YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

The official Raspberry Pi magazine

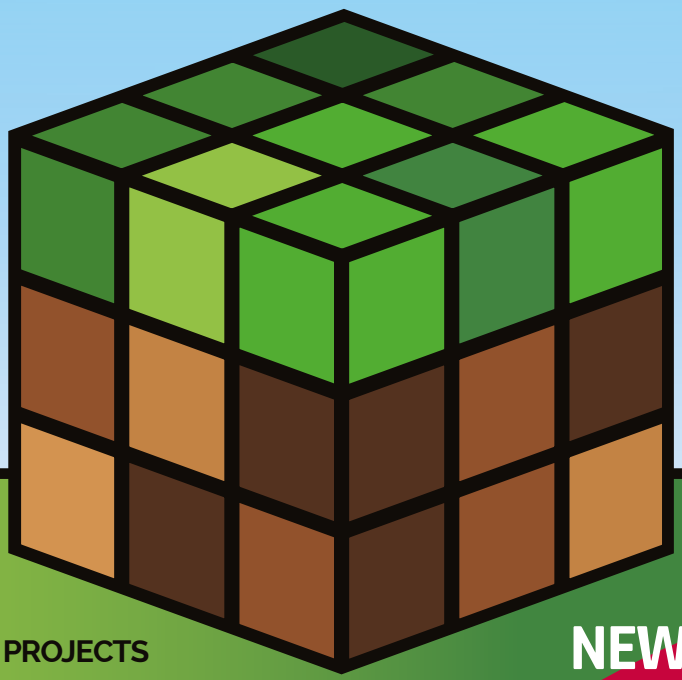
Issue 41 January 2016

raspberrypi.org/magpi

MINECRAFT MASHUPS

Amazing Pi projects that bring Minecraft's virtual world to life

Make music ❀ Control electronics ❀ Simulate natural disasters ❀ and much more



#PIZERO PROJECTS

More amazing creations with the $5 Raspberry Pi!

MAKE MUSIC WITH SONIC PI

...with its creator, Sam Aaron

SENSE HAT SPECIAL

10 pages of awesome games & guides

PI IN SPACE!

It's a giant leap for Raspberry Pi kind...

Also inside:

- MORE OF YOUR AMAZING PI PROJECTS
- ALL YOUR #PIZERO QUESTIONS ANSWERED
- BUILD A RETRO CONSOLE IN A CONTROLLER
- GET STARTED WITH THE CAMJAM ROBOT KIT

NEW YEAR, NEW YOU!

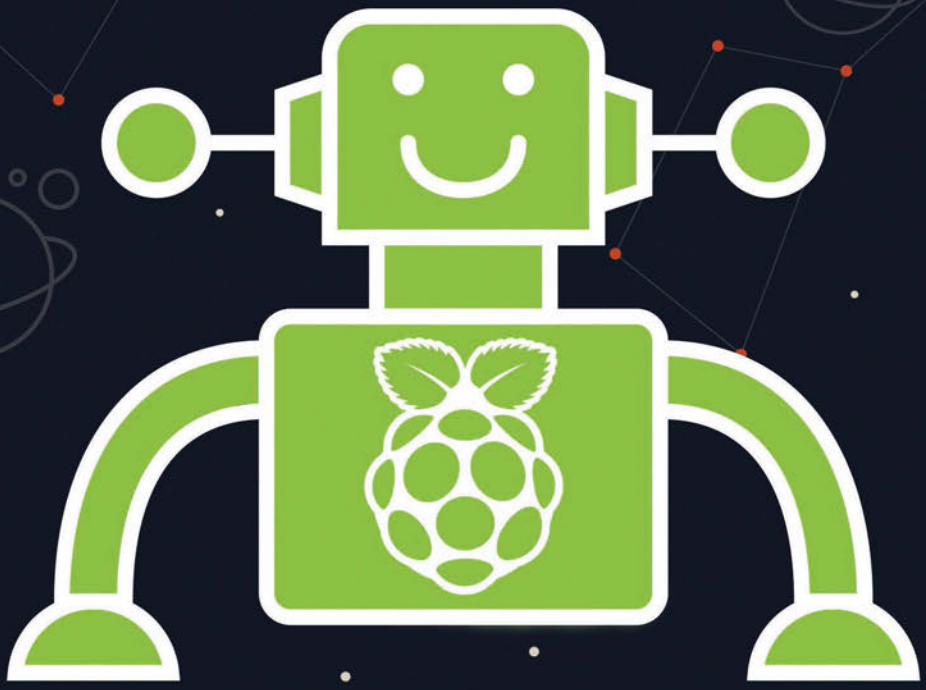
New Year's resolutions made possible with your Raspberry Pi

THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY

Issue 41 • Jan 2016 • £5.99

9 772051 998001 01

VNC SDK now available for RASPBERRY PI!



With VNC SDK you can connect VNC Viewers with VNC Servers easily over the Internet. Using VNC Cloud, no complicated network configuration at either end is required!

What will you create with VNC SDK?
Check it out here: https://developer.realvnc.com/

REAL VNC

VNC DEVELOPER

Getting connected: www.realvnc.com/products/vnc/raspberrypi/
For more information contact vncdeveloper@realvnc.com

Expand your Pi

Stackable expansion boards for the Raspberry Pi

Serial Pi Plus

RS232 serial communication board. Control your Raspberry Pi over RS232 or connect to external serial accessories.

Breakout Pi Plus

The Breakout Pi Plus is a useful and versatile prototyping expansion board for the Raspberry Pi

ADC Differential Pi

8 channel 18 bit analogue to digital converter. I²C address selection allows you to add up to 32 analogue inputs to your Raspberry Pi.

