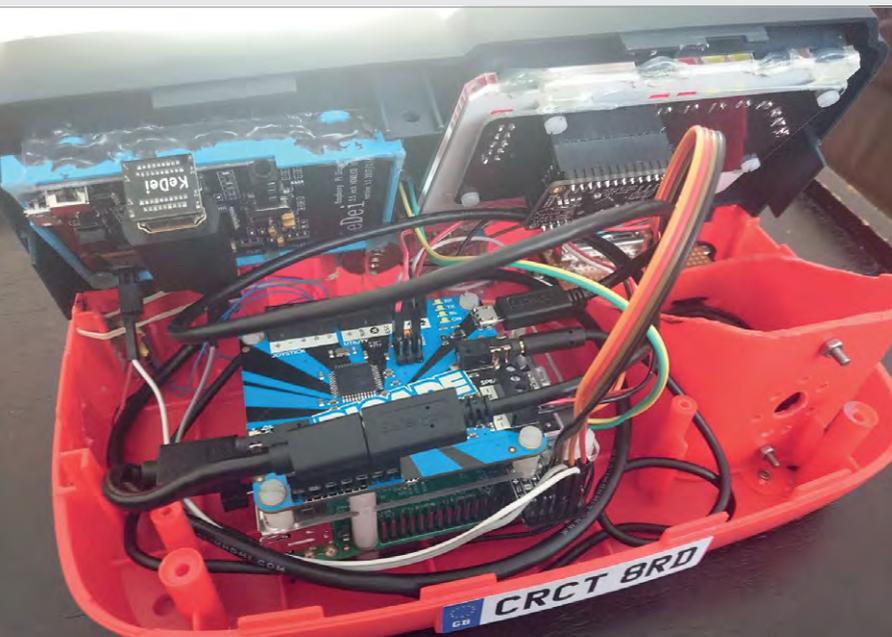
In the end, Matt used a KeDei 3.5-inch HDMI display ([kedei.net](http://kedei.net)).

This classic Tomy Turnin' Dashboard is hacked into a Sega arcade machine

The screen displays an Out Run clone called Cannonball

All of the controls and dashboard components are wired up to the Out Run game via a Raspberry Pi





A lot of components are carefully squeezed inside the original plastic case

The original ignition key from the game was already wired as an SPST switch, so Matt wired it up to the Raspberry Pi. To manage the safe powering up and down, he used a PowerBlock ([magpi.cc/2mLOf4P](http://magpi.cc/2mLOf4P)). “I simply hooked up the switch and installed the daemon to watch for the shutdown command and it all worked perfectly. A Picade PCB ([magpi.cc/29DpDCz](http://magpi.cc/29DpDCz)) was used to

time and thinking things through, then trying things out.”

### Cannonball running

You might be expecting the Raspberry Pi inside to be running an emulator like RetroPie, but no. It’s actually running an Out Run clone called Cannonball ([magpi.cc/2mL3bAi](http://magpi.cc/2mL3bAi)). Matt explains: “Cannonball is a C++ port of the original Out Run arcade

“ Ultimately it was just taking my time and thinking things through, then trying things out ”

hook up the rest of the controls. Mike tells us it was “really easy to hook these up”, especially because the Picade PCB supports analogue controls for the steering wheel.

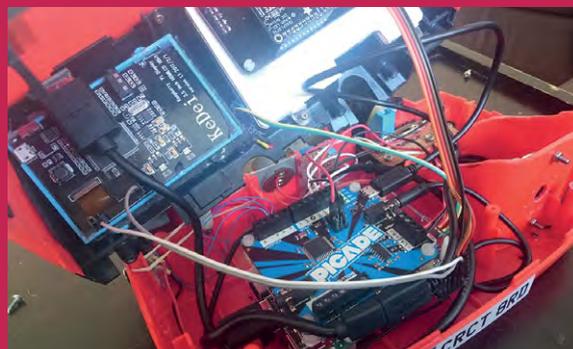
### Turn the key

“There was a lot of luck involved as well,” explains Matt. “Lucky that the slide potentiometer happened to sit at the perfect height, lucky that there was just enough room for some microswitches to fit down the side of the gear shifter, and lucky I could balance the shifter rubber bands to keep the shifter centred... Ultimately it was just taking my

game and luckily ran perfectly on a Raspberry Pi 3. The reason I went with this rather than, for example, MAME is that it allowed me to change the core code and intercept the variables I wanted to use to update the dashboard. You could probably achieve something similar in MAME by inspecting the register values, but having access to the raw source code is way easier.”

Matt tells us it’s fun to play. “It was a little squeaky to start with, but I had just tightened everything up a little too much. It runs nicely and brings back those memories as a kid. It’s how I thought it felt to play with back then.”

## HACKING A TOMY TURNIN’



### >STEP-01

#### Hooking up controls

A Picade PCB controller is used to manage the steering wheel and gearstick, while the ignition key is connected directly to the Raspberry Pi via a PowerBlock microcontroller.



### >STEP-02

#### Fitting a screen

Getting a screen that fitted was the biggest challenge. This KeDei 3.5-inch screen fits directly into the cabinet and connects to the Pi’s HDMI socket (leaving the GPIO pins free).



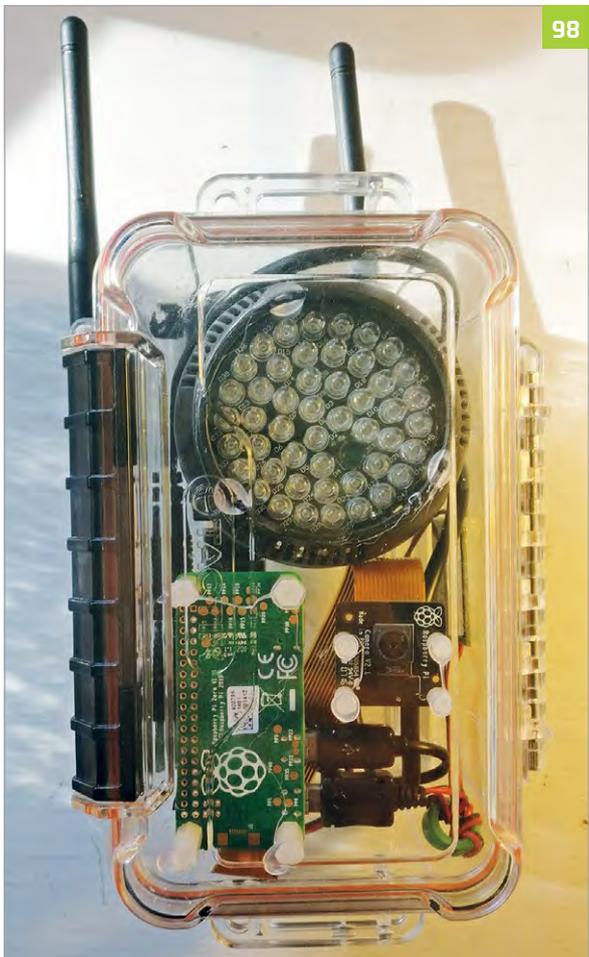
### >STEP-03

#### Connecting the dashboard

Matt ended up recreating the entire dashboard with specific cutouts for LEDs and other components. These respond to the in-game action for an authentic arcade experience.

# TUTORIALS

Get making with our expert guides and learn how to make the project of your dreams





132



134



154

## Tutorials

- 96 MAKE A PI SPY CAM**  
This simple project lets you create a nanny cam in your own home
- 98 NIGHT-VISION WILDLIFE TRAP**  
Create a camera that works in the dark to snap animal photos
- 102 BUILD A WHOOP! CUSHION**  
This classic prank has been updated for modern times with a Raspberry Pi
- 106 SAVE YOUR WINE**  
A room temperature sensor lets you know if your wine cellar is OK
- 108 GET UP WITH A SUNRISE ALARM**  
This sunrise alarm clock gets you up with the sun for a better morning
- 110 USING HUE LIGHTS WITH PI**  
Control your Philips Hue lights using this quick Raspberry Pi guide
- 112 TELL THE TEMPERATURE WITH STAIRS**  
This LED system shows you how hot it is as you go down the stairs
- 114 GIVE A ROBOT EMOTIONS**  
Not quite artificially intelligent, but it will recognise emotions and replicate them
- 116 ADD VOICE CONTROL TO A ROBOT**  
Use Alexa to add voice commands to operate a robot
- 120 BUILD A MAGIC MIRROR**  
Make the ultimate Raspberry Pi project with this thorough step-by-step guide
- 132 TAKE DEEPCREAM PHOTOS**  
Use Google's DeepDream to make trippy photos out of what you see
- 134 MAKE ART WITH RECORD SCRATCHING**  
Beautiful art is at your fingertips – literally – with this DJ Hero mod
- 138 FIND FOLKS WITH A MARAUDER'S CLOCK**  
Discover where your loved ones are using this Harry Potter-inspired project
- 140 HACK AN AMAZON DASH BUTTON**  
You can use an Amazon Dash button to do a lot more than order loo rolls
- 142 CREATE A CUT-OUT THEATRE**  
A digital cut-out theatre using JPGs and PNGs instead of paper
- 146 MAKE AN ULTRASONIC THEREMIN**  
Create space age music with this theremin using ultrasonic sensors
- 148 SAILING PI**  
Track your boating adventures with this GPS project
- 150 ADD SIRI VOICE CONTROL**  
Use Siri to add voice control options to any Pi project
- 152 AMIGA EMULATION GUIDE**  
Learn about emulating an Amiga the proper way on a Raspberry Pi
- 154 BUILD A RETRO GAMING HANDHELD**  
Make your very own Game Boy-like console with this big build



**ROB ZWETSLOOT**

Tinkerer, sometime maker, other-times cosplayer, and all-the-time features editor of *The MagPi*.  
[magpi.cc](http://magpi.cc) / @TheMagPi

# PISPY CAM

## You'll Need

- > Pi Zero
- > PIR motion sensor  
[magpi.cc/CWnNlI](http://magpi.cc/CWnNlI)
- > Camera Module  
[magpi.cc/bASFoG](http://magpi.cc/bASFoG)
- > Pi Zero camera cable adapter  
[magpi.cc/2gT2KwE](http://magpi.cc/2gT2KwE)
- > Portable battery

Set up a motion-activated spy camera in your room to find out if anyone's coming in when you're not there

**W**e've all been there. You've gone out for the day and you know you closed your bedroom door, but you come back and it's slightly ajar. Who's been in there? Were they friend or foe?

### >STEP-01 Connect up the camera

Attaching the camera to the Raspberry Pi Zero is easy – see the camera starter guide ([magpi.cc/QrMnng](http://magpi.cc/QrMnng)). You need to first switch out the cable in the Camera Module with the adapter for the Pi Zero. Gently pull down on the clasp that keeps the ribbon on the Camera Module attached and remove the cable. Take note of the orientation of the silver connectors on the

cable you removed, and make sure your adapter cable is the same way round when you insert it. Put the other, smaller side into your Pi Zero, with the silver connectors facing towards the board.

### >STEP-02 Wire up the circuit

The circuit for this is fairly simple, especially as the PIR doesn't need a resistor as part of its setup. The PIR comes with three connections: VCC, GND, and OUT. VCC needs to be connected to a 5V power pin, GND needs to go to a ground pin, and then there's the OUT wire which will be our input. We're connecting it to pin 8, also known as GPIO 14.

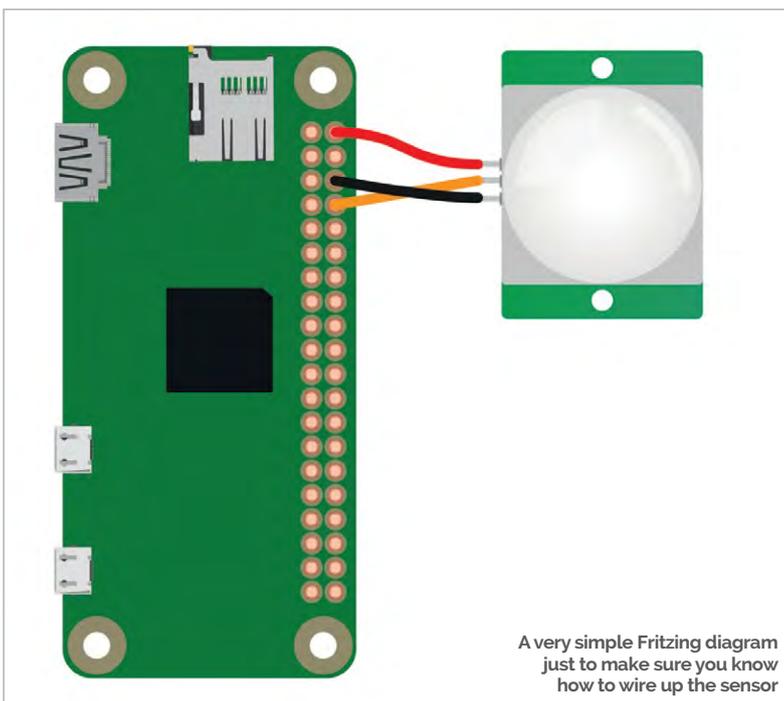
If your Pi Zero has GPIO pins attached, you can do this via a breadboard; otherwise you can loop the wire around the pin holes and use a bit of Blu Tack to keep them in place, or a dab of glue from a glue gun on a low setting. Soldering is also an option.

### >STEP-03 Get the code

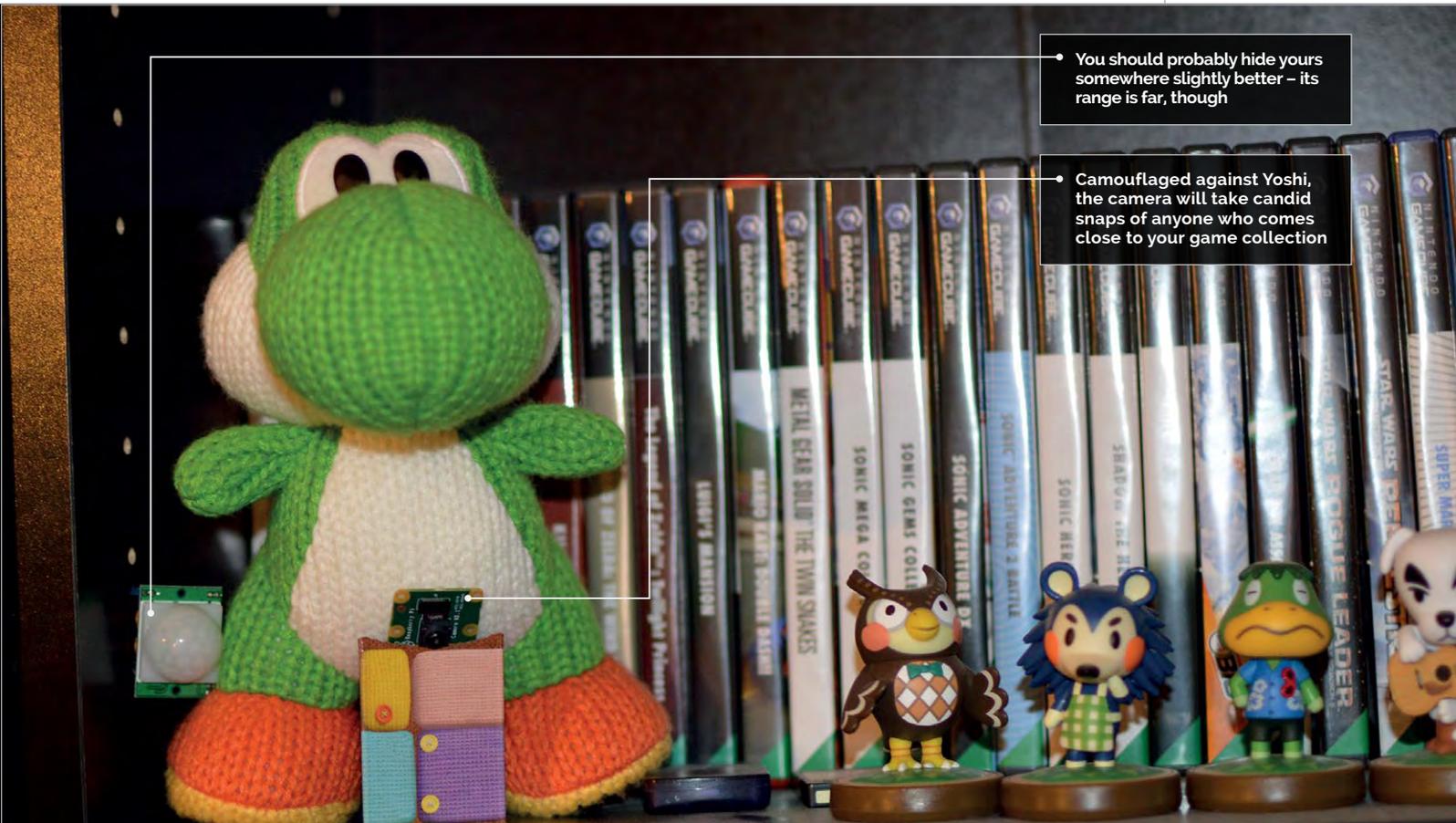
Write up or download our code for this project. This code uses two libraries: GPIO Zero and the standard picamera library. GPIO Zero can be used to get a reading from the PIR motion sensor very easily, which can then be tied into the picamera code so it takes a photo when motion is detected. We make sure the photo is given the time so you know when it was taken, and it then waits five seconds before seeing if it should take another photo. Save it as **spy.py** in the default home folder.

### >STEP-04 Final preparations

You can run it first to give it a test. You might want to change the sensitivity, which you can do by adjusting



A very simple Fritzing diagram just to make sure you know how to wire up the sensor



the little orange screws on the side of the PIR board. Once that's done, it's best to make it so that when the Raspberry Pi boots up, it logs directly into command-line mode. This means it will use up a little less power so your battery lasts longer. It's then a good idea to open up the terminal and edit the profile config file with `sudo nano /etc/profile`. To the bottom of the file, add this line:

```
sudo python spy.py
```

Now you need to find a good place to hide your camera

### >STEP-05

#### Hide your camera

Now you need to find a good place to hide your camera. The cable for the camera is limited by length, while the PIR can have its wires lengthened, so keep that in mind when building your system. Hiding the Pi and battery behind a plush toy or photo frame can work well; you could even put a dummy photo up and cut a hole in it for the camera to look through. The PIR has quite a wide range, so put it up high where people are unlikely to look.

## spy.py

```
#!/usr/bin/env python
```

```
from gpiozero import MotionSensor
from picamera import PiCamera
from datetime import datetime
from time import sleep
```

```
sensor = MotionSensor(14)
camera = PiCamera()
```

```
while True:
```

```
    sensor.wait_for_motion()
    filename = datetime.now().strftime("%H.%M.%S_%Y-%m-%d.jpg")
    camera.capture(filename)
    sleep(5)
```

Language

>PYTHON 3

DOWNLOAD:  
magpi.cc/PiSpy

### >STEP-06

#### Check for intruders

All you need to do now is plug in the battery and the Pi Zero will turn on and run the script. Do some tests to make sure the camera is facing the right way. Leave it running during the day and then when you get back, plug it into a monitor, stop the script, and run **startx** to get the GUI up. From here you can see the pictures it has taken – crucial evidence to catch your dog or sibling red-handed.



**HENRY BUDDEN**

Following the release of the Raspberry Pi when he was twelve, Henry taught himself to code and to use/break electronics, and has shared this process with the world on his website. [magpi.cc/zeCbaMf](http://magpi.cc/zeCbaMf)

**You'll Need**

- ▶ Raspberry Pi Zero v1.3 / W / WH, Raspberry Pi 1 B or B+
- ▶ Pi NoIR Camera Module [magpi.cc/2hND1tW](http://magpi.cc/2hND1tW)
- ▶ Pi Zero camera cable [magpi.cc/1V5Ngdh](http://magpi.cc/1V5Ngdh)
- ▶ USB to micro USB adapter
- ▶ Infrared illuminator [magpi.cc/2jxw3sT](http://magpi.cc/2jxw3sT)
- ▶ 12 V power supply
- ▶ Power supply adapters [magpi.cc/2hNFuEy](http://magpi.cc/2hNFuEy)
- ▶ 5V UBEC [magpi.cc/2hNMv8z](http://magpi.cc/2hNMv8z)
- ▶ WiFi dongle with antenna [magpi.cc/VPMjZm](http://magpi.cc/VPMjZm)
- ▶ Waterproof case
- ▶ Two short lengths of jumper wire

# MAKE A NIGHT-VISION CAMERA TRAP

Build a motion-detecting camera that streams to the internet and can see in the dark!



- Long-range antennae for WiFi connectivity anywhere in the garden
- See in the dark with infrared light only visible to the Pi NoIR Camera Module
- Full HD night-vision camera to record any secret antics!

**W**ith night-vision and the MotionEyeOS installed, the camera trap can be installed outside as a home security camera, a nature-trap, or a neighbour-watcher (careful!). The operating system MotionEyeOS allows the camera to be viewed online as well as alerting you when motion is detected. Built inside a waterproof case, and powered by mains electricity, the setup is also equipped with a high-power infrared illuminator to light up the darkness.

**>STEP-01**

**Prepare case**

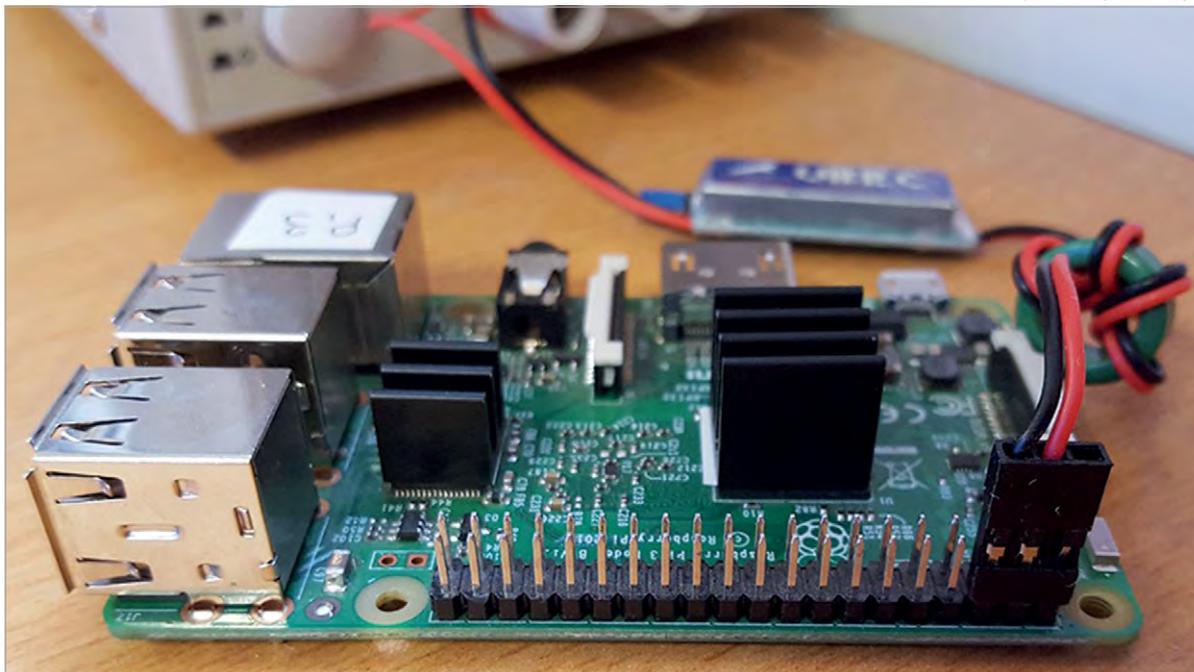
The first step is to drill all of the holes in the waterproof container of your choice (it must be transparent) in order to mount all of the necessary components. For our choice of case, we needed to drill two holes: one for the power socket and two for the WiFi antennae. Whilst the diameter and the distance needed between the holes for different WiFi antennae and power sockets may differ, we used a 7 mm bit for the power socket and drilled two 10 mm holes for the WiFi antennae. If the holes appear slightly too large once drilled, don't worry: the gaps will be filled with hot glue when mounting the components.

**>STEP-02**

**Wire the power circuit**

After preparing the case for the mounting of the components, the next step is to wire up the power circuit. The female barrel jack socket will be mounted inside the hole drilled earlier, with two connections from each screw terminal. One pair of wires connected to the negative and positive terminals will be connected to the male barrel jack connector screw terminals, with the other pair of wires connecting to the corresponding negative and positive inputs

Image courtesy of ModMyPi



## MOTION IN BACKGROUND

Ensuring that there are no moving objects in the background of the shot (e.g. trees swaying) will prevent hours of blank footage from being recorded.

**Left** Connecting the UBEC to the Pi via a pre-soldered header

on the UBEC, where the wires are already attached. Connecting the UBEC to the Raspberry Pi in order to supply power will be covered in the next step.

## >STEP-03 Powering the Raspberry Pi

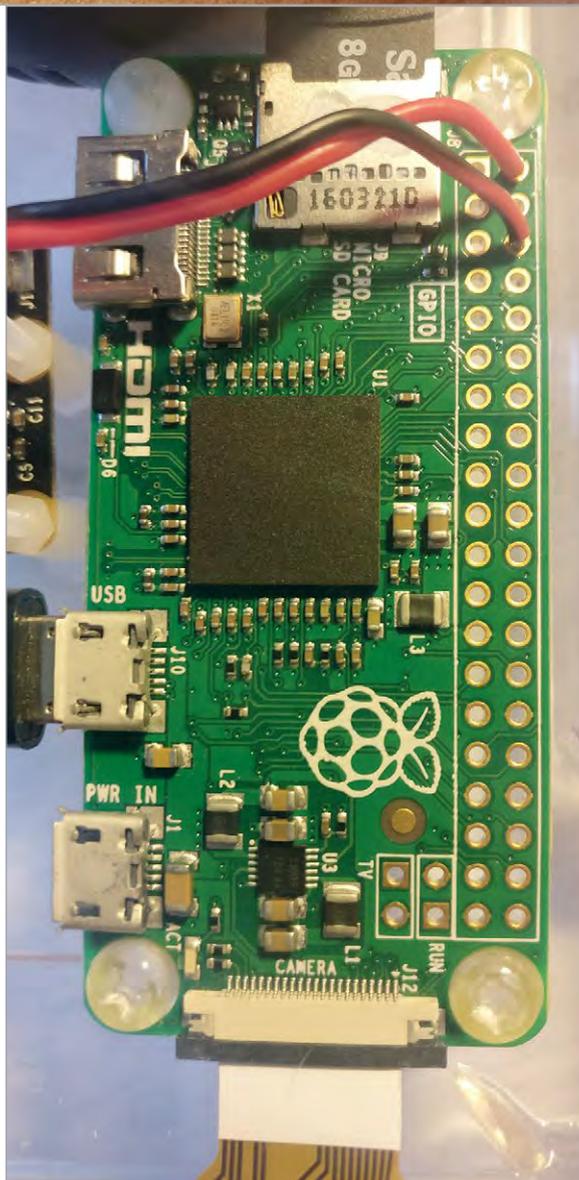
In order to supply power to the Raspberry Pi Zero, the UBEC can be connected to your Pi in two different ways: either by soldering the connections directly to the GPIO, or by attaching the three-pin socket on the UBEC to the Pi via a pre-soldered header. The photos show the output wires from the UBEC soldered directly onto the Pi's 5V and GND pins (pins number 2 and 6). Alternatively, you could solder a GPIO header onto the Pi and simply plug the UBEC's three-pin socket into the GPIO header – pins 2, 4, and 6 – making sure the black wire is connected to pin 6.

## >STEP-04 Connect the power circuit

The two input leads of the UBEC should be screwed into the corresponding terminals on the female barrel jack adapter, as well as two jumper wires connected to the correct terminals on the male barrel jack adapter. Using a hot glue gun, the female barrel jack socket can be attached to the hole that was drilled in the rear of the casing. Secondly, the UBEC can be attached to the inside of the casing in the same way. Finally, the male barrel jack adapter can be attached to the opposite side of the casing as the UBEC (see photos) to later be attached to the infrared illuminator.

## >STEP-05 Mount the IR illuminator

Using a few small dots of hot glue, the infrared illuminator can be fixed to top of the case's lid, as



**Left** Connection from the output of the UBEC to the Pi's 5V and GND GPIO pins

## MOTION ALERTS

For a security setup, there is an option under the Motion Notifications menu for an email to be sent when motion is detected.



**Above** The inner layout and connections of the camera setup



**Above** The camera works well when positioned in a spot where interesting activity is to be expected, like a hole in a fence

### CLOUD BACKUP

In order for the recorded files to be backed up, you can select Google Drive or Dropbox from the File Storage menu, clicking Obtain Key to gain permissions.

shown in the photographs. The power lead from this module can then be connected to the male barrel jack adapter now fixed to the side of the case. More hot glue may be needed to ensure a good fix, especially accounting for the stress applied to the cable when the box is opened or closed.

### >STEP-06 Mount the WiFi and antennae

With the two holes for the antennae already drilled the correct distance from each other, the WiFi module can easily be attached to the casing by pushing the antennae through the holes from the inside of the case. Once the WiFi antennae are in position, they can be fixed (and waterproofed) by using the hot glue gun to create a circular seal around the outside of the holes. The WiFi module is connected to the Pi by connecting the USB to micro USB adapter to the socket marked 'USB' on the Pi, and the USB plug on the WiFi module.

### >STEP-07 Mount the Raspberry Pi and Camera Module

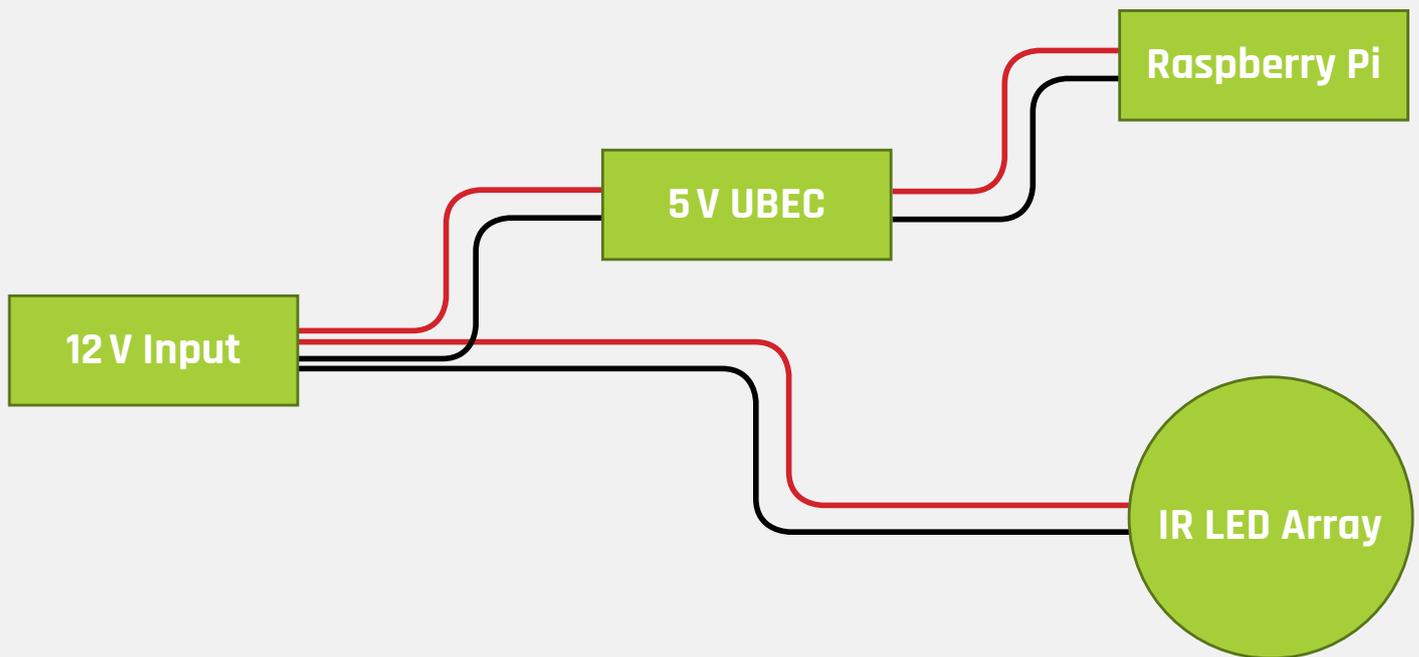
In this tutorial, we mounted both the Raspberry Pi Zero and Pi NoIR Camera module to the casing using eight hexagonal stand-offs; however, this is not necessary. With some improvisation, makeshift stand-offs can be created from common materials like wooden dowel. Both the Raspberry Pi and Camera Module are attached to the casing using hot glue. As the pictures show, we mounted the Camera Module upside-down in order to remove any kinks in the camera cable. The software can correct this inversion at a later time.

### >STEP-08 Install MotionEyeOS

On a computer, visit [magpi.cc/1UCw1jk](http://magpi.cc/1UCw1jk) and download the image shown as compatible with the Raspberry Pi Zero. Once downloaded, extract the IMG file from the **.tar.gz** archive. We did this using a program called 7zip. Finally, using a disk imaging tool like Etcher or Win32 Disk Imager, write the IMG file to a microSD card. As the OS does not create an interface on the Pi, initial setup will need to be performed on a Raspberry Pi 1 Model B or B+, as it works with the same version of the OS as the Zero and has Ethernet connectivity. This is because WiFi cannot be configured yet, unless you use one of the two methods for preconfiguring it before first boot ([magpi.cc/PHnHWK](http://magpi.cc/PHnHWK)), in which case the older model Pi isn't needed and you can insert the card straight into the Pi Zero (skip to Step 11).

### >STEP-09 Connect to MotionEyeOS

Insert the microSD card into the older Raspberry Pi model and connect the Camera Module. Power on, ensuring it is connected to your home network via Ethernet. After approximately two minutes, use an



**Above** A simple diagram illustrating the power distribution circuit

IP address scanner like the mobile app Fing and look for the IP address of a device name beginning with 'meye-'. Enter this IP address into your browser's URL bar, which will direct you to the page generated by the Raspberry Pi, showing the live feed from the camera, if connected to the older model Pi. The next step will show you how to set up a WiFi connection so that the Raspberry Pi Zero (and WiFi dongle) can be used.

## >STEP-10

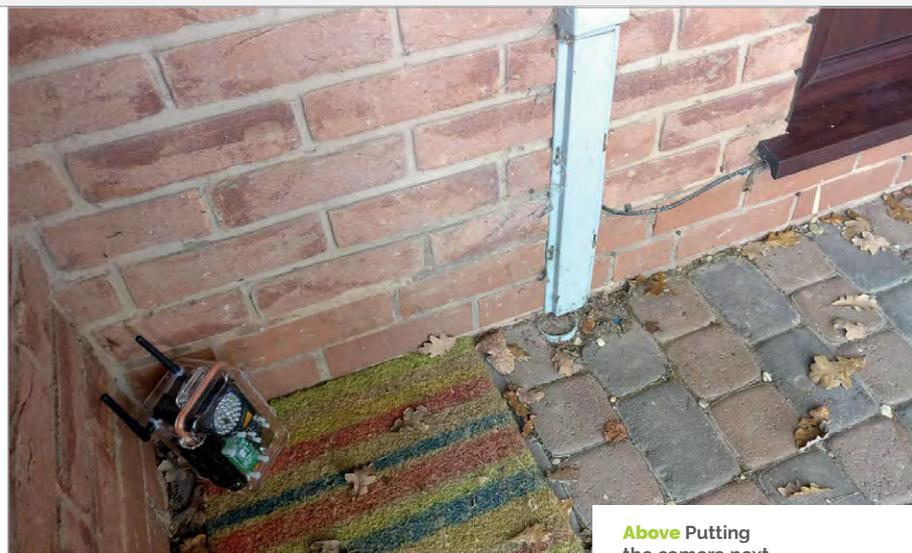
### Set up WiFi connection

When your browser is presented with the MotionEyeOS interface provided by the older Raspberry Pi model, navigate to the settings menu in the top-left corner of the screen. You may need to set up an admin user name and password beforehand. Under the Networks tab, switch Wireless Connection on, and fill in the Wireless Network Name and Wireless Network Key fields with the name and password for your wireless network, ensuring that the details are added with 100% accuracy. Once completed, click Apply and allow the system to reboot. Once rebooted, power off and insert the microSD card into the Pi Zero in the casing, then boot it by connecting the 12V supply to the socket in the rear of the case.

## >STEP-11

### Configure MotionEyeOS

Connect to the Pi Zero in the same way as before, although you may need to rescan to find the IP address if it's changed. Once connected, there are a couple of settings that need tweaking before the camera can be let loose! Firstly, as the camera is mounted with a 180° flip, the Video Rotation setting under the Video



**Above** Putting the camera next to the front door allows you to see who's been visiting during the day!

Device menu should be updated to 180. Finally, if you want the camera to record video whenever it detects motion, make sure that the Motion Detection and Movies settings are switched on.

## >STEP-12

### Using the camera

Now that everything is ready, the camera is free to be placed in the wild! When choosing a location for the camera, make sure that an extension power cable is able to reach it, and that the camera is pointing in a direction where you would expect some interesting animal activity, such as a gap in a fence. Finally, check that where you are placing the camera has a decent WiFi signal, otherwise the camera will continuously reboot until it finds a connection, and no footage will be collected. Good luck!

## SET UP WITHOUT EXTRA PI

In order to set up MotionEyeOS without the need for an additional earlier model Pi, follow the instructions found here: [magpi.cc/1UCvYwV](https://magpi.cc/1UCvYwV)



CARRIE ANNE PHILBIN

Carrie Anne is the Director of Education at the Raspberry Pi Foundation and also the author of *Adventures in Raspberry Pi*. [raspberrypi.org](http://raspberrypi.org)

You'll Need

- > 2× crocodile clips
- > 2× male-to-female jumper leads
- > A speaker
- > Craft stuff: paper plates, cardboard, aluminium foil, glue, scissors, sponge, bulldog or paper clips, sticky tape

# MAKE A WHOOPI CUSHION

A digital Raspberry Pi-powered version of a classic prank to play on your friends and relatives

In the bad old days before TV and computers, the most popular form of family entertainment was the whoopee cushion, a tooty balloon originally made from a pig's bladder. The whoopee cushion would be inflated and hidden under Grandad's chair cushion. When he sat down, it would make a loud

'PARP!' noise, causing him to jump into the air and making his false teeth fly out. It was the best thing ever. This project brings the whoopee cushion up to date: there's no bladder and no need to blow it up. Thanks to the Raspberry Pi, you can also add whatever noises you want!

Behind the sofa may not be the best place to hide, but you'll need to see your prank in action

This look of discontent is what every prank should strive for



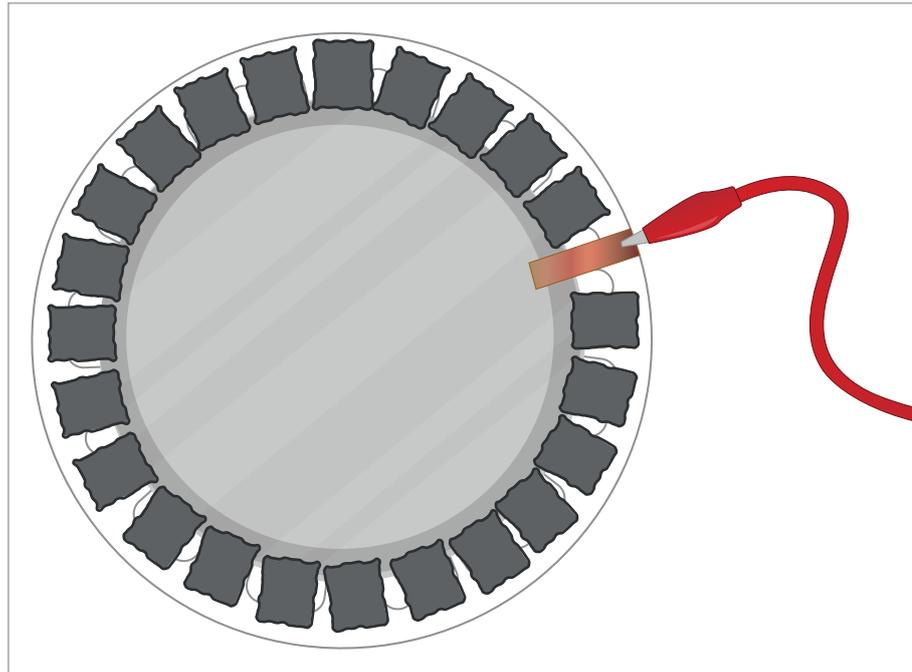
An electrically powered trump machine, ready to prank

### Making the Whoopi cushion

Cut a circle from the cardboard to fit the centre of the ‘eating’ side of each paper plate. Tape or stick squares of foil onto the cardboard. These are your contacts: when they touch, they will complete a circuit. Using some copper tape, connect the foil to the edge of the plate. Chop the sponge up into cuboid chunks and glue them around the foil on one of the plates: this will stop the foil squares touching each other until someone sits on the plate. It should look something like **Figure 1**.

Put the two plates together so that the foil pieces are on the inside and facing each other. Offset the paper plates so that the copper tape sections are not touching.

You now have a ‘cushion’ made of two plates. You can tape the plates together, or you might choose to use bulldog or paper clips so that you can test that your cushion works correctly, and easily debug any hardware problems.



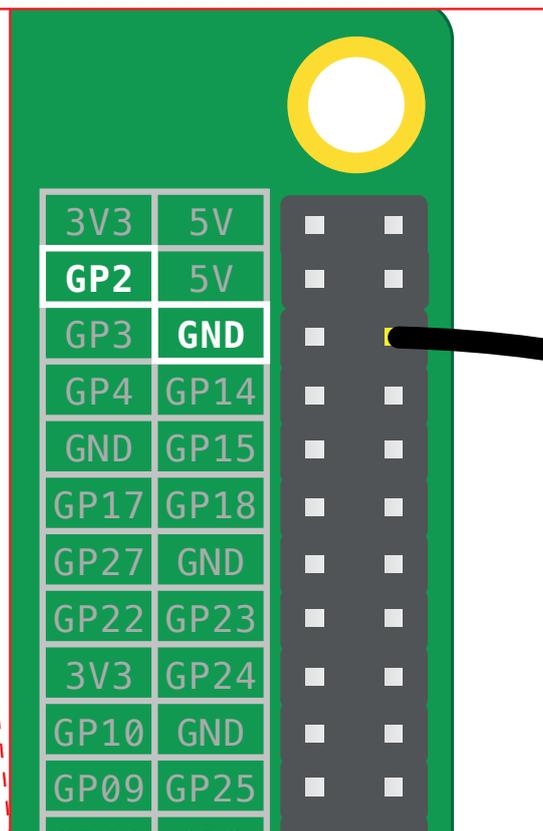
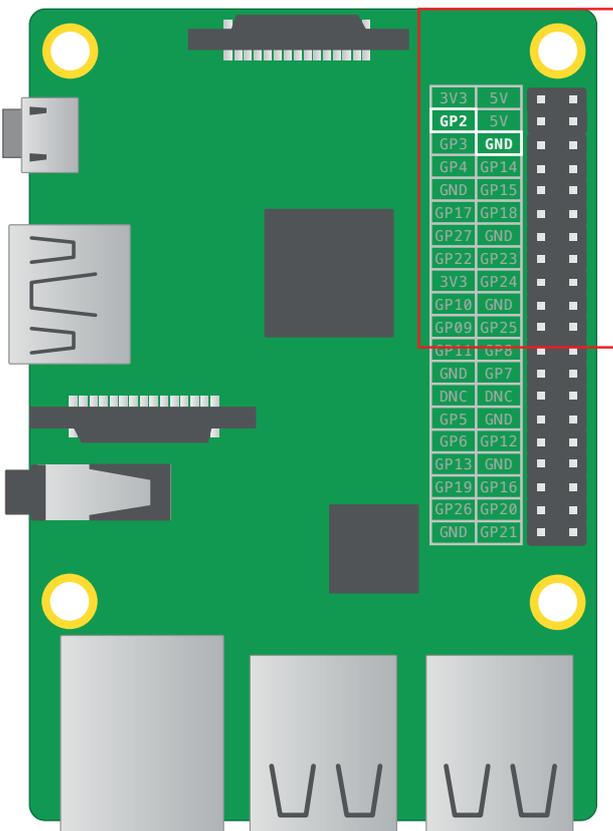
**Figure 1** It looks a bit like some futuristic tech. We’ll use it to make farting sounds

### Connect the Whoopi cushion to the Pi

Plug one header lead (it doesn’t matter which) onto a ground (GND) pin on the Pi (refer to **Figure 2**). Plug the other wire onto the GPIO 2 pin (**Figure 3**, overleaf) and then clip a crocodile clip cable to one of the paper plates’ copper tape sections. Connect the other end of the crocodile clip cable to the male pin of one of the connected jumper wires (**Figure 4**). Repeat these last two steps to connect the other plate to your Raspberry Pi. Your setup should look similar to **Figure 5**.

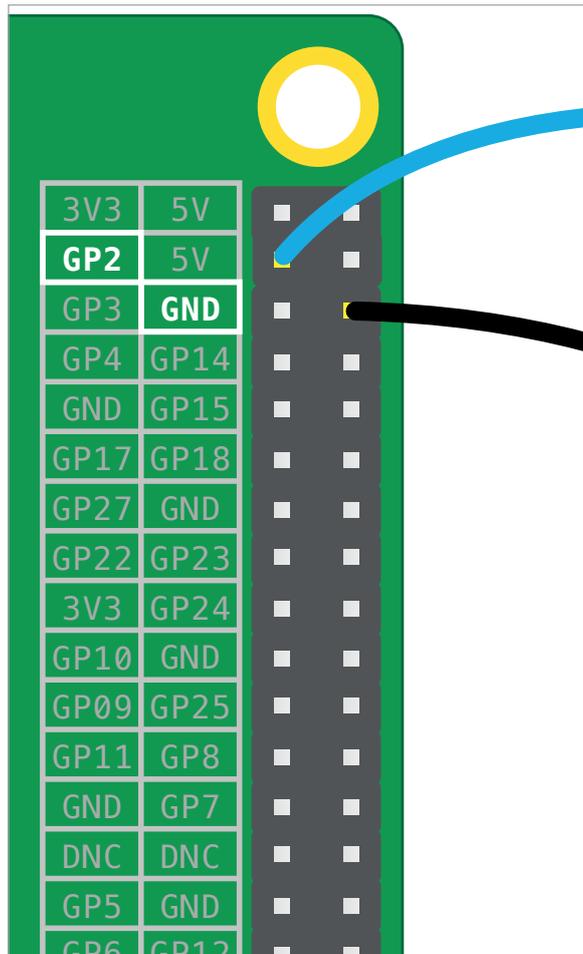
### Test the sound

That’s the hardware complete. Now for the software! We’re going to use Python. Don’t worry if you haven’t used it before: just follow the instructions and you’ll pick it up. You’ll be using the command line to enter commands. To do this you’ll need to open a terminal window by clicking on the terminal icon: it looks like a computer screen, and is found three icons along from the Menu button on your desktop.