IO Pi Plus

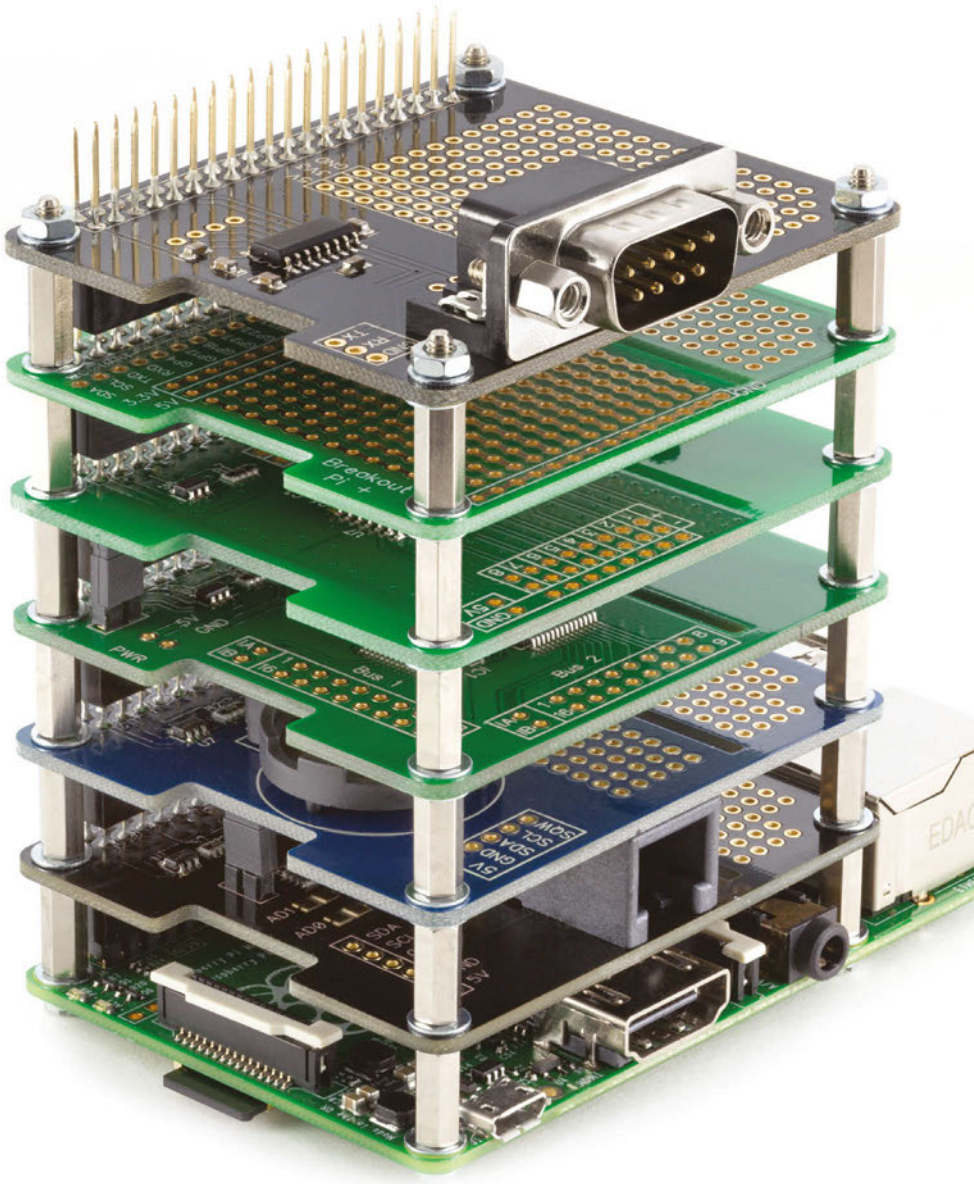
32 digital 5V inputs or outputs. I²C address selection allows you to stack up to 4 IO Pi Plus boards on your Raspberry Pi giving you 128 digital inputs or outputs.

RTC Pi Plus

Real-time clock with battery backup and 5V I²C level converter for adding external 5V I²C devices to your Raspberry Pi.

1 Wire Pi Plus

1-Wire® to I²C host interface with ESD protection diode and I²C address selection.



We also stock a wide range of expansion boards for the original Raspberry Pi models A and B

AB electronics UK

www.abelectronics.co.uk

WELCOME TO THE OFFICIAL PI MAGAZINE!

We're still reeling. We had an inkling last issue was going to be popular, but nothing prepared us for the overwhelming response it received. #40 stock was snapped up in minutes online and was sold out in most stores around the UK within hours. We had such a massive influx of subscribers, we even had to reprint the magazine just to keep up. At the time of writing, we've still got a few subscription copies left, so if you're yet to grab one, see pages 26–27 for details. You can elect to receive #40 as your first issue, but only while stocks last.

Which now brings us to this issue. The difficult second album. How do you top the first magazine issue in history to give away a free computer on its cover? It turns out you can't. So we're just going to carry on where we left off. On #40's cover we said you could use #PiZero to learn to code while playing *Minecraft*, and this month we're proving it.

In our Minecraft Mashups cover feature (starting on page 16) we've cherry-picked some of our favourite recent ideas and experiments, some of which cross the divide between *Minecraft*'s virtual world and our own. Make Steve move with a wave of your hand, create amazing music and visual effects, and even have your virtual avatar take control of your electronic devices.

To celebrate the successful arrival of ESA astronaut Tim Peake to the International Space Station, we've also got another ten pages of amazing projects and games you can make with the Sense HAT, the same hardware in the Astro Pi flight units Tim will be using over his next six months on the ISS. You can keep up with everything happening in orbit at **astro-pi.org**.

Russell Barnes

THIS MONTH:

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

The MagPi

Available on the App Store

Get it on Google play

CC BY NC SA

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

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Mike Kay**
Font used & inspired by: **PurePixel**

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

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

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

CONTRIBUTORS
Sam Aaron, Boris Adryan, Mike Cook, CrazySqueak, Tony Goodhew, Gareth Halfacree, Lucy Hattersley, Dave Honess, Jasper, Phil King, Simon Monk, Matt Richardson, Lucy Rogers & Richard Smedley

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

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

Contents

Issue 41 January 2016 raspberrypi.org/magpi

TUTORIALS

IN THE NEWS

#PIZERO IN PICTURES

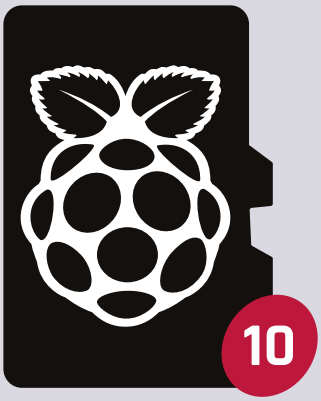
We've been collecting the #PiZero pictures you've sent us on Twitter. Here are some of our favourites

COVER FEATURE

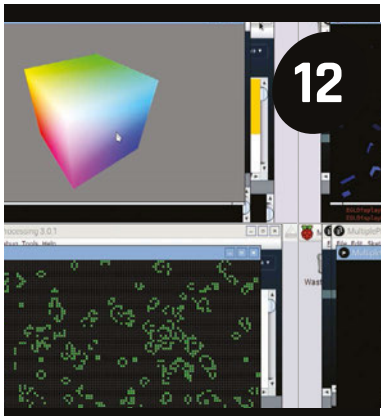
MINECRAFT MASHUPS

At a loose end? Grab your toolkit and give one of these quick-fire projects a try

RASPBIAN: WHAT'S NEW?



Raspberry Pi's Simon Long talks us through the latest upgrades and updates to the Raspbian image

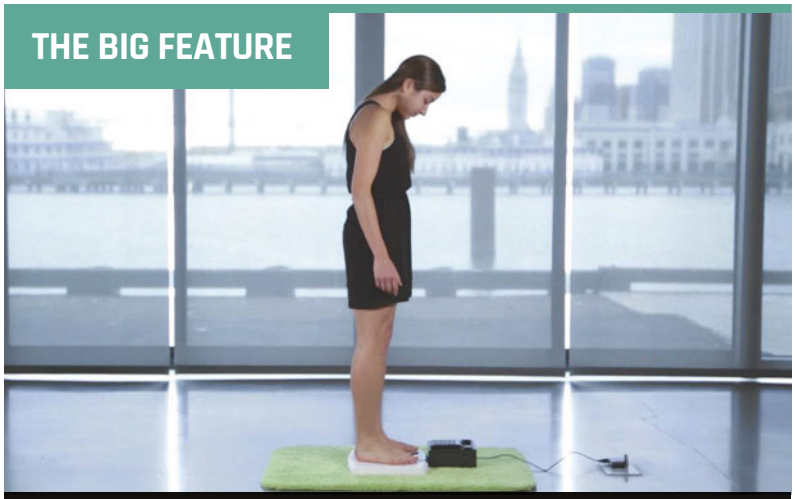


GET PROCESSING WITH YOUR PI

The popular programming language Processing is now supported in Raspbian. We speak to its co-founder to learn more

Contents

THE BIG FEATURE



NEW YEAR, NEW YOU!

10 amazing Raspberry Pi projects that could help you make your New Year's resolutions a reality

72

WIN A BBC-MICRO INSPIRED FUZE T2-SE WORTH £230!

95

In association with:

FUZE Teaching kids to code

BLAST OFF!

Tim Peake and Astro Pi are in space! Here's how the historic occasion played out...

14

YOUR PROJECTS

28

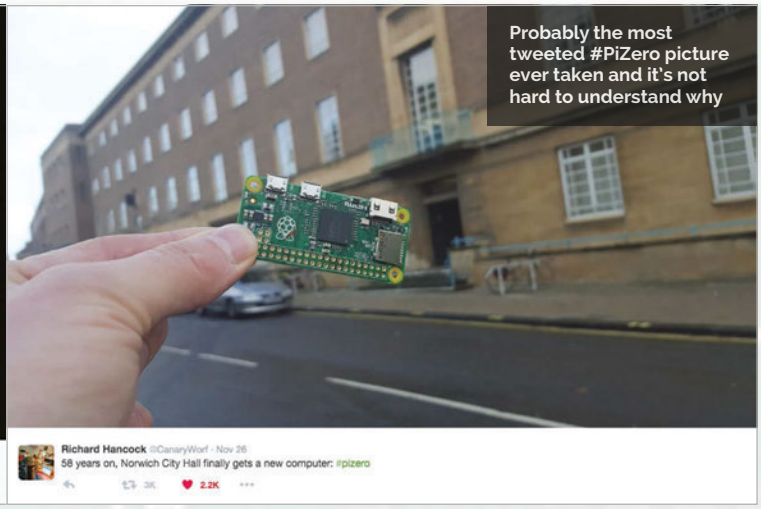
RASPBERRY PI ZERO PROJECTS

A collection of some of the best #PiZero projects we've found so far

REGULARS

COMMUNITY

REVIEWS



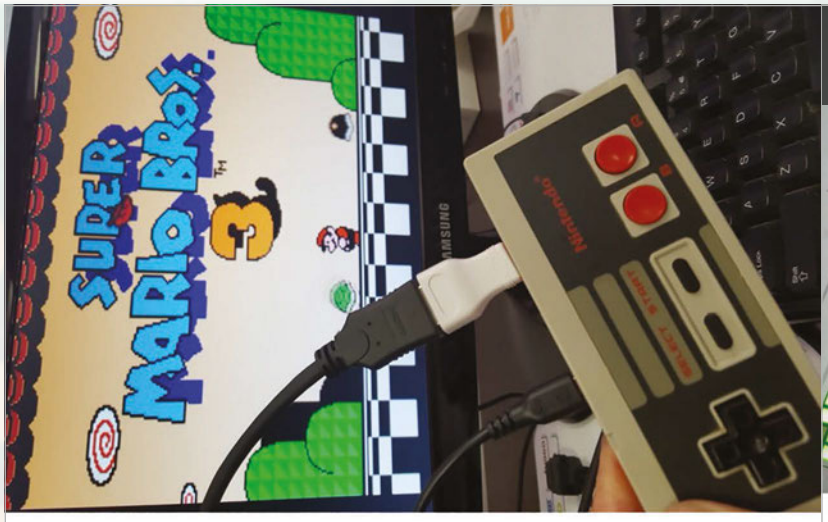
Probably the most tweeted #PiZero picture ever taken and it's not hard to understand why

#PIZERO IN PICTURES

You've been buying your special, #PiZero-laden Christmas issues in your thousands. Here's just some of our favourite pictures from Twitter...