**Figure 2** If you have an older Raspberry Pi model you’ll only have 26 pins, but they have the same layout

**Figure 3** Again, this will be the same pin on any Raspberry Pi



Connect the speaker to the Raspberry Pi using the sound jack. Create a new folder called **whoopee** by typing the following command in the Terminal and pressing **ENTER** on the keyboard:

```
mkdir whoopee
```

Next, use **cd whoopee** to enter the folder you have just created. We're going to need a sample sound file for this project, so we'll use one from Sonic Pi. Download this burp sample using the following command:

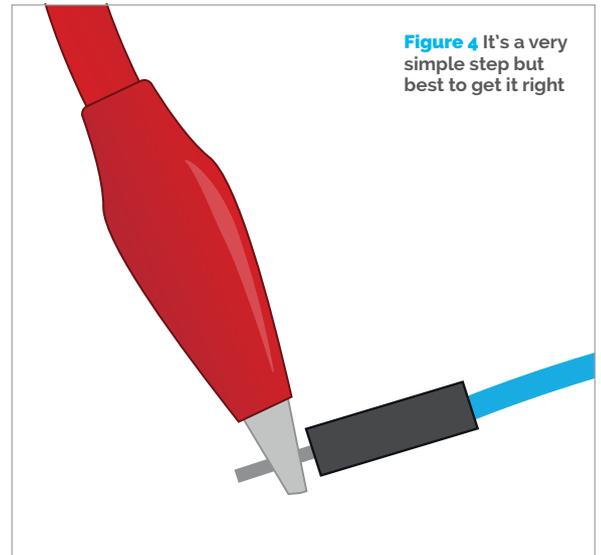
```
wget http://rpf.io/burp -O burp.wav
```

Now, test that you can play the sound file using **aplay** by typing:

```
aplay burp.wav
```

You should hear it from the speakers or headphones connected to your Pi. If you can't hear anything, make sure that your speakers are connected correctly. If this still doesn't work, you'll need to change your audio configuration.

To switch audio to the headphone jack, return to the terminal window and type the following command:



**Figure 4** It's a very simple step but best to get it right

```
amixer cset numid=3 1
```

If your Raspberry Pi is connected to the internet, you could search for some suitable trumping sounds. They need to be in WAV format to work. Alternatively, download our example sounds here: [rpf.io/farts](http://rpf.io/farts).

### Write a program in Python

Open Python 3 (IDLE) from the Programming menu and click on File > New Window. This will open a blank file. Click on File > Save As, name the file **whoopee.py** and then type the following code into your file:

```
import os
import random
from time import sleep
from gpiozero import Button
```

This part of the code pulls in all the libraries that you're going to use to write your program. You'll then need to use the Button class in your code. You'll have to tell it that the button is on pin 2. To do this, write the following code in your new file:

```
button = Button(2)
```

Now create a list of all your sound effects and store them inside a variable that you can call later on in your code:

```
trumps = ['ben-fart.wav', 'ca-fart.wav', 'marc-fart.wav']
```

In Python, square brackets are used to create a list. Each item in the list is separated by a comma.

Once all the setup needed in the code is complete, you can move on to writing the part of the program that will make something happen when the button is pressed. Begin by creating a loop using **while**

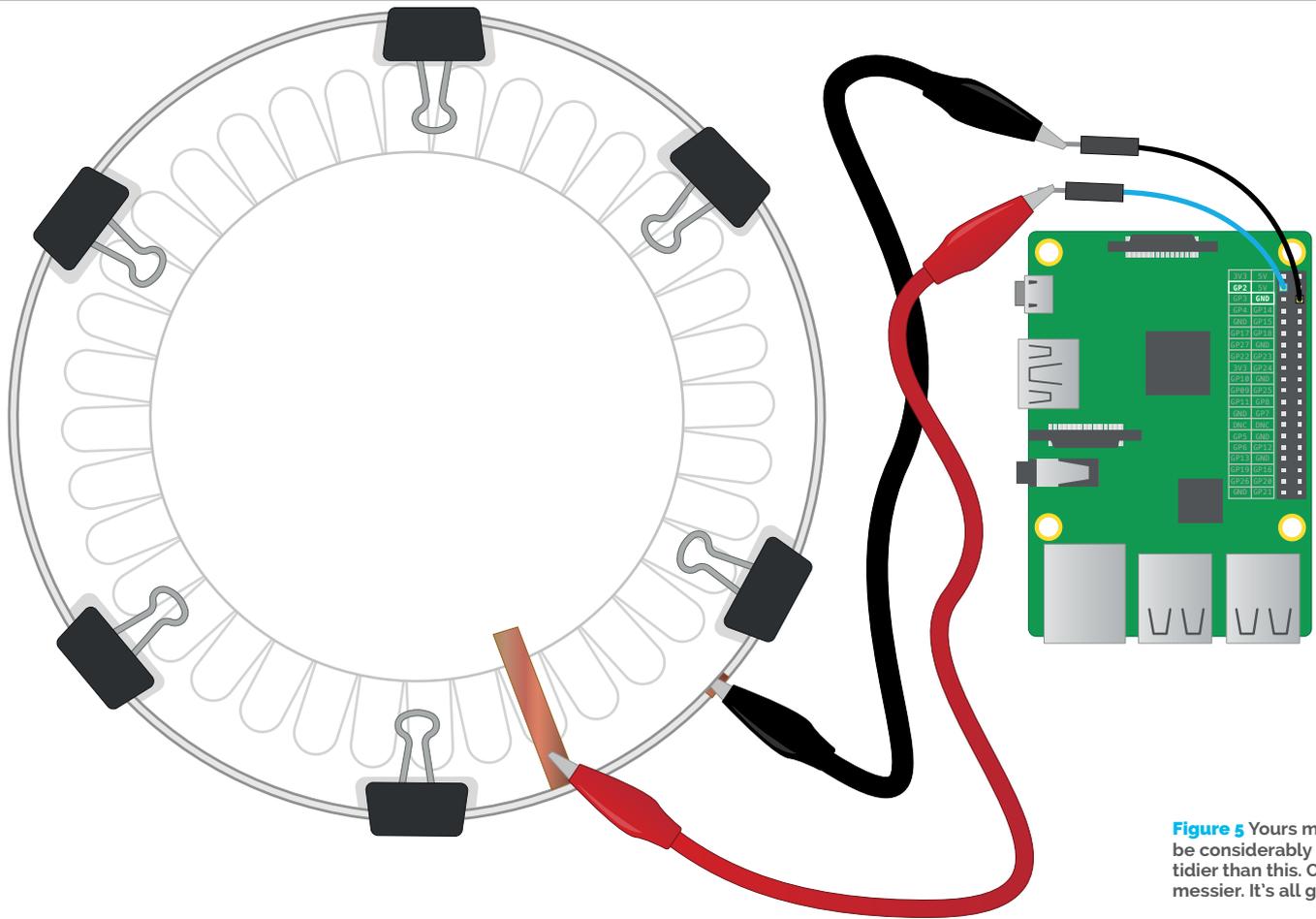


Figure 5 Yours may be considerably tidier than this. Or messier. It's all good

**True:** Then, add `button.wait_for_press()` inside the loop by indenting by four spaces. Each time around the loop, the computer waits for the button to be pressed.

On the next line, use the `random.choice` function to select a sound at random from the list you created earlier. That selected sound needs to be stored inside another variable which you can call `parp`! Type `parp = random.choice(trumps)`.

The next line will play the sound selected at random using `aplay`, as used earlier when testing sounds. Type `os.system("aplay {0}".format(parp))`. Finally, add `sleep(2)` to pause the program before it starts the loop again. Your code should look like this:

```
while True:
    button.wait_for_press()
    parp = random.choice(trumps)
    os.system("aplay {0}".format(parp))
    sleep(2)
```

Save the file by clicking on File > Save. Test that your code works by clicking on Run > Run Module. Use your hand to push the top plate of your Whoopi cushion down to make a connection between the foil sheets, and you should hear a fun sound. If it doesn't work first time, don't worry. Check your code through. Have you typed your code out exactly as you see it here?

## whoopee.py

```
import os
import random
from time import sleep
from gpiozero import Button

button = Button(2)
trumps = ['ben-fart.wav', 'ca-fart.wav', 'marc-fart.wav']

while True:
    button.wait_for_press()
    parp = random.choice(trumps)
    os.system("aplay {0}".format(parp))
    sleep(2)
```

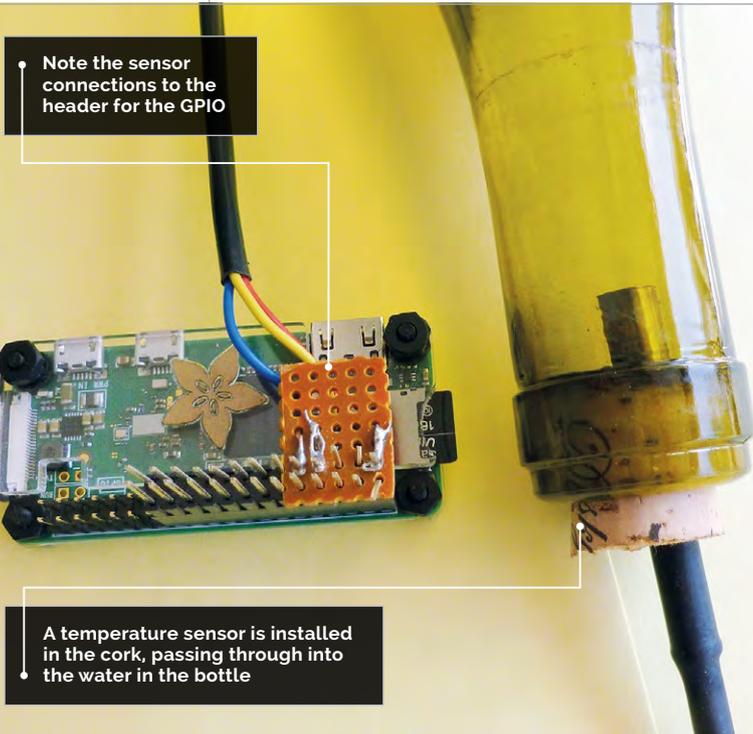
### Setting it up

Carefully place your cushion in a spot where your victim will sit on it (obviously!), but not under a really heavy cushion where it will be squashed straight away. The tricky bit is setting up the Pi so that it can't be seen: remember, you'll need a plug socket to connect it to the power, unless you're using a battery pack. Run the program, and wait. Here's a hint: whistle tunelessly and look around at the ceiling. This will make you seem innocent, and will help to attract potential victims.

Language

>PYTHON 3

DOWNLOAD:  
magpi.cc/  
WhoopiCushion



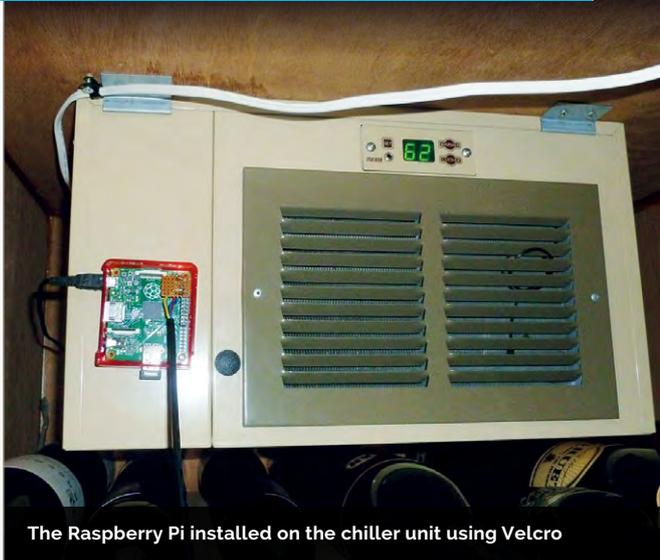
Note the sensor connections to the header for the GPIO

A temperature sensor is installed in the cork, passing through into the water in the bottle



**BILL BALLARD**

Bill is a retired physicist who got back to his programming roots to find fun things to do with his ten Raspberry Pi boards. [github.com/wpballa](https://github.com/wpballa)



The Raspberry Pi installed on the chiller unit using Velcro

# WINE SAVER: ROOM TEMPERATURE MONITOR

## You'll Need

- > Pi Zero W
- > 4GB (or larger) microSD card
- > Suitable case for the Pi
- > Waterproof DS18B20 temperature sensor
- > Stacking header
- > Gmail account
- > Empty wine bottle and a cork that fits it

This project monitors the temperature of a wine storage room hourly, and sends an email when it exceeds an 'alarm' or 'fail' level

**Y**our author has a wine storage unit in his garage, and over the last 20 years the compressor has failed twice during extreme heatwaves. We weren't constantly checking the temperature display, so had no warning of the impending failures, which would have allowed us to move the wine to an air-conditioned location and have the compressor repaired. This project provides hourly logging of the temperature, and sends warning emails if the temperature exceeds set limits.

### Software setup

Download and install the latest version of Raspbian Jessie Lite on your microSD card. Initialise Raspbian with `sudo raspi-config` to the proper international options (time zone, keyboard, wireless country code, etc.), change the default password, and set the host name in advanced options to something like 'winerom'. Make sure that you enable both SSH and 1-Wire in the **Interface** section of `raspi-config`.

Reboot (otherwise your keyboard settings may not be correct), and set up the wireless by editing the relevant config file: `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`. Add these lines to the end.

```
network={
    ssid="your ssid"
    psk="your password"
    key-mgmt=WPA-PSK
}
```

Be sure to replace "your ssid" and "your password" with the actual values for your network, and keep the quotes. Now reboot again.

Use the `ifconfig` command to see the IPv4 address that you will need to SSH into the Pi later. Write it down (192.168.1.27 or 10.0.0.27 are typical values). Now you should run the usual update and upgrade to the operating system:

```
sudo apt-get update && sudo apt-get upgrade -y
```

Next, install the mail and ssmtp applications:

```
sudo apt-get install ssmtp heirloom-mailx
```

Edit the ssmtp configuration file to point to your Gmail account as shown. Add the text below to the end

of the configuration file with `sudo nano /etc/ssmtp/ssmtp.conf`.

```
# Wine Room Monitor settings - for gmail
hostname=winerroom
root=your-mail@gmail.com
mailhub=smtp.gmail.com:587
FromLineOverride=YES
AuthUser=your gmail name (leave out the @
and stuff after that)
AuthPass=your gmail password (no quotes
needed here)
UseSTARTTLS=YES
UseTLS=YES
AuthUser=LOGIN
```

Now save the file and exit. As we will run the monitor program with crontab, we must also change the `/etc/ssmtp/revaliases` file or authentication errors will occur. So, `sudo nano /etc/ssmtp/revaliases` and add the following line:

```
root:your-email@gmail.com:smtp.gmail.com:587
```

Now reboot.

## The program

Log back into the Pi. First, create a directory called `winerroom` and move into it:

```
mkdir winerroom
cd winerroom
```

Download the `ds18b20.c` file from GitHub ([magpi.cc/2uBzLR4](https://magpi.cc/2uBzLR4)) and customise it for your email address and your alarm temperature levels. To compile:

```
gcc ds18b20.c -o ds18b20 -Wall -std=gnu99
```

## Hardware setup

Take the wine bottle cork and cut it to half its length with a hacksaw. Carefully clamp it and use a  $\frac{1}{4}$ " bit to drill a hole lengthwise through the centre of the cork. Next, push the DS18B20 sensor's stainless steel through the cork so that it is protruding from the other end – this is why we cut it in half. Wetting the sensor may make insertion easier. Now rinse the wine bottle with hot water and fill it nearly full of water. Wet the cork and push it carefully into the bottle, then put the bottle on its side to check for leaks. You may need to add a bit of silicone to seal around the sensor.

On the stacking header, solder a 4.7 kΩ resistor to the 3V3 and GPIO 4 pins. Solder the sensor red wire to 3V3; the white wire to GPIO 4; the black wire to GND.

Now plug the header onto the Raspberry Pi (make sure it's off!), being careful to get the location of pin 1 (3V3) correct. Boot the Pi up, log in, and test it with:



```
cd winerroom
sudo ./ds18b20 &
```

Look at the file in your `/home/pi/winerroom` directory called `wine.log`. There should be one entry. You are way over the temperature range and should get an email showing an alarm or alert. This will test whether you've got the email settings correct. If you don't get an email in a few minutes, take a look at `/etc/ssmtp/ssmtp.conf` again to see whether there are any errors.

Now we create a short script to rename the log file and add the year and month to the log file name at the end of every month. Of course, since we run it on the first of the month, we need last month's date. Download `logupdate.sh` from GitHub, edit it to incorporate your email address, then make it executable:

```
chmod +x logupdate.sh
```

Next, we automate things with crontab by editing with `sudo crontab -e` and adding two lines to the end of the file. The first line runs the code at five minutes after the hour, every hour of every day, and the log file renaming happens at four minutes after midnight on the first of the month.

```
5 * * * * /home/pi/winerroom/ds18b20
4 0 1 * * /home/pi/winerroom/logupdate.sh
```

Save the file, then to get the new crontab entries working:

```
sudo service cron start
sudo update-rc.d cron defaults
```

Use some Velcro on the bottom of the Raspberry Pi Zero W case to attach it to the cooler unit. If there is no power plug inside the wine cooler, remove a bit of the sealant around the cooler and run a micro USB power cable through. Then replace the sealant to close the system back up, power it up, and let it run!

Language

&gt; C AND BASH

DOWNLOAD:  
[magpi.cc/2uBzLR4](https://magpi.cc/2uBzLR4)CREATE A  
UNIQUE GMAIL

Create a brand new Gmail account just for this purpose. A unique Gmail account for the wine room prevents a lot of spam.



**HENRY BUDDEN**

Following the release of the Raspberry Pi when he was twelve, Henry has taught himself to code and use/break electronics. [magpi.cc/zeCbaMf](http://magpi.cc/zeCbaMf)

**You'll Need**

- > 2× LEDs
- > Light-dependant resistor (LDR)
- > 330 nF ceramic capacitor
- > Buzzer
- > 2× 1kΩ resistors
- > 2× 470Ω resistors
- > 6× Male-to-female jumper wires
- > 3× Male-to-male jumper wires
- > Or Monk Makes Raspberry Pi Electronics Starter Kit [magpi.cc/zeC95jz](http://magpi.cc/zeC95jz)

# DIY SUNRISE ALARM

Use a breadboard and simple components to sense light levels and activate a loud (and blinky) alarm to wake you up each morning!

**W**ith the help of some exciting code and some clever electronics, you can use your Raspberry Pi to read an analogue signal from a photoresistor without the need for a conversion chip! Armed with this power, you can measure the ambient light level and trigger an effective alarm. The project is designed as an extension to the projects found in the Monk Makes Raspberry Pi Electronics Starter Kit.

**>STEP-01**

**Connect the resistors**

Once you have the components, begin to make the circuit by connecting up the resistors. Do this as shown in the circuit diagram, pushing each component's legs into the holes in the breadboard. Ensure that the bottom two resistors are 470Ω (yellow, purple, and brown), and the top two are 1kΩ (brown, black, and red).

**>STEP-02**

**Add the rest of the components**

Next, add the LEDs, making sure that the long legs are connected towards the bottom of the diagram, as shown. The flat side of the LEDs should be facing towards the 1kΩ resistors. When connecting the buzzer, the longest leg should be facing the bottom of the breadboard. Finally, connect the LDR and capacitor. These can be connected in any orientation.

**>STEP-03**

**Connect to the Pi**

Make sure your Raspberry Pi is turned off and unplugged before you do this. Using the three male-to-male jumper wires, connect the ground to the two LEDs, the buzzer, and the LDR as shown in the diagram. Next, use the remaining male-to-female jumper wires to connect the breadboard to the Raspberry Pi's GPIO pins.

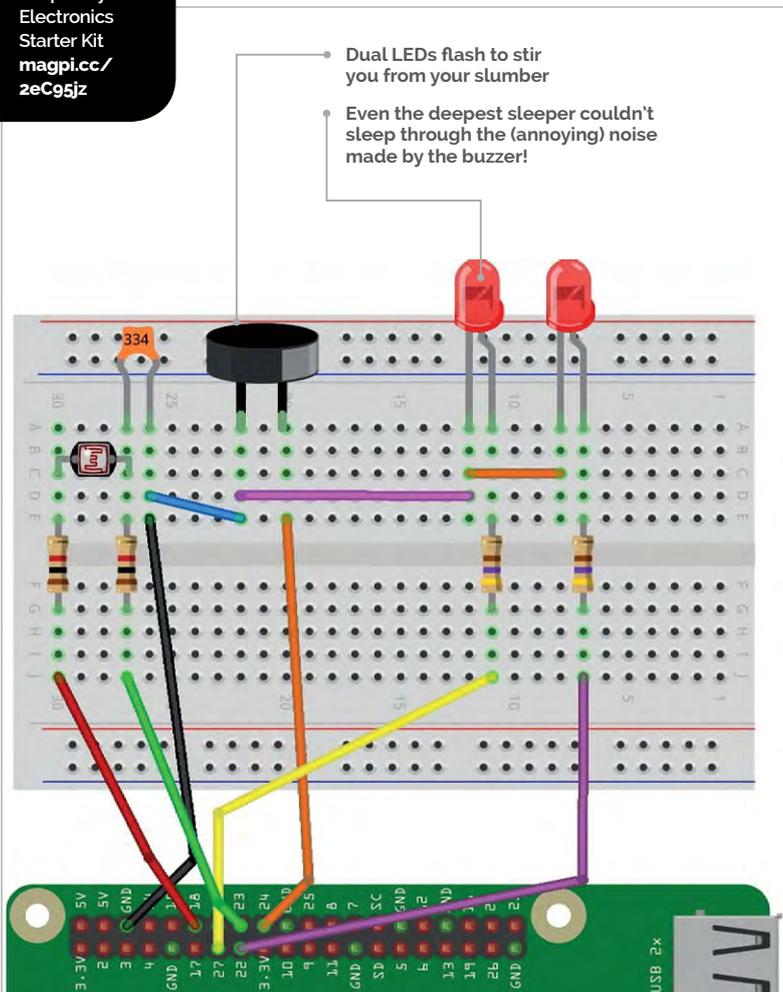
**>STEP-04**

**Install the software**

Turn on your Raspberry Pi, and ensure it is connected to the internet. Using the Terminal, clone the GitHub repository containing the code to your Pi's SD card using the command:

```
git clone https://github.com/henrybudden/rpsk-advanced/
```

Once the files have downloaded and you have returned to the command prompt, change into the



# Sunrise\_alarm.py

```

from Tkinter import *
import RPi.GPIO as GPIO
import time, math

GPIO.cleanup()

GPIO.setmode(GPIO.BCM)

sunrise = 50

a_pin = 18
b_pin = 23

buzzer_pin = 24
red_pin1 = 27
red_pin2 = 22

GPIO.setup(buzzer_pin, GPIO.OUT)
GPIO.setup(red_pin1, GPIO.OUT)
GPIO.setup(red_pin2, GPIO.OUT)

def discharge():
    GPIO.setup(a_pin, GPIO.IN)
    GPIO.setup(b_pin, GPIO.OUT)
    GPIO.output(b_pin, False)
    time.sleep(0.01)

def charge_time():
    GPIO.setup(b_pin, GPIO.IN)
    GPIO.setup(a_pin, GPIO.OUT)
    GPIO.output(a_pin, True)
    t1 = time.time()
    while not GPIO.input(b_pin):
        pass
    t2 = time.time()
    return (t2 - t1) * 1000000

def analog_read():
    discharge()
    return charge_time()

def read_resistance():
    n = 20
    total = 0;
    for i in range(1, n):
        total = total + analog_read()
    reading = total / float(n)
    resistance = reading * 6.05 - 939
    return resistance

def light_from_r(R):
    return math.log(1000000.0/R) * 10.0

while True:
    GPIO.output(red_pin1, False)
    GPIO.output(red_pin2, False)
    light = light_from_r(read_resistance())
    print light
    x = 0
    if light > sunrise:
        GPIO.output(red_pin1, True)
        GPIO.output(red_pin2, False)
        while True:
            x = x + 1
            GPIO.output(buzzer_pin, True)
            time.sleep(0.001)
            GPIO.output(buzzer_pin, False)
            time.sleep(0.001)
            if x == 250:
                x = 0
                break
        GPIO.output(red_pin1, False)
        GPIO.output(red_pin2, True)
        while True:
            x = x + 1
            GPIO.output(buzzer_pin, True)
            time.sleep(0.001)
            GPIO.output(buzzer_pin, False)
            time.sleep(0.001)
            if x == 250:
                x = 0
                break

```

Language

&gt;PYTHON 2

**DOWNLOAD:**  
[magpi.cc/zeCbCtZ](http://magpi.cc/zeCbCtZ)

directory containing the code by using the command **cd rpsk-advanced**.

## >STEP-05

### Run the code

The time has come to run the code and test out the alarm! After checking that the circuit you have built is an exact replica of the one shown in the circuit diagram, and that it is connected to the Pi, run the command **sudo python 01\_sunrise\_alarm.py**. You can test that the alarm works by shining a torch at the photoresistor from within a fairly light room. The LEDs should start to flash, and the buzzer should sound. If this happens, congratulations!

## >STEP-06

### Make it your own

The default threshold for the alarm's activation is when the photoresistor reaches a value of 50, which works well for testing as described previously. However, in order to use the alarm to accurately detect the sunrise in the morning, this value can be changed by entering the file editor with the command **nano 01\_sunrise\_alarm.py** and then changing the value found on line 13. We would recommend that you use 30 for fairly accurate detection of dawn. Save this change by pressing **CTRL+X**, followed by **Y**, then the **ENTER** key. This code can now be run again as in Step 05.

**TIM RICHARDSON**

One half of the CamJam and Pi Wars organisation team, and author of the CamJam EduKit worksheets.  
@Geeky\_Tim

# PIHUE CONTROL PHILIPS HUE LIGHTS

## You'll Need

- ▶ Pimoroni Touch pHAT [magpi.cc/2pafoPg](http://magpi.cc/2pafoPg)
- ▶ Philips Hue Bridge and Lights [magpi.cc/2uBNFsk](http://magpi.cc/2uBNFsk)
- ▶ phue Python library by Studiomaginaire [magpi.cc/2uBWOF](http://magpi.cc/2uBWOF)

## Learn how to control Philips Hue lights with a Raspberry Pi and the Pimoroni Touch pHAT

**W**e're all guilty of spending too much time geeking out! GeekyTim is no exception; he spends his time in 'Hut 8' (his log cabin in the garden) tinkering with Raspberry Pi boards, 3D printing, and laser cutting. His wife complains when she has to come out in the rain to call him in!

After buying some Hue lights, he wondered whether they could be controlled with a Pi. Philips provides APIs, and other Python libraries have been developed to access the Bridge. So, he put them all together and created the PiHue – a Raspberry Pi Zero W with Touch pHAT that his family can use to call him in from play...

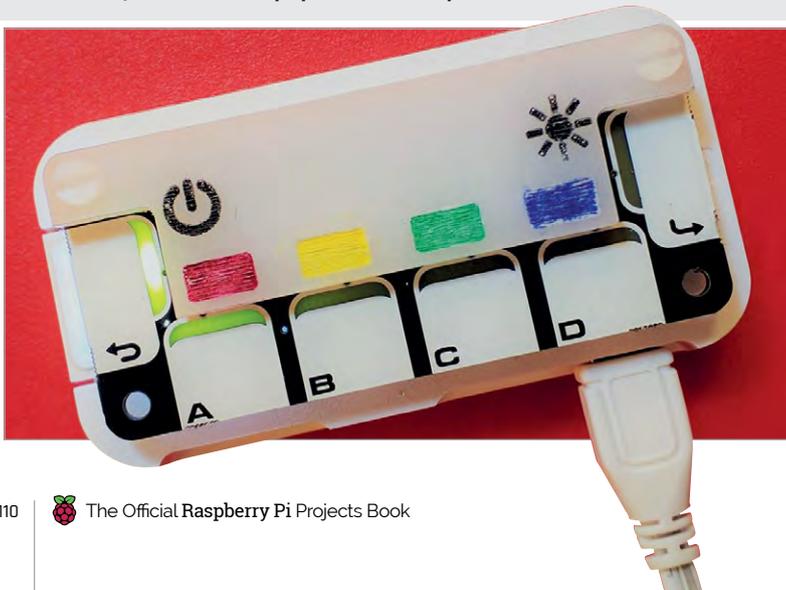
### Setting up

We start by connecting our Pimoroni Touch pHAT to the Pi, burning the Raspbian Jessie Lite image onto an SD card, booting up, and connecting to the same network that the Hue Bridge is on.

We need to install a few prerequisites before downloading the PiHue code, including Python 3, pip, Git, the **Touch pHAT** library, and the **phue** library that gives us access to the Hue lights from Python:

```
sudo apt-get update
sudo apt-get install python3 python3-pip git
sudo apt-get install python3-touchphat
sudo pip3 install phue
```

**Below** The PiHue, with Touch pHAT and custom laser-cut cover to show the colour options



Next, we need to download the PiHue code from GitHub:

```
git clone https://github.com/GeekyTim/PiHue
```

The PiHue code is in the **PiHue** directory:

```
cd PiHue
```

### Configuring PiHue

Each of the Touch pHAT's six capacitive buttons will have a different function. The Back (left arrow) button turns the lights on and off; four A, B, C and D buttons flash the lights red, yellow, green, and blue respectively; and Enter (right arrow) makes them all bright.

#### There are two versions of the code:

- ▶ **PiHueRoom.py** controls a Room as defined in the Hue app.
- ▶ **PiHueLightList.py** controls individual lights in a list.

The Room version works best with a room that contains more than one light, especially if one is coloured and one is dimmable.

#### We have to edit the code to make some changes for our Hue setup:

- ▶ Set the IP address of the Philips Hue: **bridgeip**. We can find this by accessing our router's IP address list or using a network sniffer tool. The Bridge's MAC address can be found in the Hue app.
- ▶ For **PiHueRoom.py**, set the name of the room to be controlled: **roomname**.
- ▶ For **PiHueLightList.py**, change the names of the lights in the Python list: **lights**.

Optionally, we can change the list of Hue **xy** colours by adding to those already listed under the comment **# Hue 'xy' colours**.



Above It's alive!

We can also change the **alert** patterns in the section following **# Alert Patterns**. These are Python lists; the first number is how many times to repeat the pattern. The second is the delay between changes – the recommended minimum delay is 0.4 (seconds).

The remaining values are Python dictionaries of the light status changes.

Use any valid Hue status, e.g. **on**, **bri**, **xy**, **ct**, **sat**, **hue**, **transformationtime** (see the Hue API for details: [magpi.cc/2vEpXfH](http://magpi.cc/2vEpXfH)).

Dimmable lights do not accept any of the colour changes (**xy**, **ct**, **sat**, **hue**), so just change their brightness instead.

We can specify as many changes as we want.

For Light Lists, the status changes are Python dictionaries that contain the changes for each light type (e.g. **Extended color light** and **Dimmable light**). Other light types can be added if you have them (e.g. **Color Light**, **Color Temperature Light**).

There is one special dictionary that is used by the Touch pHAT's Enter button to set all lights to bright white: **allwhite**. We could redefine this to be a preferred colour and brightness.

## Running the code

The line **# b.connect()** must be uncommented the first time the code is run. Just before running it, we must press the Bridge connect button. Run with:

```
python3 PiHueRoom.py
```

...OR:

```
python3 PiHueLightList.py
```

The Pi should connect to the Bridge and save the Bridge details in the file **/home/pi/.python\_hue**. If we ever need to change the Bridge, we just need to delete this file. We can now re-comment the **b.connect()** line by adding **#** at the front..

## Running on boot

There are a few ways to run Python code when the Raspberry Pi boots. A good method is to use **systemd**. In this case, we need to create a configuration file (aka a 'unit' file) that defines a new **systemd** service:

```
sudo nano /lib/systemd/system/PiHue.service
```

...and type in the following (replacing **PiHueRoom.py** with **PiHueLightList.py** if required):

```
[Unit]
Description=PiHue Service
After=multi-user.target

[Service]
Type=idle
User=pi
Restart=always
ExecStart=/usr/bin/python3 /home/pi/PiHue/
PiHueRoom.py

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

Exit and save using **CTRL+X**, **Y**, then **ENTER**.

The permission of the unit file needs to be set to 644:

```
sudo chmod 644 /lib/systemd/system/PiHue.
service
```

We need to instruct **systemd** to start the service during the boot sequence:

```
sudo systemctl daemon-reload
sudo systemctl enable PiHue.service
```

When we reboot the Pi, the PiHue service should run.

## Check service status

We can check the status of the PiHue service using:

```
sudo systemctl status PiHue.service
```

The last line should look like:

```
Jul 10 23:26:52 PiHue systemd[1]: Started
Start the Touch pHAT Hue controller.
```

Sometimes the Raspberry Pi disconnects from the network, especially when using WiFi. Once disconnected, it will remain disconnected unless the interface is restarted or the Pi is rebooted. That makes this light switch a bit useless. But help is at hand: we can reboot the Pi if it cannot see your home router!

Follow the instructions by Thijs Bernolet on his blog to enable reboot on network loss: [magpi.cc/2uCoiny](http://magpi.cc/2uCoiny).

Language

>PYTHON 3

NAME:

PiHueRoom.py  
PiHueLightList.py

DOWNLOAD:

[magpi.cc/2vBxtol](http://magpi.cc/2vBxtol)

CONTROL  
YOUR OWN  
WORLD!

Don't just use  
the code as  
is – why not  
experiment  
and make  
your lights  
match your  
own colour  
schemes?

LIGHT  
CHANGES

For rooms with  
both dimmable  
and coloured  
lights, alter  
the brightness  
as well as  
the colour of  
the room.



**LORRAINE UNDERWOOD**

Lorraine runs the Kendal Pi Jam. She is an Irish maker living in Yorkshire with a background in web development  
 @LMcUnderwood  
 magpi.cc/2ovDD9z

# TEMPERATURE CONTROLLED STAIRLIGHTS

## You'll Need

- > 5V RGB strip  
[magpi.cc/2pzIMyC](http://magpi.cc/2pzIMyC)
- > 5V mains power supply  
[magpi.cc/mUxjzL](http://magpi.cc/mUxjzL)
- > Female jack connector plug  
[magpi.cc/2q9Hf6](http://magpi.cc/2q9Hf6)
- > Male-to-female jumper wires

Give every day a colourful and useful start by displaying the outside temperature using coloured lights on your stairs

**W**hen you wake up in the morning, wouldn't you like to know whether you need to wear your woolly hat or your sundress? Now you can find out on your way to breakfast, thanks to Lorraine's stairlights project! The Raspberry Pi hidden under the stairs connects to the web and checks the temperature. It then controls the strip of 240 lights running up the stairs. If it's colder than 0°C, the bottom 35 lights come on in white; under 5°C, and the next 35 lights light up in blue; and so on up to 25°C and red, although that probably won't happen in Yorkshire, where it was built!

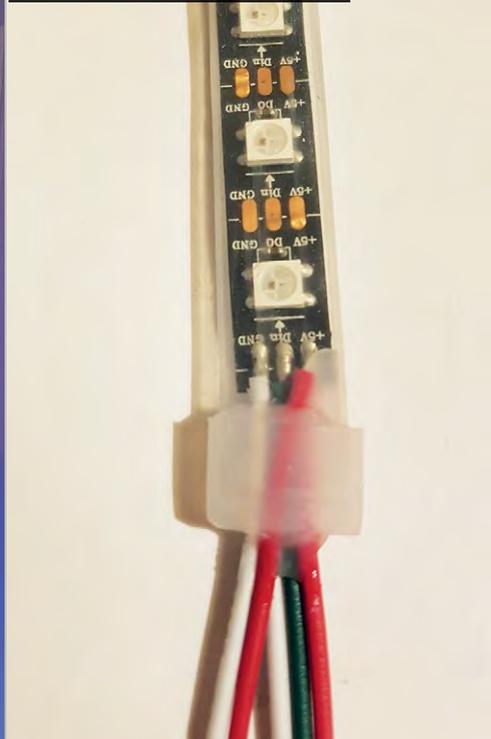
### >STEP-01 Set up the lights

Firstly, work out which end is which on your RGB strip. We are looking for the Data In end. It should be labelled as Din. In this strip there are five cables coming from three connections: two from GND, one from Din, and two coming out of 5V.

Connect the 5V wire to the '+' block on the female jack connector plug by placing the bare wire under the terminal, then screwing the terminal down with a screwdriver. Connect the GND wire to the '-' block in the same way. Pull gently on both wires to check that they're connected.



This is the end you're looking for...



## >STEP-02

### Set up the Pi

Connect the Din and GND wires to the male ends of the jumper wires. Connect the female end to the Pi as follows: Din to GPIO pin 18; GND to any ground pin.

You could power the Pi through the other 5V wire, but this can be dangerous for the board; it is best to use a normal power supply.

Follow the steps at Adafruit to install the NeoPixel library, `rpi_ws281x`: [magpi.cc/1nRSyYk](http://magpi.cc/1nRSyYk).

Plug the jack connector into your power supply. Plug in the power supply and test your strip using the scripts from the **examples** folder for some shiny lights!

## >STEP-03

### Set up the weather API

You will need a developer account for a weather API to get the outside temperature for your area. We used **forecast.io** as it allows users to request 1000 forecasts per day at no cost:

- > Go to [forecast.io](http://forecast.io)
- > Select Developer
- > Select 'Sign up' to register an account
- > Once you confirm your email address, you will get a secret key: you will need this key in Step 04

You could set up another Pi outside your house and get the real temperature for your neighbourhood. It depends on how much you want to spend on the project, and whether you trust the API's accuracy.

## >STEP-04

### Let there be light!

Download **stairlights.py** from the box at the top of this page to the **home/pi** directory and change:

**Line 6:** enter your secret key.

**Line 7:** change **longi** and **lati** to your longitude and latitude coordinates. You can use Google Maps to find them: right-click on your location and select 'What's here?'

Set up a cron task to check the outside temperature every five minutes and update the lights.

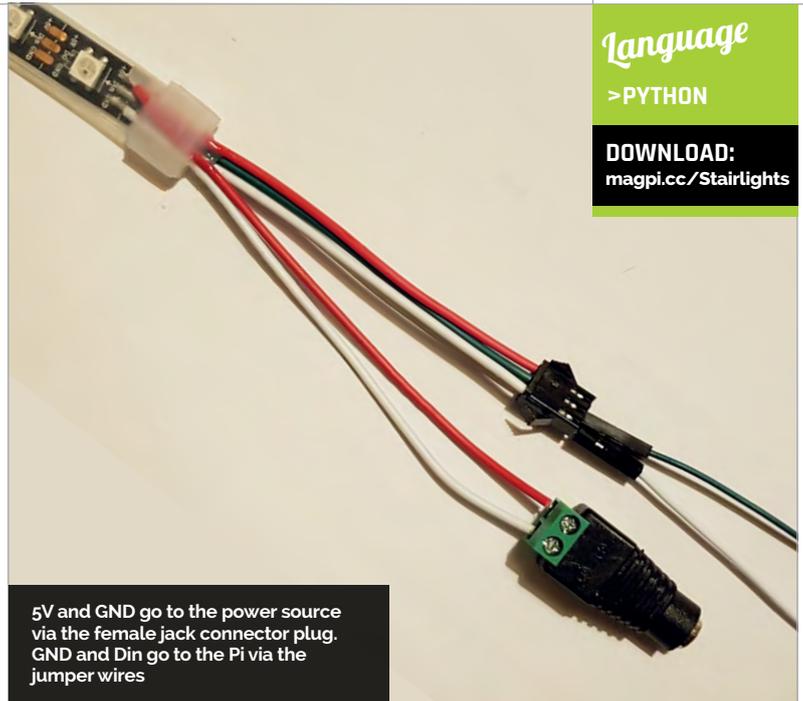
In a Terminal window, type:

```
sudo -E crontab -e
```

At the end of the file, enter:

```
* /5 * * * * /usr/bin/python3.4 /home/pi/stairlights.py
@reboot /usr/bin/python3.4 /home/pi/stairlights.py
```

Save the changes.



Language

>PYTHON

DOWNLOAD:  
[magpi.cc/Stairlights](http://magpi.cc/Stairlights)

## >STEP-05

### Debugging your lights

If the examples from the NeoPixel library didn't work, check all your connections. Make sure you have plugged the data wire into the correct GPIO pin on your Pi. Are you connected to Din or Dout? The lights will power on Dout, but nothing will happen.

If the lights are displaying strange colours, some people have reported the NeoPixel library not working with Pi's setup for audio. See [magpi.cc/2ovIQhy](http://magpi.cc/2ovIQhy).

Make sure your script is running properly before adding it to cron. There's nothing like sitting on your stairs waiting for five minutes for a cronjob that isn't working to run!

## >STEP-06

### What next for your lights?

We changed the cron tasks to stop the lights displaying through the night:

```
* /5 7-21 * * * /usr/bin/python3.4 /home/pi/stairlights.py
* /5 7-21 * * * /usr/bin/python3.4 /home/pi/nightynight.py
```

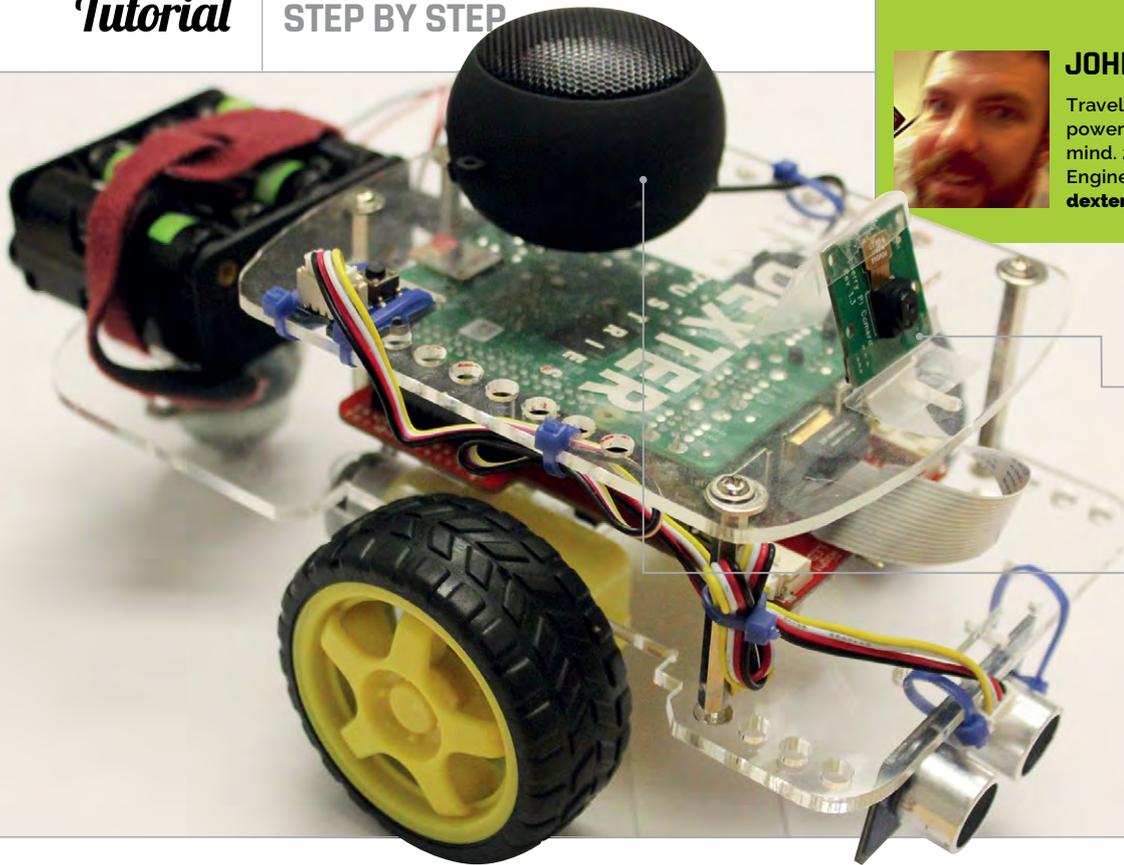
...where **nightynight.py** is a simple script that turns the lights off using the first 26 lines of **stairlights.py** and this line:

```
colorMe(strip, Color(0, 0, 0), 0, 239)
```

Now, with a strip of lights on your stairs, you could play all kinds of games!

Add some coloured arcade buttons for a quick reaction game.

Maybe when your phone connects to the Bluetooth on the Pi, your stairs could flash a welcoming hello dance. Experiment and have fun!



**JOHN COLE**

Travelling bearded nerd with a heart powered by coffee and an unfocused mind. 2016 Presidential candidate. Engineer at Dexter Industries. [dexterindustries.com](http://dexterindustries.com)

The Raspberry Pi Camera Module takes a picture of the human subject

The speaker is used to say some words back from the robot to the human subject, depending on their mood

# EMPATHYBOT: THE RASPBERRY PI ROBOT THAT READS EMOTIONS

In this project we'll build a Raspberry Pi robot with emotional intelligence using the Raspberry Pi Camera Module to tell if a person is happy, sad, angry, or surprised

## You'll Need

- ▶ GoPiGo kit [magpi.cc/2hA8f6i](http://magpi.cc/2hA8f6i)
- ▶ Speaker [magpi.cc/2hAgCj7](http://magpi.cc/2hAgCj7)
- ▶ Ultrasonic sensor [magpi.cc/2hAazds](http://magpi.cc/2hAazds)
- ▶ Button sensor [magpi.cc/2hAhqE1](http://magpi.cc/2hAhqE1)
- ▶ Raspberry Pi Camera Module [magpi.cc/28lJlsz](http://magpi.cc/28lJlsz)

**W**hat if you could build a robot with some empathy? This tutorial will show you how to build a robot that can read a person's face for emotions. In this tutorial we'll use the Raspberry Pi, the Raspberry Pi Camera Module, a GoPiGo, and a speaker to read some human faces and say something appropriate back. Our robot will roll up to its human master, take a picture, analyse the face with a free Google Cloud Vision account, and then say something appropriate to the human's current mood.

### >STEP-01 Build the GoPiGo

You will need a small Phillips head screwdriver; the rest of the parts are included with the GoPiGo. There are written directions and video instructions on the Dexter Industries site showing how to assemble the GoPiGo robot kit. Attach the Raspberry Pi, and

add eight AA batteries to power the GoPiGo. While programming the stationary GoPiGo, you may want to use a USB power supply to power the Pi.

### >STEP-02 Add the camera

Add the Raspberry Pi Camera Module to the GoPiGo. In this tutorial, we'll use one of the slots on the top of the GoPiGo canopy to support the camera. You can also use the servo accessory to move the camera side to side.

### >STEP-03 Add the speaker

The speaker can be mounted to the top of the GoPiGo using a few zip ties. Place the speaker on top of the GoPiGo, and connect the speaker aux cable to the Raspberry Pi headphone port. You can charge the speaker using the Raspberry Pi USB cable.

## >STEP-04

### Attach the sensors

We use the button sensor to give us an easy way to start the robot up once the code is running, and the distance sensor to judge how far we are from the human. Hang the ultrasonic distance sensor from the front of the GoPiGo, using small zip ties, and attach the button to the top of the GoPiGo, also using zip ties.

## >STEP-05

### Setup a free Google Cloud Vision account

You can use your Gmail or Google account to set this up. At the time of publication, Google offers a 60-day free account – [magpi.cc/2hAkotI](https://magpi.cc/2hAkotI).

## >STEP-06

### Create a new project

This is an abbreviated version of the setup process. You can see a pictorial walkthrough of how to set up a new project in Google Cloud Vision online here: [magpi.cc/2hAhHqc](https://magpi.cc/2hAhHqc).

Create a new project called 'vision1'. Enable the Cloud Vision API for vision1.

## >STEP-07

### Download and install your JSON credentials

Head back to the Console in Google Cloud. Find the box titled 'Use Google APIs' and click 'Enable and manage APIs'. Click on Credentials and Create Credentials. Credentials is on the left-hand side, with a picture of a key next to it. Select 'Create a Service Account Key'. Under Service Account, select New Service Account. We'll call this 'vision'. Finally, create a role. We'll give the new role full access, so select Project and Owner to give the Pi full access to all resources. A pop-up window should appear telling you that you have created a new key,



**Above** This project was tested on babies and it works just fine on them too



**Above** This project really only works well on non-bearded humans. Beards interfere with the software

Language

>PYTHON

DOWNLOAD:  
[magpi.cc/2hAgoqIT](https://magpi.cc/2hAgoqIT)

and an automatic download of the JSON key should begin. Keep track of this file! You should now use an FTP program (such as FileZilla) or Samba (see our tutorial at [youtu.be/CEYwYqkAPfA](https://youtu.be/CEYwYqkAPfA)) to move the JSON file over to your Raspberry Pi. Place the JSON file in the home directory `/home/pi`.

## >STEP-08

### Prepare the Raspberry Pi

We need to run a few commands to prepare the Raspberry Pi. We only need to do this one time. In the command line, run the following commands:

```
sudo pip install --upgrade pip
sudo apt-get install libjpeg8-dev
sudo pip install --upgrade google-api-python-client
sudo pip install --upgrade Pillow
sudo apt-get install python-picamera
```

Finally, make the credentials we downloaded in the previous step available to Python:

```
export GOOGLE_APPLICATION_
CREDENTIALS=filename.json
```

Here, replace 'filename' with the name of your JSON file.

## >STEP-09

### Run the code

If you've taken the code from our GitHub repo, or typed it out yourself, you can now run the code:

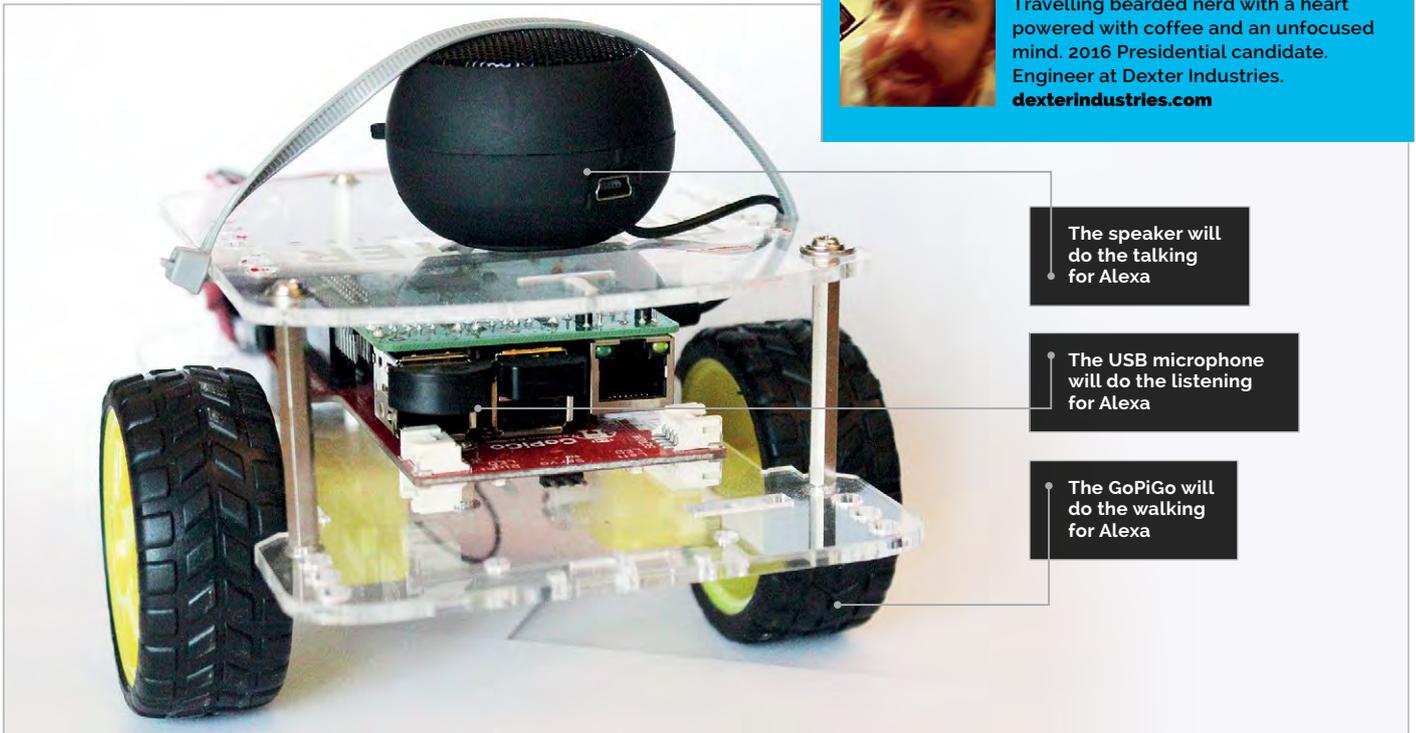
```
sudo python empathybot.py
```

Point the robot towards your human subject, press the Grove button on the GoPiGo, and let your robot start interacting with some humans!



JOHN COLE

Travelling bearded nerd with a heart powered with coffee and an unfocused mind. 2016 Presidential candidate. Engineer at Dexter Industries. [dexterindustries.com](http://dexterindustries.com)



The speaker will do the talking for Alexa

The USB microphone will do the listening for Alexa

The GoPiGo will do the walking for Alexa

You'll Need

- > Raspberry Pi 3
- > The GoPiGo [magpi.cc/2hA8f6i](http://magpi.cc/2hA8f6i)
- > Raspbian for Robots [magpi.cc/2kixPxB](http://magpi.cc/2kixPxB)
- > Speaker for the Raspberry Pi [magpi.cc/2hAgCi7](http://magpi.cc/2hAgCi7)
- > Microphone [amzn.to/299mm1V](http://amzn.to/299mm1V)
- > A free Amazon Developer account
- > A free IFTTT account

# HOW TO MAKE AN ALEXA-POWERED ROBOT

Build an Amazon Alexa-controlled robot using the Raspberry Pi, allowing you to operate it with just your voice



Finally, a Raspberry Pi robot to do your bidding, controlled only by the sound of your voice! Amazon Alexa software can be put on the Raspberry Pi robot, the GoPiGo. In this project, we'll create an Alexa-based robot that will respond to voice commands, as well as answer your questions. You can ask Alexabot "How hot is it in Dubai?" or "What's the weather like in London?". What makes Alexabot really interesting is that you can order it around with your voice, using the Alexa Voice Service (AVS).

The first step towards robot world domination and building our Amazon Alexa-controlled robot is to set up AlexaPi.



## M Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

### URL

`http://580b2886.us.ngrok.io/forward`

Surround any text with "<<<<" and ">>>>" to escape the content

+ Ingredient

### Method

GET



The method of the request e.g. GET, POST, DELETE

### Content Type (optional)

text/plain



Optional

### Body (optional)

forward

Surround any text with "<<<<" and ">>>>" to escape the content

+ Ingredient

## Alexabot.py

```
from flask import Flask
import gopigo
import time

app = Flask(__name__)

@app.route('/')
def index():
    return 'Hello world'

@app.route('/forward')
def forward():
    print("Forward!")
    gopigo.fwd() # Send the GoPiGo Forward
    time.sleep(1) # for 1 second
    gopigo.stop() # then stop the GoPiGo
    return 'Alexabot moved forward!'