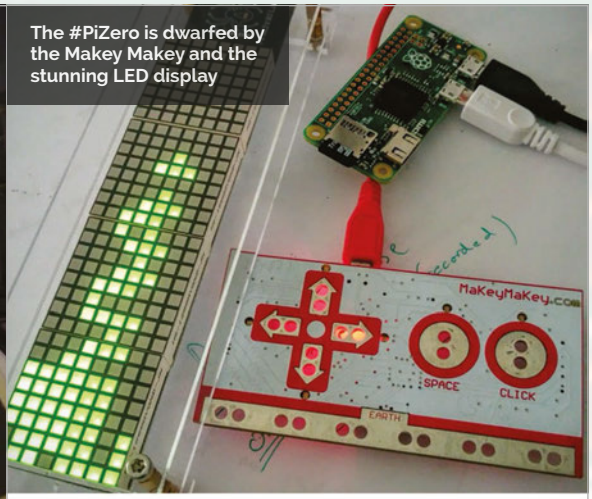
Where Eben Upton's story began, right up to the pinnacle of home computing...

Mic Wall's inspired Lego shot captures the size of the #PiZero perfectly



Tom Van den Bon @geekc0der · Dec 11
#RaspberryPi #pizero hacked into a nes controller... lots of fun will be had this weekend ;) #retropie #3dprinting

The #PiZero is dwarfed by the Makey Makey and the stunning LED display



Chetan Padia @chetbox · Dec 14
Kicking back retro style 😎
github.com/chetbox/minish...
@Raspberry_Pi @TheJoyLabz @ArachnidLabs

hyuecn @hyuecn · Dec 13
@TheMagP1 Found a Pi Zero when tucking into a bag of (micro) chips...

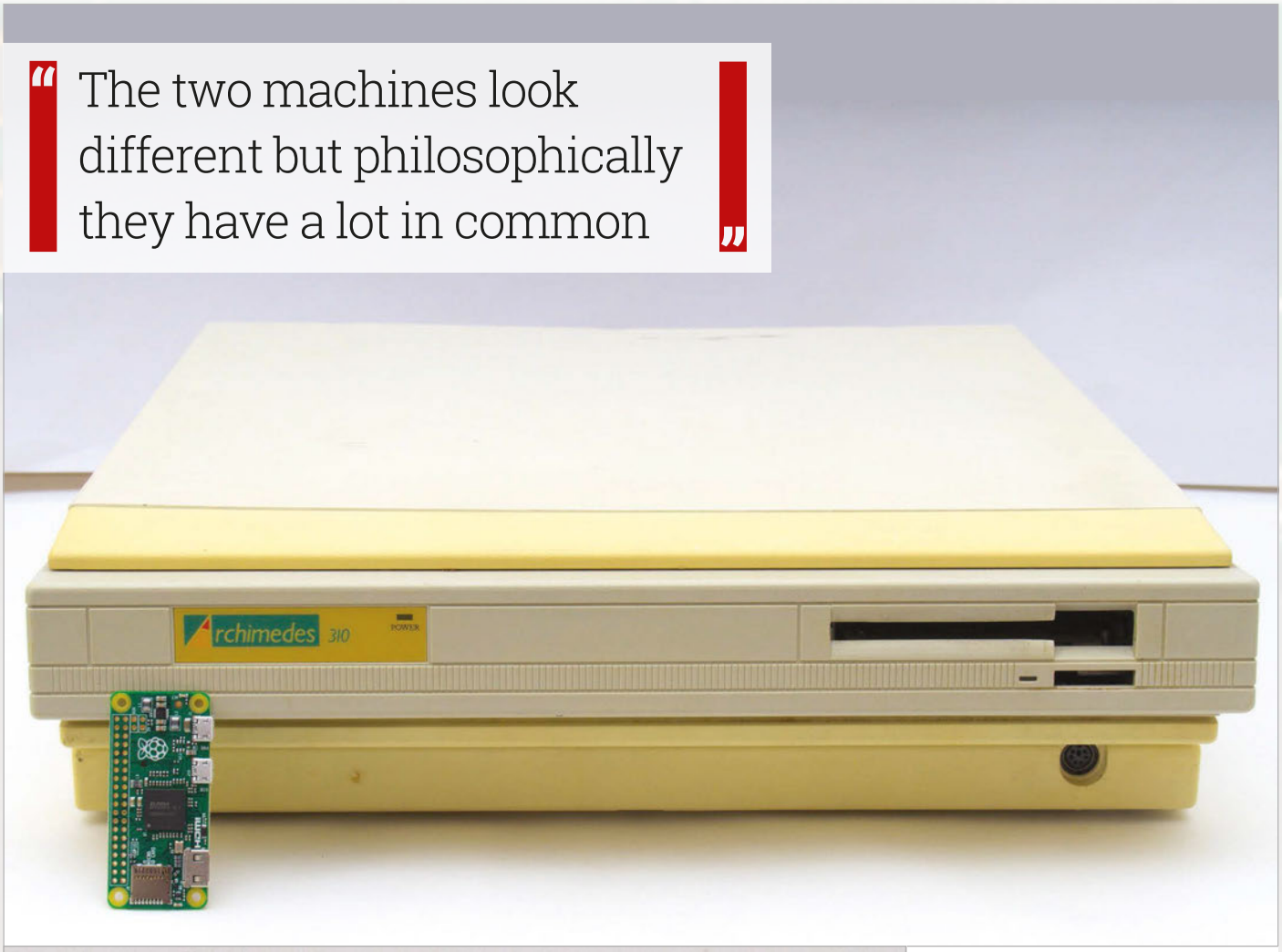
Claire Kelly @MissCKelly19 · Dec 12
Day 12 - Tech Elf
Learning all about #PiZero in @TheMagP1 #TechEdScot

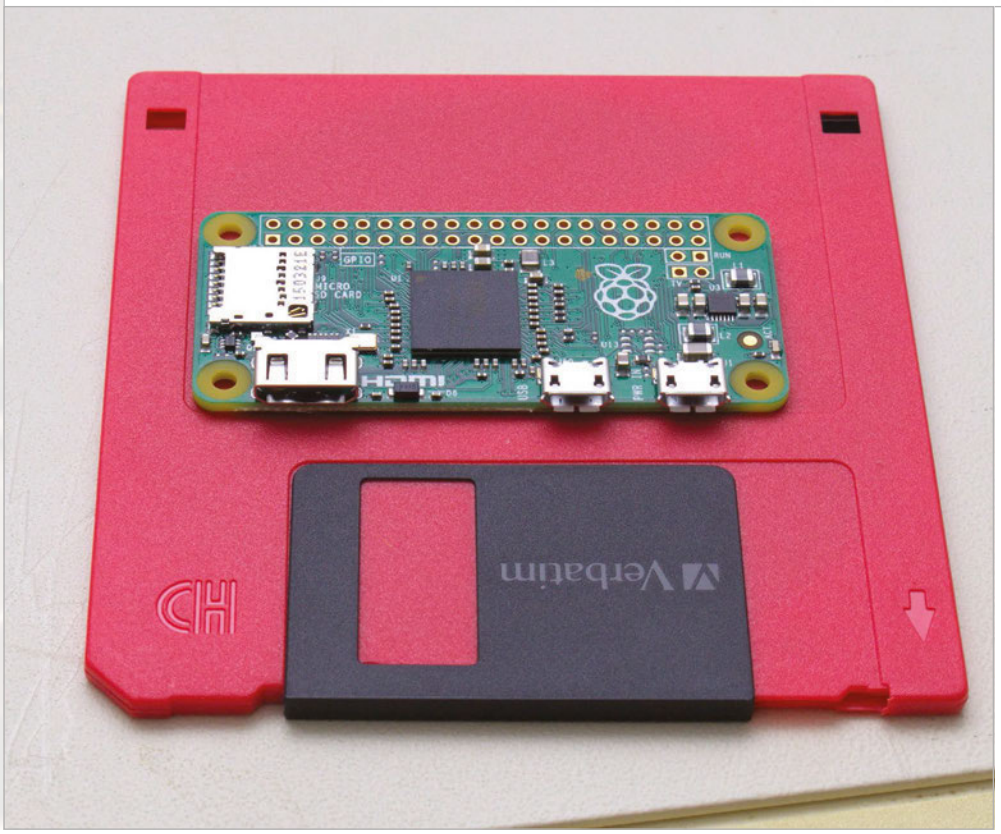
Carles Fernandez @carlestyle · Dec 12
Just received my @TheMagP1 issue 40 with the #pizero. Surprised to see how small it is compared to a #RaspberryPi

OUR FIRST READER OF #40!

Our very first reader was one Tom Kirby-Green. He managed to snag the first issue off the shelves the night before (somehow)! Here's his wonderful daughter handling the #Pi Zero for the first time!

> "The two machines look different but philosophically they have a lot in common"





Friend of *The MagPi* Michael Reed has taken some fantastic shots with his Acorn Archimedes. Here's what he said: "Released in 1987, the A310 was one of the fastest computers that money could buy, and it was the first computer system that made use of an ARM processor. Almost 30 years later, the Pi Zero, one of the newest ARM-based computers, is about a hundred times faster. The two machines look different but philosophically they have a lot in common, in that they're both begging to be prodded, experimented with, and learned from. Basically, with a Pi, if you have an idea that involves programming, electronics, or any crazy combination of the two, even the sky's not the limit – just ask the crew of the International Space Station!"

FUZE

Teaching kids to code

"The FUZE is what the Raspberry Pi was designed for"

THE NATIONAL MUSEUM OF COMPUTING

The FUZE T2 Case Special Edition

- Protect your Pi from physical & static damage
- UK keyboard* & 4 Extra USB ports
- FUZE I/O Board with 40 way GPIO pass-through
- Clearly labelled input output ports
- 2 Amp power supply and on/off switch!
- Adds analogue ports, 4 in & 1 out
- 840 pin solderless breadboard (black)