@app.route('/backward')
def backward():
    print("Backward!")
    gopigo.bwd() # Send the GoPiGo Backward
    time.sleep(1) # for 1 second
    gopigo.stop() # then stop the GoPiGo.
    return 'Backward!'

@app.route('/left')
def left():
    print("Left!")
    gopigo.left()
    time.sleep(1)
    gopigo.stop()
    return 'Left!'

@app.route('/right')
def right():
    print("Right!")
    gopigo.right()
    time.sleep(1)
    gopigo.stop()
    return 'Right!'

@app.route('/dance')
def dance():
    print("Dance!")
    for each in range(0,5):
        gopigo.right()
        time.sleep(0.25)
        gopigo.left()
        time.sleep(0.25)
        gopigo.bwd()
        time.sleep(0.25)
    gopigo.stop()
    return 'Dance!'

@app.route('/coffee')
def coffee():
    print("Coffee!")
    return 'coffee!'

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Language

>PYTHON

DOWNLOAD:  
magpi.cc/2kiFLZz

Figure 1 An example of how to build your own command

There are a few tutorials on how to turn your Raspberry Pi into an Amazon Echo using Amazon Alexa. However, AlexaPi ([magpi.cc/2kiy0x0](http://magpi.cc/2kiy0x0)) is the easiest way to get Amazon Alexa on your Raspberry Pi. The project has prepared everything to set up Alexa Voice Services on your Pi in a streamlined and simple manner. The hardest part is setting up your Amazon Developer account and gathering your credentials (and really, that's not that hard!).

At this point, with your speaker and your microphone plugged into your Pi, you should be able to run Alexa on your GoPiGo, just as you would on an Amazon Echo. Ask it for a news update: "Alexa, what's the news?"

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      connecting
Version             2.1.18
Region              United States (us)
Web Interface        http://127.0.0.1:4041

Connections          ttl      opn      rt1      rt5      p50      p90
                    0        0        0.00    0.00    0.00    0.00
```

Figure 2 Your ngrok readout should look a bit like this

TRIGGER WORDS

Remember that your trigger words, the words that get Alexa going, start with "Alexa trigger".

Next, we'll set up IFTTT. If This Then That (IFTTT) is a service that lets us connect different parts of the web together. In our case, IFTTT helps us couple Alexa together with the Raspberry Pi. IFTTT lets us bridge the AlexaPi responses back to the Raspberry Pi. First, you'll need to set up an IFTTT account. ([ifttt.com/join](http://ifttt.com/join)). Again, this is free.

Next, you'll need to add the IFTTT service to your AlexaPi. You can do this by first installing the Alexa app on your phone or tablet ([magpi.cc/2kyIjho](http://magpi.cc/2kyIjho)); it's available for iOS, Android, and Fire OS. With your phone on the same network as your Pi, connect to your AlexaPi and add the IFTTT service.

Next, in IFTTT, we will need an applet for each command we want to send to the GoPiGo. First, connect Amazon Alexa to your IFTTT account ([magpi.cc/2kizD9X](http://magpi.cc/2kizD9X)). Then we'll do the following to create an applet that will handle a single command:

01. Create an IFTTT applet ([magpi.cc/2jTYyn2](http://magpi.cc/2jTYyn2)) for our first command. We will create an Alexa applet.
02. Select 'Say a Specific Phrase'.
03. Specify the command. Say "Backward".
04. Next, select 'That'.
05. We'll use the action service, Maker.
06. Click 'Make a web request' and specify the information.
07. The URL is either your provided ngrok URL or your custom domain. We will find this in our next step, but for now we can enter [ngrok.io](http://ngrok.io).
08. Our applet will be doing a 'GET' method, the content will be text, and don't put anything in the body.
09. Click 'Create Action' and then 'Finish'.

RUNNING THE FLASK SERVER

The Flask server runs in the background and needs to be set up in a different command-line window from the ngrok server.

Figure 1 shows how this is done in more detail. We will need to do this for all the moves you want your Amazon Alexa-controlled robot to make! In our project, we made seven: one each for Forward, Backward, Left, Stop, Right, Dance, and Coffee.

Next, we'll connect to the ngrok service. We will have IFTTT contact a server on the Pi, but the tricky part will be setting up a server on the Pi that's visible to the world outside our local wireless network. For this, you can use ngrok ([ngrok.com](http://ngrok.com)), a service that allows you to connect to your Raspberry Pi through

" A robot that will respond to voice commands "

any network. IFTTT needs an internet-based URL to contact, so we'll use ngrok to make our server accessible to the outside world.

First, set up an account with ngrok. You can get away with a free account, but it will be much easier to set up Alexabot with a paid account. A paid account will allow you to set up named servers, rather than ngrok's randomly assigned server.

Next, install ngrok on your Raspberry Pi using the directions here: [ngrok.com/download](http://ngrok.com/download).

This should install all the software. You will next need to get your token, which will authorise your Raspberry Pi to ngrok. If you log into your account you will see step 2, with your token already populated. It should start with `./ngrok authToken`. Copy the command, and paste it into your command line.

Test out ngrok

You can type the following into the command line:

```
./ngrok http 80
```

You should then see a server start up, as shown in Figure 2. When an IP address appears, try typing that IP address into your browser. You should see your

default server on Port 80 come up. We will start our server with the following command:

```
./ngrok http -subdomain=dexterindustries 5000
```

...where 'dexterindustries' is a reserved domain set up with our basic account on ngrok. If you go with the free account, you'll now need to go back and re-enter the domain name that ngrok gave your Pi into each of your IFTTT commands.

## Set up the Flask server

The final step in getting our Amazon Alexa-controlled robot working is setting up the Flask server on the Raspberry Pi.

We'll set up a Flask server in Python to listen to IFTTT. To install Flask on the Raspberry Pi, enter the following into your command line:

```
sudo pip install flask
```

You can try running the Flask server we've provided in our GitHub code and see what happens:

```
python alexabot-flask-app.py
```

You should see the Flask server start!

To get an idea of how the code works on the Flask server, you can type the following in your web browser: <http://alexabot.ngrok.io/forward>. You should get this response in your browser: 'Alexabot moved forward!'.

This illustrates how the Forward command works; each command will work this way, with its own directory and custom code for moving the GoPiGo.

Now, with all the services set up, we should be able to say a command like "Alexa trigger forward"; Alexa will alert IFTTT, which will send an HTTP message through ngrok back to our GoPiGo and post to the web server running in Flask. The Flask program will command the GoPiGo to move forward.

The quick-start method to get running with Alexabot is to first start AlexaPi:

```
sudo python /opt/AlexaPi/src/main.py
```

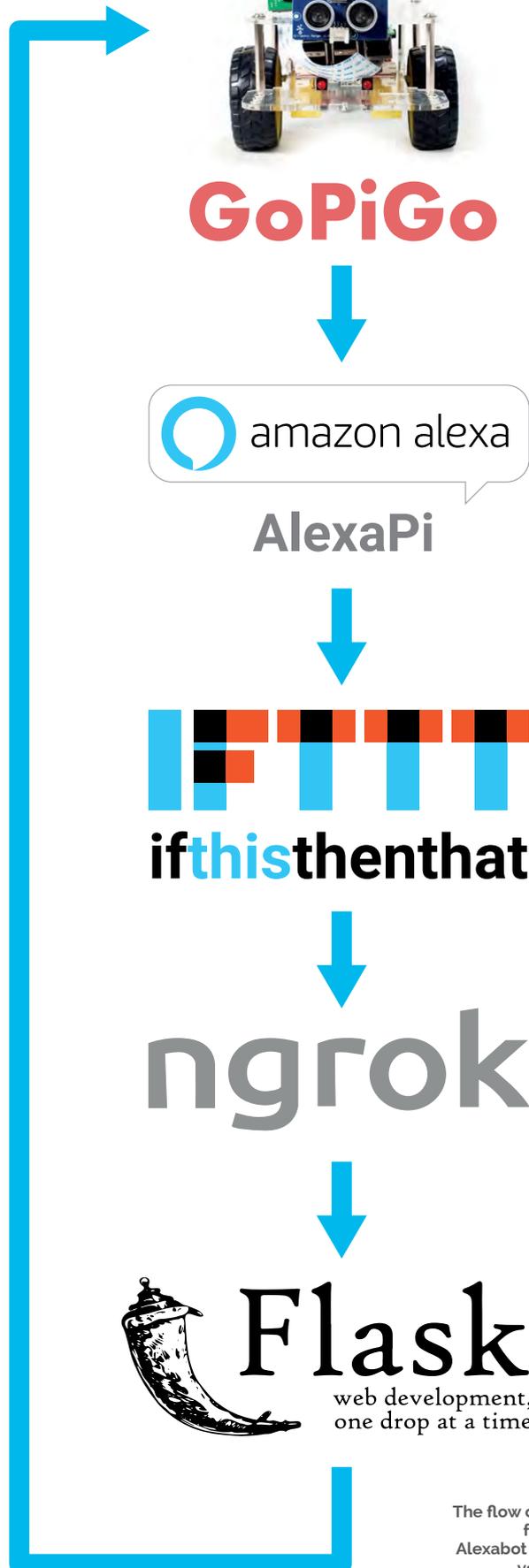
Next, start ngrok in a separate window:

```
sudo ~/ngrok/ngrok http
-subdomain=dexterindustries -log=stdout
5000 > log.txt &
```

Finally, start the Flask server:

```
sudo python alexabot-flask-app.py
```

Start talking! Remember, your commands need to start with "Alexa trigger..." and then the command you want the robot to carry out.



The flow of information for getting the Alexabot to respond to your command

# BUILD YOUR OWN MAGIC MIRROR

Live in the future with your own internet-connected mirror straight from science-fiction

**W**e live in a world where more and more items are becoming part of the Internet of Things. Fridges sending an email about buying some more milk. Washing machines tweeting you when they're done. An app on your phone that lets you turn on a light. A lot of this seems gimmicky, though, which is probably why the magic mirror concept has taken

off so well in the maker community – getting all the information you need at a glance while checking your appearance before you leave the house. It's passive and useful. Michael Teeuw pioneered the concept on the Raspberry Pi, making it easy enough for everyone to make one. All you need to build your own is a bit of spare time and a trusty saw.

**THE  
PROCESS  
STEPS  
TO BUILD  
A MAGIC  
MIRROR**



### GET THE PARTS

Everything you need to make your own mirror

122



### BUILD YOUR MIRROR

How to go about putting together the frame and electronics

124



GET ALL THE SOFTWARE AND INFO YOU NEED ONLINE AT:

MAGICMIRROR.BUILDERS

**THE CLEVER CODING ASPECT**

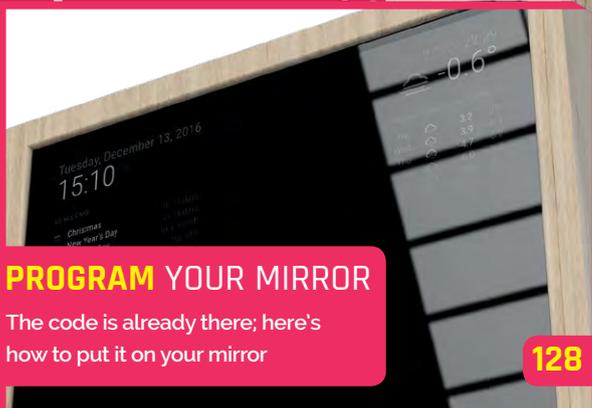
The code for the mirror is all there and ready to be installed with one line

**TWO-WAY MIRROR**

Used in the right way, you can make an info-laden mirror

**BUILDING THE FRAME**

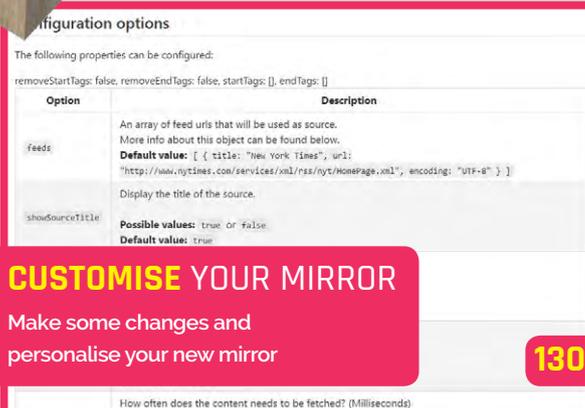
Get out your tape measure and put a pencil behind your ear



**PROGRAM YOUR MIRROR**

The code is already there; here's how to put it on your mirror

128



**CUSTOMISE YOUR MIRROR**

Make some changes and personalise your new mirror

130

# COMPONENTS

What you need to build your mirror

## BUILDING MATERIALS



### WOOD FOR FRAME

Plywood is a good option for this. If you're a little more skilled at carpentry, however, you can have a look at pine or another material. Make sure it's sturdy and deep enough to contain all the electronics when built – have a look at our build steps for an idea of the size in comparison.

### WOOD FOR FRONT

The fascia can be made from skirting board or moulding – make sure it's wider than the wood you plan on using for the frame so it can keep everything in place.



### NAILS

We'll use these to attach the front of the frame; 15–20 mm should do.



### SCREWS

A small selection of wood screws to build the frame. They don't need to be huge, though: about 20 mm longer than the depth of the wood should be fine.



### WOOD PAINT & FILLER

You'll want to make your frame look good, so get some filler to help smooth it out, and some wood paint to finish it off. If you like the wood you've used, get some varnish instead.



### WOOD GLUE

We're using this to make sure everything stays together as intended. Think of it as a backup for the screws.



### TWO-WAY MIRROR

Buy one the same size as the monitor, which will also be the same size as your frame. You can get some made from acrylic.

# ELECTRONICS

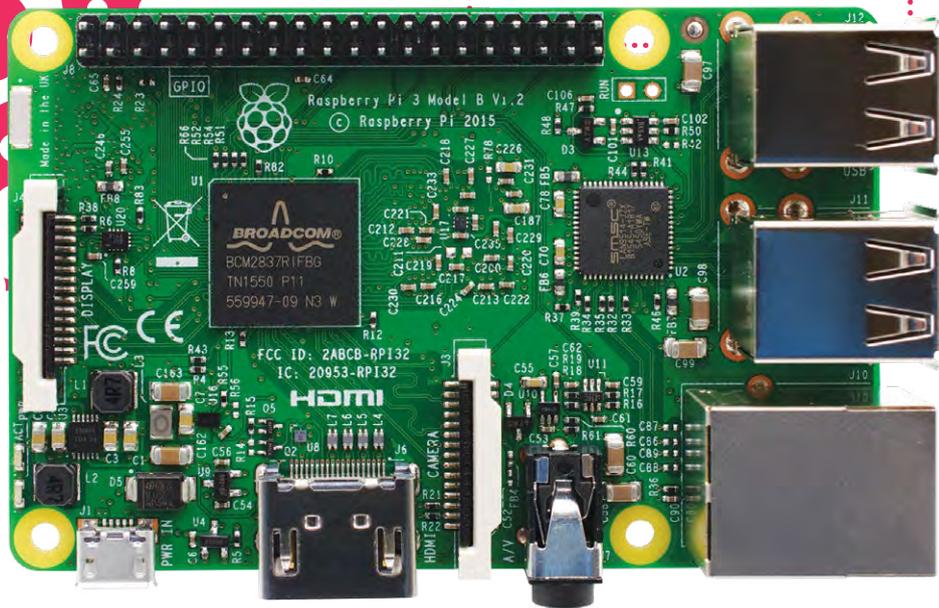


## MONITOR/TELEVISION

A large monitor or old LCD television should do the trick for this. The lighter the better, though. You may want to remove the outer casing on your monitor if you can, to save on space and weight.

## RASPBERRY PI

You'll need one to power the TV, of course. A case (and a WiFi dongle if you're using a Pi 2) should help make things a little more tidy and secure.



## CABLES

You'll need power for the monitor and the Pi, as well as an HDMI for the picture output.



## TOOLS

You'll need a saw, a hammer, a drill, some clamps, and various painting tools to do this. Research what you'll need for specific bits.



## STEP 01:

### MEASURE THE MONITOR

Measure the dimensions of the monitor. It may be an idea to see how deep it is as well, to ensure your wood selection is correct; it's best to have a little space between the back of the monitor and the wall, so make sure your wood will allow for this.



## STEP 03:

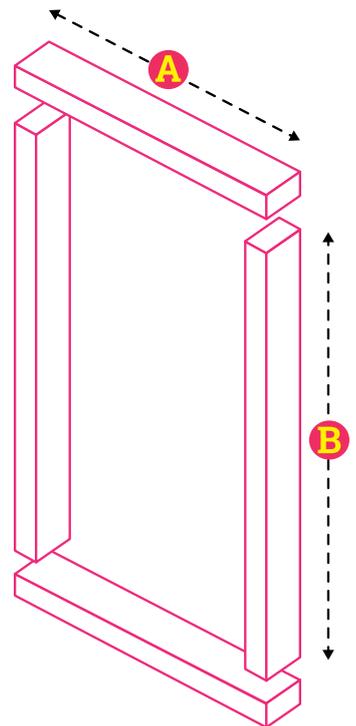
### ASSEMBLE THE FRAME

As this is the main frame, it's best to have this as sturdy as possible. Screwing the wood into place with two screws will keep it nice and strong, but adding a little wood glue will make sure it stays together even better.

## STEP 02:

### CUT THE WOOD

The basic frame is made up of four pieces of wood. You can view it as two side pieces and a top and bottom piece. The sides should be slightly longer as the mirror will be in portrait orientation. The sides should be the same length as the long edges of the monitor, while the top and bottom should be the length of the short sides plus the width of the wood pieces so that they can fit neatly on top as a rectangle. Make sure to not make the fit too snug for the monitor – allow for an extra millimetre or two.



**A** Width of monitor + depth of wood × 2

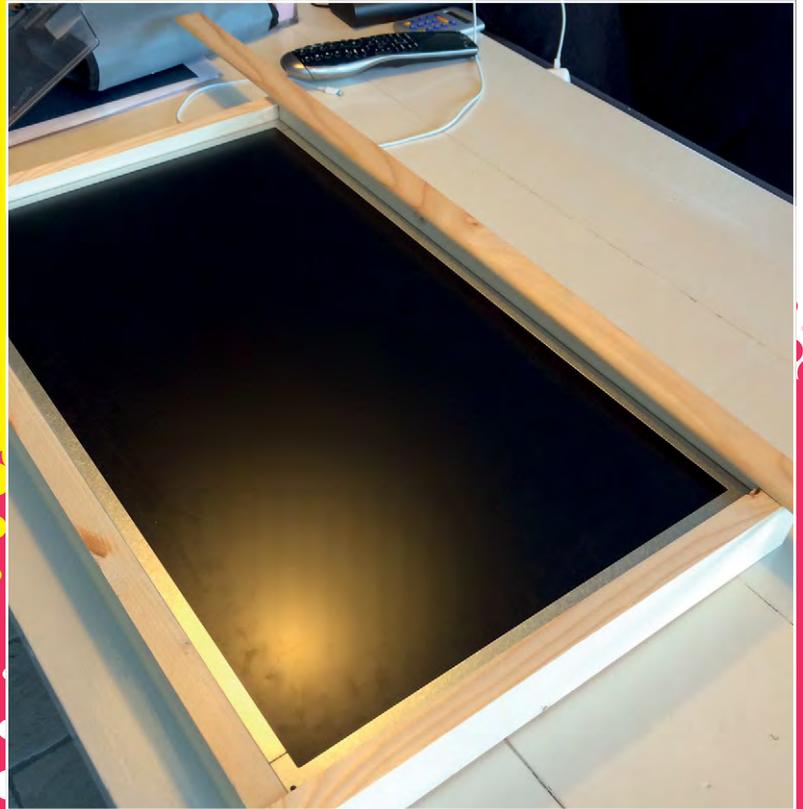
**B** Height of monitor

**REMEMBER:** MEASURE TWICE, CUT ONCE



**STEP 04:****CUT THE FRONT**

With the main frame done, we can now add the front of the frame. This has two functions: it covers the bezel on the monitor, making it look a bit better aesthetically, and acts as a lip to hold the monitor and mirror in place. Make sure the material is deeper than the wood for the frame to make this lip, and cut the ends at an angle for when you join them all together.

**STEP 05:****ATTACH THE FRONT**

Carefully nail on the front pieces, making sure they're flush to the edge of the outside. Do one nail at each end first to make sure it is orientated correctly (don't nail it in fully until you're sure), or use a vice lightly gripping it all together. Once you've done a couple of the pieces, the other two will be easier to get right.



## STEP 06:

### FINAL PIECES

Drill a few holes through the top and bottom parts of the frame, as shown – this aids ventilation. Nothing should be getting toasty hot in there, but it's better to have some air going through. You'll also need a piece, as shown, with some slots to hang over screws in the wall. You should make a little indentation on the back of the bottom piece of wood so the power cables can run through to the monitor and Pi. Finally, create some small brackets that you can fasten to the frame to keep the monitor from falling out of the back.



## STEP 07:



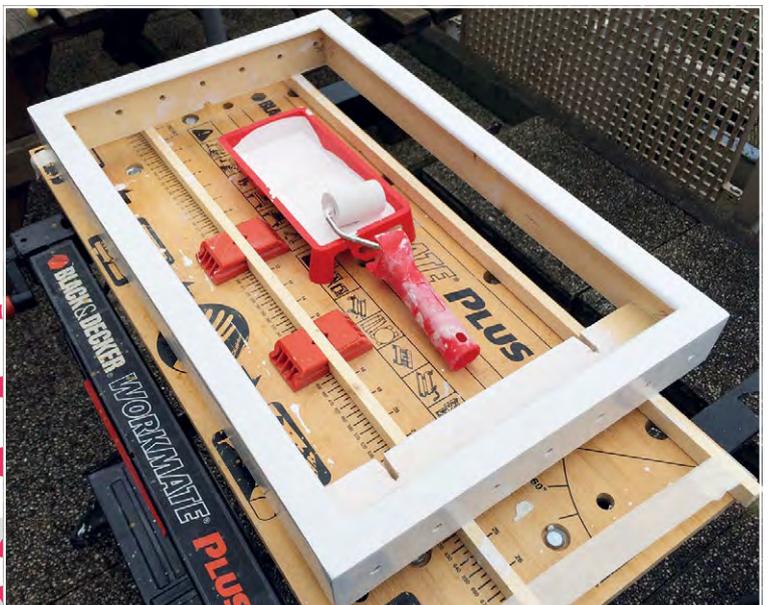
### SANDING & SMOOTHING

Use a little filler to cover any dents, as well as the nail heads on the front of the frame. Give it a sand as well, to smooth everything off and prepare for painting.

## STEP 08:

### GET PAINTING

In this example, the frame has been painted white with some wood paint. Make sure to do this in a well-ventilated room.



**STEP 09:**

**SLOT IN THE MIRROR**

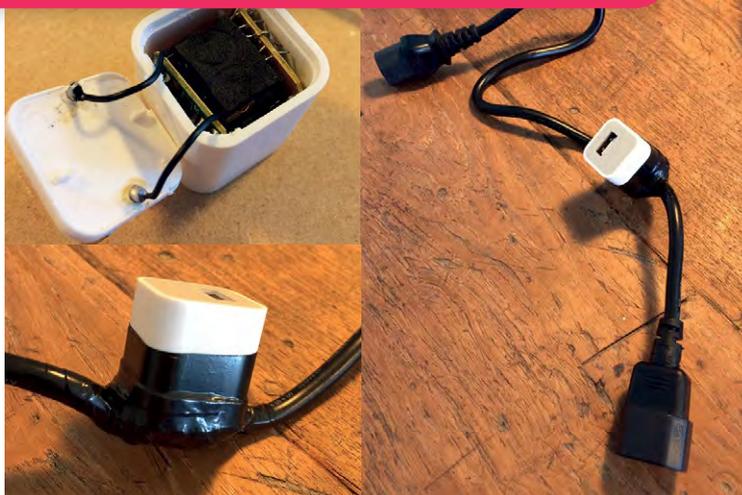
When the paint is dry, flip the frame over so it's resting on its front and carefully lower the mirror in. We're going to keep it in place using the overhang of the front moulding and the monitor, effectively sandwiching it between them.

**STEP 10:**

**INSTALL THE ELECTRONICS**

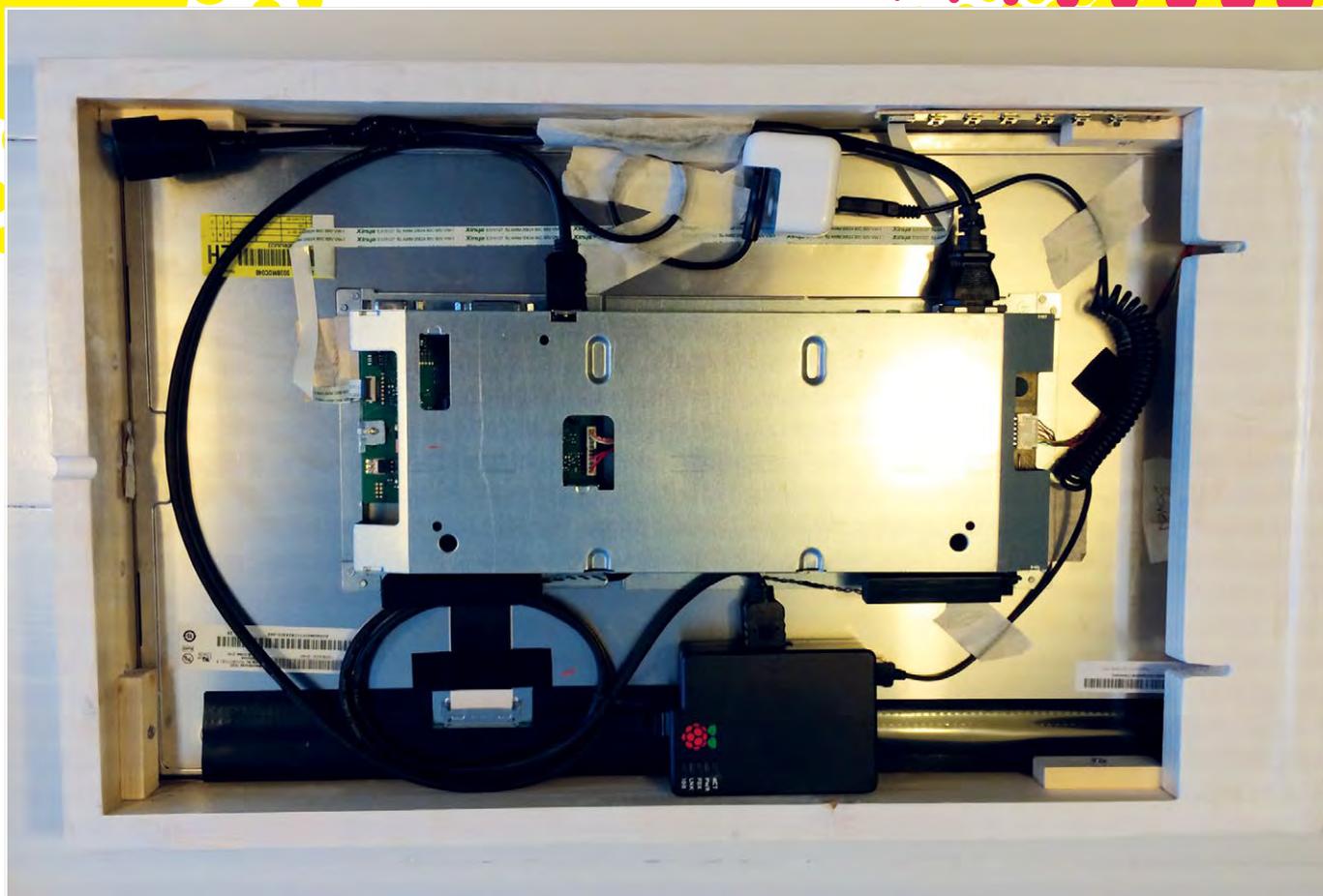
This bit is pretty easy: just put the monitor into the back, fasten it in with the little wooden brackets, and hook up the Raspberry Pi to the HDMI. Run power cables through the indentation you made.

**STEP 11: EXTRA TIP!**



**ONE CABLE, TWO DEVICES**

One thing the more advanced maker could do is combine the power cable to the TV with a plug-to-USB adapter. This way you only need one cable running through to the mirror.



# PROGRAM YOUR MIRROR

Install the software to your Raspberry Pi and make your mirror truly magical

Once you've hung your mirror on the wall, or placed it wherever it's now going to live, it's time to install the software. Michael has made the process incredible easy, and all you need to do is type the following command:

```
curl -sL http://magpi.cc/MirrorInstall | bash
```

It will go through the installation process and set up some defaults... and that's it! Your mirror is ready to use.

## PERSONALISATION

Well, almost ready: you may want to have a quick look at editing those default values to make sure it's running as you'd like it to. The settings are kept in a config file which you can create by using:

```
cp ~/MagicMirror/config/config.js.sample ~/MagicMirror/config/config.js
```

You can now access and modify it with:

```
nano ~/MagicMirror/config/config.js
```

Here are some of the options you can modify...

```

Magic Mirror Config Sample
* By Michael Tenow Email: michaeltenow.nl
* MIT Licensed.
*/

var config = {
  port: 8080,
  ipWhitelist: ["127.0.0.1", "::ffff:127.0.0.1", "::1"],
  language: "en",
  timeFormat: 24,
  units: "metric",

  modules: [
    {
      module: "alexa",
    },
    {
      module: "updateNotification",
      position: "top_bar"
    },
    {
      module: "clock",
      position: "top_left"
    },
    {
      module: "calendar",
      header: "Up Holidays",
      position: "top_right"
    }
  ]
};

```

The sample configuration file gets you started, but there's way more you can do

OPTION	DESCRIPTION
<b>port</b>	The port on which the MagicMirror server will run on. The default value is 8080.
<b>address</b>	The IP address of the mirror you use to connect to it.
<b>ipWhitelist</b>	The list of IPs which are allowed to connect to the mirror. The default value is ["127.0.0.1", "::ffff:127.0.0.1", "::1"]. It is possible to specify IPs with subnet masks (["127.0.0.1", "127.0.0.1/24"]) or define IP ranges (["127.0.0.1", ["192.168.0.1", "192.168.0.100"]]).
<b>zoom</b>	This allows scaling of the mirror interface with a given zoom factor. The default value is 1.0.
<b>language</b>	The language of the interface. Possible values are en, nl, ru, fr, and so on, but the default value is en.
<b>timeFormat</b>	The style of clock to use. The accepted values are 12 and 24. The default is 24.
<b>units</b>	The units that will be used in the default weather modules. Possible values are metric or imperial. The default is metric.
<b>modules</b>	An array of active modules. There must always be an object in here.
<b>electronOptions</b>	An optional array of Electron (browser) options. This allows configuration of e.g. the browser screen size and position (defaults <b>.width = 800</b> and <b>.height = 600</b> ). Kiosk mode can be enabled by setting <b>.kiosk = true</b> , <b>.autoHideMenuBar = false</b> , <b>.fullscreen = false</b> .

**DATE**

What's the time and today's date? Better stop admiring the reflection or you will be late for work!

Tuesday, December 13, 2016  
15:10

**TEMPERATURE**

What's the weather going to be like today? Is it umbrella weather? Better make it back before sundown as well

**CALENDAR**

Is today a holiday? Is there a meeting coming up? Is this colour of tie appropriate for it?

US HOLIDAYS  
 Christmas in 11 days  
 New Year's Day in 18 days  
 M.L. King Day in a month  
 Valentine's Day in 2 months  
 Presidents' Day in 2 months  
 Good Friday in 2 months  
 Easter Sunday in 2 months  
 Mother's Day in 2 months

2° -0.6°  
22:29

WEATHER FORECAST NEW YORK, US

Tue	3.2	-0.9
Wed	3.9	-0.3
Thu	-4.7	-6.2
Fri	-6.0	-8.6
Sat	8.5	8.1
Sun	10.1	9.1

**NICE MESSAGE**

Thanks mirror, you're looking pretty good yourself! My compliments to the amazing person that made you

You look sexy!

**NEWS**

Any important news to catch up on? Is there traffic on the commute? Oh, look: *Crystal Maze* is coming back!

Rex Tillerson, Exxon C.E.O., Chosen as Secretary of State

# CUSTOMISING YOUR MIRROR

Make your mirror truly yours by adding and customising modules

As well as the pre-installed default modules, you can add third-party modules that have either been created by the community or yourself. They're easy to add: you just need to download the files and then update the configuration file to use them.

First of all, take a look on the page for the MagicMirror modules in the GitHub repo here: [magpi.cc/WKRaXQ](https://magpi.cc/WKRaXQ). You'll find a list of great modules to add, such as a Bitcoin monitor and something that displays today's XKCD comic. Pick one you like and copy the link location to it.

To install the module, first move to the **modules** folder with `cd~/MagicMirror/modules` and then download the data for it with:

```
git clone https://github.com/[author]/[module-name]
```

...with the GitHub link pasted from the link you copied. Check out the readme for the module and see if there are any other steps to perform, otherwise open up the **config.js** file from before and add the module to the module section. You'll need to format it something like:

```
{
  module: 'module name',
  position: 'position',
  header: 'optional header',
  config: {
    extra option: 'value'
  }
},
```

Here's the full list of options to use...

OPTION	DESCRIPTION
<b>module</b>	The name of the module. This can also contain the subfolder. Valid examples include <b>clock</b> , <b>default/calendar</b> , and <b>modules/[module name]</b> .
<b>position</b>	The location of the module on the mirror. Possible values are:  <b>top_bar</b> , <b>top_left</b> , <b>top_center</b> , <b>top_right</b> , <b>upper_third</b> , <b>middle_center</b> , <b>lower_third</b> , <b>bottom_left</b> , <b>bottom_center</b> , <b>bottom_right</b> , <b>bottom_bar</b> , <b>fullscreen_above</b> , and <b>fullscreen_below</b> .  This field is optional, but most modules require it to be set. Check the documentation of the module for more information. Multiple modules with the same position will be arranged based on the order in the configuration file. Some trial and error is required.
<b>classes</b>	Additional classes which are passed to the module. The field is optional.
<b>header</b>	To display a header text above the module, add the header property. This field is optional.
<b>disabled</b>	Set disabled to true to skip creating the module. This field is optional.
<b>config</b>	An object with the module configuration properties. Check the documentation of the module for more information. This field is optional, unless the module requires extra configuration.

**DON'T WANT TO DISMANTLE YOUR MIRROR?**  
Use SSH from another computer to access the Pi inside

# SUPPORT & COMMUNITY

You've made your mirror, now here are some ways to keep up on its developments

## MAGICMIRROR SITE

[magicmirror.builders](http://magicmirror.builders)

Your first stop for MagicMirror info is the home site for the software. You'll find some handy links, such as those to the blog for updates on the project, GitHub for the source code and more in-depth configuration options for the mirror, as well as a link to the forum and available modules. It's also a good way to quickly introduce a friend to the concept.

## MAGICMIRROR FORUM

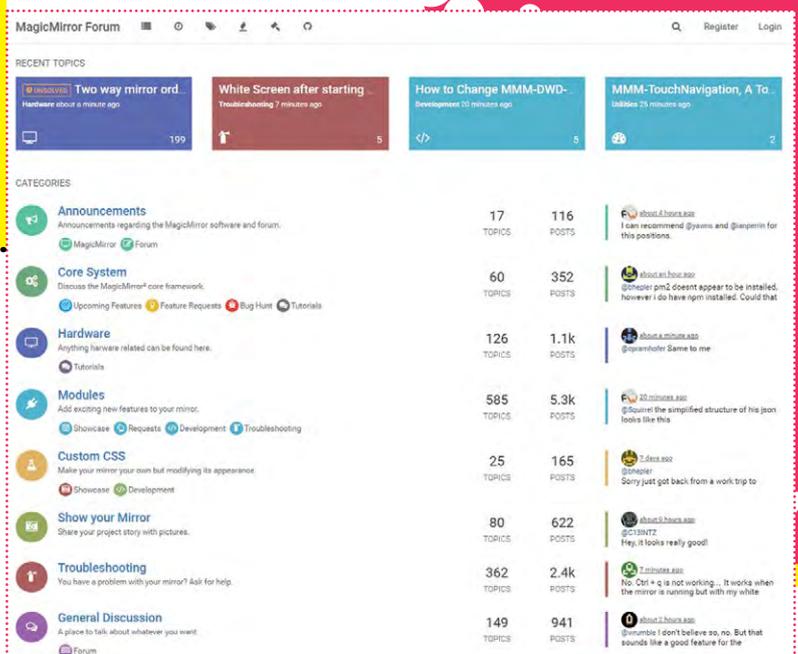
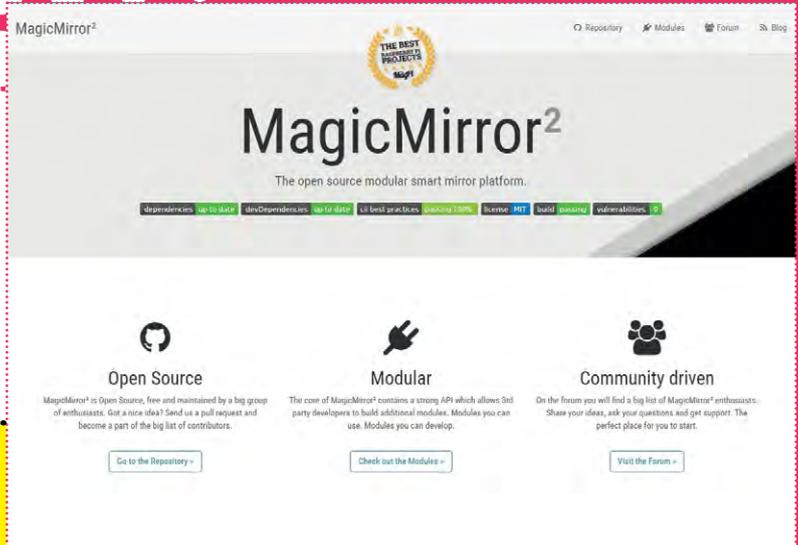
[magpi.cc/2je2dXI](http://magpi.cc/2je2dXI)

The community for the MagicMirror lives here, and the answer to many problems you might face in the long term have likely already been answered there. They're a friendly bunch, though, so if you can't find a solution, you can always have a chat with them to find out what might be going wrong. You'll also be able to compare notes about any modules you decide to make or any other little upgrades or building tips you might want to know about.

## MAGICMIRROR MODULE DEVELOPMENT DOCUMENTATION

[magpi.cc/2jebFux](http://magpi.cc/2jebFux)

Fancy making your own custom module for your mirror? You'll need to know how the API and code generally works, and you can do that with the documentation provided on the MagicMirror GitHub repo. We've seen some excellent add-ons for the mirror code, including seasonal variations and live train times. All you need is a way to get the data!



## MagicMirror<sup>2</sup> Module Development Documentation

This document describes the way to develop your own MagicMirror<sup>2</sup> modules.

### Module structure

All modules are loaded in the `modules` folder. The default modules are grouped together in the `modules/default` folder. Your module should be placed in a subfolder of `modules`. Note that any file or folder you create in the `modules` folder will be ignored by git, allowing you to upgrade the MagicMirror<sup>2</sup> without the loss of your files.

A module can be placed in one single folder. Or multiple modules can be grouped in a subfolder. Note that name of the module must be unique. Even when a module with a similar name is placed in a different folder, they can't be loaded at the same time.

### Files

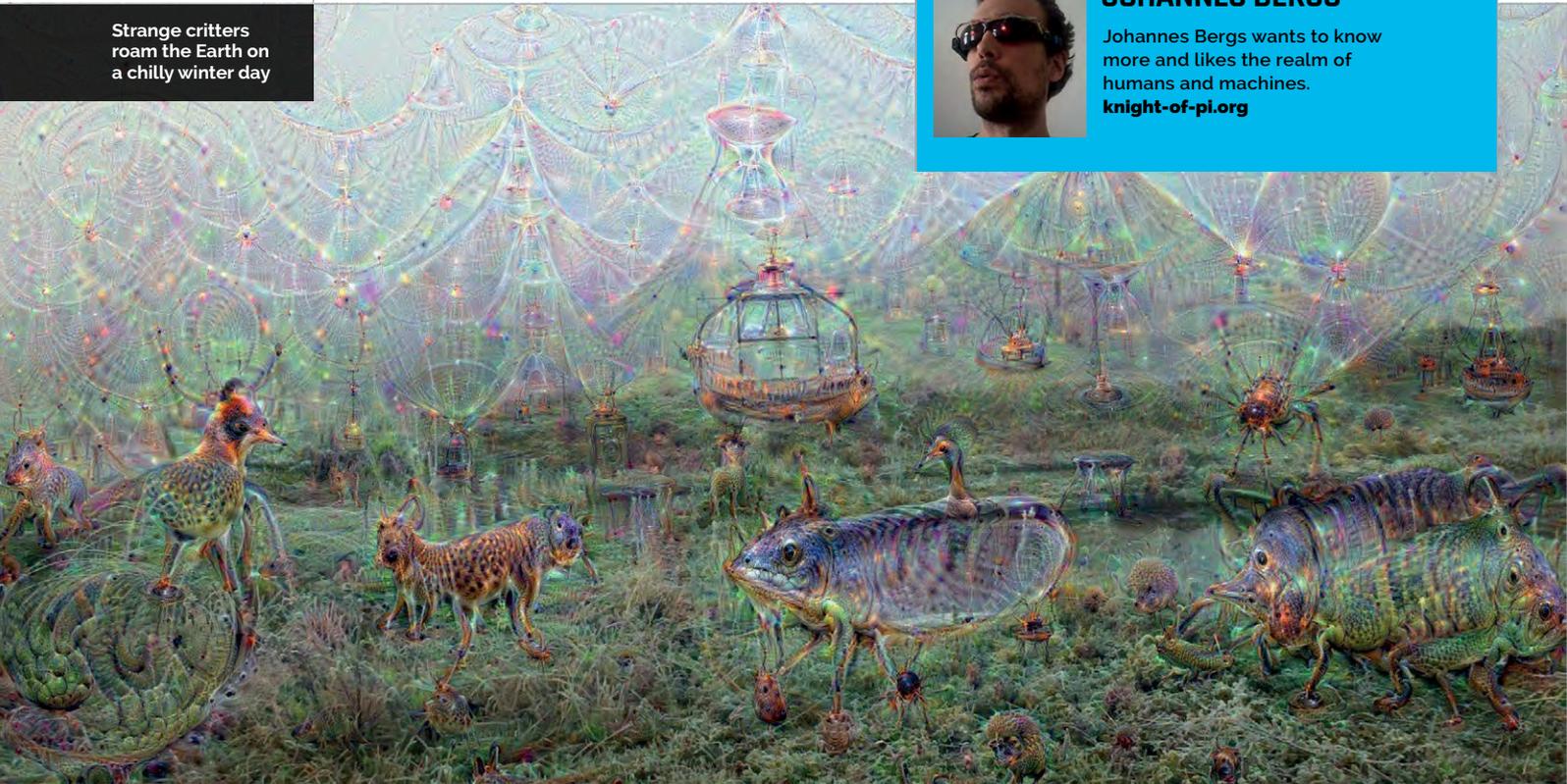
- `modulename/modulename.js` - This is your core module script.
- `modulename/node_helper.js` - This is an optional helper that will be loaded by the node script. The node helper and module script can communicate with each other using an intergrated socket system.
- `modulename/public` - Any files in this folder can be accessed via the browser on `/modulename/filename.ext`.
- `modulename/anyfileorfolder` Any other file or folder in the module folder can be used by the core module script. For example: `modulename/css/modulename.css` would be a good path for your additional module styles.

Strange critters roam the Earth on a chilly winter day



### JOHANNES BERGS

Johannes Bergs wants to know more and likes the realm of humans and machines.  
[knight-of-pi.org](http://knight-of-pi.org)



# GOOGLE DEEPAIDREAM ON THE RASPBERRY PI

## You'll Need

- ▶ Camera Module [magpi.cc/bASFoG](http://magpi.cc/bASFoG)
- ▶ PsyCam [magpi.cc/zeCSQOD](http://magpi.cc/zeCSQOD)

## RASPBERRY PI 2 COMPATIBILITY

Although the Pi 3 /3B+ with its advanced processing power is preferable, the project will run on a Pi 2 without modifications.

Do computers dream of electric sheep? Find out by making your Raspberry Pi dream with the amazing Google DeepDream algorithm

**D**reams can be beautiful, surreal, or even scary. While we can assume that they play a role in learning, their exact purpose remains unknown. What would a computer see if it could dream? You can find out by installing the amazing Google DeepDream software on your Raspberry Pi!

DeepDream is derived from a neural network for object recognition, which has been modified to express every object it vaguely recognises. The resulting images are impressively surrealistic, bearing more than a passing resemblance to the paintings of artists like Salvador Dalí.

Neural networks are a very versatile technology. They can be applied to a variety of complex problems like facial recognition, text translation, or even playing games like Go.

Caffe, the neural network upon which Google built DeepDream, was originally developed for identifying objects in images. To achieve this, the network is trained on a large set of images to identify the main features of the objects presented.

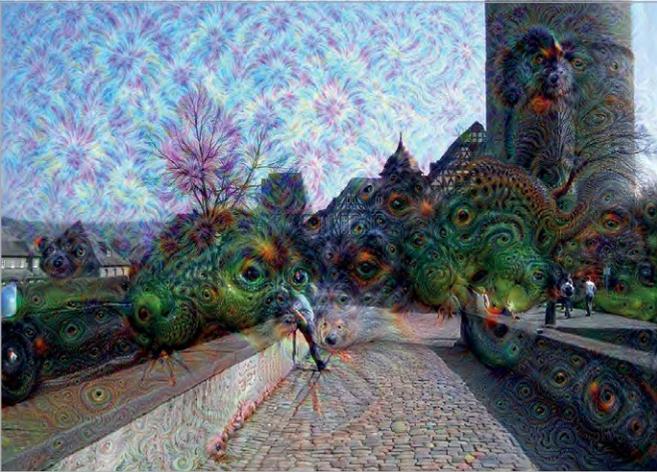
## Artificial neurons

Artificial neurons mimic their natural counterparts, the neurons in the human brain. Every artificial neuron has an activation threshold and numerous weighted incoming and outgoing connections to other neurons. If the sum of the weighted signals from the incoming connections exceeds the activation threshold, the neuron fires a signal.

## Neural networks

Generally, neural networks consist of at least three layers of neurons. The input layer reads the input, one or more hidden layers process the information, and the output layer shows the result.

Consider a neural network for recognising the numbers zero to nine in images with a size of 28×28 pixels. The input layer would have 784 neurons, one for every pixel. If an image is presented to the input layer, every neuron in it produces a signal with a strength corresponding to the greyscale value of its pixel, with a darker pixel generating a stronger signal.



**Above** A magical city in a far-off land

Every input neuron is connected to all neurons of the first hidden layer, but every connection has a distinct weight. The higher the weight, the more of the input neuron's signal reaches the hidden neuron. If the total signal strength arriving at a hidden neuron tops its activation threshold, the hidden neuron fires a signal to all neurons of the next layer which, in this simplified example, is the output layer. Again, the signal intensity depends on the weight of the connection and the incoming signal strength.

The output layer has one neuron for every object to be classified, so ten neurons are used to identify the numbers from zero to nine. If the activation threshold of an output neuron is surpassed by the weighted incoming connections, the resulting signal strength is a measure of confidence in the classification.

## Machine learning

The weights and thresholds are initialised randomly, which can cause very bad classification results.

If an output neuron is activated incorrectly (e.g. if an image from the training set shows a five, but the output neuron for two produces a strong signal), its activation threshold and all the weights of its incoming connections are adjusted. Then, the error is propagated back proportionally through all connected neurons lower in the chain, from the highest hidden layer down to the input layer. This process is repeated for all of the images in the training data set until the results given by the network begin to improve.

Instead of accurately detecting objects in images, by contrast, DeepDream actually changes the input image to make it more similar to the objects learned from the training data.

Imagine a cloud as an input image. Some structure in the cloud might have a vague similarity to the features DeepDream associates with a 'dog-like' object. If this is the case, the input image is changed to look more dog-like.

Additionally, DeepDream allows for selection of the hidden layer depth. Since a layer has more detail the closer it is to the output level, output images can range from basic shapes to detailed dream-like creatures.

## Have some DeepDreams

Insert an SD card with Raspbian installed into the Raspberry Pi, attach the Camera Module and peripherals, then boot it up. To install the DeepDream software, enter the following in a Terminal:

```
mkdir ~/deepdream && cd ~/deepdream
git clone https://github.com/JoBergs/PsyCam
cd PsyCam
python install_tools.py packages
python install_tools.py caffe
python install_tools.py protobuf
python install_tools.py camera
sudo reboot
```

After downloading the project from GitHub, use the custom installer to first install all packages with **pip** or **apt-get**. Then, install the open-source neural network framework Caffe. Because speed matters, you should also install the serial data processor protobuf from Google. Finally, activate the camera and reboot.

All in all, the installation takes a few hours, so you'll need to be patient! The installer should also work for any newish Ubuntu operating system. If you encounter problems, try using the manual installation instructions at [magpi.cc/2eCSxDt](http://magpi.cc/2eCSxDt).

Enter the following, and your Pi can start to dream:

```
cd ~/deepdream/PsyCam
python psycam.py
```

The network parameters depth (**-d**), octave (**-o**), and type (**-t**) are randomized. Add a **-c** to dream continuously. Pass **-i IMAGE.jpg** to use an image as the base for the dream instead of a snapshot. Find more information on input arguments by checking the command-line help:

```
python psycam.py -help
```

When the Pi finishes dreaming, the dream and the original photo are stored in the directory **/home/pi/deepdream/PsyCam/dreams** with a timestamp. You can watch them by opening the directory in the file browser and double-clicking the image.



Alice in a very colourful wonderland

Language

> PYTHON

DOWNLOAD:  
[magpi.cc/2eCSQOD](http://magpi.cc/2eCSQOD)

ORIGINAL  
DEEPDREAM  
ARTICLE

Have a look at the amazing original article about DeepDream by Alexander Mordvintsev, Christopher Olah, and Mike Tyka here: [magpi.cc/2eCRH3](http://magpi.cc/2eCRH3)

NEURAL  
NETWORK  
TUTORIAL

If you want to learn more about neural networks, why not check out this interesting and well-written introduction by Michael Nielsen?: [magpi.cc/2eCU01u](http://magpi.cc/2eCU01u)

KIVY GUI

Try the powerful Python GUI framework Kivy ([kivy.org](http://kivy.org)) for building a GUI: it is slick enough to embrace DeepDream.

# 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](http://magpi.cc/259aT3X)

# MAKE STUNNING SCRATCH ART

Hack a DJ Hero turntable to simulate a harmonograph and make intricate line art by scratching

In this guide we're going to look at hacking an unlikely interface: we found a DJ Hero turntable at a local charity shop going cheap, so we had a go at seeing if it could be interfaced with the Raspberry Pi. There are a few versions of the turntable for different consoles, but the one we're using here is designed for the Wii. This is the simplest design to interface to, because to the Pi it simply looks like an I<sup>2</sup>C device. It's designed to plug into the auxiliary port of the Wii controller, but here we wire it directly onto the Pi's GPIO pins. You can cut off the end and solder

## You'll Need

- > DJ Hero turntable – Wii version
- > WiiChuck Nunchuk adapter or similar [magpi.cc/FeAozQ](http://magpi.cc/FeAozQ)
- > Length of 4-way flat ribbon cable

- Harmonograph display driven by the Raspberry Pi
- Control screen, drawing depth, and saving images

- Control parameters with turntable and buttons



Nunchuck Adapter

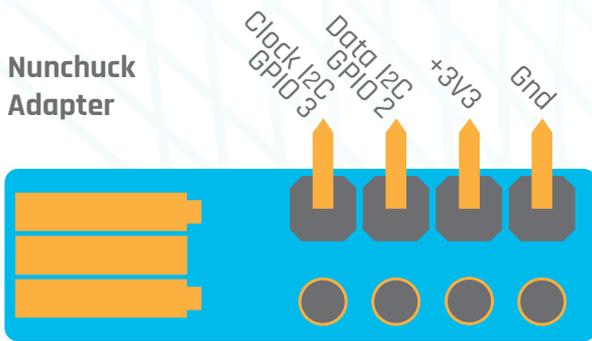


Figure 1 Wiring the Wii Nunchuk adapter to the Pi

extension wires to it, or get a small 'Wii Nunchuk adapter' PCB and wire that to the Pi. Figure 1 shows the wiring you need, as there are only four wires, you can use a length of ribbon cable to attach it to the Pi, with the cover neatly clipping over the top.

Talking to the DJ Hero

The DJ Hero comprises a collection of buttons, knobs, a joystick, and a turntable. Figure 2 shows these controls labelled up as the functions they're going to perform in this project. The I<sup>2</sup>C address of this device is 0x52 and consists of a number of registers. To initialise the device on power-up, you must write the value 0x55 to register 0xF0. Note that each successive read will increment the address of the register read from. So, to read the interface status, you need to set the register to zero and then read data six times – these six bytes contain the information from the DJ deck. The assignment of bits is shown in Figure 3 overleaf, and at first sight looks a bit complex. However, this diagram has been simplified, in that the DJ Hero is capable of having a second turntable on the left-hand side, and the bits corresponding to this second turntable have been omitted.

Language

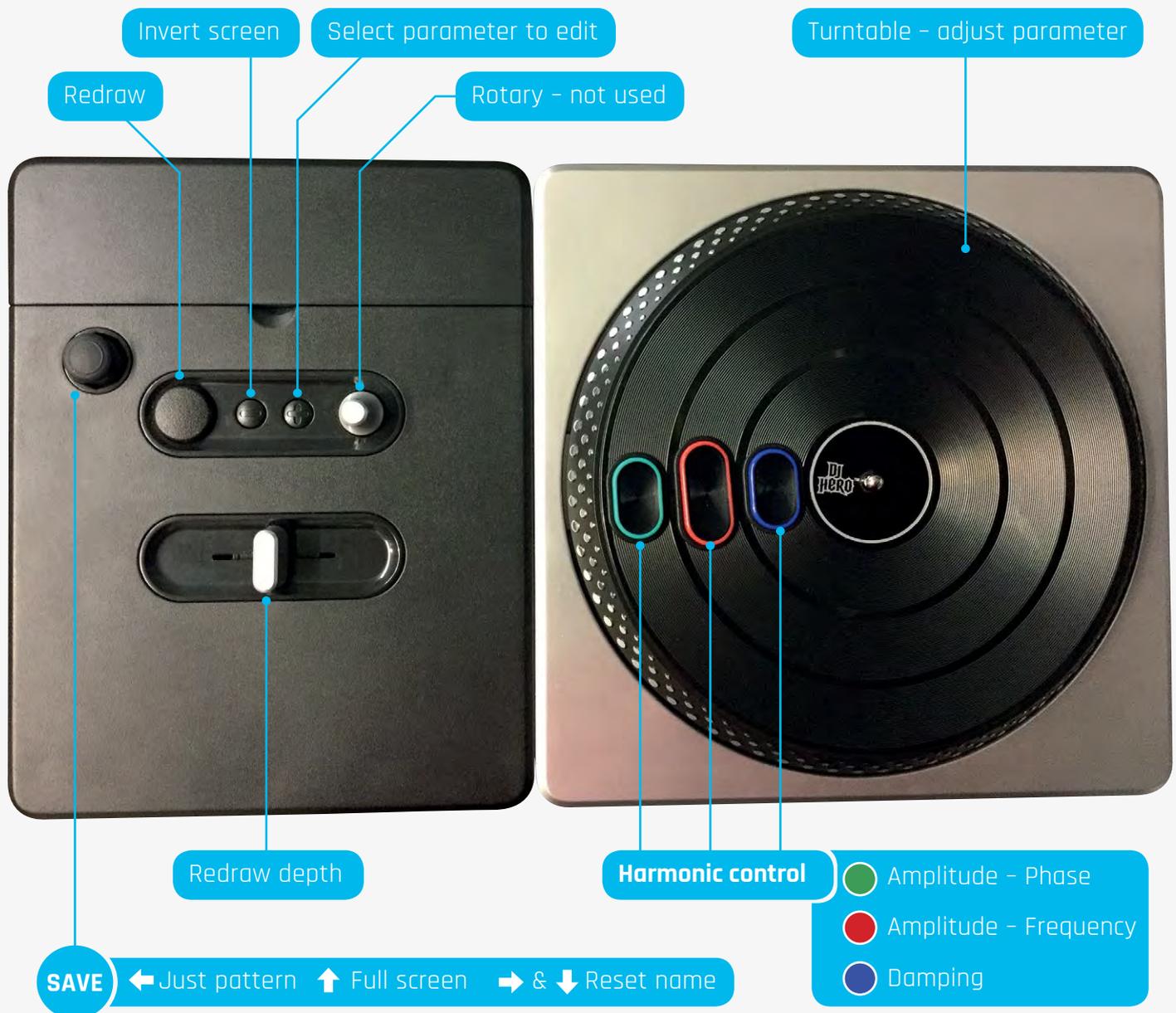
>PYTHON 2

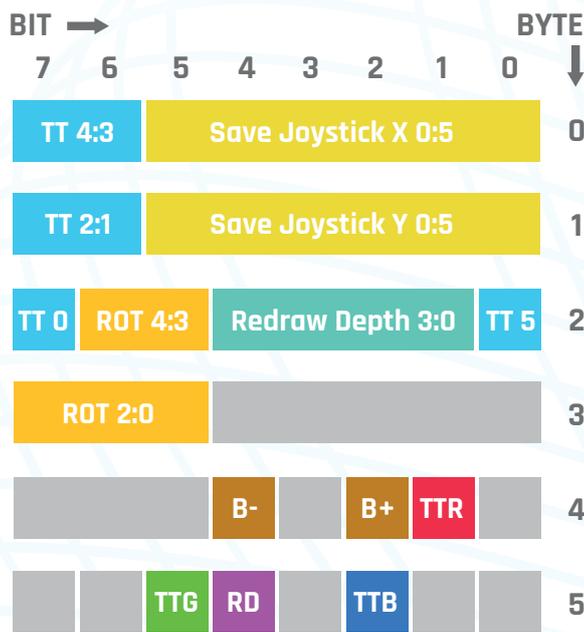
DOWNLOAD THE CODE: [magpi.cc/1NqJmV](http://magpi.cc/1NqJmV)

PROJECT VIDEOS

Check out Mike's Bakery videos at: [magpi.cc/1NqJnTz](http://magpi.cc/1NqJnTz)

Figure 2 The full set of controls on the DJ Hero controller





- TTR Turntable Red Button
- TTG Turntable Green Button
- TTB Turntable Blue Button
- ROT Rotary - Not Used
- B- Invert Screen Button
- B+ Select Parameter to Edit Button
- RD Redraw Button



Figure 3 The data block returned from the DJ Hero

The buttons are simple: one bit in one byte of the returned data block, and other controls are spread over several bits. The two axes of the joystick are the first six bits in the first two bytes, whereas the values from the slider are in bits 4 to 1 of byte number 2. The rotary control’s five bits are split up over two bytes, whereas the turntable is scattered all over the place in four locations. You’ll need a bit of software to pull out the data you want from this block.

The turntable returns a value, not of its position, but of the change in the position since the last time it

was read. This is a five-bit signed value, with the most significant bit being the sign bit. To be useful, this must be converted into the sort of number that Python can understand, so bit 5 of the turntable number is propagated to all the higher bits in the word; this is known as sign extension. Also, for negative values, the number must be in the ones’ complement format; that is, with bits 0 to 4 inverted and 1 subtracted. This format is universally used in computer languages and prevents you from having two different bit patterns for plus zero and minus zero, meaningless concepts.

### The graphics

We’re going to use this controller to control a simulation of a harmonograph. These devices were originally a very popular Victorian contraption, consisting of a pen on the end of a compound pendulum that was sent swinging and produced Lissajous-like patterns. Here, however, we’re going to simulate a system with four pendulums that will exceed the flexibility of any mechanical device. Two pendulums control the X position of the pen, and two control the Y position, the final position being the simple sum of each of the two pendulums on each axis. There are four parameters that control the exact path of each pendulum.

01. The Amplitude or extent of the swing
02. The Frequency of the swing, determined by the pendulum’s length
03. The Phase of the swing, a fixed value added to the frequency
04. The Damping or decay, which is the slow reduction of amplitude as the friction takes energy out of the pendulum

It’s the subtle interaction of these four parameters that produces the near-infinite number of patterns that can be obtained. There’s also the factor of how long you let it swing for, producing a less or more dense pattern. Eventually the damping will make all the pendulums stop, although in this simulation you can set the damping value to zero if you wish.

### The software

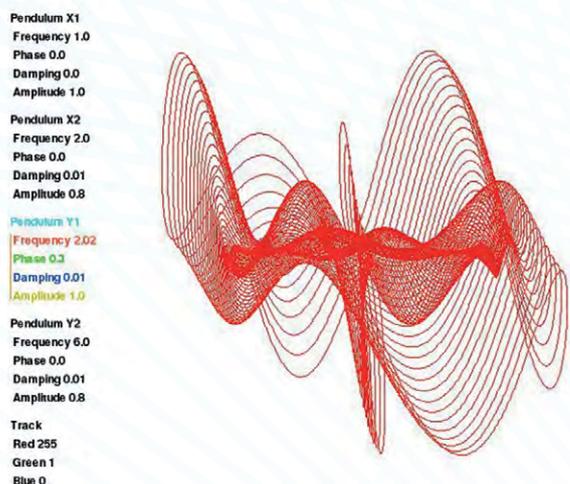
The code for this is **DJArt.py** and is written under the Pygame framework. The drawing area is set by the **screenSize** variable, and you could make the window size larger simply by altering this value if you have a higher-resolution monitor. In addition, the window is wider by 100 pixels given by the **controlBar** variable, in order to have room for the editable parameters. The **swing** function is the one that actually draws the path of the pendulums on the screen, with the points calculated by the **calcNewPoint** function. Rather than

showing every point on the screen as it is plotted, we speed things up considerably by drawing for 40 ms and then displaying the result. All the rest of the code is devoted to editing the parameters of the program. The `initValues` function sets the startup parameters; you might want to change this as you explore new combinations of pendulum frequencies.

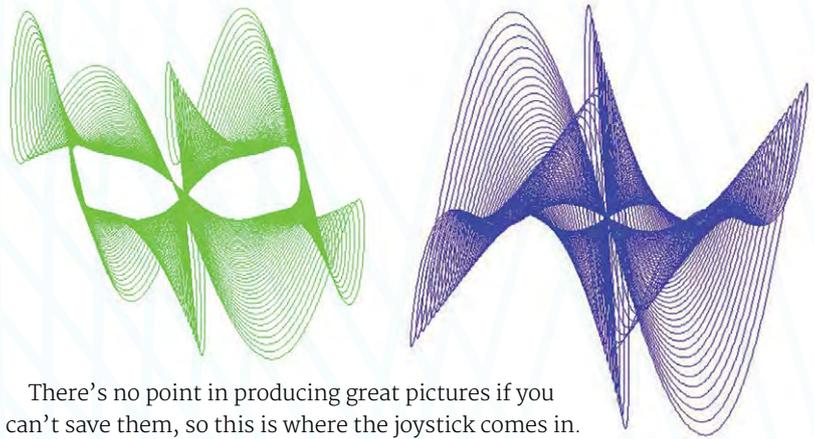
The DJ Hero peripheral is handled by the `readDJ` function, and the `buttonFix` function returns a one if the button is pressed, or else it returns a zero. The raw data returns the bit as zero when a button is pressed, and a one when not. The `adjustPends` function imposes a wraparound value limiting the range on the parameters. Keeping the amplitude at one or less ensures that the drawing stays within the screen area.

## Using the software

When running, the software starts out by having a starting set of parameters producing a pattern on a black background. The - (minus) button will toggle the background between black and white. **Figure 4** shows the full program window against a white background. The Redraw button will run the simulation for a time determined by the 'Drawing Depth' slider; with the slider to the left, you get the shortest time. The + button allows you to edit the parameters of each of the four pendulums and the track colour. The selected pendulum's parameters are displayed in that colour, with the others shown in the foreground colour. Changing a parameter is simple: just press the appropriate colour button on the turntable and move it clockwise for an increase, or anticlockwise for a decrease. For example, the pendulum's frequency is shown in red, so holding down the red button and moving the turntable will change the frequency. However, there are four parameters and only three buttons. The fourth parameter is amplitude and coloured yellow; to change this, simply hold down both the red and the green keys at the same time.



**Figure 4** The control screen, and the result of different parameters taken from the controller



**Figure 5** Designs created using the DJ Hero controller in this tutorial

There's no point in producing great pictures if you can't save them, so this is where the joystick comes in. Move it fully left to save just the picture and fully up to save the picture along with the parameter values. The first time you do this, a dialogue box will prompt you for the name and place to store the image. After the first one is saved, subsequent saves append a number to the file name. Once a picture is saved, there is a 'saved' message in the lower left-hand corner of the window which lasts until you press another control. If you want to use a new name or new folder, then move the joystick fully to the right or fully down and you will get the file dialogue box next time you save an image.

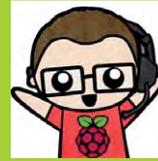
## Getting good results

The patterns look best on the screen with a black background, but if you're going to print it out then it looks a lot better on a white background. We think the best patterns are obtained when the frequency of two pendulums are set as a simple ratio of each other, or slightly off a simple ratio, like **1:2.01**. Unrelated frequency values tend to produce scribble-like, chaotic results. It's amazing how small changes in phase or damping can radically change the picture. This is especially true of the phase, which wraps round every cycle; as this is shown in radians, this is at a value of **5.3**. When changing a parameter, there's a short real-time preview of the resulting shape. Other parameters have lower limits of either zero or one, while the colour has an upper limit of 255. You can actually get Lissajous figures if you want by setting the amplitude of pendulums 1 and 3 to zero, and the amplitude of the other two to one, along with a decay of zero. **Figure 5** shows a selection of patterns we have obtained so far.

## Taking it further

The only control not used is the rotary one. You might like to try to add some code so that this controls the thickness of the plotted line. Altering the short variable value to a larger number alters the depth of the pattern preview; you might like to make this bigger if you're using a Raspberry Pi 3 / 3B+, because it can draw faster. You could also add software to change the colour of the track according to the absolute X or Y position or change it between two colours, either abruptly or fading from one to the other over a length of track.





**SPENCER ORGAN**

Chemistry and physics teacher, Raspberry Pi enthusiast and Certified RPi Educator from the West Midlands with a passion for running workshops and building fun, educational, and practical things with the Pi!  
[themakercupboard.space](http://themakercupboard.space)

# BUILD THE MARAUDER'S CLOCK

Know when your family get home with this beautiful Harry Potter-inspired clock

## You'll Need

- > Servos (continuous and position)
- > Watch hands
- > Arduino Uno [magpi.cc/2lO4uNA](http://magpi.cc/2lO4uNA)
- > 2× Mini breadboard [magpi.cc/FidIZR](http://magpi.cc/FidIZR)
- > Transparent medium breadboard [magpi.cc/wRHunb](http://magpi.cc/wRHunb)
- > Pimoroni Explorer HAT Pro [magpi.cc/2lO5ZZ](http://magpi.cc/2lO5ZZ)
- > 2× 5V mini DC relay [magpi.cc/2lOere6](http://magpi.cc/2lOere6)
- > 4× LED, male-to-male and male-to-female jumper cables
- > A box or clock body
- > Background graphics

## TESTING THE CLOCK MECHANISM

The clock mechanism can be tested by connecting digital input 1 on the Arduino to ground.

**W**e all love the many visual elements in the Harry Potter movies, and this project combines two of the most iconic: the Marauder's Map and the Weasley 'who is home?' Clock. The project uses a Raspberry Pi to detect the presence of people on the network and an Arduino Uno to control the clock hands.

### >STEP-01

#### Preparing the setup

We start the project by setting up the Raspberry Pi and Explorer HAT Pro. Install the libraries for the Explorer HAT Pro using the tutorial on the Pimoroni GitHub page here: [magpi.cc/2lOfD0k](http://magpi.cc/2lOfD0k). In this project we will be using two of the 5V DC outputs from the HAT to activate a relay. Note: These ports don't actually output 5V, but connect to ground when activated.

### >STEP-02

#### Configure the code

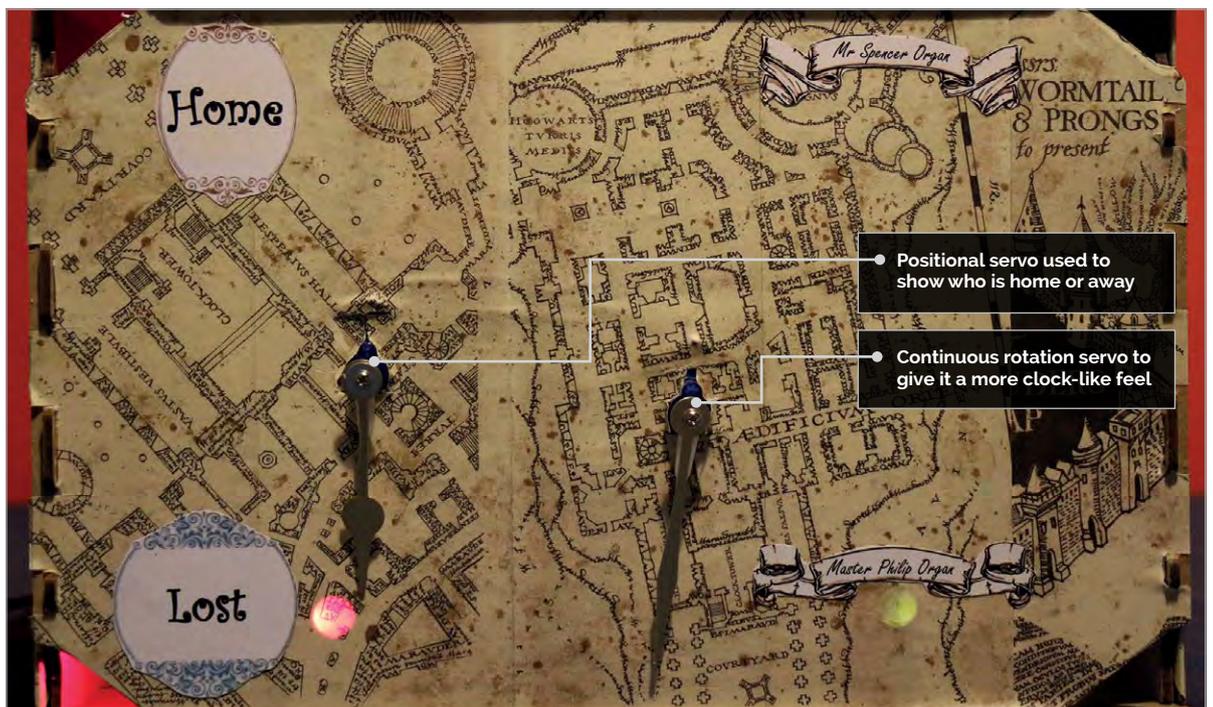
Clone the Python code from [magpi.cc/2lOfkTJ](http://magpi.cc/2lOfkTJ). You will need to find the MAC addresses for each

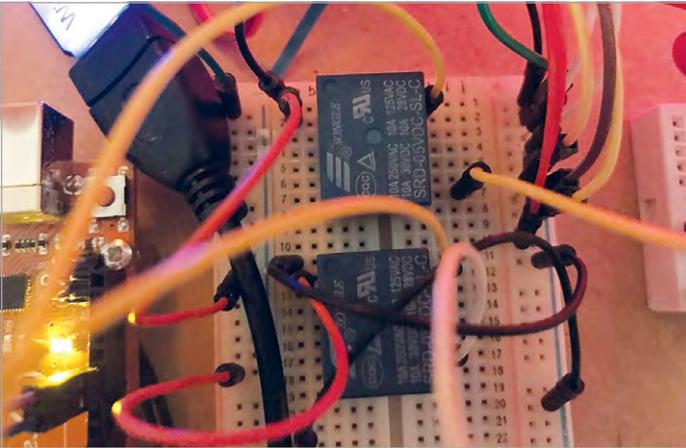
of the devices you will be detecting on the network, and substitute them into the code. It is important that all the devices are on the same network as the Pi. On an iOS device you can find your MAC address under Settings > General > Wi-Fi address; on an Android device, it's in Settings > About phone (or tablet) > Status.

### >STEP-03

#### Connect the servos

We then need to connect the two servos to the Arduino. Use a breadboard to connect a common 5V and ground line from the Arduino. Connect the red power cables on each of the servos to the common 5V line, and the brown wire to the common ground. Use jumper cables to connect the orange signal wire of the continuous servo to digital pin 12 on the Arduino, and the orange signal wire of the position servo to digital pin 11. The continuous rotation servo will become our second hand on the clock face, and the position servo will show whether people are home or away.





**Above** The coil connectors of the relay are connected to 5V and output on the Explorer HAT Pro. The switch connectors are connected to digital input 1 (and 2 for the second relay) and the ground on the Arduino

## >STEP-04

### Connect the LEDs

We will be using four LEDs to light up different parts of the Marauder's map. Connect the short leg (cathode) to a common ground. Connect the long leg (anode) to digital pins 5,6,7, and 8.

The following LED pins are used:

- Digital Pin 5 = Person 2
- Digital Pin 6 = Away notification
- Digital Pin 7 = Person 1
- Digital Pin 8 = Home notification

## >STEP-05

### Link the Pi and Arduino

Before connecting the relays, we will need to look at the pin guide and identify the two pins for the coil and the two pins for the switch. One side of the coil on both relays should be connected to the 5V output on the Explorer HAT Pro. The other side of the of the relay coil should then be connected to output 1 and 2 on the Explorer HAT Pro. The common switch connector on both relays should then be connected to the common ground on the Arduino. The switch connector on the first relay is connected to Digital 1 on the Arduino, and the switch connector on the second relay is connected to Digital 2. We will be using the switch connector, which is activated when the relay receives power.

## >STEP-06

### Assembling the project

We will need to download the code for the Arduino and upload it to the Uno from here: [magpi.cc/2lOfkTJ](https://magpi.cc/2lOfkTJ). We then need to assemble our project. In our example we have used a fruit box, but you could use an old clock, a cardboard box, or a hobby box. We need to (carefully) screw clock hands to the two servos; take care not to move the servo as it can cause damage. Mount the two hands so that that the continuous rotation servo can move freely and the positional servo can move up and down.

## clock.py

```
#!/usr/bin/env python
```

```
import subprocess
import time
import explorerhat
```

```
occupant = ["Person 1 device", "Person 2 device"]
```

```
# MAC addresses for our devices - in order of people above
address = ["XX:XX:XX:XX:XX:XX", "XX:XX:XX:XX:XX:XX"]
```

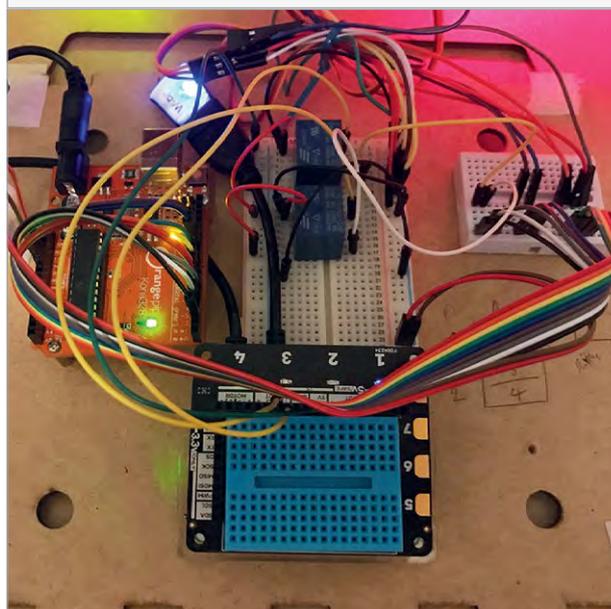
```
while True:
    print("starting loop")
    output = subprocess.check_output(
        "sudo arp-scan -1", shell=True)
    print("starting scan")
    for i in range(len(address)):
        if address[i] in output:
            print(address[i])
            print(occupant[i])

            if "XX:XX:XX:XX:XX:XX" in output:
                #Person 1 MAC address
                explorerhat.output[0].on()

            else:
                explorerhat.output[0].off()

            if "XX:XX:XX:XX:XX:XX" in output:
                #Person 2 MAC address
                explorerhat.output[1].on()

            else:
                explorerhat.output[1].off()
    time.sleep(60)
```



### CONNECTING THE RELAYS

If the output from the relay is reversed (i.e. on when it should be off), try swapping the pole of the switch on the relay.

**Left** The Arduino, Raspberry Pi, and relays are mounted on the rear of the case with long jumper cables so that the box can be opened



**CLIVE WEBSTER**

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



£5 for a WiFi smart button? Yeah, we'll have seven

Buy different buttons so you can identify which is which later

You'll need a low-power, always-on computer with wireless – that's the definition of a Pi Zero W

# HACK AN AMAZON DASH BUTTON

## You'll Need

- ▶ Dash Buttons [magpi.cc/2mUKaQT](http://magpi.cc/2mUKaQT)
- ▶ A smart thing
- ▶ Some patience

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

**T**he 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](http://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.

## Let there be Lite

We used Raspbian 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 LIFX 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

```
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](https://magpi.cc/2mt3zmo)). Or use a service like Pushover ([pushover.net](https://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](https://magpi.cc/2msVvC5)). We'd also thank Ted Benson for being one of the first Dash hackers ([magpi.cc/2msUzhh](https://magpi.cc/2msUzhh)), although updated Amazon firmware is incompatible with his code.

Language

&gt;PYTHON

**DOWNLOAD:**  
[magpi.cc/DashButtonCode](https://magpi.cc/DashButtonCode)


# 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](http://magpi.cc/259aT3X)

# DIY CUT-OUT THEATRE

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

## You'll Need

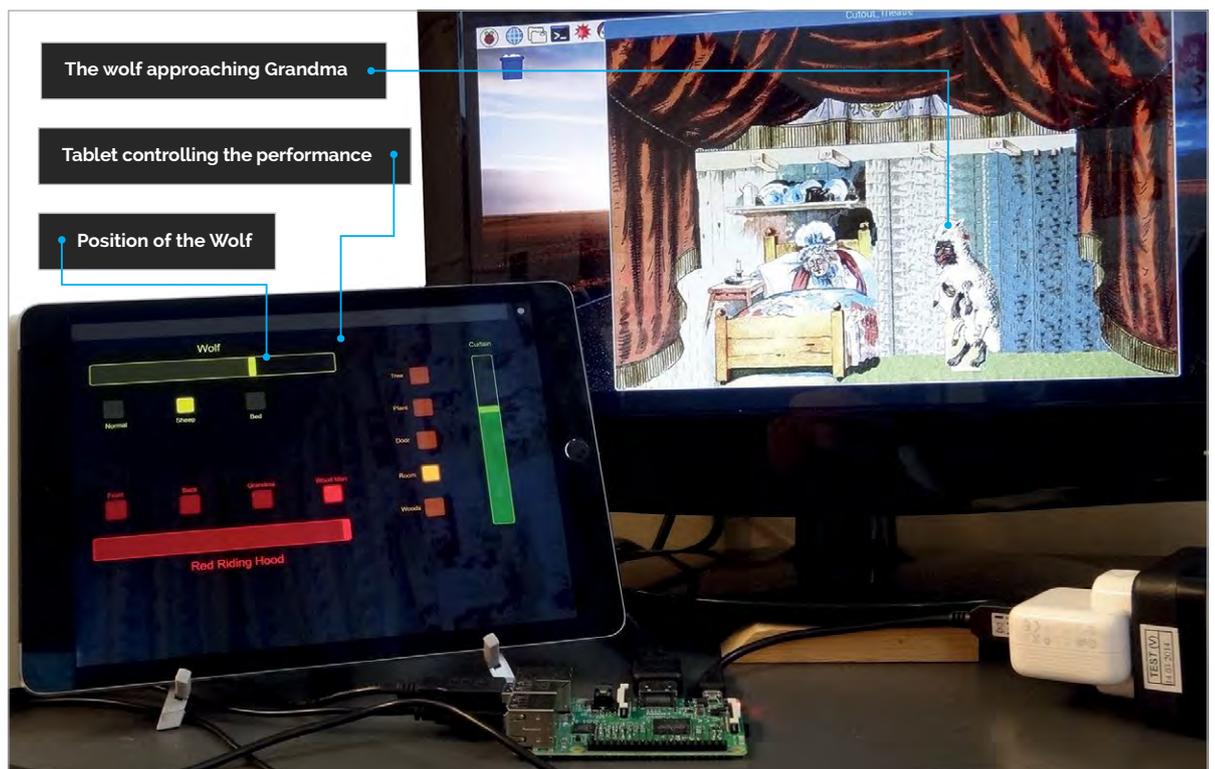
- ▶ Apple or Android mobile device
- ▶ TouchOSC app
- ▶ Raspberry Pi 3

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 than £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](http://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.



The wolf approaching Grandma

Tablet controlling the performance

Position of the Wolf

Language

&gt; PROCESSING

DOWNLOAD:  
magpi.cc/1NqJmVPROJECT  
VIDEOSCheck out Mike's  
Bakery videos at:  
magpi.cc/1NqJmTz

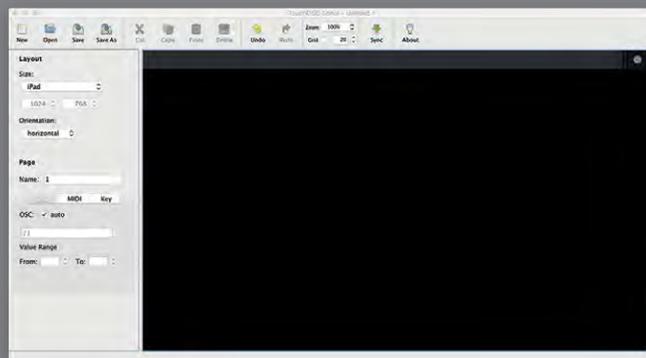
## CREATING THE CUSTOM INTERFACE

### >STEP-01

#### Installing the editor

First, download the free TouchOSC editor. There are versions for macOS and Windows, and a Linux version which will run on the Raspberry Pi. Download the ZIP file by going to [magpi.cc/2mPqfsz](http://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:

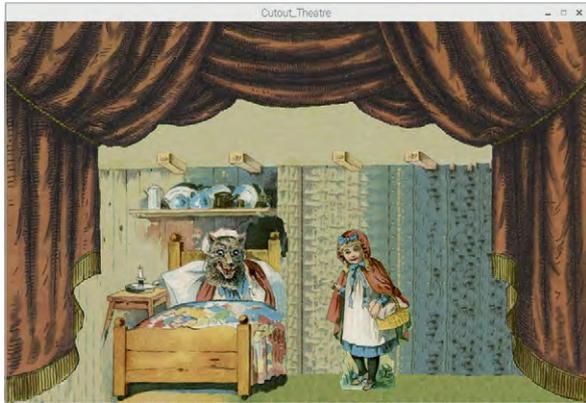
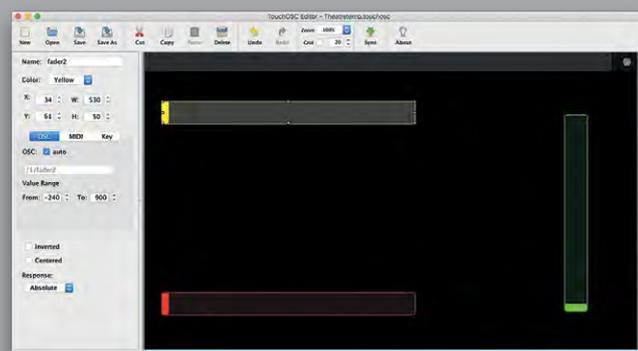
```
java -jar TouchOSCEditor.jar
```



### >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 50, 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.



## 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





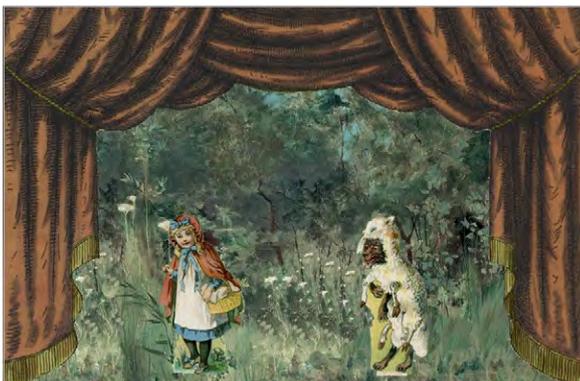
**Fig 1:**  
**HOST:** Set this to the address of the Raspberry Pi.  
 Hover the pointer over the WiFi icon to find it.

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

in *The MagPi* issue 49 ([magpi.cc/49](http://magpi.cc/49)). 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 Pi. This is shown in **Figure 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 were 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! **Figure 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

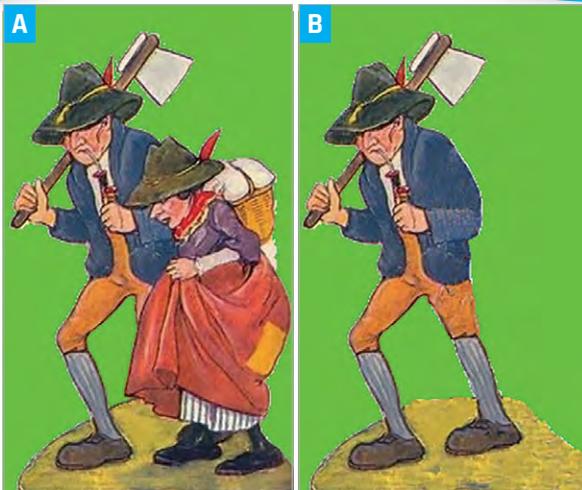


Figure 2 Removing the wife with the clone tool

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 than 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.

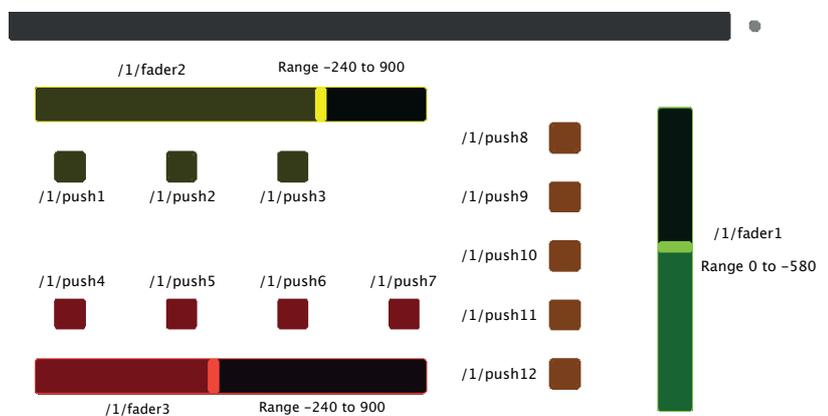
## Taking it further

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

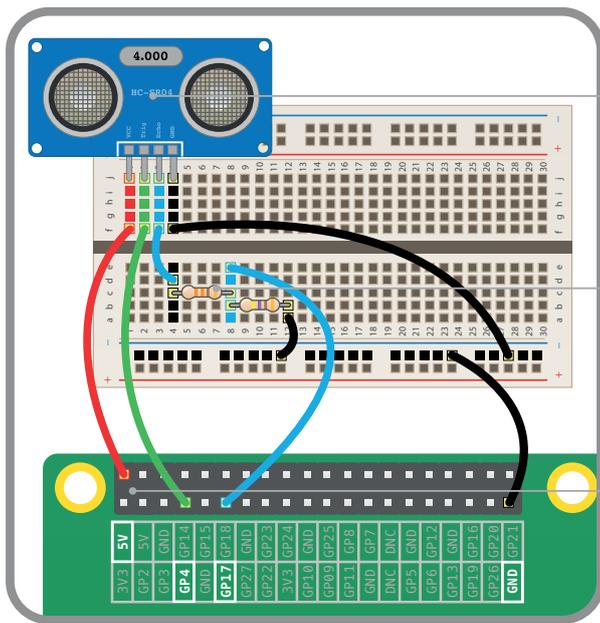
Finally, create labels for each element. The Wolf and Red Riding Hood should have a size 25 font. Save the layout, then open OSC on your tablet. Go to the layouts page and click on Add. Then click on the Pi, and click on the green Sync icon. You will see the Raspberry Pi appear on your tablet. Click 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.





**MARC SCOTT**

Marc used to run a Raspberry Pi and Minecraft club at his old school, where he taught Computer Science, and Systems and Control. He's now Head of Curriculum at the Raspberry Pi Foundation. [magpi.cc/2pOFrJB](http://magpi.cc/2pOFrJB)



HC-SR04 ultrasonic distance sensor detects the distance to an object, such as your hand

A voltage divider, comprising two resistors, is used to reduce the voltage of the Echo pin current to 3.3V

The sensor has four connections to the GPIO header on the Pi, including 5V power and GND

# THE RASPBERRY PI ULTRASONIC THEREMIN

## You'll Need

- > HC-SR04 ultrasonic distance sensor [magpi.cc/2q3cZ9J](http://magpi.cc/2q3cZ9J)
- > Breadboard [magpi.cc/2q34ZFz](http://magpi.cc/2q34ZFz)
- > Jumper wires [magpi.cc/2q2Kqcm](http://magpi.cc/2q2Kqcm)

Build your very own theremin musical instrument using an ultrasonic distance sensor and a little bit of Python and Sonic Pi code

**A** theremin is a unique musical instrument that produces sound without being touched.

In this tutorial, you will use an ultrasonic distance sensor to control the notes played by Sonic Pi.