Reviewed model FUZE T2-R

Computer *act!ve*

BUY IT AWARD

PC PRO Recommended

...makes the Pi more accessible than ever

micro mart

EDITOR'S CHOICE

... it's certainly the **best** we've **ever** tested

£99.99

FUZE T2-C-SE

Compatible with Raspberry Pi V1 & V2

BONUS

Includes 8GB SD pre-configured with FUZE BASIC

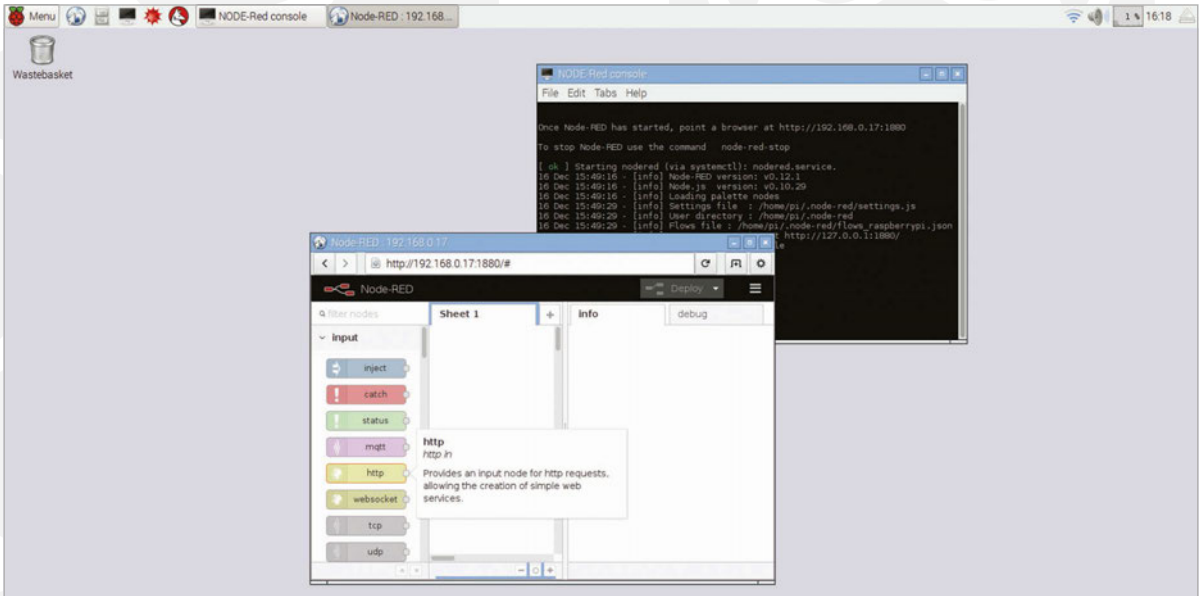
* USA & German keyboard layouts are also available. Prices include VAT but not shipping - see fuze.co.uk for details

FUZE Technology Ltd
+44 (0) 1844 239 432 - contact@fuze.co.uk

Available from www.fuze.co.uk

RASPBIAN: WHAT'S NEW?

Raspberry Pi recently released an update to the popular Raspbian image. **Simon Long** brings us up to date on his latest handiwork...



Node-RED

IBM's Node-RED Internet Of Things application is now included in Raspbian Jessie. This allows you to rapidly create IoT applications by connecting blocks in a graphical editor. We're going to run a full blog post on this in the near future, but for now we'll show you how to get started. Run the Node-RED application from **Programming** in the main menu, and then use the web browser to access port 1880 at your Pi's own address to see the editor.

To install Node-RED on an existing Raspbian image, use Apt:

```
sudo apt-get install nodered
```

New package manager

Under **Preferences** in the Raspbian desktop main menu, you will also now find an option for **Add/Remove Software**. This launches a modified version of the Gnome Packages application, which allows you to add and remove software on your Pi.

The window shows a list of categories on the left, and there's a search box at the top – type in a term to look for a particular package or feature. Software already installed appears in the list with a tick in the box next to it; software which is not installed appears unticked – simply click on the box to tick or untick it, and hit Apply to install or uninstall.

One limitation is that you can only either install or uninstall in one 'Apply' – once a package is selected for installation, you can't select other packages for removal at the same time (but you can choose to install additional packages). Similarly, once a package is selected for removal, you can't select other packages for installation at the same time.

You will be asked for your password to confirm any installation or removal operations.

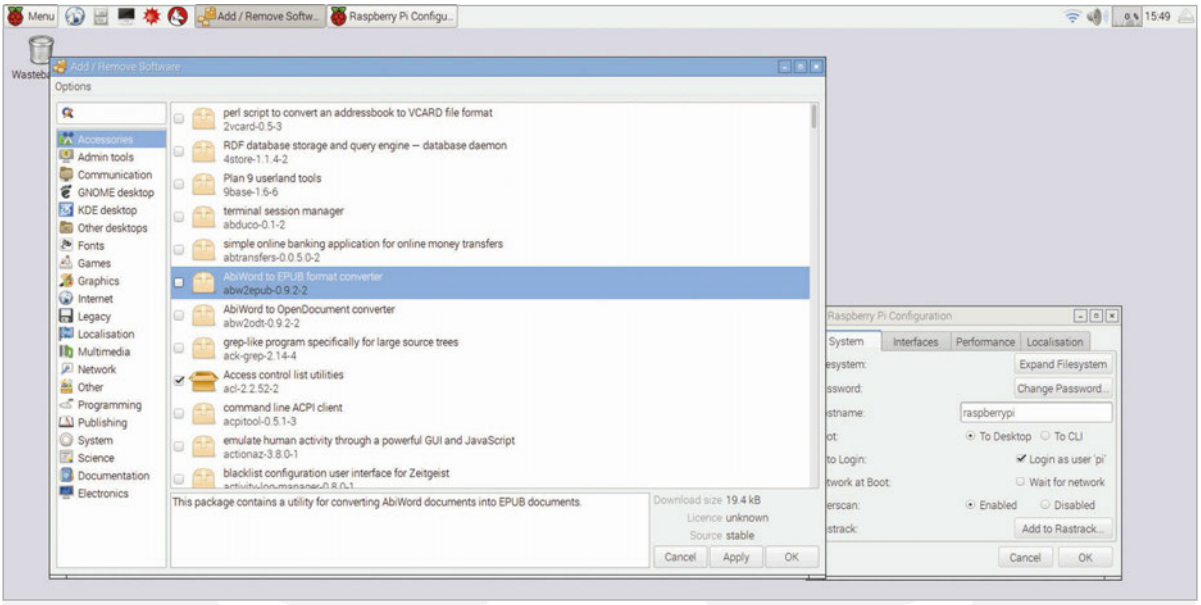
To install the package manager on an existing Raspbian image, use Apt:

```
sudo apt-get install pi-package
```

GPIO ZERO

The new GPIO Zero libraries are included, which vastly simplify access to GPIO pins from Python scripts. You can learn more about GPIO Zero on Ben Nuttall's blog at: **magpi.cc/1Zdxics**. To install GPIO Zero on an existing image, use Apt:

```
sudo apt-get install python-gpiozero python3-gpiozero
```



UPDATED SCRATCH

Scratch has been updated. Changes include support for MIDI general instruments, improved support for the PiFace interface board, and fixes for the Scratch mesh server, along with some bug fixes and minor tweaks, plus updated Japanese translation files.