An ultrasonic distance sensor has four pins: Gnd (ground), Trig (trigger), Echo (echo), and Vcc (power).

To use the sensor, you need to connect its Gnd pin to a GND (ground) pin on the Raspberry Pi, the Trig pin to a GPIO pin on the Pi, and the Vcc pin to the 5V pin on the Pi.

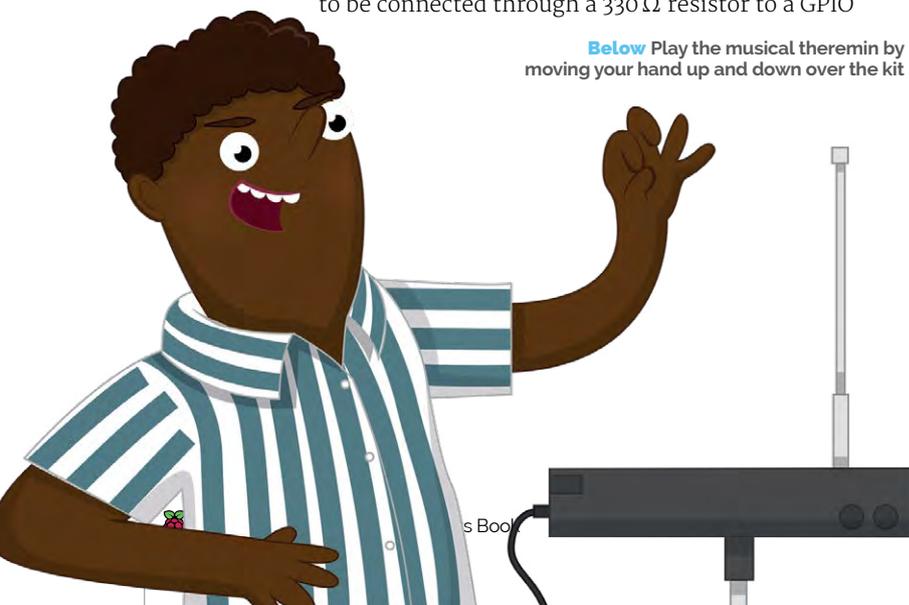
The Echo pin is a little more complicated. It needs to be connected through a 330Ω resistor to a GPIO

pin on the Raspberry Pi, and that pin needs to be grounded through a 470Ω resistor. The diagram above shows one suggested arrangement. If you've wired up the sensor as shown in the diagram, your echo pin is 17 and your trigger pin is 4.

Click on Menu > Programming > Python 3 (IDLE), to open a new Python shell. Click on New > New File. The code to detect distance is listed in **theremin1.py**. Type it into your new file, then save and run it.

The **sensor.distance** is the distance in metres between the object and the sensor. Run your code and move your hand backwards and forwards. You should see the distance changing, as it is printed in the shell.

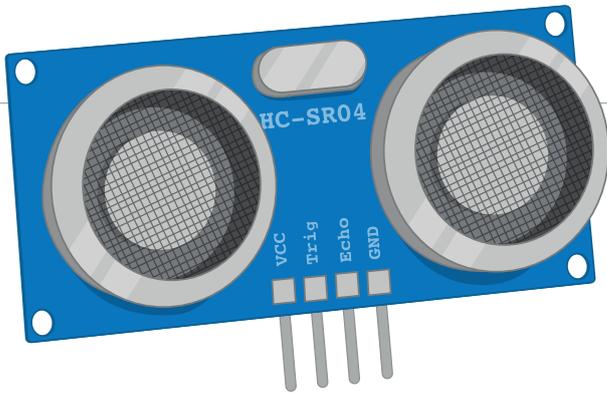
**Below** Play the musical theremin by moving your hand up and down over the kit



## Getting Sonic Pi ready

Sonic Pi will receive messages from your Python script. Open Sonic Pi by clicking on Menu > Programming > Sonic Pi. In the buffer that is open, you can begin by writing a **live\_loop**. This is a loop that runs forever, but can easily be updated, allowing you to experiment. You can add a line to reduce the time it takes for Sonic Pi and Python to talk.

```
live_loop :listen do
  set_sched_ahead_time! 0.1
end
```



**Above** The distance sensor sends out a high-frequency pulse and measures how long it takes the echo to reflect back.

Next, you can sync the live loop with the messages that will be coming from Python.

```
live_loop :listen do
  message = sync "/play_this"
end
```

The message that comes in will be a dictionary, containing the key `:args`. The value of this key will be a list, where the first item is the MIDI value of the note to be played.

```
live_loop :listen do
  message = sync "/play_this"
  note = message[:args][0]
end
```

Lastly, you need to play the note.

```
live_loop :listen do
  message = sync "/play_this"
  note = message[:args][0]
  play note
end
```

You can set this live loop to play straight away, by clicking on the Run button. You won't hear anything yet, as the loop is not receiving any messages.

## Sending notes from Python

To finish your program, you need to send note MIDI values to Sonic Pi from your Python file. You'll need to use the OSC library for this part.

```
from gpiozero import DistanceSensor
from time import sleep

from pythonosc import osc_message_builder
from pythonosc import udp_client

sensor = DistanceSensor(echo=17, trigger=4)

while True:
  print(sensor.distance)
  sleep(1)
```

Now you need to create a sender object that can send the message.

## theremin1.py

```
from gpiozero import DistanceSensor
from time import sleep

sensor = DistanceSensor(echo=17,
  trigger=4)

while True:
  print(sensor.distance)
  sleep(1)
```

## theremin.rb

```
live_loop :listen do
  message = sync "/play_this"
  note = message[:args][0]
  play note
end
```

## theremin.py

```
from gpiozero import DistanceSensor
from time import sleep

from pythonosc import osc_message_builder
from pythonosc import udp_client

sensor = DistanceSensor(echo=17, trigger=4)
sender = udp_client.SimpleUDPClient('127.0.0.1', 4559)

while True:
  pitch = round(sensor.distance * 100 + 30)
  sender.send_message('/play_this', pitch)
  sleep(0.1)

sensor = DistanceSensor(echo=17, trigger=4)
sender = udp_client.SimpleUDPClient('127.0.0.1', 4559)

while True:
  print(sensor.distance)
  sleep(1)
```

You need to convert the distance into a MIDI value. These should be integers (whole numbers), and hover around the value 60, which is middle C. Round the distance to an integer, multiply it by 100, and then add a little bit, so that the note is not too low in pitch.

```
while True:
  pitch = round(sensor.distance * 100 + 30)
  sleep(1)
```

To finish off, you need to send the pitch over to Sonic Pi and reduce the sleep time. The final code is listed in **theremin.py**. Save and run your code and see what happens.

Language

&gt;PYTHON &amp; RUBY

FILE NAMES:  
theremin.py  
theremin.rbDOWNLOAD:  
magpi.cc/  
2qj7qTN

**BILL BALLARD**

Bill is a retired physicist who went back to his programming roots and found fun things to do with his ten Pi boards, even while sailing.  
[github.com/wpballa](https://github.com/wpballa)

**You'll Need**

- ▶ Raspberry Pi Zero W with case and GPIO header
- ▶ Ultimate GPS breakout [magpi.cc/zqLDUpB](https://magpi.cc/zqLDUpB)
- ▶ Pimoroni Scroll pHAT [magpi.cc/zqMVMks](https://magpi.cc/zqMVMks)
- ▶ 40-pin stacking header for the Scroll pHAT
- ▶ USB to micro USB cable for power
- ▶ 12 V USB power adapter suitable for a car

# SAILING PI

Track and map your sailing adventure, wherever you go

**M** y weekly sailing crew asked for a way to visualise where we had been, and for a live display of the true speed over ground to see whether we were fighting tides. This project delivers both, without relying on internet connections or cell phone data. Add a battery and it would also be useful for car rallies or cyclists. All you need is a Raspberry Pi Zero W with some additional hardware, Python, and Mathematica!

## Assembly

Solder the dual male header to the Raspberry Pi Zero W, and solder the stacking header to the Scroll pHAT. Cut off the header wires except for pins 4, 6, 8, and 10 (see [pinout.xyz](https://pinout.xyz)). Double-check the pin numbering before you do this! Bend these four pins through 90 degrees so that they point outwards.

Take four wires of different colours, about 10 cm long, and solder them into the GPS breakout Vin, GND, TX, and RX connectors, bringing the wires in from the top of the board. Next, solder the Vin wire to pin 4 of the GPIO, GND wire to pin 6, RX wire to pin 8, and TX wire to pin 10. There should be no crossover wires.

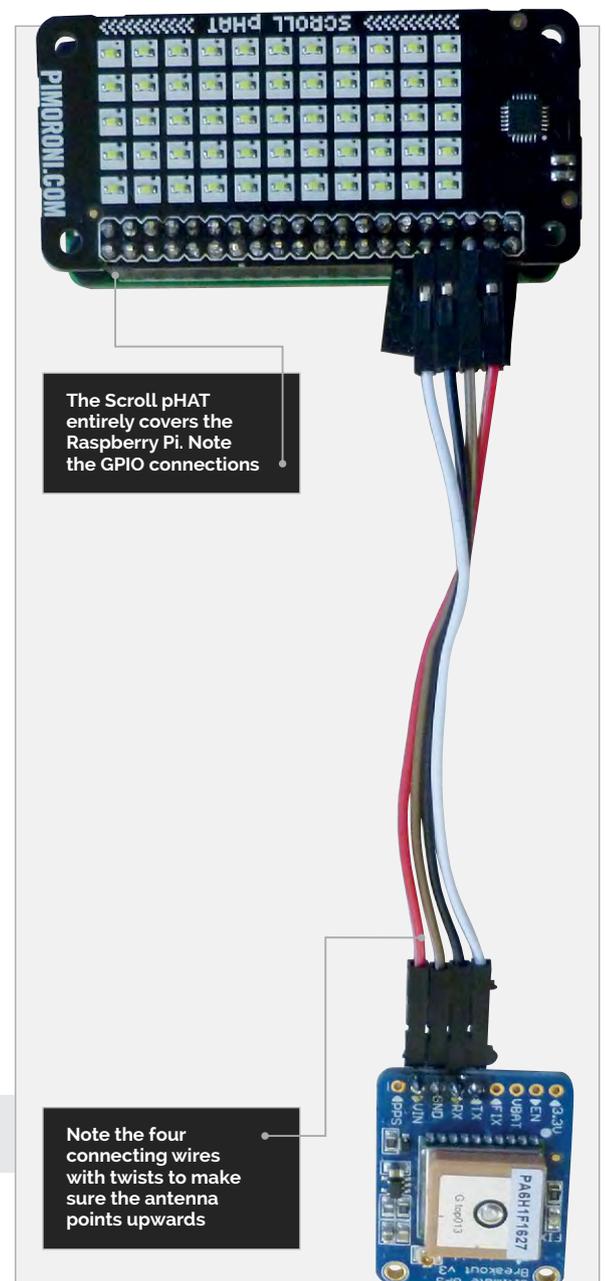
## Software setup

Boot up your Raspberry Pi Zero without the GPS attached. Go to Preferences > Raspberry Pi Configuration. In the Interfaces tab, be sure to enable SSH, Serial, and I<sup>2</sup>C. In the System tab, you will also ensure the Boot option is 'To CLI' (command-line interface). Next, set up the wireless and reboot.

Now load some special software for the Scroll pHAT and the GPS:

```
sudo apt-get install python-scrollphat gpsd
gpsd-clients python-gps
```

The GPS uses the TX/RX pins that are the console defaults, so you need to make some modifications to the system:

**SPEED UP GPS ACQUISITION**

Installing the optional battery on the GPS board will significantly speed up satellite acquisition when starting the system.

```
sudo nano /boot/cmdline.txt
```

Remove the `console=serial0,115200` portion of the line. Then save the file and exit the editor.

```
sudo systemctl stop serial-getty@AMA0.service
sudo systemctl disable serial-getty@AMA0.service
sudo halt
```

When the system stops, remove the power and install the scroll pHAT with the GPS breakout. Power the system up. The Fix light on the GPS board will blink once every second with no satellite fix, and every 15 seconds with a fix. If you don't have a fix, move the system to a window with a good view of the southern sky (northern sky in the southern hemisphere) where the antenna can see the satellites.

While you are acquiring a lock, we need to disable the standard gpsd socket. In a Terminal window, type:

```
sudo systemctl stop gpsd.socket
sudo systemctl disable gpsd.socket
```

## Program

Download the Python program `gpsd-boat.py` and Mathematica notebook `LatLonPlot.nb` from [magpi.cc/2qN6BmG](http://magpi.cc/2qN6BmG). The system will not have a live internet connection while you are sailing, and so it will not know the date straight away. As a result, the program waits for the GPS to get a fix and uses the UTC date information to generate file names with the current sailing date. Automate everything so that the program starts at boot time:

```
sudo nano /etc/rc.local
```

Add the following two lines to the file just before the `exit 0` line:

```
gpsd /dev/ttyS0 -F /var/run/gpsd.socket
python /home/pi/gpsd-boat.py > /home/pi/gpsd-boat.log 2> /home/pi/gpsd-boat.err &
```

Then save and exit. The last line will start the Python program as a background job, redirect output to a log file, and redirect any error messages to an error file for later debugging.

## We are sailing

Now go sailing, or perhaps driving. A 12V automotive USB plug works in a car or in the boat, in one of the many 12V outlets. Above deck these are likely to be corroded, so be prepared to clean the contacts (Scotch Brite scouring pads work well), and keep the Raspberry Pi and GPS in a plastic bag to ward off water. Power everything up and go for a sail. When you've finished sailing, unplug the system and take it home.



Language

>PYTHON &  
MATHEMATICA

FILE NAMES:

gpsd-boat.py  
LatLonPlot.nb

DOWNLOAD:

[magpi.cc/2qN6BmG](http://magpi.cc/2qN6BmG)

When you get home and have internet access again, connect the Raspberry Pi to a monitor and keyboard. The CSV output file is designed to be easily read with Mathematica. However, because we crashed the Pi to power it off on the boat, you will need to edit the file with nano and remove the last line or two, which will contain some garbage. If the application crashed and restarted at any point, there will be extra headers you should search for and delete. These files were written as root, so, where '2017-05-04' is replaced by your activity date, enter:

```
sudo nano 2017-05-04-latlon.csv
```

Remove the last few lines and any errant blank lines or additional headers. Then save the file and exit.

Start Mathematica and use the `LatLonPlot.nb` notebook you downloaded to visualise your sail, but replace the date with the date identifier for your file. The first line of the file imports the data from the comma-delimited file and loads the header and data separately. The output of this command should be a partial list of all the latitude and longitude pairs. The second line converts the latitude-longitude data into a `GeoPosition` set of variables, and then a `GeoPath` construct for plotting. The output of this should be a small graph of the path taken, but with no map. The third line places the `GeoPath` on an automatically sized map and places the output in the file `image.jpeg`. `PlotStyle` controls the colour and thickness of the sailing path plot.

Hit `SHIFT+ENTER` to force an evaluation and wait a while (the header will show running), particularly if it was a long sail. It takes quite some time for Mathematica to load the map data over the internet, so be patient. When the calculation is complete, open the `image.jpeg` file to have a look.

**Above** The unit installed on the boat's instrumentation console

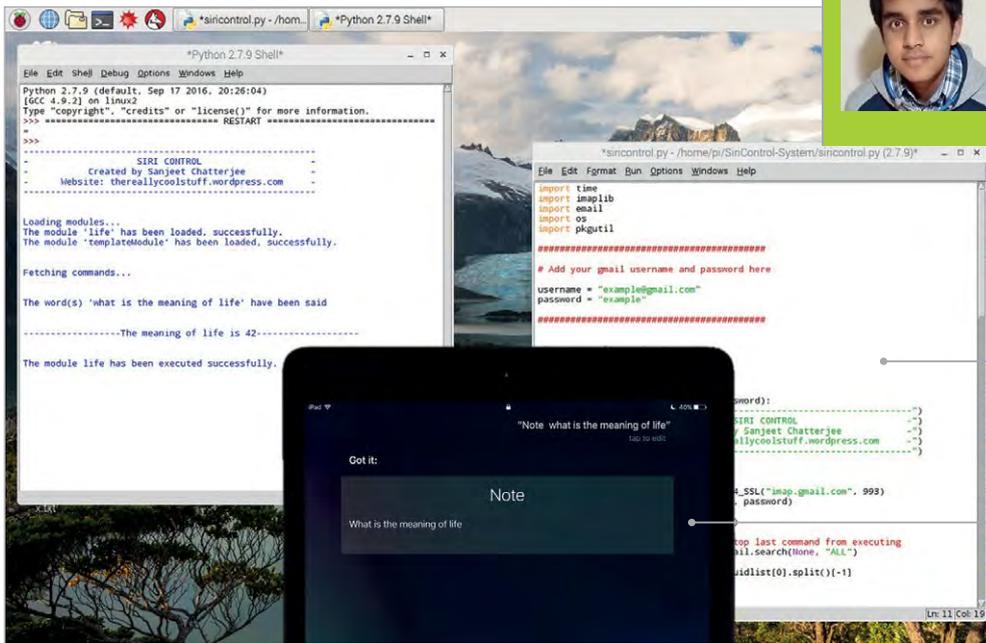
## MOUNTING THE UNIT

The final product is held in place with Velcro strips on the back of the Pi and on the boat console.



### SANJEET CHATTERJEE

16-year-old technology enthusiast and self-taught programmer with a knack for creativity. Passionate about studying robotics and AI.  
[magpi.cc/2t3oJtK](http://magpi.cc/2t3oJtK)



- SiriControl fetches the commands from your Gmail account, and runs the appropriate module
- Siri creates a note with your command, which syncs with your Gmail account

# SIRICONTROL: CONTROL YOUR PI WITH SIRI

### You'll Need

- > iOS device
- > SiriControl: [magpi.cc/2t3Bh4v](http://magpi.cc/2t3Bh4v)

SiriControl is a framework, written in Python, which implements an ingenious hack to enable you to add Siri voice control to any project

**S**iri is an intelligent personal assistant, integrated with Apple devices. From setting reminders to hailing taxis, Siri can do many things to make life easier. However, wouldn't it be awesome if you could control anything with Siri?

Welcome to SiriControl – a Python framework which provides a simple way of using Siri voice commands to add fantastic voice control to any project. The possibilities for SiriControl are endless, and as no extra hardware is required: you can get started right away!

### >STEP-01 How it works

Siri can create Notes by using the command word 'note'. The new Note is then synced with the linked Gmail account. SiriControl fetches the new Note from the Gmail account, and executes the appropriate function from the dynamically loaded modules created by the user. This ingenious hack enables you to add fantastic voice control capabilities to any project, with minimal setup required.

### >STEP-02 Configure Gmail

A Gmail account is required for SiriControl. For security reasons, we suggest creating a new Gmail account specifically for SiriControl, as you will have to enter your credentials in a Python script.

Access for less secure apps needs to be enabled for your new account as Google identifies the connection between the Python script and mail servers as less secure – see [magpi.cc/2u3gprx](http://magpi.cc/2u3gprx).

IMAP must also be enabled. This is found in the Gmail account settings, as shown in the image (on next page). This is the protocol SiriControl uses to fetch new Notes.



Above You will need to add your Gmail account so that the Notes can be synced with it

## >STEP-03

### Prepare your iOS device

The Notes need to be synced with your Gmail account so that SiriControl can fetch the voice commands that you say, through Siri. So navigate to Settings > Notes > Accounts > Add Account on your iOS device, and add your new Gmail account.

After turning on Notes, ensure that the default account for creating Notes is the new account. Now, if you say to Siri, “Note this is awesome,” it should appear under the Notes section, under your Gmail account.

## >STEP-04

### Set up SiriControl

Finally, you will need to clone the SiriControl repository using:

```
sudo apt-get update
```

```
sudo apt-get install git-core
```

```
git clone https://github.com/theraspberryguy/SiriControl-System
```

Edit **siricontrol.py** and enter your Gmail account credentials. This should be self-explanatory once the script is opened up.

Next, run **siricontrol.py** and say to Siri: “Note meaning of life.”

You should get the answer. That’s it! You have finished the SiriControl setup. Now, let’s add your own voice commands.

## >STEP-05

### Create your own modules

SiriControl uses a modular approach for adding your own commands. Each voice command, along with its action, is separated into different modules, found in the **modules** folder.

Every module must have the following:

**moduleName** – this is the name of the module, which can be anything you want.

**commandWords** – this array will contain the words which need to be spoken in order to call the **execute()** function.

**execute(command)** – this function is called when all the words in the **commandWords** array are spoken. The **command** parameter is the complete command spoken by the user.

Take a look at the **life.py** module, which comes with SiriControl, to gain a better understanding of how it works. Also, keep in mind that all modules you

## life.py

#You can import any required modules here

```
moduleName = "life"
```

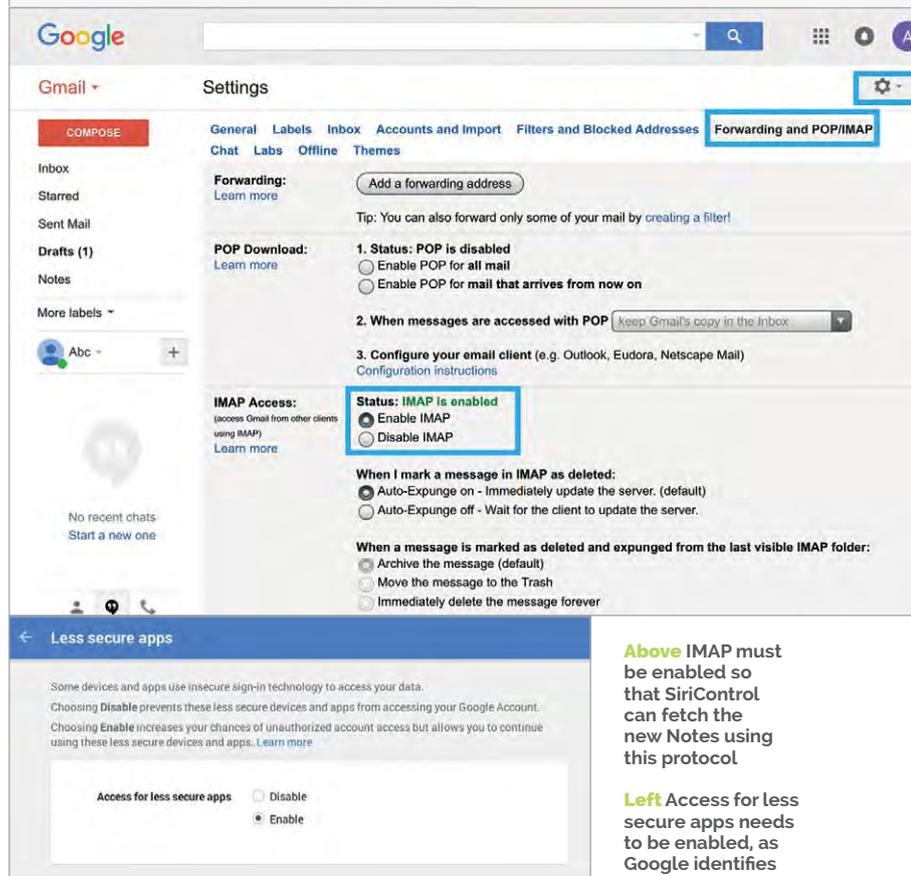
```
commandWords = ["meaning", "life"]
```

```
def execute(command):
    print("\n")
    print("-----The meaning of life is 42-----")
    print("\n")
```

Language

&gt;PYTHON

DOWNLOAD:  
magpi.cc/zt3BAwb



**Above** IMAP must be enabled so that SiriControl can fetch the new Notes using this protocol

**Left** Access for less secure apps needs to be enabled, as Google identifies this connection as being less secure

create must be stored in the **modules** directory for SiriControl to load when it starts up. The template **Module.py** is also included with SiriControl, which can be used as a reference.

## >STEP-06

### Next steps

Now you know how to create your own voice commands, what next? Well, whether it’s as simple as turning on an LED or controlling your TV using infrared signals – with the power of Siri behind it, the possibilities are endless! You could integrate SiriControl into any project. Due to the nature of the hack, you can control your Raspberry Pi from anywhere in the world, as long as you have an internet connection. This opens up many possibilities, including home automation and IoT. Anything is possible!





**K.G. ORPHANIDES**

K.G. has been tinkering with computers since the mid-'80s and writing about them since the late '90s. They've mostly survived the experience. @KGOOrphanides

**You'll Need**

- > microSD card
- > USB stick
- > Wired Xbox 360 controller
- > Amiga Kickstart ROMs [amigaforever.com](http://amigaforever.com)
- > Amibian [magpi.cc/rcFTIK](http://magpi.cc/rcFTIK)

# TURN YOUR PI INTO AN AMIGA

Recapture the glory days of 16-bit computing by turning your Pi into a faithful Amiga emulator

**T**he Commodore Amiga's top-notch sound and graphics made it one of the most desirable home computers of the 1980s and early 1990s, at a time when your average IBM PC was still plodding along with EGA graphics and an internal beeper. Amiga games from the era have aged incredibly well, and look and play brilliantly on everything from a portable display to a widescreen TV. We'll take you through turning your Raspberry Pi 3 into a perfect modern-day Amiga emulator. You'll need a Windows, macOS, or Linux desktop operating system to copy the Amibian Linux distribution to your microSD card and unpack the Kickstart ROMs required to make it work smoothly.

Start by downloading the Amibian distro. Format a microSD card, decompress the Amibian RAR file, and use Win32DiskImager or Linux's `dd` command to copy the IMG file to the card. A 4GB card should be plenty, as Amibian only occupies around 300MB.

Slot the microSD card into your Pi and power up. It'll boot directly into the UAE4ARM emulator, but there's some extra configuration to do before we start playing. Quit UAE4ARM to get to the command line and run:

```
raspi-config
```

Select **Expand Filesystem**, which will give you access to the entirety of the SD card's capacity for storage, then **Exit** and select **Yes** to reboot.

If your Pi won't output sound via HDMI properly, enter this at the command line:

```
nano /boot/config.txt
```

Make sure the following lines are present and aren't commented out with a preceding hash (`#`):

```
hdm1_drive=1
hdm1_force_hotplug=1
hdm1_force_edid_audio=1
```

Save and return to `raspi-config`:

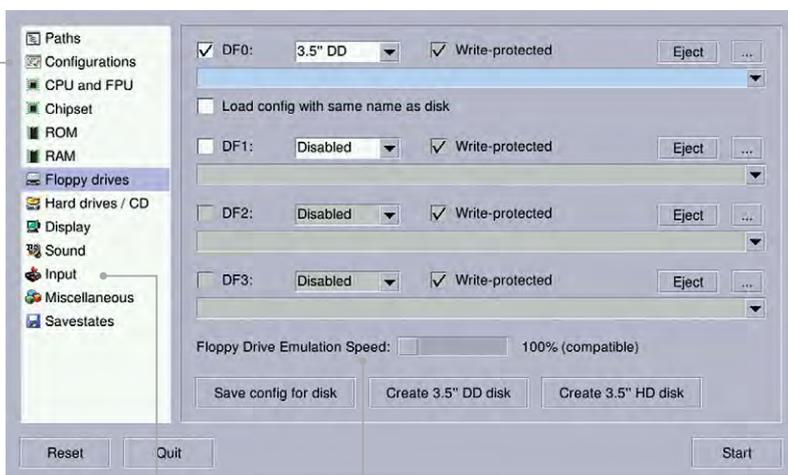
```
raspi-config
```

Select **Advanced Options > Audio > Force HDMI** and then reboot.

## Kickstart me up

To run Amiga programs, you'll need a Kickstart ROM – firmware from the original computers. UAE4ARM comes with the open-source AROS ROM, which can run only some Amiga programs, so we recommend using genuine Amiga Kickstarts for reliability.

The Amiga's Kickstart ROMs and Workbench GUI are still being maintained, thanks to Italian firm Cloanto. Amiga Forever Plus Edition, priced at €29.95, gets you a complete, legal set of Kickstarts for every Amiga computer and console. Cloanto is still working



You can load and create emulated hardware configurations for specific Amiga computers

Game controllers, mouse, and keyboard configurations can be selected and tweaked in Input

Once you've set up your emulated hardware and firmware config, just mount a floppy disk image and click Start


**PUBLISHER-APPROVED  
GAME DOWNLOADS**

Amigaland  
amigaland.de

Ami Sector One  
magpi.cc/2dDLELL

Dream17  
dream17.abime.net

Games Coffe  
gamescoffer.co.uk

Gremlin Graphics World  
magpi.cc/2dDKZ3S

on a Raspberry Pi edition, so you'll currently have to install Amiga Forever on a Windows PC or Wine and copy the files onto a USB stick.

There are other ways of obtaining Kickstart ROMs, but most are legal grey areas. You can extract them from an Amiga using a tool such as TransRom or find them on abandonware sites, but we strongly recommend supporting Cloanto's continued development of Amiga Forever.

Classic Amiga software is even easier to find. You'll get 50 games with Amiga Forever Plus, while some major publishers have made the Amiga versions of their games available for free (see above).

Many more games are only available online as legally dubious abandonware. They're easily found using any search engine, but inform yourself of the legal status of such software in your region before you download.

## One true path

As Amibian doesn't include a window manager, it's easiest to download and copy everything to a USB stick using your operating system of choice. Helpfully, UAE4ARM can read Amiga ADF floppy images even if they're in a zip file.

We recommend copying everything to your microSD card. Fire up your Pi, exit UAE4ARM, and run:

```
mc
```

Copy your game files from `/media/usb` to `/root/amiga/floppys`, and your Kickstart ROMs, including a Cloanto rom.key file if you have one, to `/root/amiga/kickstarts`. Quit Midnight Commander and reboot:

```
reboot
```

In the latest version 1.313 of Amibian, two different versions of UAE4ARM are supplied. If you plan on using two Xbox 360 controllers, button mapping on controller two works best using the 'old' version, although the 'new' edition generally provides more options. To switch between the two, at the command line type either `rpioold` or `rpinew`. The following configuration instructions work with both versions.

## Configure UAE4ARM

First, go to the Paths tab and click Rescan ROMs so UAE4ARM knows where to find everything.

The Configurations tab lets you select from several preset hardware emulations, with the default being an A1200 – just select and Load your chosen computer. You can tweak your virtual hardware in the CPU and FPU, Chipset, and RAM tabs.

Your configuration selection doesn't always set the relevant Kickstart ROM for you, so check the ROM tab, where you can choose Kickstarts from a pull-down menu. Note that many games require a specific ROM or hardware configuration to work properly, depending on which system they were originally released for.

To run most software, you'll need the Floppy drives tab. Just press the ... icon next to drive DFO's Eject button, select the desired disk image, and click Start.

By default only drive DFO is active, and most titles expect this configuration. To swap disks when prompted, press **F12**, eject the disk image in DFO, select the disk image you're asked for, and click Resume.

**F12** will always pause and return you to UAE4ARM's main interface, so you can create a save state – a stored image of your progress in a game – or give up and load something new. The Reset, Quit, and Start/Resume buttons are always visible in UAE4ARM's GUI. Reset completely reboots your emulation and Resume returns you to your current game.

UAE4ARM automatically detects Xbox controllers. You can use two controllers for multiplayer gaming – if the second is unresponsive, you may need to press **F11** to disable your mouse and switch control to the pad. If you're running the 'new' version of the emulator, first select your controllers from the pull-down Porto and Port1 menus in the Input settings.

Now you've got your Amiga emulator up and running, there's plenty of scope to build on the project, from setting up virtual hard disks to install Workbench and other software onto, to creating floppy images from your own original Amiga disks and using the Pi's GPIO to connect a classic 1980s joystick.

# RETRO GAMING BLISS

Build upon Adafruit's amazing **Pi GRRL 2** using the brand new **Pi Zero W** to create the ultimate retro handheld console...

**P**ortable gaming has been hugely popular ever since the Nintendo Game Boy was released in the late eighties. Building on the work of the Game & Watch LCD games, the Game Boy allowed you to take one machine wherever you went. It played a multitude of games via handy cartridges filled with code, including video game classics such as Tetris, The Legend of Zelda: Link's Awakening, and Pokémon.

Since then, handheld gaming and computers have evolved. Mobile phones have become a great source for providing quick hits, while Nintendo is going all out with its hybrid Switch system. On the computing side, processing power has advanced to such a degree that

the Raspberry Pi is powerful enough to emulate several popular retro home consoles, while also being small enough to carry around.

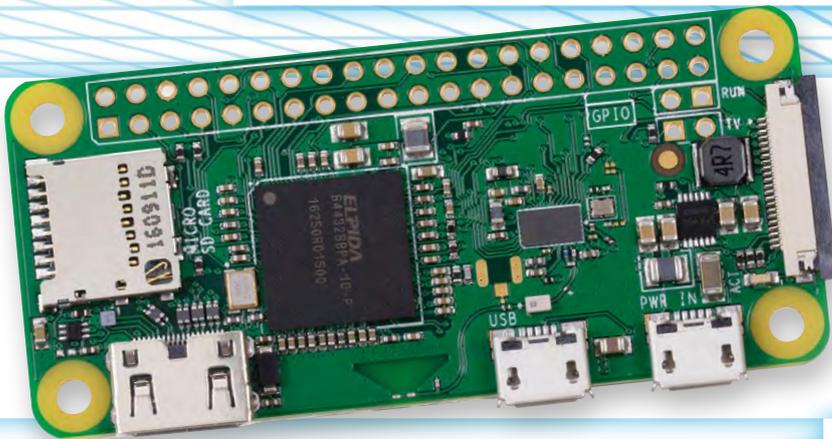
This is why the PiGRRL projects from Adafruit are popular: handheld, old-school consoles you can use on the go. There are many versions of them, based on everything from original models of the Raspberry Pi to the Pi Zero.

With the release of the Pi Zero W, these projects can go even further thanks to the built-in wireless LAN, and it also leaves more space for a bigger battery. And a bigger battery means longer play time. In this guide, we're going to show you how to take the PiGRRL 2 and do just this. Grab your work dungarees and let's-a go!



# ASSEMBLE YOUR PARTS

Here's what you'll need to make the **Pi GRRL Zero W**



## Pi Zero W

> [magpi.cc/2l6zurq](https://magpi.cc/2l6zurq) <

The key to this project is the Pi Zero W. The wireless-enabled version of the Pi Zero saves a load of space thanks to the radio chip included in it, so no WiFi dongle is needed. It runs at 1GHz, which makes it powerful enough to run emulators of many eighties and nineties consoles.



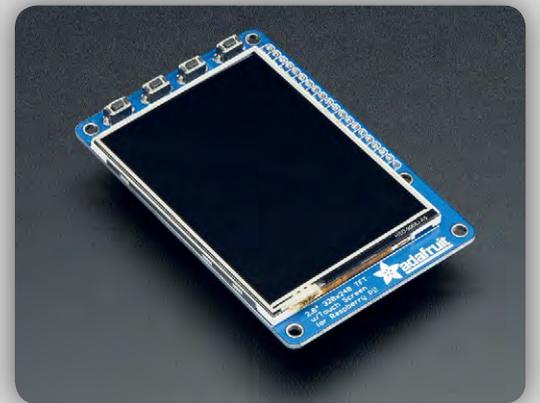
## TOOLS FOR THE JOB

- Soldering iron
- Wire
- Heat shrink
- Glue
- Blu Tack
- Wire strippers
- Hobby knife

## 3D-printed case

> [magpi.cc/2kSgK1f](https://magpi.cc/2kSgK1f) <

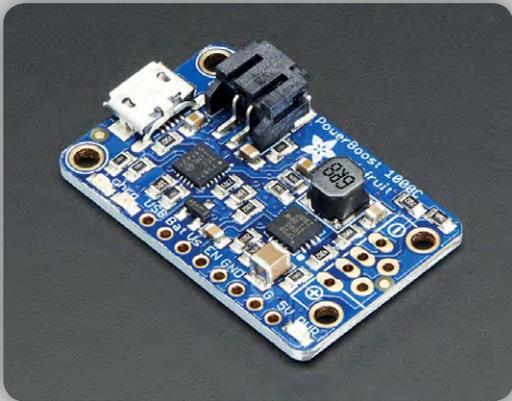
The PiGRRL 2 case repurposed for our needs. It has some spaces for USB and Ethernet on a B+/2/3 board, but the Pi Zero will require extensions if you want to make use of the gaps.



## Adafruit PiTFT 2.8"

> [magpi.cc/2lob5Ky](https://magpi.cc/2lob5Ky) <

This is actually a touchscreen, although we won't be making use of it in that way. It fits neatly in the case and provides four extra buttons to use when playing games. You can also assign system and UI shortcuts to the buttons.



## PowerBoost 1000C

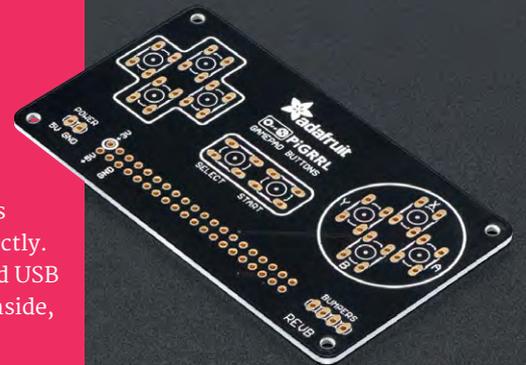
> [magpi.cc/2lo5aFg](http://magpi.cc/2lo5aFg) <

This is one of the cool bits: we're going to use this PowerBoost to actually charge a battery within the handheld. With the low power draw of the Pi and advancements in modern battery tech, you'll get a lot out of one charge.

## PIGRRL 2 controller board

> [magpi.cc/2lohZzr](http://magpi.cc/2lohZzr) <

This Custom Gamepad PCB is designed to fit the case perfectly. We did try to see if a standard USB SNES or NES PCB would fit inside, but they were far too big.



## Slide switch

> [magpi.cc/2lojfTb](http://magpi.cc/2lojfTb) <

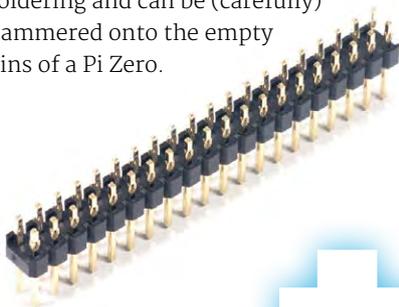
This switch allows you to turn the power on and off. It's best to do the software shutdown of RetroPie first before switching the power off, though.



## GPIO hammer headers

> [magpi.cc/2lohN2U](http://magpi.cc/2lohN2U) <

A wonderful innovation from Pimoroni, these GPIO headers require zero soldering and can be (carefully) hammered onto the empty pins of a Pi Zero.



## Microswitches

> [magpi.cc/2lo8goi](http://magpi.cc/2lo8goi) <

The beauty of proper retro gaming is tactile controls. You'll need ten 6mm switches, and a couple of 12 mm buttons.



## 2000mAh battery

> [magpi.cc/2lQzVmr](http://magpi.cc/2lQzVmr) <

We squeezed the biggest battery into here that we possibly could. This way, it should last for hours and hours.

## Bluetooth audio (optional)

There's no dedicated Audio Out on the Pi Zero W, but it does have Bluetooth. RetroPie doesn't support Bluetooth audio yet, but there are workarounds: see [magpi.cc/IyxKkA](http://magpi.cc/IyxKkA).



## SCREWS

A selection of screws to attach the parts to the case. This includes 14x #4-40 and 6x #2-56 3/8 machine screws.

# PRINT THE CASE

What you need to know about 3D-printing the **Pi GRRL 2** case

**T**he proliferation and advancement of 3D printing has been a huge boon for the maker community, enabling you to create wonderful chassis and cases for your final products. The PiGRRL series has a number of cases built around the Raspberry Pi that allow for maximum

efficiency in size, while also allowing for a fully operational handheld.

For this project, we're going to make use of the slightly larger PiGRRL 2 case for maximum comfort, and also so we can use the extra space to install a bigger battery into it. Here's how to make your own.

## HOW TO PRINT YOUR 3D CASE

### >STEP-01 GET THE FILES

The full PiGRRL 2 case files can be downloaded from [magpi.cc/2kS9K1f](http://magpi.cc/2kS9K1f), although there are more files here than what you actually need to print. The ones you'll need from the pack are:

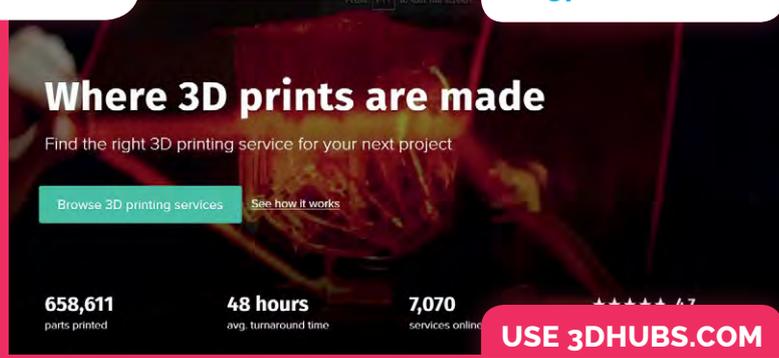
- pigrrl2-top.stl
- pigrrl2-bot.stl
- pitft-buttons.stl
- dpad2.stl
- action-btns.stl
- pause-start.stl
- shoulder-btns.stl
- shoulder-mount.stl



GET THE FILES AT  
[magpi.cc/2kS9K1f](http://magpi.cc/2kS9K1f)

### >STEP-02 FIND A 3D PRINTER

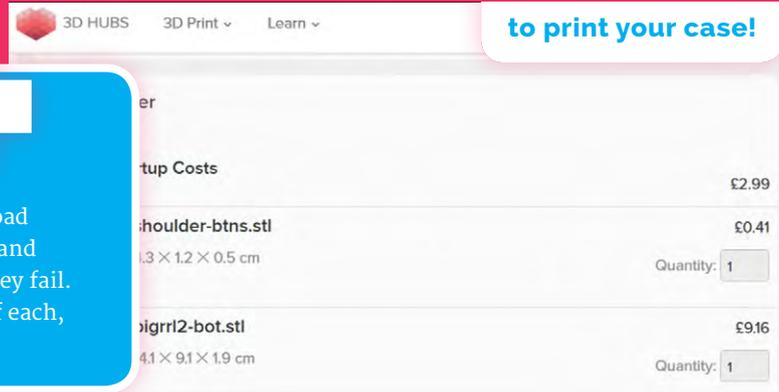
It can be tricky to find a good 3D printing service online, so unless you have access to a 3D printer, we highly recommend using [3DHubs.com](http://3DHubs.com). It lists local 3D printing services, along with an estimated completion time and reviews. The files we downloaded also work with the service.



USE 3DHUBS.COM  
to print your case!

### >STEP-03 UPLOAD FILES

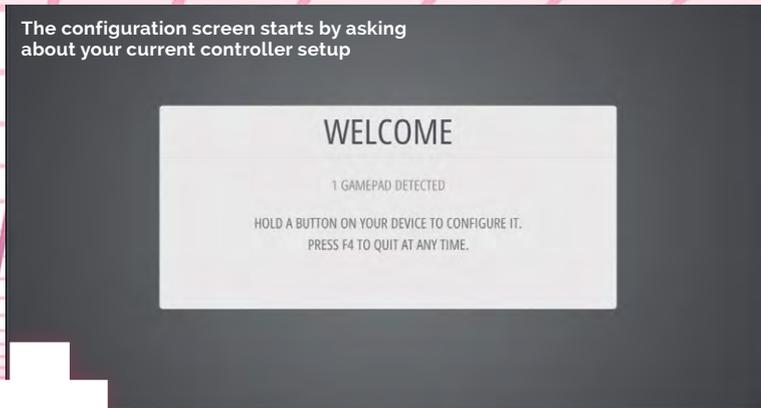
Once you've selected your printer, you'll be asked to upload the files. Double-check you've selected the correct ones and make sure they upload properly – you'll get an error if they fail. Usually, trying again will work. You also only need one of each, and ABS or PLA are great materials to use for the parts.



# PREPARING YOUR PI

Get your **Pi Zero W** ready to be made into a retro gaming treasure

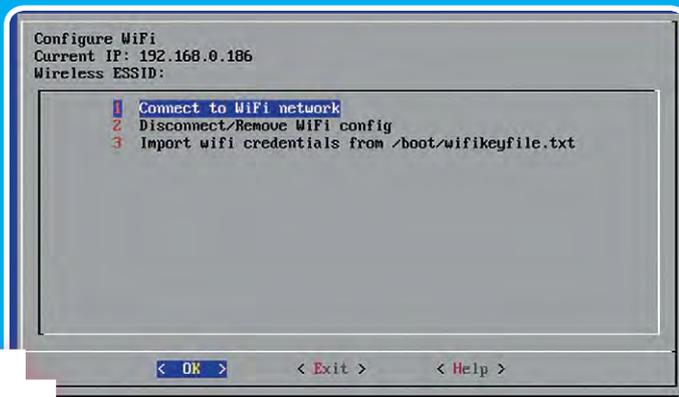
The configuration screen starts by asking about your current controller setup



When you come back to reconfigure your buttons, refer to this handy guide

## SETTING UP WIRELESS

Wireless connectivity on the new Pi Zero W is great, but the configuration method for wireless LAN on the RetroPie is very basic. You'll need to know the name of your wireless network (SSID) along with the password, as it won't be able to search for available networks. You can also import details by loading a .txt file onto the boot partition.



**F**or this project, we're going to be using the excellent RetroPie to power our emulation software. You'll first need to download the image for RetroPie from its website here: [magpi.cc/25UDXzh](https://magpi.cc/25UDXzh). Write it to a microSD card and pop it into your Pi Zero W. Connect that to a monitor along with a keyboard, and we can get it ready.

You'll first need to map some buttons – as the PCB controller isn't hooked up yet, we'll have to quickly use the keyboard for the initial setup. Make sure the directional keys, Start, Select, A, and B are assigned a key and just hold down the SPACE bar to skip anything else. Once that's done, connect to wireless using the info in the 'Setting up wireless' boxout.

To get our final build working, we need to make sure to install support for the PiTFT, as it's not supported natively. SSH into the Pi Zero W at [retropie.local](http://retropie.local) or press F4 to enter the command line on RetroPie, and enter the following:

```
cd
curl -O https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/pitft-fbcp.sh
sudo bash pitft-fbcp.sh
```

Select PiGRRL 2 and don't reboot. Now we need to add support for the custom buttons. Back in the command line, use:

```
cd
curl -O https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/retrogame.sh
sudo bash retrogame.sh
```

Select PiGRRL 2 again and then reboot the system. Once we've put all the parts together, and before it's assembled in the case, we'll need to configure the controls for the buttons we've made and added to the project. Press the button assigned to Start and select Configure Input, and then go through the configuration process again.

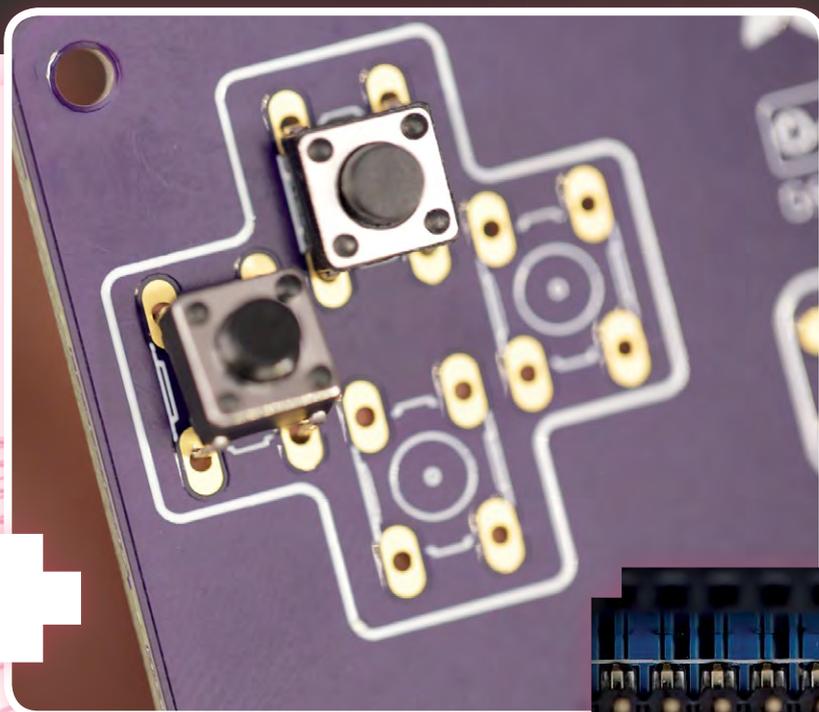
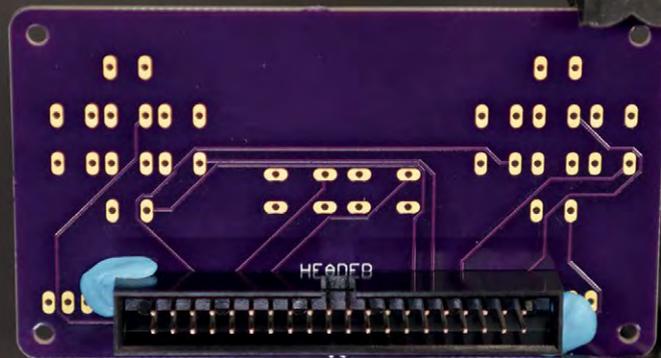
# BUILD THE SYSTEM

Follow along and build your **retro handheld**

## >STEP-01

### PREPARE THE GAMEPAD BOARD

Our first job is to solder the header pins onto the Gamepad board. You can keep it all in place with a little Blu Tack before soldering it on. Make sure you're soldering the header onto the correct side of the board.



## >STEP-02

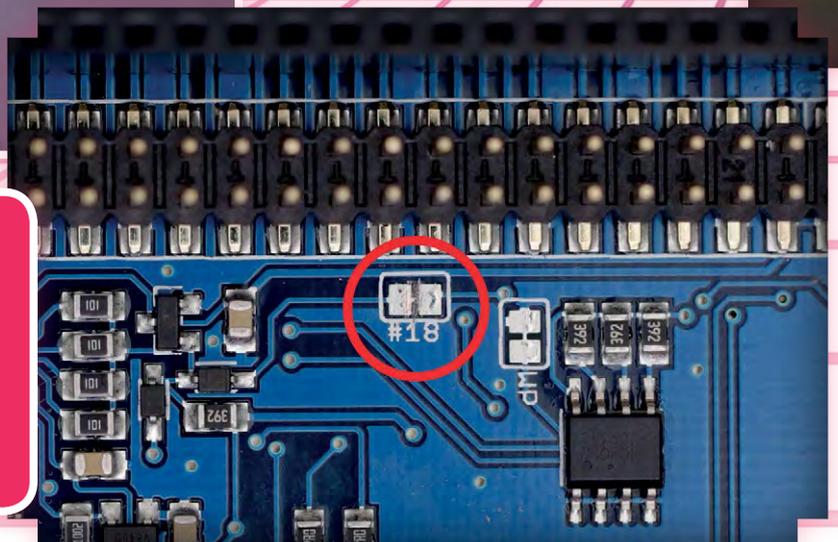
### ADD THE BUTTONS

Now it's time to carefully solder the ten 6 mm microswitches to the front of the board. Some helping hands would be good here.

## >STEP-03

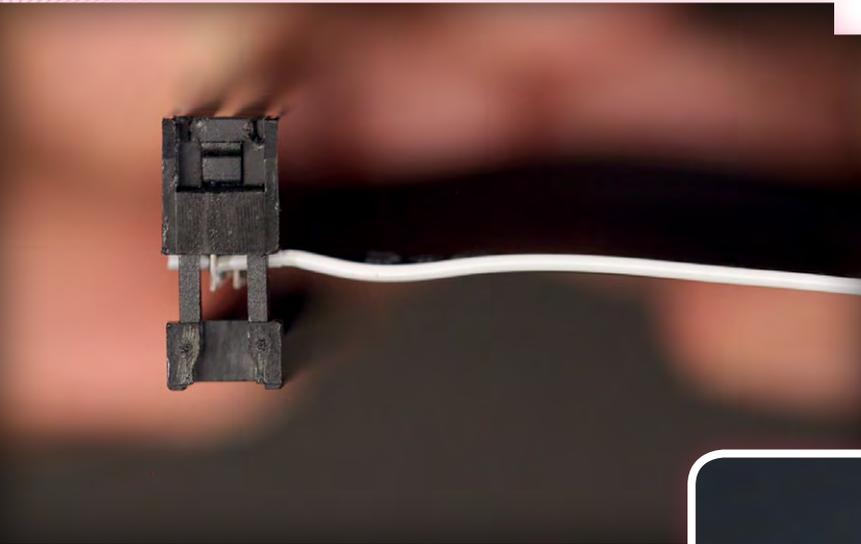
### TURN ON THE BACKLIGHT

By default, the PiTFT doesn't have its backlight turned on. You need to take a craft knife and cut the circuit between the blocks in the #18 box that you can see circled in the picture.



>STEP-04  
**RESIZE  
THE  
RIBBON  
CABLE**

It's a good idea to shorten the ribbon cable. 108 mm is apparently the perfect size, but you can go a little longer. Once you've measured it, cut the cable.

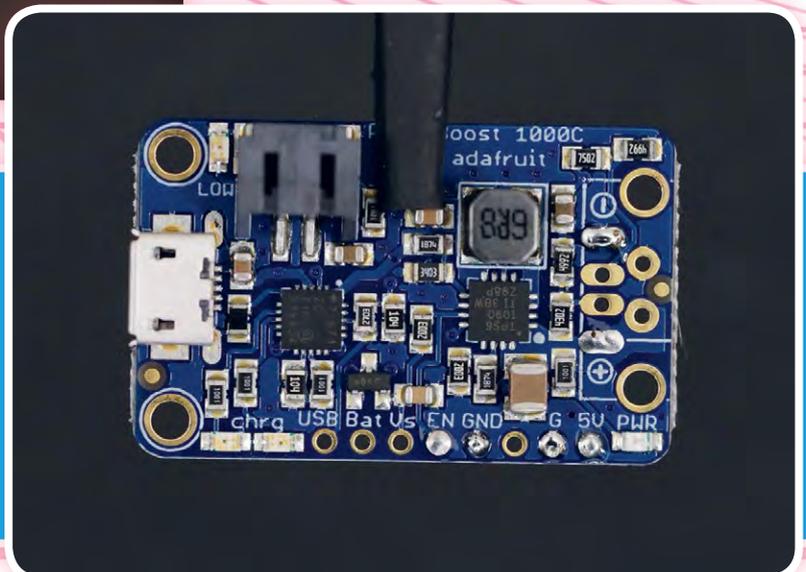


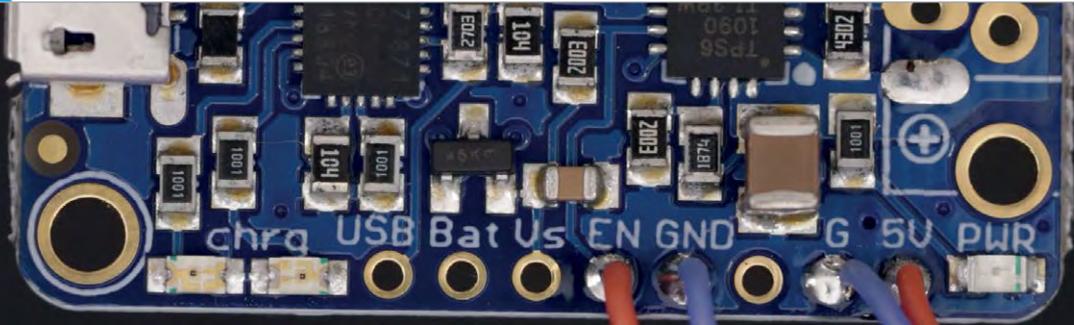
>STEP-05  
**ASSEMBLE  
THE CABLE**

Using something like a pen or pin, you'll need to push in the clip that holds the connector in place on the part you're discarding. Very carefully remove the cable and install it at the end of your newly trimmed cable.

>STEP-06  
**PREPARE  
FOR POWER**

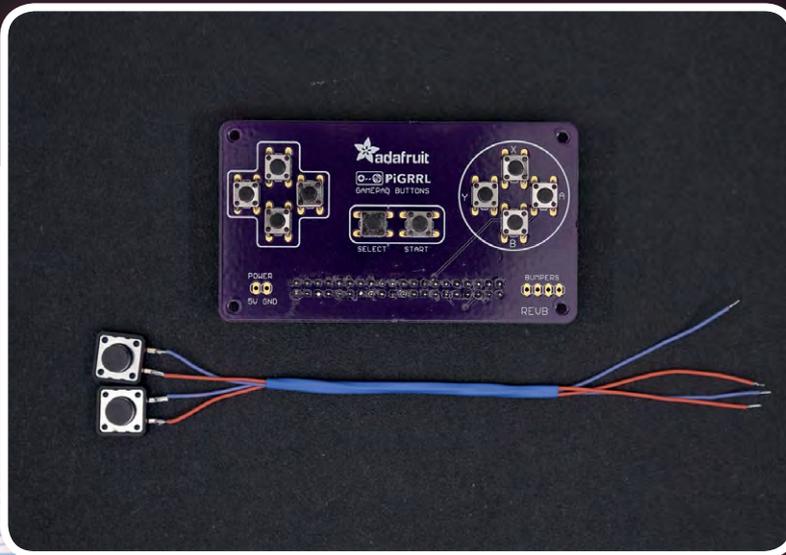
To make all our soldering easier, we'll dab some solder onto the spots where we need it for the moment. On the PowerBoost 1000C, add some solder to the positive and negative pins, and the EN and GND pins. Cut one of the legs off the power switch and put some solder on the other two.





## >STEP-07 SOLDER ON THE SWITCH

Make sure the power switch will fit in the hole for it in the case – it’s on the side of the bottom part. You may need to file away the plastic a bit. Once that’s done, trim two short bits of wire to about 7 cm long and solder one to each leg. Solder those to EN and GND – it doesn’t matter which way around they go.

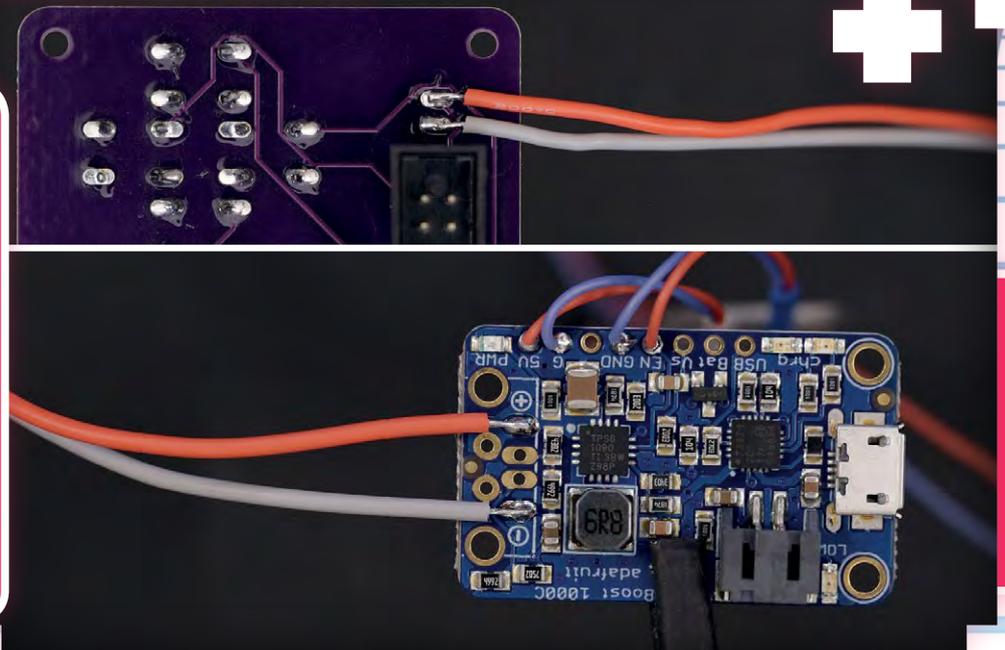


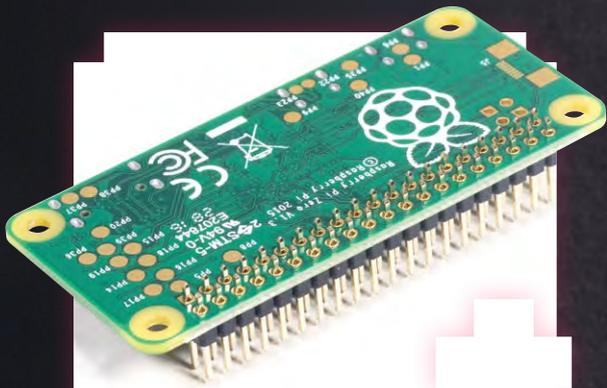
## >STEP-08 SHOULDER BUTTONS

The shoulder buttons (the 12 mm ones) need to be attached to the controller board much like the switch was connected to the power board. Clip two legs off each and use a pair of pliers to flatten the remaining two. Solder wires to each pin (about 14cm long) and then solder the other end to the bumper pins on the controller board. Again, polarity doesn’t matter, but keep each one as a pair in the row.

## >STEP-09 WIRE IT ALL UP

Now we can combine the power with the controller board, which will allow us to provide power to the whole system. Solder two wires (about 14 cm long) to the underside of the 5V and GND pins on the controller board. The 5V wire should then be soldered to the positive of the PowerBoost board, with the GND to the negative.





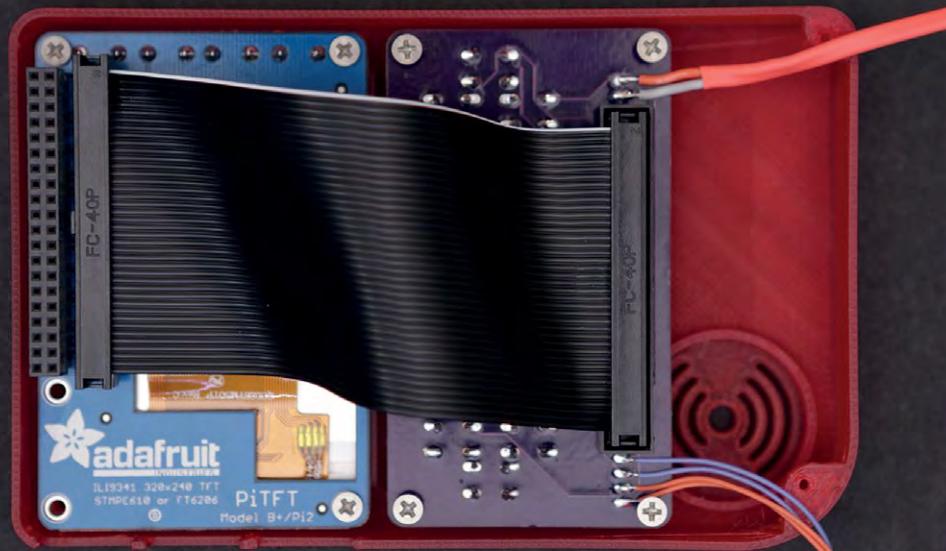
>STEP-10

## HAMMER THE HEADS

It's now a good idea to add the GPIO headers to the Pi Zero W. Gently hammer them in until they're secure, and you're done.

TEST IT ALL OUT!

At this point you can test the entire system, otherwise we'll move onto final construction...



>STEP-11

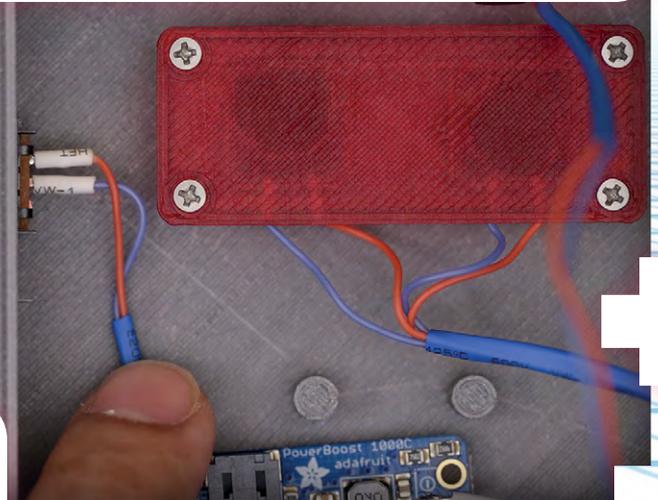
## BEGIN CONSTRUCTION!

Now the easy bit. Take the top of the case and insert all the button 3D prints. Once that's done, insert the screen and screw it in, followed by the controller board. Finish up the top bit by connecting the two with the ribbon cable, and then insert the Pi Zero into the header on the PiTFT.

>STEP-12

## BACK PLATE

You'll now need to mount the back buttons and the PowerBoost. The rear buttons have a plate that keeps them in place, and Adafruit suggests using a little Blu Tack to stick the switches in position. You'll attach the power switch and then finally screw on the PowerBoost. Be careful with the wires that you've soldered on.



>STEP-13

## CLOSE IT UP

Now you can finally close it up! Insert the battery and tape it down if need be, before screwing it shut. Make sure all the cables are safely inside the case before tightening it, though!

### WANT TO KNOW MORE?

Prefer to use a full-sized Pi? Check out the original PiGRRL 2 guide [magpi.cc/2loALzQ](http://magpi.cc/2loALzQ)

# USING A WIRELESS PIGRRL

Here's some of the amazing advantages of having the **PiGRRL** powered by **Pi Zero W**

## CONNECT VIA SSH

Physically connecting to the Pi Zero inside is a massive hassle once the case is screwed together. With the Pi Zero W connected to your home network, though, it's easy to connect to it remotely from another computer using SSH.

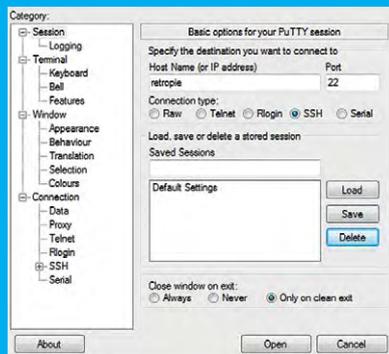
On a Mac or Linux machine (which includes another Raspberry Pi!), you can simply open the terminal or command line and enter the following to connect:

```
ssh pi@retropie
```

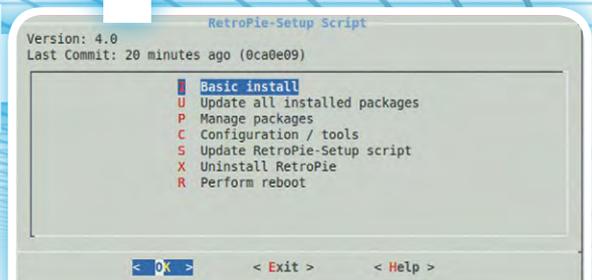
It will prompt you for a password, which is **raspberry**. The user name in this instance is **pi**, with **retropie** being the default name for the system on the network.

Windows 10 has an SSH client built in; for older Windows machines, you'll need to use PuTTY ([magpi.cc/2lBHCRm](http://magpi.cc/2lBHCRm)) to connect. Once it's installed, you need to set the host name to **retropie**, the port to 22, and then click Open. You'll need to put in **pi** as the user name and **raspberry** as the password.

Once inside, you can control many aspects of the system via the command line. If you've used the terminal in Raspbian, you'll know how it works: `sudo reboot`, `ls`, `cd`, etc.



You can ignore a lot of the PuTTY interface when just connecting to the handheld



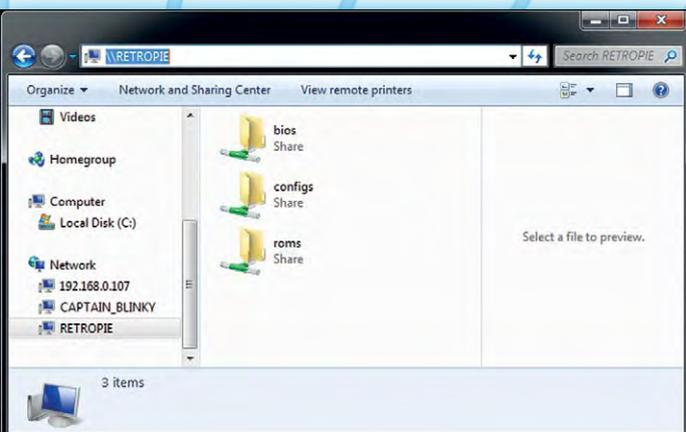
## UPDATING RETROPIE

You can update RetroPie from the handheld itself or via SSH in the command line. The keyboard might be better suited for this, so if you're by your computer it wouldn't hurt to use it instead.

From the handheld, you need to go to the RetroPie menu in EmulationStation and activate the setup script. From the terminal (if you've SSHed in), you can use the following command:

```
sudo ~/RetroPie-Setup/retropie_setup.sh
```

From here, look for the 'Update All Installed Packages' option. There are many options here that you can select from, including managing the individual packages in case you want to remove or add any. To update, you can select the option 'Update all installed packages' (which will also update the **RetroPie-Setup** script as well) or you can go to 'Manage packages' and update the packages individually. This could be useful if any packages have some problems updating, or if you want to do the essential updates before running out of the house.



## UPLOADING FILES

If you want to upload ROMs to the handheld, you can do so with the Pi Zero W's wireless connection. Otherwise, you'd have to manually load them onto the microSD card, which would require dismantling the console to get to the Pi Zero – not particularly easy or practical to do. Luckily, RetroPie includes Samba and SFTP, which allow you to transfer the files over the network.

For Samba it's nice and easy: when your handheld is connected to your network, you can go to your main computer and find it on the network shares as \\RETROPIE. Here you can upload any necessary extra files to the handheld with minimal hassle.

Windows 10 has an SFTP client built in; for older Windows machines, the RetroPie team recommend WinSCP ([magpi.cc/2lCwRhZ](http://magpi.cc/2lCwRhZ)); for Mac, you can try Cyberduck ([magpi.cc/2lCwjs9](http://magpi.cc/2lCwjs9)).

Once booted up, you can use the same SSH settings as we used for PuTTY. You can then drop the files into the corresponding folder in the **roms** directory.

## PLAYING ONLINE

Otherwise known as netplay in emulation circles, this allows you to play multiplayer games with friends, even if they're on the other side of the world! Let's see the SNES do that! Not every emulator supports it and if it does, you need to follow three extra rules: both parties need to be running the same version of RetroArch, both must be running the same emulator, and both need to be running the same ROM.

You then need to configure netplay from the setup script. If you're hosting, change your Netplay Mode to host, make the host IP your IP address, and pick a nickname. The client (other player) needs to go to the same menu, change Netplay Mode to client, change the host IP to the other handheld, and pick your nickname. You may need to open up a specific TCP/UDP port on the host's router, which you then need to set as the same on both systems.

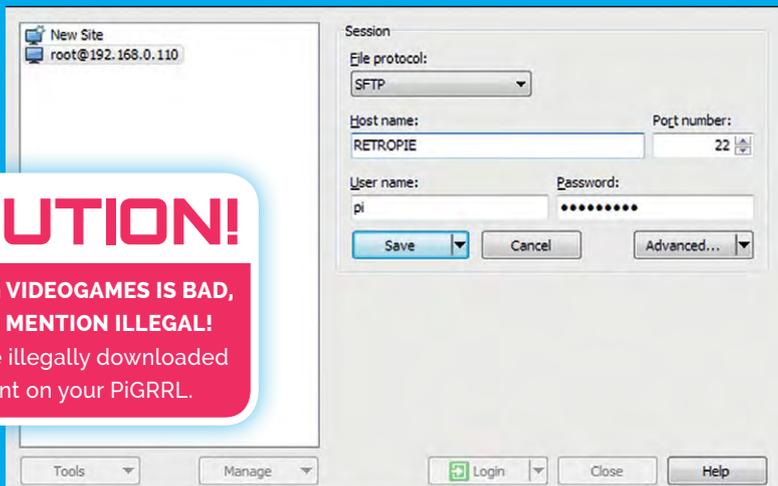
Now both of you need to open the same ROM using the 'jso' key (which should be X on a SNES layout) and select 'Launch with netplay enabled'. If it's been set up correctly, you'll connect!

```
System: snes
Emulator: lr-snes9x-next
Video Mode:
ROM: Mask, The (U)

1 Select default emulator for snes (lr-snes9x-next)
2 Select emulator for rom ()
3 Select default video mode for lr-snes9x-next ()
4 Select video mode for lr-snes9x-next + rom ()
5 Select RetroArch render res for lr-snes9x-next (640x480)
6 Edit custom RetroArch config for this rom
7 Launch
8 Exit (without launching)
9 Launch with netplay enabled
```

### CAUTION!

**PIRATING VIDEOGAMES IS BAD, NOT TO MENTION ILLEGAL!**  
Don't use illegally downloaded content on your PiGRRL.



# MEET THE RUIZ BROTHERS

We talk to the original creators behind the **Pi GRRL** and many other amazing **Adafruit** projects

## PIGRRL HISTORY

How the **PiGRRL** project has evolved



### PIGRRL

The original PiGRRL celebrated the 25th anniversary of the Game Boy and used the original Raspberry Pi Model B to power it. It's a lot bigger than the version we're building, although it more accurately matches the size of the original Game Boy.



### PIGRRL 2

This should look familiar – this is the version we've based ours on! It's an upgrade over the PiGRRL as it uses a lot more custom components, including a custom PCB for the controls instead of a repurposed SNES board. You could easily switch a Pi 3 / 3B+ in there if you wanted a bit more power.

Two of the superstars of the maker scene are Noé Ruiz and Pedro Ruiz, otherwise known as the Ruiz Brothers. They've done many amazing projects for Adafruit, including a lot of 3D printing and wearables, which always go down well with the community. So it's surprising to hear they've only been in the maker scene for about five years.

"My brother and I purchased our first 3D printer in 2012 and quickly started using it in our work," Noé tells us. "While looking for a way to integrate lighting into our 3D-printed designs, we discovered Adafruit and the Arduino platform. We built some projects using their parts and came up with some unique ideas. We went on Adafruit's weekly live show-and-tell show, and the rest is history."

What started off as a load of cool hacks that added LEDs to existing products or enabled you to create great light-up projects quickly evolved into doing more. Part of this was to do with the introduction of the Raspberry Pi.

"Our first project with the Raspberry Pi was the DIY Wearable Pi with Near-Eye Video Glasses," explains Noé. "We were interested



### PIGRRL ZERO

The latest version of the PiGRRL is a tiny device, reminiscent of the GBA or the Game Boy Micro. It uses a Pi Zero and a series of other small components, all squeezed into a tiny little 3D-printed case. The Pi Zero is still powerful enough to run a lot of emulators, though.



## MORE FROM THE RUIZ BROTHERS

Other amazing things you can find by **Noé** and **Pedro**

### RASPBERRY PI POKÉMON FINDER

> [magpi.cc/2lKXpcA](http://magpi.cc/2lKXpcA) <

Based on the Lure Modules found in Pokémon GO, this project caused a little bit of a stir at the time, as it used some APIs that people possibly shouldn't have had access to. It would tell you if there were any Pokémon in your immediate area, and even display a coloured light for how rare the Pokémon was.



### NEOPIXEL YOYO

> [magpi.cc/2lKRd4c](http://magpi.cc/2lKRd4c) <

A simple playground toy turned awesome with the use of some NeoPixel LEDs. The yoyo itself is also 3D-printed, allowing for custom parts so that you can fit the electronics inside. It even has a USB charging port. Check out the link for some cool GIFs of the yoyo in action.



### HALO ENERGY SWORD

> [magpi.cc/2lKKSWC](http://magpi.cc/2lKKSWC) <

This one isn't a completely custom build – instead, it's an upgrade/customisation of a pre-existing licensed Halo Energy Sword toy by Mattel. The Ruiz Brothers took the already pretty cool design and added a ton of NeoPixels to make it pulse with energy, similar to how the sword looks in the Halo games.

in Google Glass and thought we'd make a DIY version with the Raspberry Pi. We hacked apart a pair of video glasses, and designed a custom 3D-printed housing for the display and driver. It was a fun experiment and this is how we learned about the Raspberry Pi."

Things escalated even further when the brothers made the original PiGRRRL. Originally an idea from Limor Fried, founder of Adafruit and 'Ladyada' herself, the idea was to improve upon her earlier Game Grrl project but this time use a custom 3D-printed enclosure. It was their biggest project to date, so they were extremely happy to see that it had such a positive reaction.

The PiGRRRL projects have since become the Brothers' favourites to work on, according to Noé.

"I think it's become a classic Raspberry Pi project because it looks like an iconic device that offers lots of playtime. People love to play games, and being able to build your own gaming console is super-rewarding. Every year we create a new version with better hardware, and change the form factor to try different designs. So many folks have built one and it's really awesome to see parents building them with their kids."



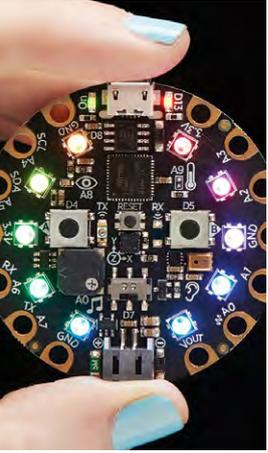
50 PROJECTS FOR MAKERS & HACKERS

**£12.99**  
**180 PAGES**  
OF HACKING  
& MAKING

# BOOK OF MAKING

VOLUME 1

BOOK OF MAKING VOLUME 1



FLY YOUR OWN  
CUSTOM DRONE

BUILD AN  
A.I. WATER CANNON

CONTROL YOUR HOME:  
SMART TECH YOUR WAY

BUILD A  
TREBUCHET

MAKE MUSIC  
ON A DIY SYNTH

COOK FOOD IN A  
HOMEMADE TANDOOR

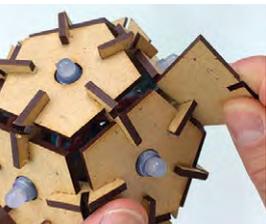
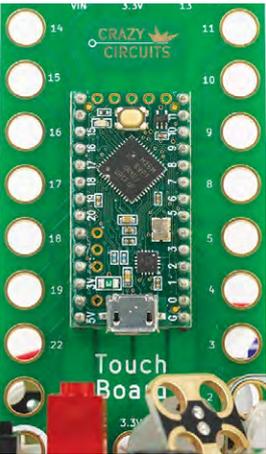
& LOTS  
MORE

FROM THE MAKERS OF **HackSpace** MAGAZINE

hackmag.co

# BOOK OF MAKING

VOLUME 1



- 50 projects for makers and hardware hackers
- Make your own creations with step-by-step guides
- Be inspired by amazing community projects
- Perfect projects for an hour, afternoon, or weekend

Available at: [hsmag.cc/store](https://hsmag.cc/store)

plus all good newsagents and:

WHSmith BARNES & NOBLE



# REVIEWS

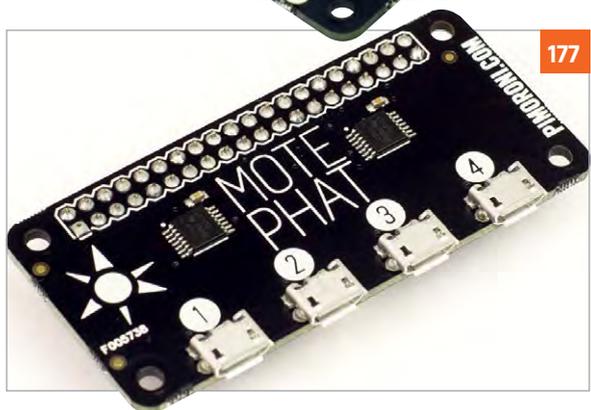
Improve your Raspberry Pi projects by discovering the best accessories and add-ons with our definitive reviews



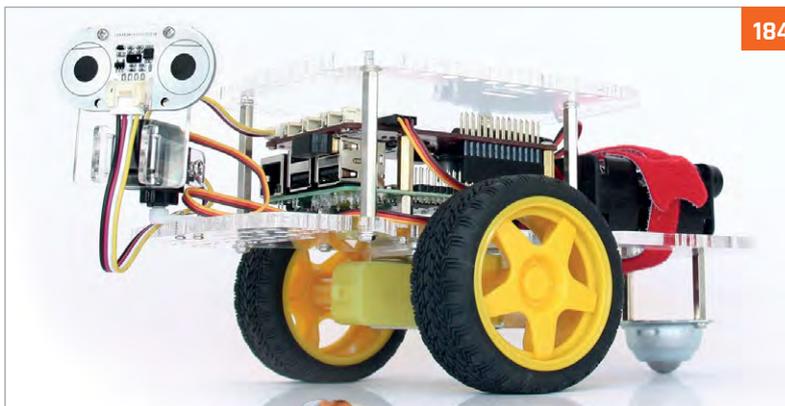
174



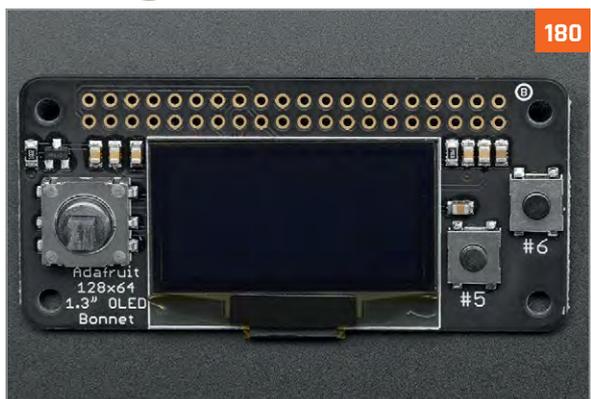
183



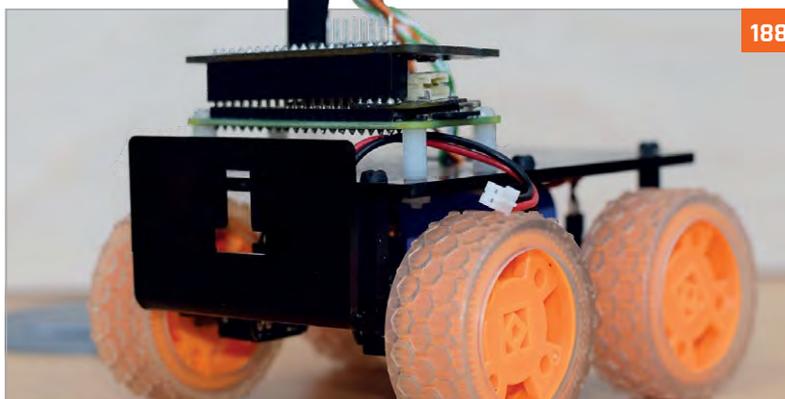
177



184



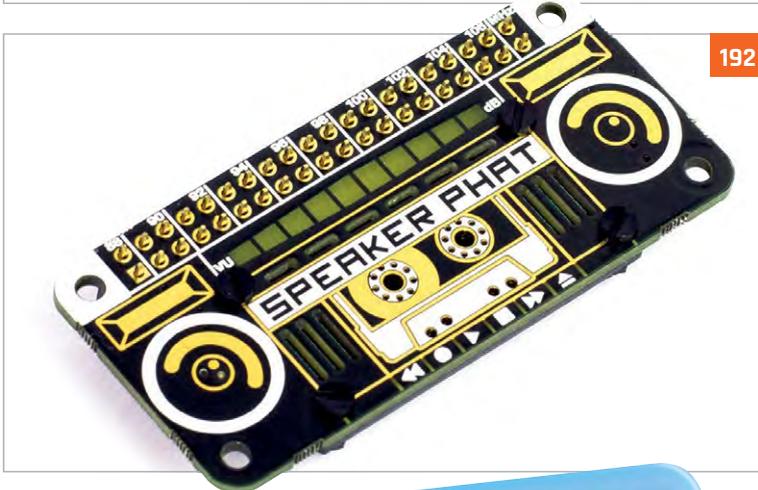
180



188



190



192



194



196

## Reviews

### 172 RASPIO INSPIRING

Programmable LED shapes for impressive 2D and 3D displays

### 173 UNICORN HAT HD

256 LEDs that create a wonderful matrix display

### 174 RAINBOW HAT

A multifunctional HAT that acts as an intro to Android Things on Pi

### 176 PI-TOP PULSE

A set of LEDs designed for pi-top laptops which are fully programmable

### 177 MOTE PHAT

This mini HAT can control up to four programmable LED strips

### 178 PAPIRUS ZERO MEDIUM

A special E Ink display that fits neatly over a Pi Zero

### 179 PAN-TILT HAT

Perfect for robots, this allows you to have a moving camera

### 180 OLED BONNET

A tiny screen and buttons that might prove handy as a mini menu display

### 181 TOUCH PHAT

Capacitive touch buttons that can be slotted above the Pi Zero

### 182 FLICK HAT

A HAT for adding motion-gesture controls to your Pi projects

### 183 INKY PHAT

This more advanced E Ink display has three colours

### 184 GOPIGO 3

A very simple robot kit for beginners, with plenty of learning resources

### 186 RASPIROVER KIT

This kit lets you create a powerful platform for robotics

### 188 TINY 4WD

A super-fast robot based on a feature written in The MagPi magazine

### 190 MEARM PI

Robot arm that's controlled with a Raspberry Pi

### 192 SPEAKER PHAT

Add music to your Pi project with this add-on that even has a volume unit meter

### 193 JUSTBOOM DAC

A powerful DAC that can turn your humble Pi into a proper hi-fi

### 194 PIRATE RADIO

Create an internet radio with this fun Pi Zero project

### 196 JOY BONNET

This tiny controller fits onto the Pi Zero for compact gaming

### 198 PIJUICE

Keep your Raspberry Pi properly powered with this add-on

### 200 ARCADE CONTROLLER KIT

Make your own plug-and-play arcade controller with a Pi at its heart

Maker Says

Connect RGB LED shapes. Sculpt your lights RasPiO



# RASPIO INSPIRING

Create impressive illuminations with this dazzling range of light kits

**T**he main selling point of RasPiO Inspiring is the ability to daisy-chain a variety of LED strips and shapes together to create 2D and even 3D displays. Available kits are: a strip of eight LEDs (£10/£13), a circle or triangle of 24 LEDs (£20), a bike turn signals kit (£48), and a bumper bundle (£83). While the original pyramid kit is no longer available, you can still create it by buying extra triangles.

Although the super-bright APA102c LEDs can be controlled by other devices – including Arduino, ATtiny, and ESP8266 – each kit comes with a pHAT-sized driver board for the Pi, communicating via SPI. Assembly takes around 20 minutes and involves soldering on a 40-pin header, two small female headers, and a socket for a buffer chip. The chip translates the Pi’s voltage to 5V logic to drive the LEDs. The power supplied by the Pi alone should be enough for most projects, but there’s the option of boosting it with an external 5V power supply if needed. Jumper wires are supplied for connecting

the light strips and shapes to the Pi, along with a couple of optional 10kΩ resistors (not needed in most cases). There’s also the option of a header for analogue inputs, so you could control your lights with potentiometers, for instance.

Male and female four-pin connectors are supplied for each LED strip and shape, enabling you to daisy-chain units together. Three sets of input/output holes are provided at each end, so you can solder your connectors at various angles: 180°, 90°, or 45°. This offers more flexibility when connecting units to create custom light displays, although you can always just wire directly to the holes if you prefer.

Indeed, that’s what you’ll be doing if you want to build a pyramid. After assembling the 3D shape from three triangles, held together using pieces of sticky tape, you’ll need to solder bare wires between the input/output holes at each vertex, which is fairly fiddly.

Once the hardware is assembled, the software is installed on the Pi with a single command and includes a range of Python examples which can easily be adapted for your own purposes.

Last word

This range of LED lighting kits offers flexible daisy-chain connectivity that enables makers to ‘sculpt’ custom arrangements of shapes. The end results are impressive: the super-bright LEDs creating some dazzling displays. The bike turn signals kit is a fun-to-make road safety aid, and would make an excellent gift.



Related

MOTE

Instead of daisy-chaining, each super-bright LED strip (no shapes) is connected separately to a pHAT or USB-connected host.

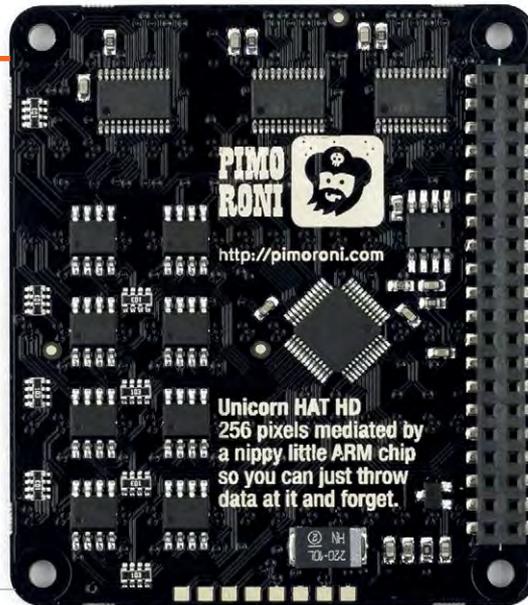
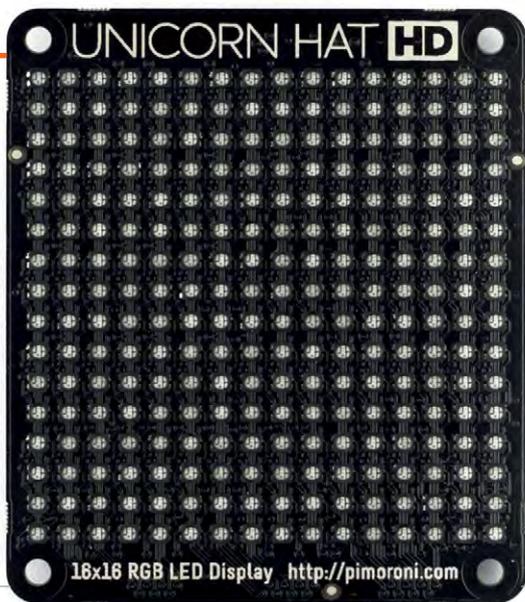


From £16/\$21

magpi.cc/2g9iyLs

magpi.cc/2ultfMn

£32/\$35



Maker Says

256 RGB LEDs in a 16x16 matrix... High-definition rainbow goodness! Pimoroni

# UNICORN HAT HD

Create brighter projects with the new Unicorn HAT HD

Related

ADAFRUIT RGB MATRIX HAT + RTC

This HAT makes it easy to control HUB75-type LED matrices (which you'll need to purchase separately) using a Raspberry Pi.



£22 / \$24

magpi.cc/2u9xlju

The Pimoroni pirates have upped their game with this new HD version of the classic Unicorn HAT, featuring a 16x16 array of super-bright LEDs. Boasting four times the number of RGB pixels of the original display, it can do a lot more and has a real wow factor.

The board comes pre-assembled with its female GPIO header, so there's no soldering required. All you need to do is attach the diffuser layer to the top of the LED array using the supplied nuts and bolts. The diffuser makes a big difference to the visual effect, filling out the gaps in the display between the rather small pixels.

Communicating with the Raspberry Pi via SPI, the HAT uses only four GPIO pins (plus power and ground). Best of all, it has its own ARM STM32F chip to do all the heavy lifting, acting as a middleman between the Pi and the three LED drivers. This means

there's no lag at all when you send data to it, as demonstrated by the camera program downloaded with the one-line software installer.

The camera demo displays the low-res live view from the Camera Module (although it was tricky to connect the latter to the CSI port on a non-Zero Pi model with the HAT sitting on top).

Other example programs also showcase the Unicorn HAT HD well, particularly the snazzy shading demo which transitions between four classic graphics effects to produce some awesome eye candy. A Snake game and Conway's Game of Life demo really benefit from the increased number of pixels, confirming how much more versatile this display is compared with an 8x8 version.

There's even a version of Unicorn Paint that you interact with via a remote web browser, your painted pixels appearing almost instantly on the HAT.

The Unicorn HAT HD has its own Python library, including functions to set the brightness and rotation. Scrolling text across the display isn't as easy as using a single function, requiring a fairly complex loop, but you could adapt either of the impressive multicoloured text demos for this purpose.

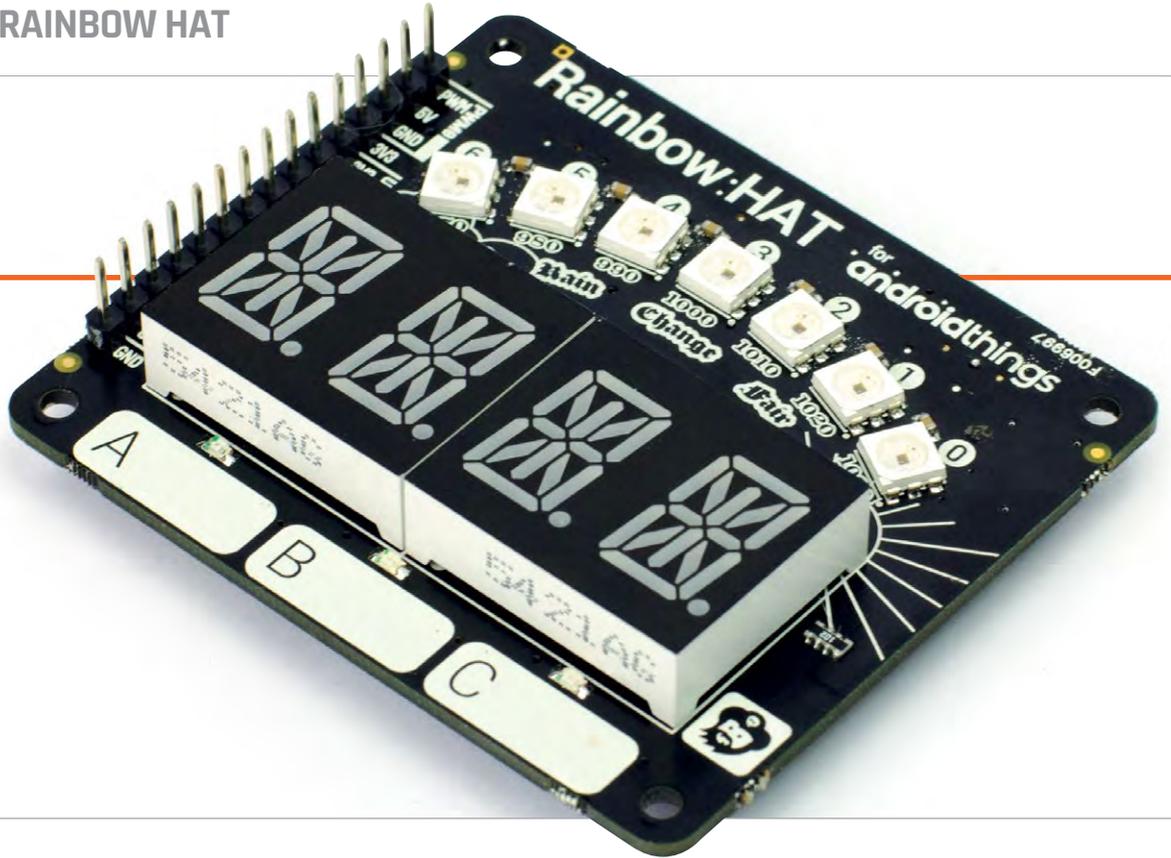
Last word

The increased number of pixels means the Unicorn HAT HD is a more versatile display than its predecessor. The performance is excellent, too, as it reacts near instantaneously to whatever data you throw at it. Since the HAT's ARM chip does all the processing, it's even technically possible to drive multiple displays from one Pi.