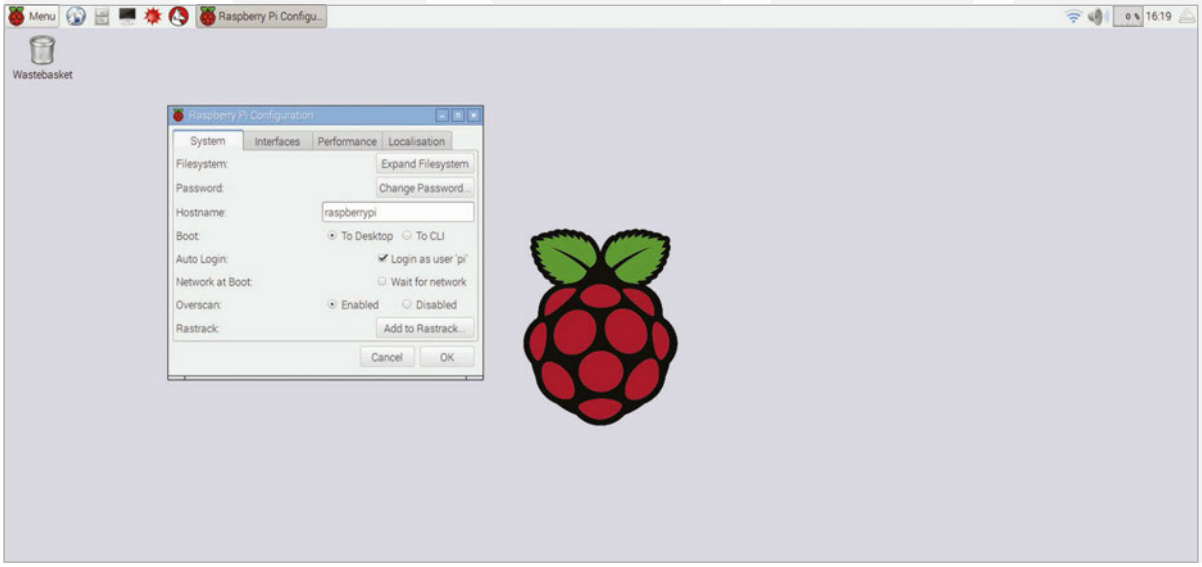
Updated Raspberry Pi Configuration application

This adds the option to 'wait for network' at boot. By default, the Jessie boot process runs as quickly as possible, but this can mean that the boot completes before a network connection is up and running, and this can cause problems in some applications that run at boot. By selecting this option, boot will not complete until a network connection is established, but at the cost of making bootup take longer – this is disabled by default, so if you are having problems with applications which need a network connection, try enabling it. This option has also been added to the raspi-config command-line application.

There are various other small bug fixes, tweaks, and performance improvements, but no significant UI changes this time around – it should all look and feel the way the previous version did. To update your existing image, run:

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

Bear in mind, though, that this won't add the new packages listed around these pages by default – use the individual **apt-get install** commands to download them.



UPDATED EPIPHANY

The Epiphany web browser has been updated – most of the changes are to improve playback of videos from sites such as YouTube and Vimeo. Many more videos should now play than in previous releases, and overall stability when playing video should be improved.


Gottfried Haider

PI GETS PROCESSING

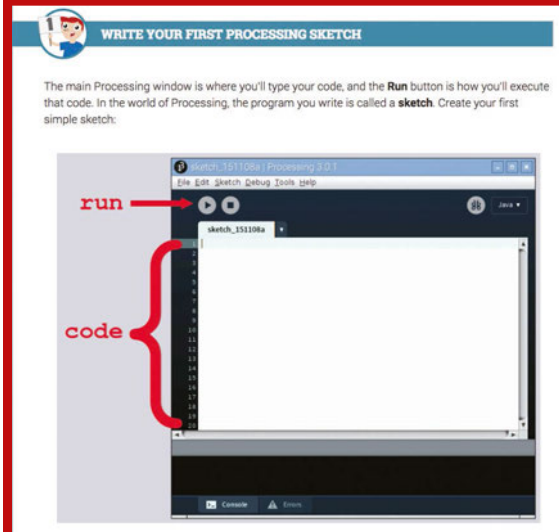
The Processing language now officially supports Raspberry Pi. We chat to co-founder **Ben Fry**, along with **Gottfried Haider**, who led the conversion process...

When it comes to tools for learning how to code, we draw the line at Processing... literally! Mainly used in the realm of the visual arts, this free and open-source programming language enables users to draw lines and shapes on the screen with just a few lines of code in a 'sketch' (program). While it's easy to get started with, Processing is a very flexible and powerful tool for design prototyping, and enables the creation of sophisticated animated simulations that respond to inputs. So we're really excited that there's now an official ARMv6hf version available to download for the Raspberry Pi, particularly since it includes a Hardware I/O library so you can use it out of the box with the Pi's GPIO pins for physical computing projects.

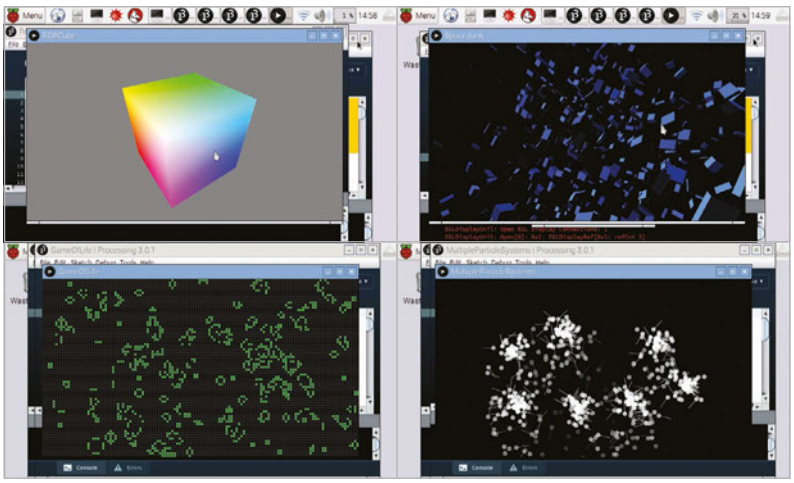
Speaking to us about the importance of Pi support, Processing co-founder Ben Fry explains: "The project has always been about lowering the barrier of entry for people getting started with code (or coders getting started with creating visual things), and the Pi is an extremely inexpensive and therefore very accessible platform." The Pi also reminds him and fellow founder Casey Reas of how they taught themselves to code many years ago using machines like the Apple II, which had built-in BASIC and diagrams for its entire schematic at the back of the manual. "Merging those two together is the sort of experimentation that comes along with an open platform: the most interesting work comes from areas where you have that openness in the platform, mixed with a community that supports it."