## Maker Says

A buffet of sensors, inputs, and displays to explore Android Things  
Pimoroni



# RAINBOW HAT

A versatile HAT to try out Android Things or use with Python

Originally known as Project Brillo, Google's recently relaunched Android Things operating system is designed to be used with IoT devices including the Raspberry Pi 3 / 3B+. To save you the trouble of connecting various individual electronic components to your Pi to try out the new system, Pimoroni has designed the all-singing, all-dancing Rainbow HAT. Jam-packed with LEDs, buttons and sensors, it enables you to experiment with Android Things and use it with the wide range of protocols available on the Raspberry Pi, such as GPIO, I<sup>2</sup>C, SPI, and PWM.

To show numbers and letters, there's an I<sup>2</sup>C-connected four-digit alphanumeric display, which is a cut above the standard type with 14 segments per digit, enabling it to accurately show the full alphabet. GPIO is used for the three capacitive touch buttons and their built-in LEDs (red, green,

and blue respectively). The most visually impressive feature is the SPI-operated 'rainbow' arc of seven super-bright APA102 RGB LEDs that gives the HAT its name. PWM is employed for the piezo buzzer, which can be used to play buzzy tunes. I<sup>2</sup>C is also used for the BMP280 sensor, which measure temperature and pressure. Finally, there's a line of 15 breakout pins for servo, I<sup>2</sup>C, SPI, and UART – all powered by 3V3. One notable omission, however, is an accelerometer/magnetometer, as featured in Pimoroni's Enviro pHAT – possibly there wasn't room on the board for one.

The Rainbow HAT can be programmed using Python in Raspbian, but we first tried it out with Android Things (see boxout for more details). Since the coding – via a remote computer – is mainly done in Java, it helps if you have a working knowledge of the language.

Alternatively, like us, you can import one of the example projects from the Android Things GitHub repo ([magpi.cc/2mFNxTe](https://github.com/magpi/cc/2mFNxTe)). The first time you do this, you'll be prompted to install and update various tools and plug-ins, which is a bit of a hassle, but after that it works smoothly.

## Android examples

To start with, we tried out the sample-button app from the Android Things repo. All it does is light up the LED of the Rainbow HAT's capacitive button A when you press it. By digging into the code, you can change the GPIO pin numbers to use a different button and LED, although you'll need to refer to [pinout.xyz](http://pinout.xyz) to find the correct numbers. Next, we imported the weather station project, which showed off more of the HAT's capabilities. By default, the segment display displays the current temperature, although

## Related

### ENVIRO PHAT

This Zero-size board features four different sensors, for measuring temperature, pressure, light level, colour, motion, and direction, plus analogue inputs.



£16 / \$20

[magpi.cc/29NHB3T](https://magpi.cc/29NHB3T)

magpi.cc/2lX1r6h

£24 / \$26



## Installing Android Things

Once you've downloaded the latest image of the Android Things OS (see [magpi.cc/ssGoaw](http://magpi.cc/ssGoaw)), you can flash it to a microSD card to insert in the Raspberry Pi. On booting up, it shows a logo with network connection information underneath – you'll need to hook the Pi up to the router via Ethernet at first, as there's no way to set up wireless connectivity beforehand. With the Android Studio application installed on a remote computer, you can then connect to the Pi from a terminal using the adb tool from the SDK. You can then start building an app in Android Studio and deploy it to the Pi.

we soon noticed it rising. As usual with a board-mounted sensor, this is due to the heat from the board and the Raspberry Pi just below it, but you could always calibrate it by comparing the real ambient temperature (using a standard thermometer) to discover the average difference. A press of a button switches the display to pressure, which is also indicated by

the rainbow arc and showing AHQY, YARR or GROG on the alphanumeric display depending on the capacitive button pressed! Other examples include a temperature gauge and an RGB/HSV lamp whose shade is set using the touch buttons.

A look at the library example code – or the getting started tutorial at [magpi.cc/2lX6Jh](http://magpi.cc/2lX6Jh) – reveals a simple syntax for using the HAT's features,

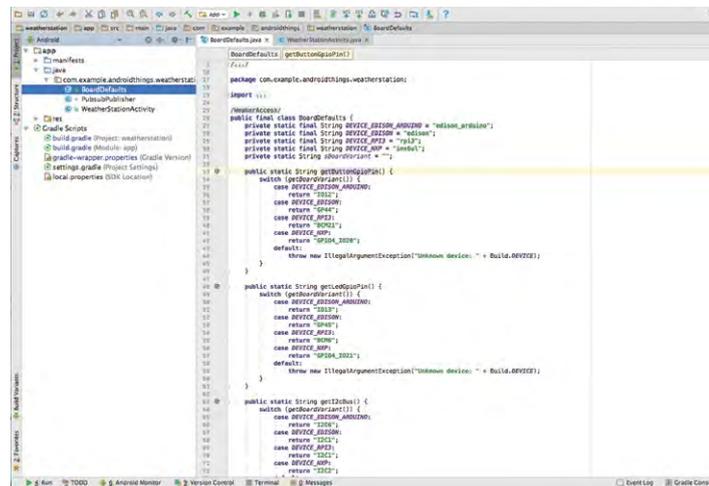
“ The most visually impressive feature is the SPI-operated ‘rainbow’ arc ”

the rainbow arc of LEDs. You can also configure this example app to publish sensor data to Google Cloud Pub/Sub, from which you can process it and visualise it using various Google tools.

Alternatively, in the Raspbian OS, Python can be used to program the HAT. There's a dedicated library, added using the one-line installer. Example programs include a pirate-themed demo that plays the *What Shall We Do with the Drunken Sailor?* sea shanty on the piezo buzzer while gradually changing the colour of

such as `weather.temperature` to obtain temperature data and `display.print_str('AHQY')` to show a string on the four-digit alphanumeric display.

Whether programmed using Python or Android Things, the Rainbow HAT has an impressive number of features crammed into it, making it useful for all sorts of projects. Suggestions from Pimoroni include a weather station, stopwatch or countdown timer, three-note piano, reaction time game, and mood light.



Above Android Studio helps you program the Rainbow HAT

## Last word

A potpourri of LEDs, buttons, and sensors makes the Rainbow HAT very useful for experimenting with Android Things, or programming with Python in Raspbian. It may be overkill for some projects, but it's a versatile add-on that offers good value for money.





*Maker Says*

pi-top PULSE brings the arts into STEAM pi-top

# PI-TOP PULSE

A new multimedia add-on that brings lights and sound to a pi-top or a Raspberry Pi. **Rob Zwetsloot** takes a look

**W**e love the pi-top range of computers. They provide true PC or laptop conversion kits for the Raspberry Pi, customisable with plug-in modules and aimed at educational users. Usually these add-ons are designed to enhance the experience of using the pi-top, but that's all changing with their latest and most ambitious add-on, the pi-topPULSE.

It may be ambitious, but it is also very cool. From the top, it's an unassuming white plastic square. It includes a 7x7 LED display under the plastic, plus a microphone and speaker. The PCB is visible on the underside, but it is lovely and well designed.

On its own, it can be plugged into a pi-top using the special magnetic rails inside the slide-out

compartment, but it can also be used as a HAT on the Raspberry Pi. You'll need to plug in a header to the underside of the PULSE before popping it on the Pi, but it works with Raspbian after installing a single piece of software.

It's a very tight fit on the Pi. The case barely misses the Ethernet port, and the connectors for the pi-top hang over the board, increasing the footprint of the Pi. As an educational tool it is excellent, with a very robust Python library for creating light displays and games.

The microphone and speaker give you access to other functions. The PULSE acts as a fully integrated speaker for Sonic Pi, and is even optimised for use with Amazon Alexa on the Raspberry Pi. There's a great demo on the

product page that shows how it can be used with Alexa.

It is a little tricky to get your head around the programming, but there are plenty of examples that helped us figure out how to use everything.

All in all, it's one of the most fun HATs we've seen in a while, allowing users to explore many amazing project ideas.

*Last word*

An amazing idea with great execution – we just wish it was slightly easier to program. With a bit of practice you can create some astounding things with the PULSE.



*Related*

**UNICORN HAT HD**

With its programmable 16x16 RGB LED display, this is a great HAT that is easy to code and fun to use in a range of projects.

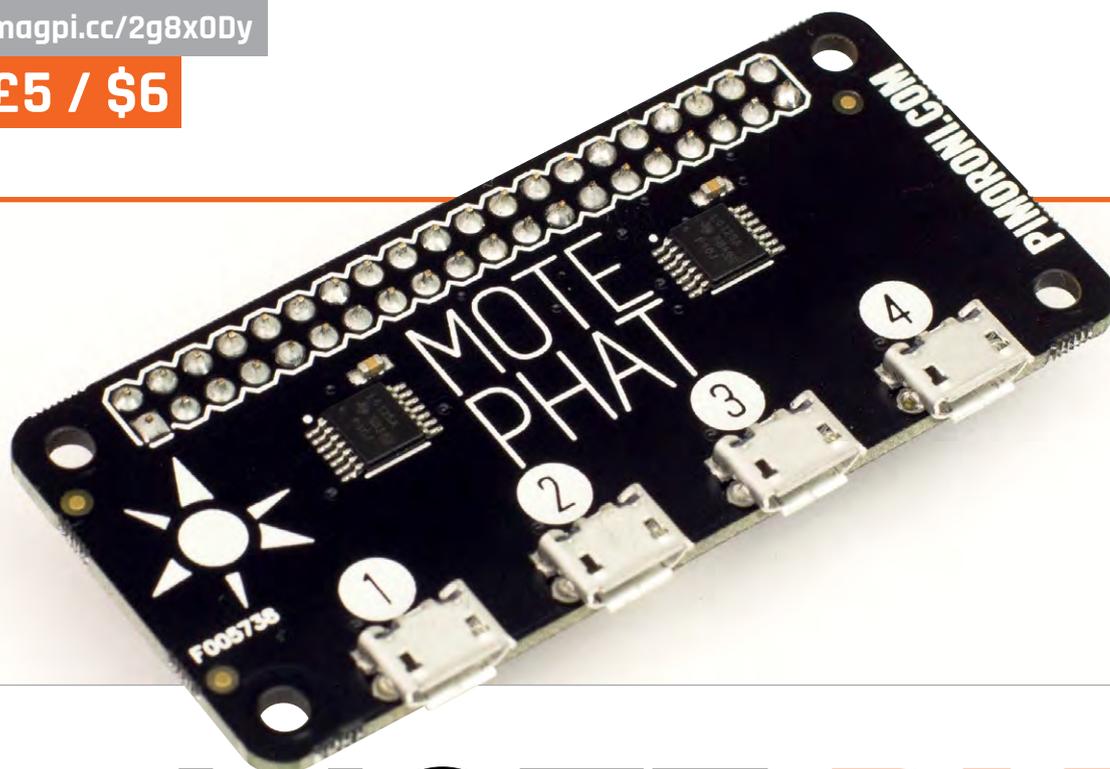


£32 / \$35

magpi.cc/UniHATHD

magpi.cc/2g8x0Dy

£5 / \$6



### Maker Says

Run our beautiful Mote lights straight from your Pi or Pi Zero  
Pimoroni

# MOTE PHAT

A more compact way to control Mote light strips

**F**eaturing 16 super-bright RGB LEDs, Pimoroni's Mote 'stick' light strips (£8.50 each) are ideal for under-shelf/cabinet mood lighting, among other applications, and can be used to create some impressive effects. Until now, a Mote Host has been required to control the sticks, plugging into any computer – including the Pi – via USB. For a far more compact and convenient setup, the Mote pHAT does the same job. Its pHAT form-factor matches that of the Pi Zero perfectly, so you can tuck it discreetly away out of sight. You will need to solder the female GPIO header onto the Mote pHAT first, although at least this does give you the option of using a stacking header to use it alongside other add-on boards: you could get Mote sticks to display sensor readings from an Enviro pHAT, for example, or control them with buttons or switches.

As with the Mote Host, there are four output channels via micro USB ports for controlling up to four Mote sticks independently of each other. A Python library ([magpi.cc/2fw4oFX](https://magpi.cc/2fw4oFX)) is provided, along with a few examples to get you started. These include an impressive rainbow effect, colour cycling, and CheerLights web control via Twitter. The library itself is easy to use, enabling you to set each individual RGB pixel on each of the four channels. They are triggered with a **show** command, and **clear** is used to turn them off. Using a **for** loop, we soon managed to create a simple chase lights effect – handily, if you exceed the highest pixel number, it wraps round to the start of the strip again. Since the LEDs are APA102 standard (aka DotStar) with a fast data rate, there are none of the timing issues you might get with NeoPixels and they respond almost instantaneously.

You can even control them from your phone or from a web browser by setting up a Flask API. Pimoroni's step-by-step tutorial ([magpi.cc/2g9g143](https://magpi.cc/2g9g143)) shows how to use HomeKit to control the lights from an iPhone, using Siri voice commands to turn them on and off and to set the colour – a good way to impress friends! The same result should be possible using Google Assistant or Alexa, too.

### Last word

The Mote sticks are excellent, even better than NeoPixels, and the new Mote pHAT offers a compact, discreet way of controlling them without the need to connect to the USB port of a laptop or Pi. At just £5 it's great value, although of course you'll still need to buy sticks (and cables) to use with it.



### Related

#### MOTE HOST

The original Mote controller plugs into the USB port of any computer, including the Pi. Like the pHAT, it features four micro USB outputs.



£11 / \$14

magpi.cc/2g9iyLs



## 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

**M**any 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. 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.



## 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.



£41 / \$54

magpi.cc/2ouxPtz

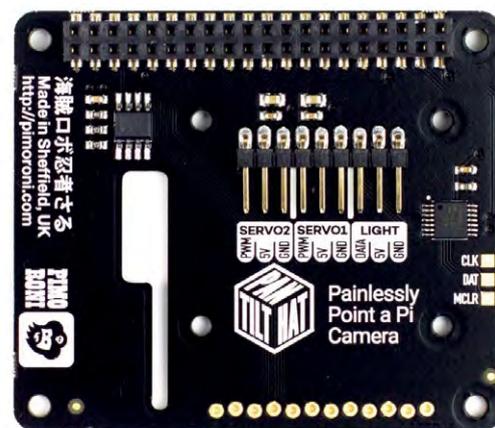
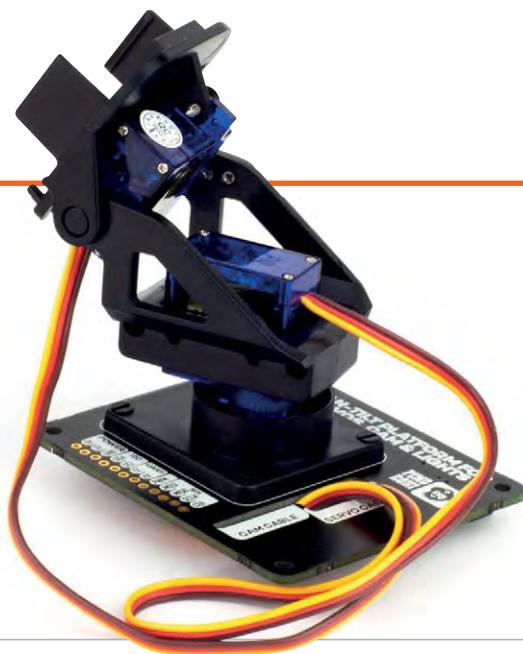
magpi.cc/raRzbi

£30 / \$33

## Maker Says

Let your Raspberry Pi and camera look at the world around you

Pimoroni



# PAN-TILT HAT

Raspberry Pi camera stand with horizontal and vertical motion servos

**T**he Raspberry Pi Camera Module is one of the best accessories you can get, enabling cheap photography on the Pi. But it doesn't sit upright on its own – a stand is required.

The Pan-Tilt HAT fulfils this function and a whole lot more. The Camera Module is mounted on the end of a robotic arm that sits on top of the HAT. Thanks to the arm's horizontal and vertical joints, the camera can be angled precisely by the two servo motors.

The finished effect is adorably cute, instantly imbuing your Raspberry Pi with personality as it looks around the room.

It's really useful too. You could set the Pan-Tilt HAT up to monitor a room, and then use VNC or SSH to adjust its viewing position remotely.

Alternatively, you can set up a Raspberry Pi with face-tracking software and connect it to the Pan-Tilt HAT. Pimoroni, the HAT's makers, also suggest mounting it on top of a robot for a set of eyes.

## Setting it up

First, you need to set the HAT up. Fortunately, there is an online setup guide ([magpi.cc/2hR4NFC](http://magpi.cc/2hR4NFC)).

The board has a GPIO connection on one side, and servo connections on the other. The two sets of cables on the arm are connected to Servo 1 and Servo 2 on the board (1 for pan, 2 for tilt). A third servo channel can be used to control an optional NeoPixel strip for lighting.

## Camera control

You can download all of the code from Pimoroni's GitHub page ([magpi.cc/2hRrjyo](http://magpi.cc/2hRrjyo)). You need to install the `pantilthat` module to access the controls.

After importing the `pantilthat` library in Python, you use `pan()` and `tilt()` methods to change the camera position. These accept any value between  $-90$  and  $90$ . To set the camera straight forward, for example, you would use:

```
pantilthat.pan(0)
pantilthat.tilt(0)
```

To look up by 45 degrees, use:

```
pantilthat.tilt(-45)
```

To look all the way to the camera's left, you'd put:

```
pantilthat.pan(90)
```

We would have dearly loved more software examples. There are ones for motion and NeoPixels, but none for recording from the camera or face-tracking. A few more sample programs and it'd be perfect.

Even so, we had a lot of fun setting up the Pan-Tilt HAT and look forward to researching and coding a face-tracking program.

## Last word

A highly enjoyable and extremely cute accessory. With a bit of research, you should be able to create some fun things with it.



## Related

### RASPBERRY PI CAMERA MOUNT

It doesn't move, but if all you're after is a camera mount, you can pick one up from The Pi Hut for a tenth of the price.

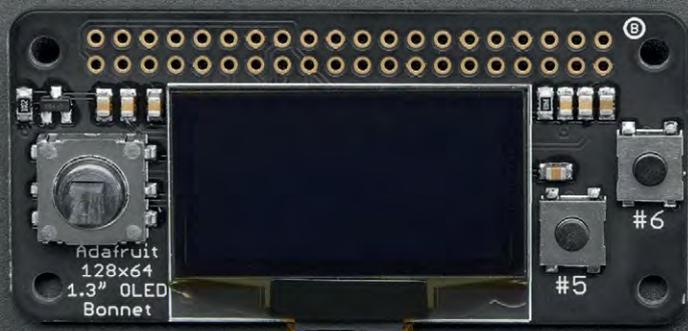


£3 / \$4

[magpi.cc/2hRtkdT](http://magpi.cc/2hRtkdT)

Maker Says

A compact display, with buttons and a joystick  
Adafruit



# 128×64 OLED BONNET

A high-contrast mini OLED display, complete with controls

Looking for a low-power yet bright mini display for your Pi project? Adafruit’s miniature OLED screen could well fit the bill. An OLED (organic light-emitting diode) display offers high contrast combined with a low power draw, since it doesn’t require a backlight.

While numerous OLED screens are available, including a range from Adafruit itself, most require you to wire them up manually to the Raspberry Pi (or whatever device you’re using). The Pi Zero-sized OLED Bonnet takes the hassle out of connection: pre-assembled with a female header, it simply slots onto the Pi’s GPIO pins.

Available from Pimoroni in the UK, the OLED Bonnet is the big sibling of the 128×32 PiOLED (magpi.cc/2xAq7po), doubling the latter’s screen area while adding a mini joystick (four-way plus

central push function) and two buttons. This would make it ideal for use as a mini menu system in, for example, a music player.

While the screen is monochrome – white on black – and obviously too low-res to use as a main Pi display, its high contrast enables it to show text with great clarity. Any standard TTF font can be used, and one of the Python examples downloaded after cloning the relevant GitHub repo is an old-school sine-wave scrolling text demo. Basic images, which may be converted to bitmaps and resized via PIL, can also be displayed.

Unlike an E ink screen, the OLED Bonnet is even able to handle basic animations. While the frame rate is rather sluggish by default, it can be speeded up to about 15 fps by raising the I<sup>2</sup>C core baud rate to 1MHz in the Raspberry Pi’s /boot/config.txt file.

As well as two GPIO pins for I<sup>2</sup>C communication with the Pi, the OLED Bonnet uses seven others for joystick and button inputs. That still leaves plenty of GPIO pins available for use in projects, although due to the full-size female header, you’ll need to break them out using something like a Pico HAT Hacker.

Last word

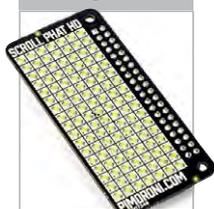
With its high contrast and clarity, the OLED Bonnet is ideal as a mini status display or – taking advantage of the joystick and buttons – menu system. The screen’s low power draw (around 40 mA on average) is also an advantage for portable projects using battery power.



Related

SCROLL PHAT HD

Packing 17×7 white pixels, with full PWM brightness control, this display is ideal for scrolling text messages.



£12 / \$13

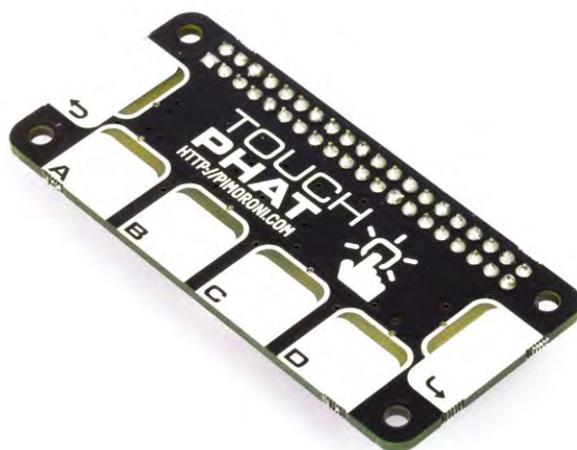
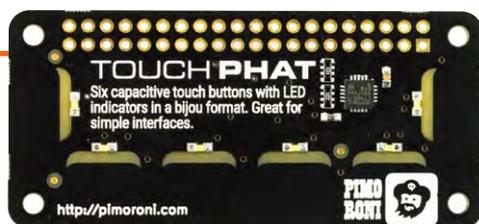
magpi.cc/2wShYcf

magpi.cc/2paf0P9

£8 / \$9

## Maker Says

Six touch-sensitive buttons to use for whatever your heart desires  
Pimoroni



# TOUCH PHAT

## An easy way to add button inputs to your projects

**N**eed to add some button controls to a project you're making? Pimoroni's Touch pHAT makes it easy. This Pi Zero-sized board boasts six touch-sensitive buttons which light up when pressed; white LEDs located on the underside produce a yellow/green glow through the translucent board sections. While the buttons are marked A, B, C, D, Back, and Enter (and referred to as such when coding), each has a large white area for custom labelling with a sticker or dry-wipe marker.

The pHAT is supplied with a female header which you'll need to solder in place. While the board has a Pi Zero form factor, it can be used with any 40-pin Pi model. Equipped with a CAP1166 capacitive touch and LED driver chip, it uses I<sup>2</sup>C for communication, and therefore requires only two GPIO pins. No standoffs are supplied, but you may want to add some to

keep the pHAT rock steady on top of the Pi as you press the buttons. Alternatively, you could combine it with Pimoroni's neat-looking Pibow Zero W case.

Like most Pimoroni add-ons, the Touch pHAT has its own Python library, which is easily installed – along with any missing dependencies – using a single Terminal command. A couple of examples are supplied: a simple button-press demo and a GUI app launcher. The code syntax is simple enough, using `on_press` and `on_release` events to register the relevant touch actions. It's then completely up to you as to what these will trigger. Possible uses for the Touch pHAT include as a control panel for a robot, a remote control for home automation, a drum machine / mini piano, and a Simon game.

Most importantly, the buttons are very responsive to touch

and will stay triggered/lit until released; you can press as many as you like simultaneously, too. They can even be activated through a thin transparent layer if needed. Note that if you wanted to attach alligator clips to the buttons to attach remote triggers (such as pieces of fruit), you'd have to scrape down to the copper on each button to make the connection work.

### Last word

The Touch pHAT makes it a lot easier to add input buttons to projects, instead of having to wire up push-buttons individually. What you use it for is completely up to you, but the touch-sensitive buttons are really responsive and the light-up effect is a nice bonus.



## Related

### RAINBOW HAT

Compatible with Android Things and Python, this full-size HAT features three capacitive buttons, a four-digit display, LEDs, buzzer, and sensor.



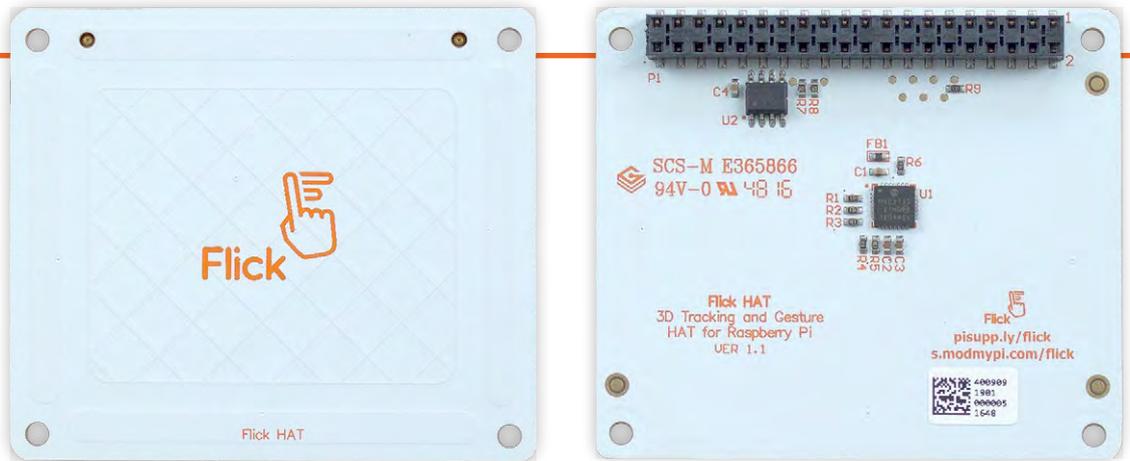
£24 / \$26

magpi.cc/2lX1r6h



## Maker Says

Swipe, tap or flick your wrist to control your Raspberry Pi Pi Supply



# FLICK HAT

One of a trio of 3D gesture-tracking boards for Raspberry Pi computers

What could be cooler than triggering actions with a mere wave of your hand? The Flick HAT allows you to do this, and more. The Flick HAT is one of three 3D gesture-tracking boards from Pi Supply – there’s also a smaller Pi Zero version, and a larger standalone pre-soldered board with jumper leads to connect it to the Pi’s GPIO header.

After securing the Flick HAT to the top of the Pi with the nylon nuts, bolts and spacers supplied, and then using a single-line installer for the software, it’s ready to use.

While we’re still awaiting a full Python API, a few example programs are supplied. The most basic of these shows the X, Y and Z values as you move your hand above the board, and it works best when you use a downward-pointing finger. While the blurb claims a vertical range of up to 15 cm, in our tests we typically achieved 5–7 cm. This distance will depend on the surroundings, and how conductive you are.

As well as tracking 3D positions, the board reads gestures including swipes (left-right and up-down), and ‘airwheel’, which involves rotating your finger in mid-air. This is ideal for an audio volume control, as used in the **volctrl** demo. Among many other uses, the swiping actions would be suitable for controlling a slideshow presentation – something covered in *The MagPi* magazine previously, using the rival Pimoroni Skywriter HAT ([magpi.cc/1PLLYMt](http://magpi.cc/1PLLYMt)).

The Skywriter and the Flick are based on the same MGC3130 3D tracking chip. Not surprisingly, their performance is very similar. The chip reads up to 200 positions

per second, enabling it to reliably detect simple gestures. Like the Skywriter, the Flick also senses touch – in its centre and on the four edges – so it’s a versatile input device that could prove useful for countless projects.

Interestingly, it even works with the Skywriter Python library, from which the Flick library is derived. We were therefore able to use the Skywriter software examples, including a fun theremin synth.

One advantage of the Flick HAT is the availability of optional cases for all three models. The cases use a non-conductive surface, which does not interfere with the Flick’s ability to track gestures.

## Related

### SKYWRITER HAT

Based on the same chip as the Flick, Pimoroni’s 3D gesture-tracking board boasts similar performance. There’s also an XL breakout version.



£16 / \$17

magpi.cc/10APeHb

## Last word

The performance of the Flick HAT is very similar to that of the Skywriter HAT, but the Flick offers an optional snug-fitting case, as do the other Flick models. Communicating with the Pi via I<sup>2</sup>C, the Flick only uses a small number of GPIO pins. With the ability to detect a wide range of gestures and touches, it would be a useful input device for many projects.



[magpi.cc/2w1RKow](http://magpi.cc/2w1RKow)

£22 / \$29



### Maker Says

A low-energy, high-falutin, red/black/white electronic paper display for your Pi!

Pimoroni

# INKY PHAT

Add a little more colour to your E Ink display, says **Phil King**

**W**ith the ability to display text and images that remain on screen without using a power supply, E Ink displays are ideal for saving precious battery life in portable projects. They're also much easier to view in bright sunlight than conventional screens.

Until now, Pi E Ink displays have been black and white – but the tri-colour Inky pHAT injects a bright shade of red or yellow into the mix. This certainly adds a dash of extra interest to images, such as the logo shown on the display when you unpack your pHAT.

Unlike the rival PaPiRus Zero, the Inky pHAT comes fully assembled, with the E Ink screen already mounted and connected to the board via a small ribbon cable. This saves fiddling about with a thin and fragile screen, although you still have to take care not to press on it when mounting the pHAT on the Pi's GPIO pins.

While the Inky pHAT has a Zero form factor, it'll happily work with any 40-pin Pi model. Communicating via SPI, it only uses six GPIO pins (plus 3V and 5V power). If you wanted to locate the screen away from the Pi, however, you'd need to use something like a Black Hat Hack3r ([magpi.cc/2fqGy0D](http://magpi.cc/2fqGy0D)).

A single Terminal command installs all the software required. The Python library comes with a few helpful examples, including a calendar and a weather display. There's a lot of flashing and pulsing as the screen refreshes, the red parts usually appearing last. While the end result looks vibrant, the downside is a much slower refresh time compared with a monochrome E Ink screen – typically around 15 seconds vs 1 second or less. The badge example takes even longer, but demonstrates the ability for a

partial update, adding text (your name) to a background image.

The slow refresh rate makes the Inky pHAT unsuitable for scrolling text, but static text can easily be displayed in any installed TrueType font, using a simple `message` function in Python. Images are a bit trickier, as you need to prepare them in a special indexed colour mode, and they must also exactly match the size of the display at 212×104 pixels.

### Last word

While the slow refresh rate makes the Inky pHAT unsuitable for some projects, it does look far cooler than a standard monochrome E Ink display. If you want a low-power, battery-saving screen with a little more pizzazz, this is the one to get.



### Related

#### PAPIRUS ZERO

Available in a range of sizes, this black-and-white E Ink display has a far faster refresh rate, plus push-buttons to control what's shown on screen.



From  
£27 / \$35

[magpi.cc/2ou5KTh](http://magpi.cc/2ou5KTh)

## Maker Says

A complete kit to build your own robot car  
Dexter Industries



# GOPIGO3

This sturdy robot kit creates a network and broadcasts lessons, making it ideal for students and teachers alike

**F**or the last few years, the GoPiGo has been one of the most impressive robot kits available for the Raspberry Pi.

The build remains excellent for the latest version. Two 3D-printed acrylic boards house a Raspberry Pi 3 and GoPiGo3 board. To this, you attach the motors, a wireless networking dongle, a battery pack, and a USB thumb drive.

The build of the robot is relatively complex (although with fewer steps than the GoPiGo2 kit it replaces). It took us 45 minutes to put in all the screws, washers, and posts that connect the various parts together. This build time is about three times as long as most kits we encounter.

There are good photographic instructions, though, and we didn't get stuck during the build process. More information on the build process can be found at [magpi.cc/2vo8oVo](http://magpi.cc/2vo8oVo).

You might be wondering why it comes with a wireless dongle when the Pi 3 / 3B+ has wireless built in. That's because GoPiGo3 broadcasts its own wireless hotspot, so the minute you switch it on, you can connect to it via a local network called 'GoPiGo'.

Thanks to this approach, as soon as you've finished the setup and inserted the eight batteries, it becomes an absolute breeze to start using the GoPiGo3. There's a whole new DexterOS operating system to play with and it's packed with features.

### Hotspot robot

The robot transmits its network called 'GoPiGo'. You connect to this network, and point a web browser to [bloxter.com](http://bloxter.com). You don't need internet access to do this. You can log into the network from a Windows, Mac, or Linux computer;

a tablet or smartphone; and of course, another Raspberry Pi.

The web interface starts with four options: Drive, Learn, Code in Bloxter, and Code in Python. Click Drive and you can move the robot around using the on-screen buttons. Tap 'Code in Bloxter' and you can use Scratch-style blocks to control the robot. The 'Code in Python' option lets you use a web-based Python IDE to program the robot with the GoPiGo3 software library (GitHub, [magpi.cc/2tWzwmwE](http://magpi.cc/2tWzwmwE)).

You can use Scratch to control the GoPiGo, but you need to switch the OS over to 'Raspbian for Robots'. We think DexterOS has the better solution with Bloxter and Python. Based on Google Blockly ([magpi.cc/2uq6MG9](http://magpi.cc/2uq6MG9)), Bloxter is similar enough to Scratch.

On the whole, DexterOS is a slick solution that enables you to

## Related

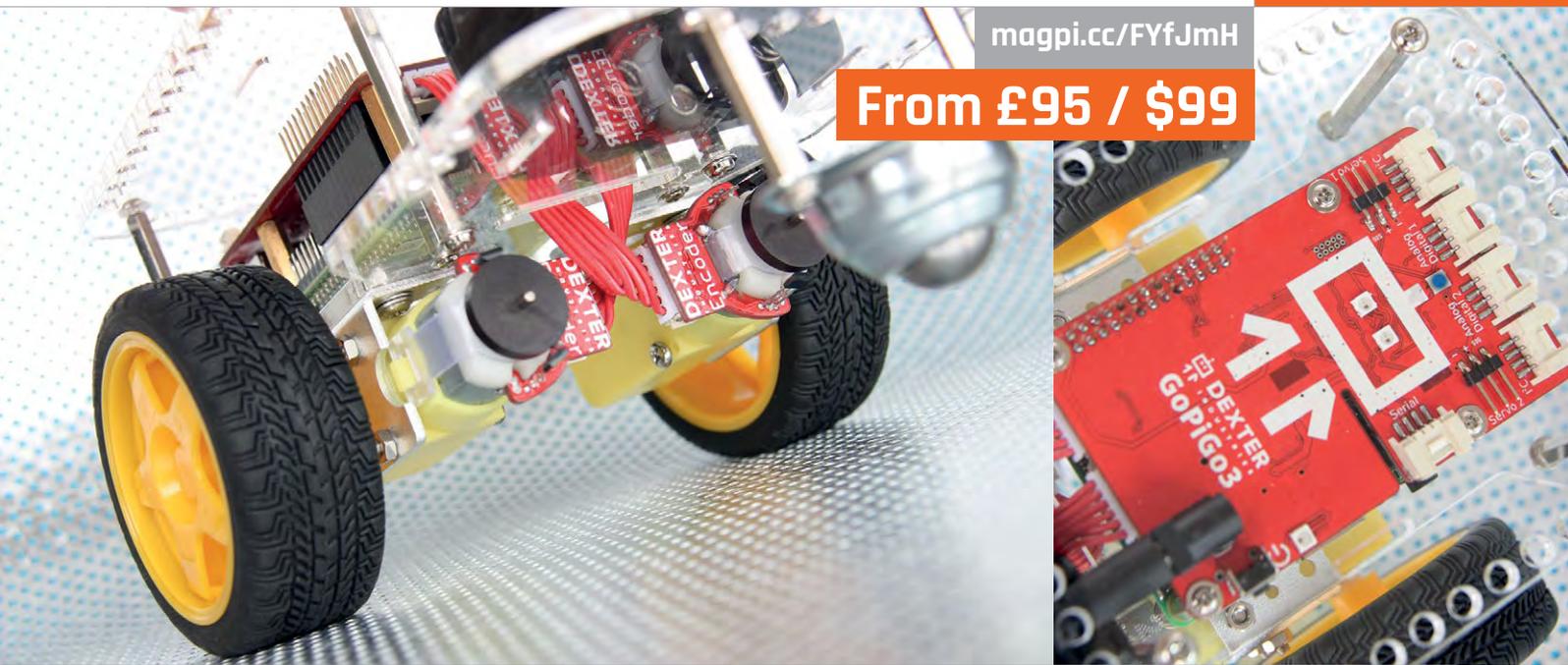
### CORETEC TINY 4WD

With twice as many wheels for around half the price, this Pi Zero W based robot is a capable rover.



£55 / \$60

[magpi.cc/0iXbKb](http://magpi.cc/0iXbKb)



start controlling and programming the robot quickly. Perhaps more importantly, it combines the lessons and programming environment into one single space that doesn't require an internet connection – the robot acts as a router. There are dozens of lessons covering movement, sensors, speakers, buzzers, and buttons.

## Ready to teach

There's a lot here for teachers. The GoPiGo is a great tool for introducing robotics to a classroom. It's relatively easy to set up, very easy to connect to from a wide range of computers, network independent, and packed full of tutorials for visual and text-based programming environments.

The two motors have encoders built in, so you no longer have to attach these separately during the build. These act as tachometers, measuring the precise rotation of the wheels. The GoPiGo3 runs forward and backwards in a straight line, can move by precise amounts (such as 10 cm or 5 in), and turn with degree precision. It's a small thing that makes a massive difference to the experience of using a robot.

There is a GoPiGo base kit for just \$99. It includes the GoPiGo3 board, chassis, wheels, motors, encoders,

and power battery pack. You need to add your own Raspberry Pi 3.

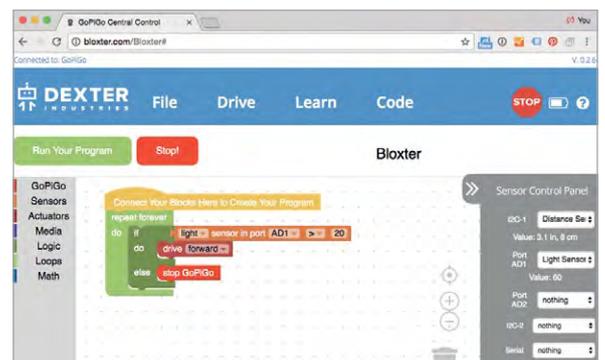
We tested the \$199 Starter Kit, which comes with a Raspberry Pi 3, mini WiFi dongle, GoPiGo servo package, distance sensor, microSD card (with DexterOS software), 8GB USB drive, and power supply. It's a useful complete package, but we think the base kit would be sufficient if you already have many of the components.

The 8GB USB drive is used to update the software. You add software updates to the USB drive, power up the GoPiGo robot, and it automatically updates its software (see [magpi.cc/2t1Q0yr](http://magpi.cc/2t1Q0yr)) – again, without the need to remove the SD card or connect to a network.

Both kits are available in the UK from ModMyPi ([magpi.cc/2tTWJqN](http://magpi.cc/2tTWJqN)).

Dexter offers a range of accessories for the GoPiGo. The Starter Kit comes with a servo motor and distance sensor, but you can also add an ultrasonic distance sensor, sound sensor, buzzer, temperature and humidity sensors, camera, and line follower. You can find tutorials for each of the sensor projects at [magpi.cc/2v08oVo](http://magpi.cc/2v08oVo).

There's a lot about this robot that makes it ideal for the classroom. Aside from the integrated lessons and programming environment,



it's a sturdy build. We'd wager it can take quite a few knocks.

Even if you're not a student or a teacher, it's a great robot kit. The built-in encoders make it more accurate than other robots, and there are plenty of holes on the board for adding your own custom equipment. We've got a lot of time for GoPiGo3, and we intend to spend a lot of time experimenting with it. Well done Dexter!

Above Control the robot's movements using Bloxter

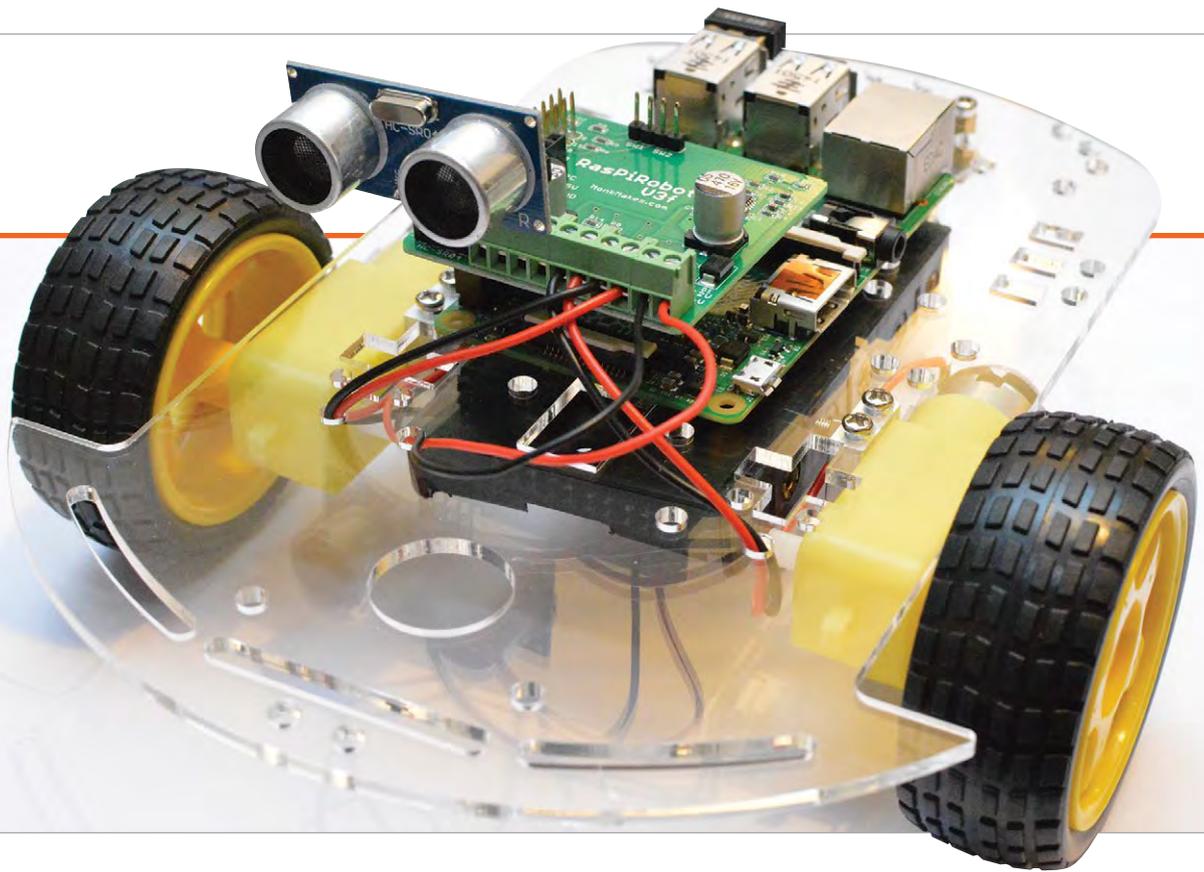
## Last word

One of the best robotics kits you can buy, especially for teachers. DexterOS is delightful, bringing programming and learning into one space. Creating its own hotspot frees the robot from network limitations.



**Maker Says**

Build your own Robot Rover for Raspberry Pi - just add Raspberry Pi and batteries  
**Monk Makes**



# RASPIROBOT ROVER KIT

**Related**

**GOPIGO**

The GoPiGo is another fairly simple-looking robot kit. However, it too can be programmed to do a lot of robotics tasks.



£95 / \$99

magpi.cc/2hA8f6i

A kit that includes more than just a robot, how well does it work for teaching youngsters about coding?

**W**e always appreciate the kits from Monk Makes. They're functional, come with great documentation, and most importantly are always a good-value price. The RasPiRobot board that you can currently buy impressed us a little while ago, so it's nice to see a kit that uses it for a Pi robot in the way Simon Monk intended.

This particular robot kit is two kits in one. As well as being able to create a basic robot that you can program, you have access to a push-button and the Raspberry

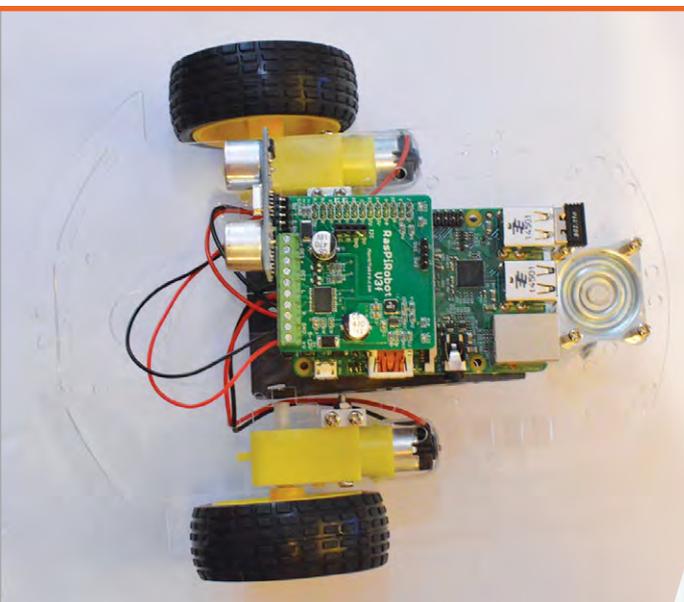
Squid RGB LED. These work on the RasPiRobot board without it needing to be built as a robot; however, it does also mean you can attach them once the robot is built. It's an interesting concept, as in theory you can use the button and LED as a 'getting started with GPIO programming' step and then move onto building the robot and doing even more.

As for the robot build itself, the process is nice and easy. The chassis is made up of one piece of laser-cut acrylic and it uses two motors to drive it. A caster wheel

is attached to the rear so it can pootle about, and the battery pack can be attached to the underside. The Pi is merely stuck to the surface using some Blu Tack or double-sided tape and the rest of the components slot neatly into the motor controller which attaches on top of the Pi. It's very simple and there's a couple of video guides from Simon that show you how to do it, although it would be nice to have a step-by-step visual guide as well. It took us an episode of the original *Star Trek* to build, so under an

magpi.cc/2i7kJzh

£42 / \$50



**Above** There's not much to build with the kit and you'll be done very quickly. There are lots of extra parts, though

hour, although that does include us struggling to peel off the protective layer of paper from the chassis and find the rechargeable batteries for it.

it connected up to the board, although that's all part of the fun.

Programming it is very easy, using the same standard Monk Makes libraries. The instruction

While a minor spin on the normal robot kit, it does open up a lot more types of customisation and play options that are much simpler to understand.

We very much like the RasPiRobot Rover and its kit. We've built a few other similar and simple robots and while they're always quite good, they don't quite go the extra mile like this. Not only that, but we can easily get our Raspberry Pi off it without having to fiddle around with tiny screws to do so. For the price, as well, it's an excellent starter kit for kids interested in the Raspberry Pi and programming electronics.

**“ You can use the button and LED as a ‘getting started with GPIO programming’ step ”**

### Tough enough

Once built, it's quite sturdy. The chassis isn't large enough to be bendy and the Raspberry Pi can be affixed firmly enough in place so it doesn't fly off. The kit also comes with speed sensors in the form of timer discs you can attach to the motors if you want to get the extra sensors to add them. Speaking of adding sensors, there are plenty of spots along the chassis to add them. The front is especially good for line and proximity sensors, as well as adding a pan-and-tilt arm. You'll have to do the research yourself to figure out how to get

booklet that comes with the robot also provides some little tutorials on getting the LED and button working, as well as programming the robot for remote control and autonomy. The ultrasonic rangefinder also has a bit in there, so you can start programming with that as well. The great part about all these instructions is that you can combine them to do crazy things. For example, you could have the button start a little mission where it drives towards the nearest wall and uses the rangefinder data to change the colour of the LED as it gets closer.

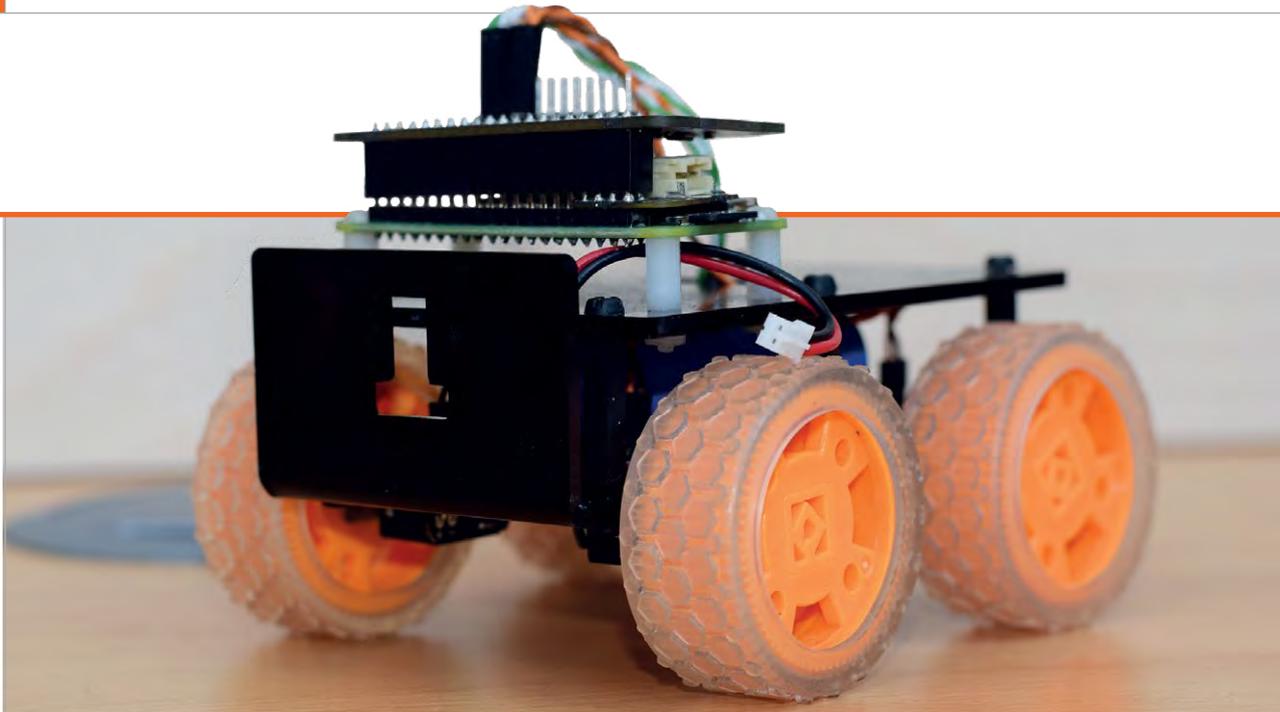
### Last word

**A wonderfully put together piece of kit, the RasPiRobot Rover is a great intro to robotics that can also be used far beyond the beginner level.**



## Maker Says

Tiny 4WD is a small but powerful robot that you build yourself  
Coretec



# TINY 4WD ROBOT KIT

Rob Zwetsloot looks at a tiny robot with a lot of power, in a kit based on a magazine project

If you read issue 51 of *The MagPi* magazine, you'll remember a robot on the cover that you could make yourself. Its designer, Brian Corteil, has taken the lessons from making that robot to build an all-new improved version, which you can buy directly from his Coretec website ([magpi.cc/Tiny4WD](http://magpi.cc/Tiny4WD)) and from [pimoroni.com](http://pimoroni.com).

The Tiny 4WD is the end result of this learning process, and is available as a kit that you build yourself. All you need to supply is a Pi and a way to power it (a mobile USB battery charger will do the trick). The kit is optimised for a Pi Zero, with specific mounting points on the chassis. However,

we tested it using a Raspberry Pi 3 and it worked well.

### Easy build

The build is fairly quick. There are only three parts to the chassis, and one of these isn't necessary in its basic state. The

the Pimoroni website:

[magpi.cc/2pXHCu8](http://magpi.cc/2pXHCu8). Online instructions aren't ideal – it would be nice to have printed instructions in the box – but it's enough to get you started.

We managed to get the robot up and running in just under two

The kit is optimised for a Pi Zero, with specific mounting points on the chassis

box recommends using the build in *The MagPi* ([magpi.cc/51](http://magpi.cc/51)) as your instructions, but they don't quite fit this kit. Luckily, there's a great guide online, linked from

episodes of *Brooklyn Nine-Nine*, so we'd estimate 30–40 minutes' build time, but it may take longer depending on your skills with a soldering iron. You'll need to

## Related

### GOPIG03

A more expensive kit, but it does come with sensors for some proper robotics. It's even easier for newbies as well.



£95 / \$99

[magpi.cc/2vpUPP8](http://magpi.cc/2vpUPP8)

magpi.cc/Tiny4WD

£55 / \$60

solder a 40-pin and a 20-pin header to the Explorer pHAT supplied with the kit, as well as soldering a GPIO header to the Pi Zero. The rest of the setup is easy: install the software on the Pi Zero (running Raspbian) and you're ready to play!

## Flexible control

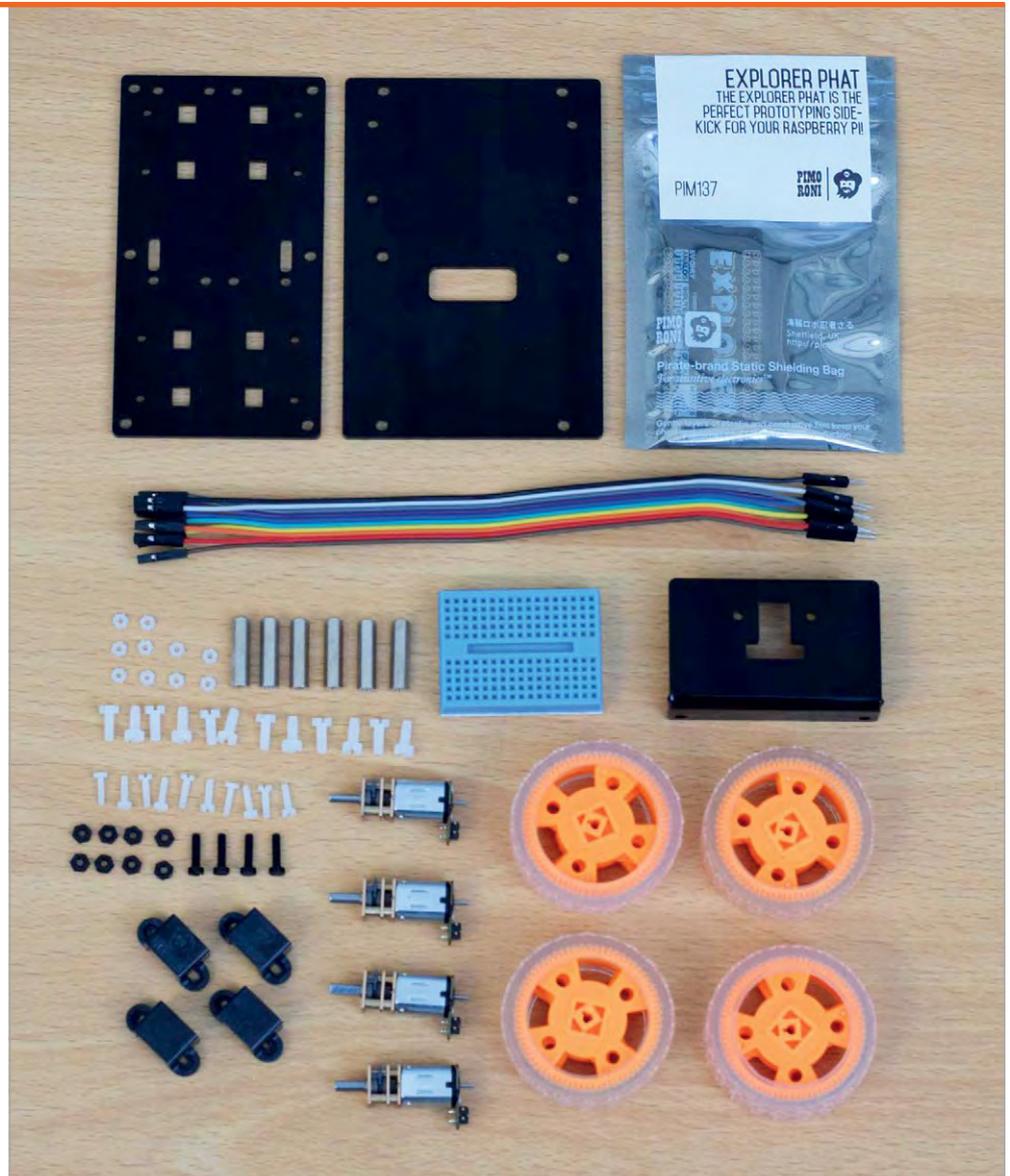
From here you can use the test scripts to control the robot using a USB controller, or start making it more autonomous. The Pimoroni library for the Explorer pHAT is easy to understand – so, with the docs open and the examples to hand, you can easily start creating your own programs.

Each side of the Tiny 4WD is powered by one of the motor outputs on the pHAT. This means that, for example, both left wheels will always run at the same speed and direction as each other. This makes it easy to turn on the spot, and gives it full power when it's moving forwards and backwards.

The Tiny 4WD is easy to customise. There are extra inputs and outputs on the board, so you can start adding additional sensors. There's a camera mount that could be used to attach an ultrasonic sensor, or a Raspberry Pi Camera Module – like the robots in the Formula Pi racing series.

## Power choices

Choosing a power supply can be tricky. The idea is to house your power source between the layers of the chassis. However, the narrow gap, partly blocked by the wires to the motors, limits the size of any power supply. You'll also need to think about power capacity – the motors will be powered from the Pi's GPIO pins, so your Pi will need more power than you might expect.



## Thumbs up

We do like this robot overall. The build quality is great, with a sturdy chassis made from thick acrylic, decent micro motors supplied with the kit, and the inclusion of a great motor controller in the Explorer pHAT. The soldering requirements make it unsuitable for an absolute beginner, but it'd be great as a step up for novices. It could even be used as a base by more advanced robotics users, as you can easily swap out the HAT and use a bigger Pi to add more functionality.

At the time of writing, supplies are limited for this kit, but more units are being made all the time. If you want one, you may just have to be patient. We think it's worth the wait, though.

## Last word

A great little robot kit which could do with some better build instructions. Once it's built, however, you have a lot of options and plenty of ways to make it your own.



### Maker Says

An easy-to-build robot arm kit that's designed to get children learning about technology, engineering, and programming

Mime Industries



# MEARM PI

This fun robot arm will grab the interest of children and adults alike

**T**he original MeArm was reviewed back in issue 33 of *The MagPi* magazine. While we loved the concept of an affordable, open-source robot arm, we did note that a missing piece of the puzzle was the need

for users find their own way to drive the servos from a Raspberry Pi. The new MeArm Pi solves the problem by including a HAT with twin on-board joysticks, so now you have everything you need in one kit.

Not only that, but the arm itself has been completely redesigned, eliminating two-thirds of the screws and using new fixings to make it easier to build. It certainly is a lot of fun to put together, like a cool motorised Meccano set. Step-by-step illustrated instructions are supplied online – two versions, for a standard-size Pi or a Pi Zero. Most of the acrylic pieces simply slot together, secured by a few rubber bands, while three lengths of screws are used for the moving joints. A hex key is supplied for this purpose, but you need to avoid overtightening the screws so that the joints can move freely. The design is well thought out, with strategically placed holes for the servo cables to pass through to keep them tidy. We found the trickiest part was sliding the arm

**Below** The joystick controls are on a HAT that slots easily onto the Raspberry Pi



### Related

#### OWI ROBOTIC ARM EDGE

This robust robot arm comes with a remote control, but can also be controlled from a Pi via any motor driver board.

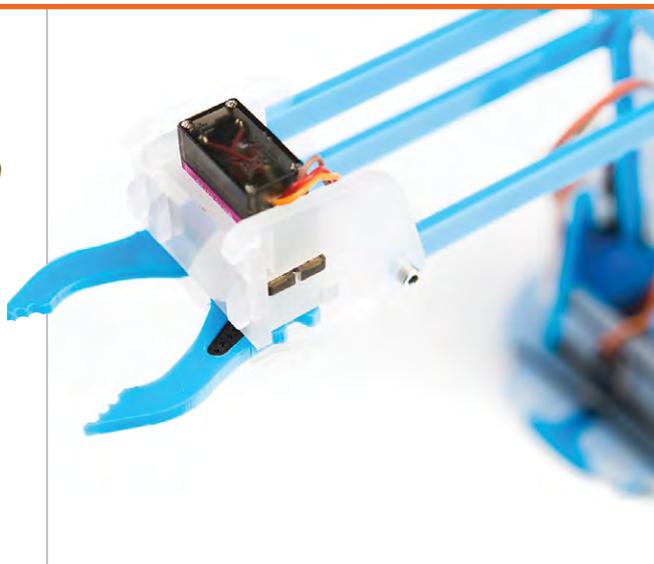


£47 / \$50

magpi.cc/KTUqXa

magpi.cc/2waqp7b

£70 / \$92



into the base, which required prising the two layers apart; we also needed to loosen the base servo horn slightly to enable it to move freely.

The case that houses the joysticks also features cut-outs for the Pi's ports, although there's no easy access to the SD card, so you'll need to download and write the special OS image to it before you enclose the Pi. Based on Raspbian, the OS includes all the required software and creates a wireless access point so that you can connect to and program the MeArm Pi directly from a remote computer or tablet. This means it can be used headless, without the need for a monitor, keyboard, and mouse. There's also the option of connecting it to your WiFi network if you prefer.

### Twist your arm

Once the system has booted up, which takes about 75 seconds on a Pi 3, the arm jerks into life and you can then control it using the twin joysticks. This is a lot of fun – like controlling a mini digger, albeit with a claw. The left stick handles base rotation and lower arm movement, and the

right stick controls the upper arm and claw grip. While the servos are pre-calibrated to be used out of the box, we did encounter an issue with the grip not closing fully at first, but this was easily

functions (and sample code) for moving the arm's servos by degrees for accurate control. Naturally, you can set up loops and conditionals to get the arm to behave in sophisticated ways.

“ Most of the acrylic pieces simply slot together, secured by a few rubber bands ”

rectified by unscrewing and repositioning the servo horn. There is quite a lot of buzzing from the servos, but we found the control was accurate enough for us to pick up small objects, such as cherry tomatoes and socks.

A more interesting and educational way to control the arm is by programming it. While you can do this directly on the Pi, an easier way to get started is by connecting a PC or tablet to the MeArm Pi's local web server, by pointing a browser to **local.headlesspi.org**. This gives you access to four apps, each for a different programming language: Blockly, Snap!, Python, and JavaScript. These include

There's even the potential to add a Camera Module to the end of the arm (using an extra-long ribbon cable) and use OpenCV for image or face recognition to control it – as demonstrated by Mime ([magpi.cc/2ffXB2J](http://magpi.cc/2ffXB2J)).

### Last word

Despite a few teething troubles, we found the MeArm Pi a joy to build and use. Manual control using the joysticks is great fun, but programming it is ultimately more rewarding. The headless setup and local web server make it very easy to get started, offering a choice of four programming languages, although you could still opt to code it directly on the Pi instead.





### Maker Says

Turn your Pi into a Lilliputian ghetto blaster  
**Pimoroni**

# SPEAKER PHAT

A tiny speaker board for adding sound to projects

**P**imoroni's Speaker pHAT may not sound all that phat, but it does offer a neat way to quickly add audio to your projects. This Pi Zero-sized board manages to cram in a mini speaker, I<sup>2</sup>S DAC, and mono amplifier. We particularly appreciate the 1980s boombox-style art on the top of the board, complete with a ten-LED bar graph.

Since the 8Ω speaker has just 2W of power, you won't exactly be rocking the neighbourhood. It sounds more like a tinny transistor radio than a ghetto blaster, offering little in the way of bass (and obviously no stereo), so it's certainly no rival for high-quality audio add-ons from the likes of IQaudio and JustBoom. But then that's not what it's intended for; instead, it should prove suitable for projects that require audio output for notification sounds or speech. Indeed, it would be ideal for the digital whoopee cushion in this book (see page 102), precluding the

need to wire up a separate powered speaker. Other use cases include a retro gaming handheld or – by pairing it with a mini microphone

command. It should then work as the default audio output, although you may need to reboot the Pi first. We tested it out with aplay and

“ It is nevertheless a great way of adding audio to projects ”

– a DIY voice assistant based on Amazon Alexa or the like.

The Speaker pHAT comes in kit form, requiring about 30 minutes to assemble. First, you need to screw the speaker to the rear of the board, using tiny spacer nuts, which is a little fiddly. Then it's time to get the soldering iron out. Soldering the speaker's two terminals to the board contacts with the supplied 24AWG wire is a bit awkward and you need to use tweezers to hold the bare wire as it gets hot! It's then simply a matter of soldering on the 40-pin female header and the pHAT is ready to use.

Installation of the software is achieved with a single Terminal

mplayer from the command line, as well as Sonic Pi and YouTube in the Chromium web browser. There's also a library for manually controlling the LEDs if you want.

### Last word

Not one for audiophiles, the Speaker pHAT offers rudimentary sound quality with little power. It is nevertheless a great way of adding audio to projects, with its tiny Zero-size footprint and no need for a separate power supply. It looks cute too.



### Related

#### MINI PORTABLE SPEAKER

Powered or charged via USB and connecting to the audio out port of a standard Pi (but not Zero), this mini speaker packs a punch.



£12 / \$16

magpi.cc/2diQyHq

magpi.cc/2l2BrFT

£30 / \$32



## Maker Says

👉 A plug-and-play, high-resolution, digital-to-analogue converter for the Raspberry Pi  
**JustBoom**

# JUSTBOOM DAC HAT

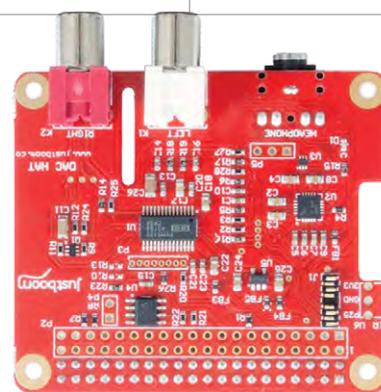
Music sounds better with a HAT on

**T**he JustBoom range of Pi products comprises three main types – DAC, Amp, and Digi – featuring different sets of outputs for different uses. Each is available in full-size HAT and Pi Zero-size pHAT form, ready-assembled. All are based on the I<sup>2</sup>S (inter-IC sound) standard.

Here we're taking a look at the DAC HAT, which delivers its sound via an amplified 3.5 mm headphone socket, or twin RCA outputs connected to a hi-fi system or powered speakers. While the HAT features a 384kHz/32-bit DAC chip (the TI PCM5122), this is limited by the Pi's Linux drivers to 192kHz. Nevertheless, the sound quality proved mightily impressive in our tests. Both CD-quality 16-bit tracks and 24-bit HD audio files were played with crystal clarity, even via a modest Sony mini hi-fi system. Classical music in particular benefits from the enhanced clarity and detail.

The DAC HAT comes pre-assembled, so you just need to mount it securely on the Raspberry Pi using the supplied plastic spacers and bolts. As well as slots for Camera Module and touchscreen cables, this well-designed HAT features a full (unpopulated) GPIO breakout. The packaging includes an optional IR receiver, for you to solder on if you want to use a remote control. There's also the option to fit the HAT in a sleek plastic case (£13 / \$17).

It's then a case of deciding which OS to use. A whole host are supported, including Volumio, OSMC, OpenELEC, Roon, and Max2Play (for which a 30-day free trial is included). Most are easy to set up with the HAT. In Raspbian, it involves commenting out one line in `/boot/config.txt` and adding three, one of which enables playback from Sonic Pi. A quick reboot and it's ready to



use, although we found we needed to boost the digital volume in alsamixer for headphone playback.

One final trick up the DAC HAT's sleeve is the ability to stack a standalone Amp board (£60 / \$79) on top, featuring block terminals so you can connect it directly to passive speakers. You'll need an additional power supply to get full power (30 W RMS) out of it.

## Last word

Delivering audiophile sound at an affordable price, the well-designed DAC HAT is an impressive piece of kit. It offers a huge improvement in quality over the Pi's standard 3.5 mm audio jack, and sounds particularly impressive when playing back lossless formats.



## Related

### IQAUDIO PI-DAC+

Based on the same TI PCM5122 chip, IQAUDIO's HAT delivers similarly impressive sound quality. It can be stacked with a Pi-AMP+ to use passive speakers, too.



£33 / \$43

magpi.cc/2mFj87h

**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](http://magpi.cc/2lNJMKO)

**T**he most expensive of a batch of four 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 I<sup>2</sup>S 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/2oi7d7](http://magpi.cc/2oi7d7)). 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

would it round the bolts at the rear of the radio.

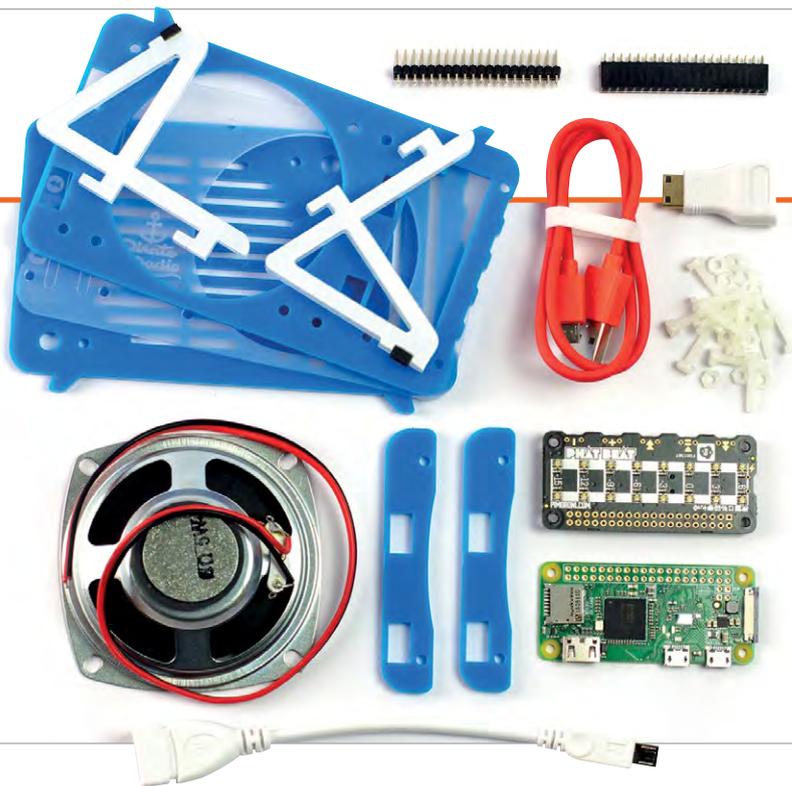
Unless you opt to buy a couple of Pimoroni's ingenious hammer headers separately, you will need to break out the soldering iron to attach the supplied standard male and female headers to the Pi Zero W and pHAT BEAT. The latter then slots onto more bolts at the rear of the radio, with the Zero W mounted 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 / \$44



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

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

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

instant headless route by adding **ssh** and **wpa\_supplicant.conf** (with our router details) files to the microSD card before first boot.

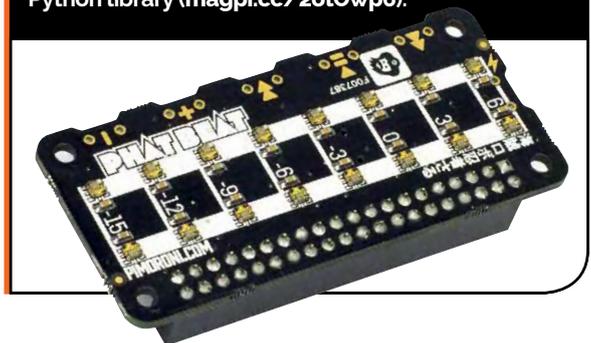
On the software side, Pimoroni has put together guides for three project examples. The first is for an internet radio based on the VLC daemon. As with the other examples, a single command is used to install all the required packages. You can then edit the playlist file to add URLs for your favourite radio stations. With this particular project, everything

dynamically by the pHAT BEAT's super-bright LED VU meter.

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/\$18) 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/2ot0wp6](http://magpi.cc/2ot0wp6)).



## 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.



Maker Says

Pocket-sized fun is the name of this game – our most fun Bonnet ever! Adafuit



# JOY BONNET

Mount a joystick on your Pi Zero to get retro gaming

Ever since its arrival, the tiny Pi Zero has been used for mini retro gaming projects, usually involving inserting one inside an old joystick. Adafuit's Joy Bonnet offers a much simpler, quicker route to pocket-sized retro gaming, however. Coming fully assembled, it simply stacks on top of your Raspberry Pi Zero. Naturally, you'll need to solder (or hammer) a GPIO header to the latter first. A couple of plastic spacers and screws keep the Bonnet firmly in place – which is pretty essential as you'll be pressing its buttons continuously and therefore pushing it down on the Pi. While it's comfortable enough to hold in your hand, you may want to add the bottom of a Pi Zero case for extra comfort – although we had problems keeping the mini-HDMI display

adapter fully inserted through the hole in an official case.

You're then ready to install a retro gaming OS. Adafuit recommends using RetroPie ([retropie.org.uk](http://retropie.org.uk)) or Emulation Station ([emulationstation.org](http://emulationstation.org)) – just flash your microSD card as usual. With wireless set up, you can then SSH in and use a single command to install the Joy Bonnet Python library and software. It takes a little while and offers options to disable overscan (to remove the black border on some monitors) and install a gpio-halt utility for safe shutdown.

Upon rebooting, the OS (we used RetroPie) should sense the Joy Bonnet. We were somewhat surprised to see it recognised as a keyboard: it turns out that the Bonnet's buttons emulate keys such as Z, X, and ENTER. Another interesting point to note is that the

mini joystick is actually analogue, although its directions produce cursor key presses – more on this later. Once you have assigned the various buttons and joystick directions to functions in RetroPie, you're ready to play – naturally, you'll need to have added a few game ROMs in the relevant system folders in RetroPie to make them appear in the on-screen menus.

## Tiny buttons

We started off with a quick game of Galaga '88 running on the MAME arcade emulator. Everything worked fine and the controls were responsive enough. Upon switching to Street Fighter II on SNES, however, we encountered a slight drawback. In place of L and R shoulder buttons, the Joy Bonnet has a couple of tiny buttons labelled 1 and 2, located in the middle of the top

Related

PICADE CONSOLE

The control unit from the Picade arcade cabinet, it can be fitted with a Pi 3 / 3B+, plugs into a TV, and features a robust full-size arcade joystick and buttons.

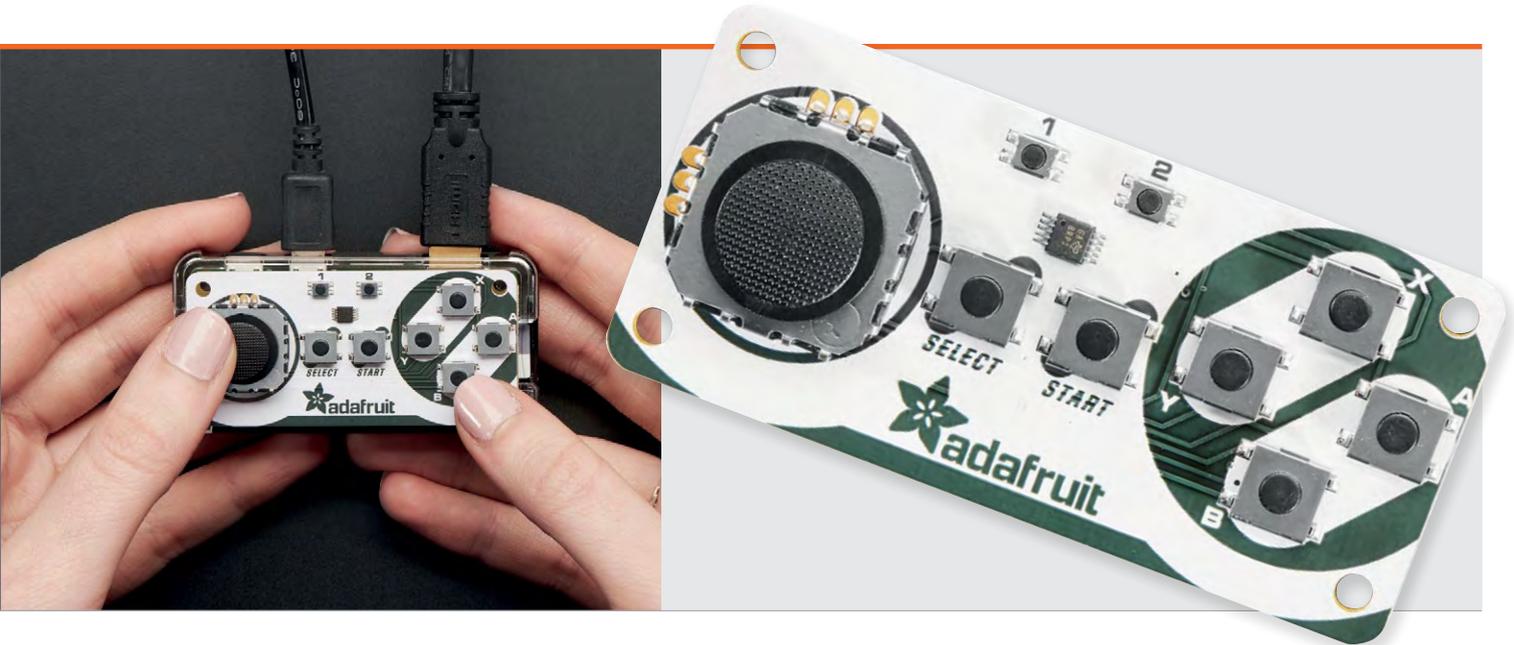


£75 / \$83

[magpi.cc/2pwjhhS](http://magpi.cc/2pwjhhS)

magpi.cc/2qaSfIT

£13 / \$15



of the board – so not that easy to reach in the heat of battle. The four main buttons (X, Y, A, and B) worked well, although they're far smaller than the ones on original

a fix. As mentioned previously, the mini joystick is analogue but emulates digital presses, and we found it extremely difficult to obtain diagonal directions for our

sensitive and were therefore able to obtain the diagonal directions. It's also possible to edit the key presses produced by the buttons in this file, which might come in useful when playing a Spectrum or C64 game with unorthodox keyboard controls.

Note that the Pi Zero is not capable of emulating more powerful consoles such as the N64 and PlayStation. You could always use the Joy Bonnet with a Raspberry Pi 3 / 3B+, although it wouldn't exactly be handheld.

## “ Joy Bonnet offers a much simpler, quicker route to pocket-sized retro gaming ”

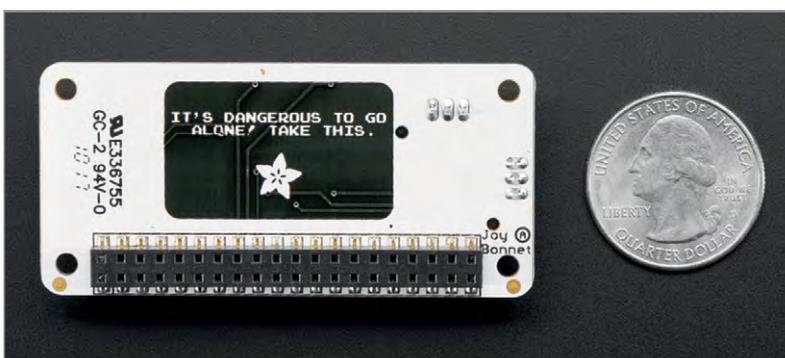
joypads, so not that comfortable. While not quite so critical, the Select and Start buttons are the same small size.

Next, we thought we'd have a blast with classic vertically scrolling shmup, 1942. Here we came across a bugbear that spoils our enjoyment until we figured out

plane in the game. Fortunately, we managed to sort this out by editing the Joy Bonnet's Python library and reducing the positive and negative thresholds for the analogue stick. Setting these at -300 and 300, rather than the original -600 and 600, we found the stick considerably more

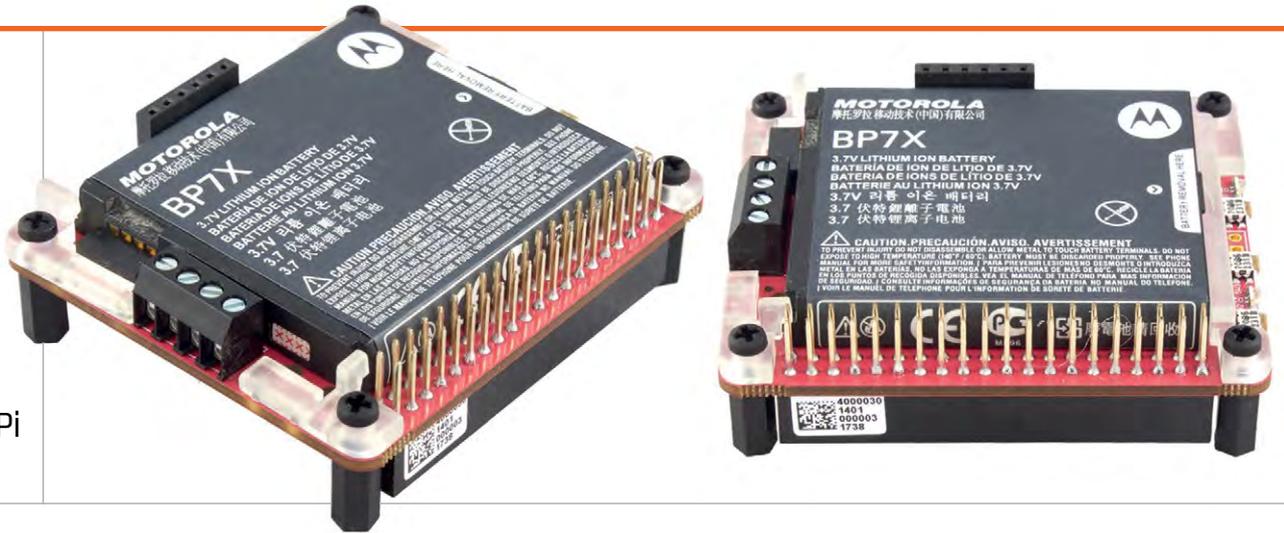
### Last word

**Not as comfortable to hold or responsive as a regular game console joystick, the Joy Bonnet is unlikely to net you many high scores. Still, it is a cute concept that makes it easy to quickly get retro gaming on a Pi Zero: a neat portable solution that you can carry around with you to plug into any TV. You might want to invest in a longer HDMI cable so you don't have to stand quite so close to the screen, though.**



Maker Says

A portable project platform for every Raspberry Pi  
PiSupply



# PIJUICE

Use your Pi anywhere with this smart portable power solution

**Y**ou may recall the PiJuice Kickstarter campaign if you have a good memory – it was back in March 2015 and proved extremely popular, achieving a funding level of over 1200%. So, why the long delay to get it into production? Well, it seems it was due to a combination of technical, manufacturing, and business issues,

but here it is at last, so let's find out if it has been worth the wait.

The PiJuice comes preloaded with a Motorola BP7X 1820 mAh phone battery, which can easily be lifted out and replaced with an alternative if needed. The board has a pre-soldered header so it fits snugly onto the Raspberry Pi's GPIO pins, but extends them to its

own full set of GPIO pins above, so you can still plug in another HAT or add-on – the PiJuice only uses I<sup>2</sup>C pins. Four stand-offs keep it sturdy atop the Pi. It's a much neater solution than most portable power methods – no messy wiring here.

Not only is it neater, but it's far smarter, thanks to its STM32-F0 microcontroller chip, real-time clock, and Pi software. It can therefore offer a range of advanced power management features – akin to those of the Witty Pi 2 – that make it more useful than simply plugging your Pi into a USB power bank. It can also be used to provide an uninterruptable power supply.

After downloading the software (with `sudo apt-get pijuice`), a battery status icon then appears in the task bar of the Raspbian desktop; hover over it to see the current charge percentage, or right-click to access a plethora of configuration settings.

Related

LIPO SHIM

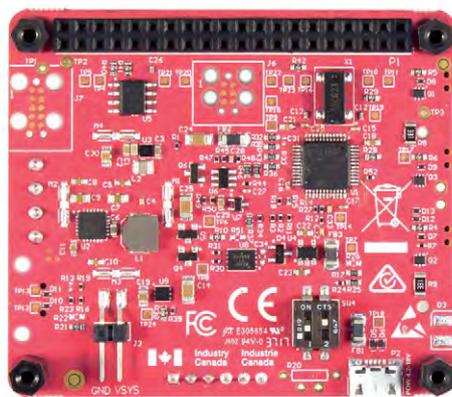
Formerly known as the Zero LiPo, this little shim can be soldered to the Pi's GPIO pins and hooked up to a LiPo battery back and charger (not supplied).



£10 / \$11

magpi.cc/2xyhCYH





## SOLAR POWER

PiSupply also sells a Solar kit (£90/\$117) which comprises a PiJuice and a specially made 6W solar panel to charge it out in the field. The latter folds out of a soft case, with a pull-out flap that can be used to hold its twin mini panels at an angle. You'll probably need bright direct sunlight to provide a good level of current (about 1 amp at most) – ours was weak when we tried it out on a grey autumn day – but it's a nice option. We were also supplied with a 40W version (with six mini panels) which should provide a greater current output and, with twin USB ports and a barrel jack, could be used to charge several devices at once.

## What's my level?

One thing we soon noticed was a disparity in the displayed battery level when charging – via the GPIO pins of the mains-connected Pi or the PiJuice's own micro USB socket. As soon as we unplugged the power, the status level dropped rapidly by around 20%. According to PiSupply, this is a known quirk of the protection circuitry in Li-ion batteries and the specific 'fuel gauge' IC used by the PiJuice, and the actual battery discharge rate is fairly linear. So, a bit confusing but nothing to worry about. In addition, an RGB status LED gives a rough



feature in the GUI config options, to turn a shutdown Pi back on at a specific time or even charge level.

Another interesting option is the watchdog timer that monitors a software 'heartbeat' and, if it's

updating the firmware, and choosing the functions of the board's two LEDs. Lastly, you can set the functions for the three tiny push buttons on the side of the PiJuice – individually for press/release, single press, double press, and two long-presses (with customisable time parameters). By default, the SW1 button can be long-pressed for ten seconds to safely shut down the Pi, and then pressed to restart it – another greatly appreciated feature.

**“ A much neater solution than most portable power methods – no messy wiring here ”**

guide to the battery level, flashing blue during charging.

So, how long does the battery last? We performed a simple uptime test with a Python script that periodically logged how long the Pi had been running. On an idling Pi 3, it averaged around four hours. A little shorter than we'd hoped for, but on a Pi Zero or A+ you should be able to achieve near double that. To extend battery life, you could also make use of the wake-up alarm

not heard for a certain period, automatically resets the Pi – ideal for when you can't physically reach it to do a hard reset following a crash. There's also an array of system event scenarios for which you can trigger events, including custom functions – just add the path to your own script under User Scripts.

Further options include changing the battery profile (to one of several presets or a custom configuration),

## Last word

**As well as an all-in-one portable power solution that's far neater than the alternatives, the PiJuice offers advanced power management features with an impressive number of settings and custom options for maximum versatility. Three user-definable push buttons and a built-in real-time clock are a major bonus.**



## Maker Says

Level up your Raspberry Pi Monster Joysticks



# DELUXE ARCADE CONTROLLER KIT

**Rob Zwetsloot** builds a mini arcade machine with this all-in-one controller kit from Monster Joysticks

**B**ack in issue 63 of *The MagPi* magazine ([magpi.cc/63](http://magpi.cc/63)) we took you through a comprehensive arcade machine build, including a complete wooden build of the cabinet itself. While it's certainly impressive, not everyone has the space, time, or money for one. This is where awesome little kits like this one from Monster Joysticks come in.

You've probably seen this type of kit before – it's an all-in-one arcade joystick and case for your Raspberry Pi that turns it into a small and portable arcade machine. Just hook it up to the nearest television and you're ready for some Elevator Action. It's like the plug-and-play mini Mega Drive you got for Christmas a few years ago, or the more recent SNES Classic Mini.

Unlike the stocking filler plug-and-play consoles, this kit requires you to build your gaming system and supply the Raspberry Pi that powers it. Construction is very simple, though: there are six acrylic panels for each side of the box and only eight screws required to fasten them all together.

### Quality components

The kit comes with nine genuine Sanwa arcade buttons and a Sanwa joystick, which just simply click into the acrylic panels as you build them.

To wire up the buttons and joystick, a little add-on board is provided with colour-coded wires. They can be a little tricky to properly attach to the connections as the connectors themselves are a bit tight,

but you don't have to worry too much about wires getting tangled up. You may also need to push down the top panel a bit due to resistance of all the wires, but otherwise it all fits fairly neatly inside. You can find the full build instructions on the Monster Joysticks website: [magpi.cc/2i3iQp8](http://magpi.cc/2i3iQp8).

The build took us just shy of three episodes of *The Simpsons*, so make sure you set aside about an hour for the job. Our only real complaint about the build is that, while all the ports and even SD card slot are readily accessible, the Raspberry Pi can only be removed by taking the case apart. It will only take a couple of minutes to remove it, but we'd have preferred it to be a little easier.

## Related

### PICADEE CONSOLE

The product that launched Pimoroni is very similar to the Monster Joysticks version, although it has a slightly nicer case.



£75 / \$83

[magpi.cc/2pwjhhS](http://magpi.cc/2pwjhhS)

magpi.cc/YTnKkS

£100 / \$132



The final part of the build involves attaching little rubber feet to the bottom – very welcome, as the case had been slipping a bit on the glass table it had been built on.

The stick feels solid and has a decent weight to it thanks to the included components, so you feel pretty safe giving the buttons and

customised: the version that comes with the kit has square four-way gates, but they can be upgraded to an octagonal eight-way gate, or any other gate style if you prefer.

### Quick configuration

Software customisation for RetroPie is also very simple.

as well as any extra controllers you've plugged into the USB ports. Co-op Contra, anyone?

This kit is a great, solid package and it looks good as well. We recommend investing in some nice, long HDMI and USB cables to power the box and don't be afraid to put some stickers or a little custom decal onto the case as well.

All in all, the Deluxe Arcade Controller Kit may just be the perfect gift for the retro games fan in your household.

“ The stick feels solid and has a decent weight to it thanks to the included components ”

joystick a proper workout. The included Sanwa components are quite important as not only are they high-quality and can survive a bit of classic Street Fighter button mashing/frame-perfect combo-timing, they're also quite customisable. For instance, if you don't fancy the button colour scheme, you can always swap them out. The joystick itself can also be

With a custom add-on board to connect the controls to the Pi over GPIO, we initially feared we'd have to download custom scripts for the job. Not so, though, and while you do need to go into the RetroPie configuration menu and install an extra driver (snesdev), it's all quick and included in the RetroPie archive. Once that's done, you can configure the stick controls,

### Last word

A great little kit. It's a fun build but also a good quality product to use. We'd prefer the Raspberry Pi to be a bit more accessible, but otherwise the high customisability is a big plus.



# ENJOYING THIS BOOK?

FIND MUCH MORE IN *The MagPi*  
THE OFFICIAL RASPBERRY PI MAGAZINE

SAVE  
UP TO **35%**



## Subscriber Benefits

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

## Rolling Monthly Subscription

- ▶ **Low Monthly Cost** (from £5)
- ▶ **Cancel at any time**  
Leave any time applies to Rolling Subscription only
- ▶ **Free delivery to your door**
- ▶ **Available worldwide**

## Subscribe for a Year

£55 (UK)      £90 (USA)  
£80 (EU)      £95 (Rest of World)  
Free Pi Zero offer applies to 12-Month Subscription only

 [magpi.cc/subscribe](https://magpi.cc/subscribe)

JOIN FOR 12 MONTHS AND GET A

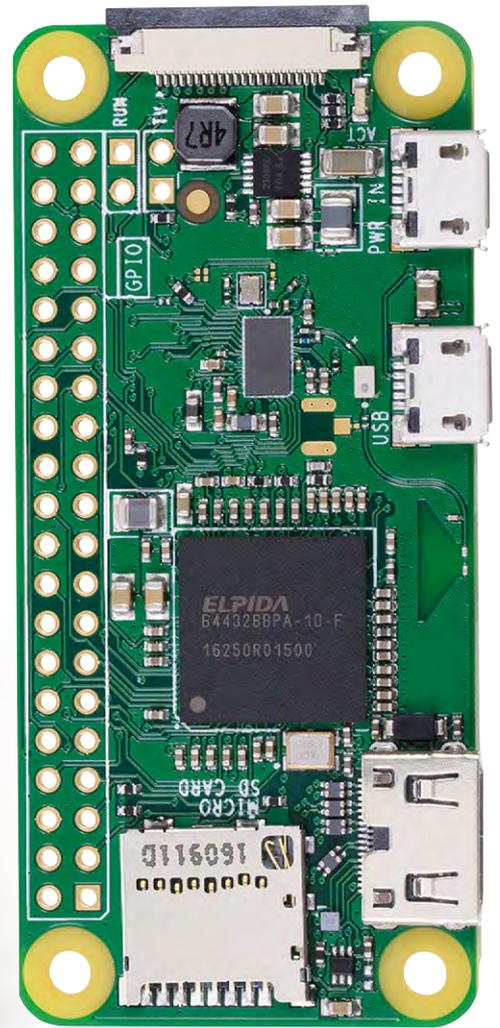
# FREE Pi Zero W Starter Kit\*

WITH YOUR SUBSCRIPTION

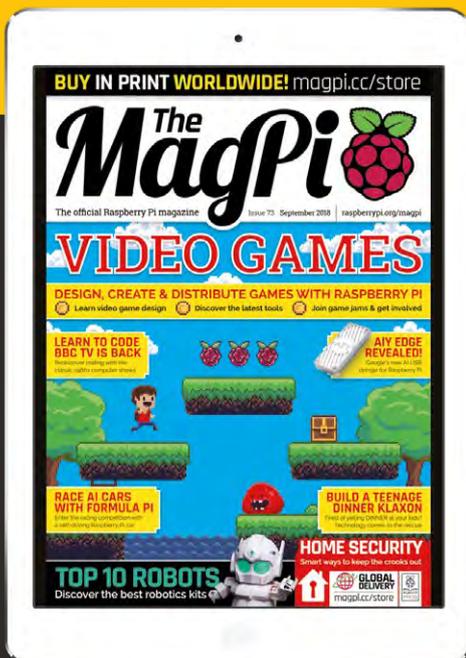
Subscribe in print for 12 months today and you'll receive:

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

WORTH  
**£20**



\*This is a limited time offer and the publisher reserves the right to cancel it at any time



[magpi.cc/subscribe](https://magpi.cc/subscribe)



**SUBSCRIBE**  
on app stores

From **£2.29**



# THE *Official* RASPBERRY PI PROJECTS BOOK

## VOLUME 4

The Official Raspberry Pi projects book returns with inspirational projects, detailed step-by-step guides, and product reviews based around the phenomenon that is the Raspberry Pi. See why educators and makers adore the credit card-sized computer that can be used to make robots, retro games consoles, and even art.

### 200 PAGES OF RASPBERRY PI

- Start your digital making adventure with Raspberry Pi
- Be inspired by amazing community projects
- Make your own creations with easy-to-follow guides
- Learn about the very best accessories for your Pi



[raspberrypi.org/magpi](http://raspberrypi.org/magpi)