Asked about what advantages Processing has over a language like Python, Ben responds: "Python is a great general language for so many things, but of course it's not designed around interactive graphics - nor should it be. While there are graphics libraries and toolkits, they're often tricky to get working in a cross-platform fashion, especially for beginners." Still, he says they do a lot of teaching and scripting with Python, and there's also a Python Mode so that you can write code with the Processing API using Python syntax (**py.processing.org**).

LEARNING RESOURCE



The Raspberry Pi Foundation's 'Introduction to Processing' guide (magpi.cc/1XOJeEo) helps newcomers to get started with the language. The first tutorial worksheet focuses on drawing and animating objects, and getting them to respond to mouse input. A second worksheet gives examples of how to use the GPIO pins to control an LED and react to a physical push button.



Hardware library

Much of the heavy lifting to get Processing working smoothly on Raspberry Pi was done by Austrian artist and self-taught programmer Gottfried Haider. "Since Processing runs on Java, and Oracle provides a version of Java for ARMv6, it was very easy to run Processing on the Raspberry Pi," he tells us. "Most of the actual work concerned 3D graphics, the Hardware I/O library, enabling JNA [Java Native Access] on Raspbian, and porting the few libraries that make use of native code themselves. No major hurdles."

While the initial plan was to focus on Eric Anholt's OpenGL 2 Mesa3D driver, in the end they opted to go with the existing OpenGL ES 2 driver. "Fortunately, the JOGL project (the OpenGL binding for Java that Processing uses) also supports the Pi's OpenGL ES 2 driver, so there wasn't much effort needed on our side," reveals Gottfried. "Thanks a lot to the JOGL team, and especially Xerxes Rånby for his work on this!"

Gottfried's biggest contribution was the creation of the Hardware I/O library. "The prospect of interfacing with a wide range of sensors and actuators opens up interesting possibilities for extending Processing's 'screen'," he tells us. "I've been toying around with, for example, drawing to very small and inexpensive OLED displays that are connected via I²C, as those have such an interesting scale to work with." He has also hooked up a digital-to-analogue converter (DAC) to a sketch via I²C that generates a precise voltage connected to touch input and other things happening on the screen: "This voltage can then control and activate other elements in analogue hardware."

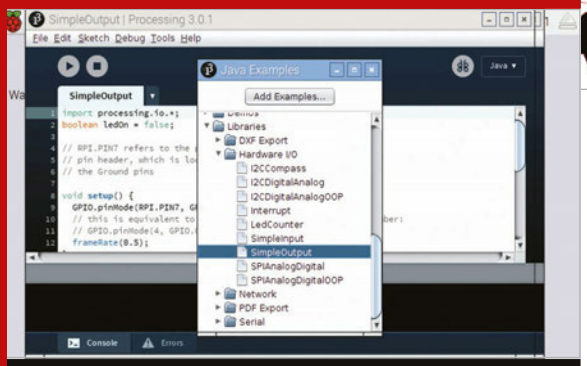
Gottfried says this is just one example of the many possibile ways it could be used. "Part of the general motivation for writing the library was also the desire to expose and make accessible the more hidden aspects of everyday infrastructure. I believe that most chipsets in consumer electronics actually have plenty of GPIO pins, and that it would be just a matter of providing access to allow them to be used in ways not foreseen by the OEMs."

While its reliance on Oracle's non-FOSS Java (bundled with it) means Processing is unlikely to be included in the official Raspbian repo for the time being, it can be downloaded from the Processing website (**processing.org**) or by opening a Terminal window in Raspbian and entering:

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

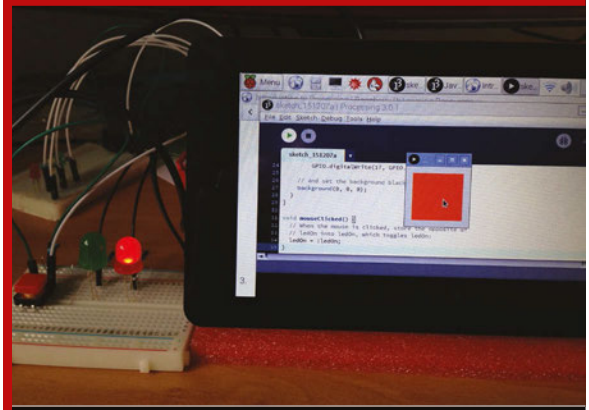
It should then appear under Programming in the desktop menu. To help you get started with Processing, the Raspberry Pi Foundation has created an introductory guide (see 'Learning resource' boxout).

HARDWARE I/O LIBRARY



Gottfried's new Hardware I/O library includes several classes, including those to support GPIO, I²C and SPI interfaces. There's also a PWM class to control hardware pulse-width modulation, and an LED class to flash the Pi's own built-in LEDs. For more information, see the reference page: magpi.cc/1XONtQu

PROCESSING WITH GPIO



As detailed in the 'Introduction to Processing' guide (magpi.cc/1XOJeEo), a basic example of using the Pi's GPIO pins with Processing involves lighting an LED whenever the sketch window is clicked (or tapped on a touchscreen). As in Python, the relevant pin is first set as an output, then an if/else block is used to turn it on when the window is clicked – using the `GPIO.digitalWrite` function to set it to HIGH – and off again when it's not.

BLAST OFF!

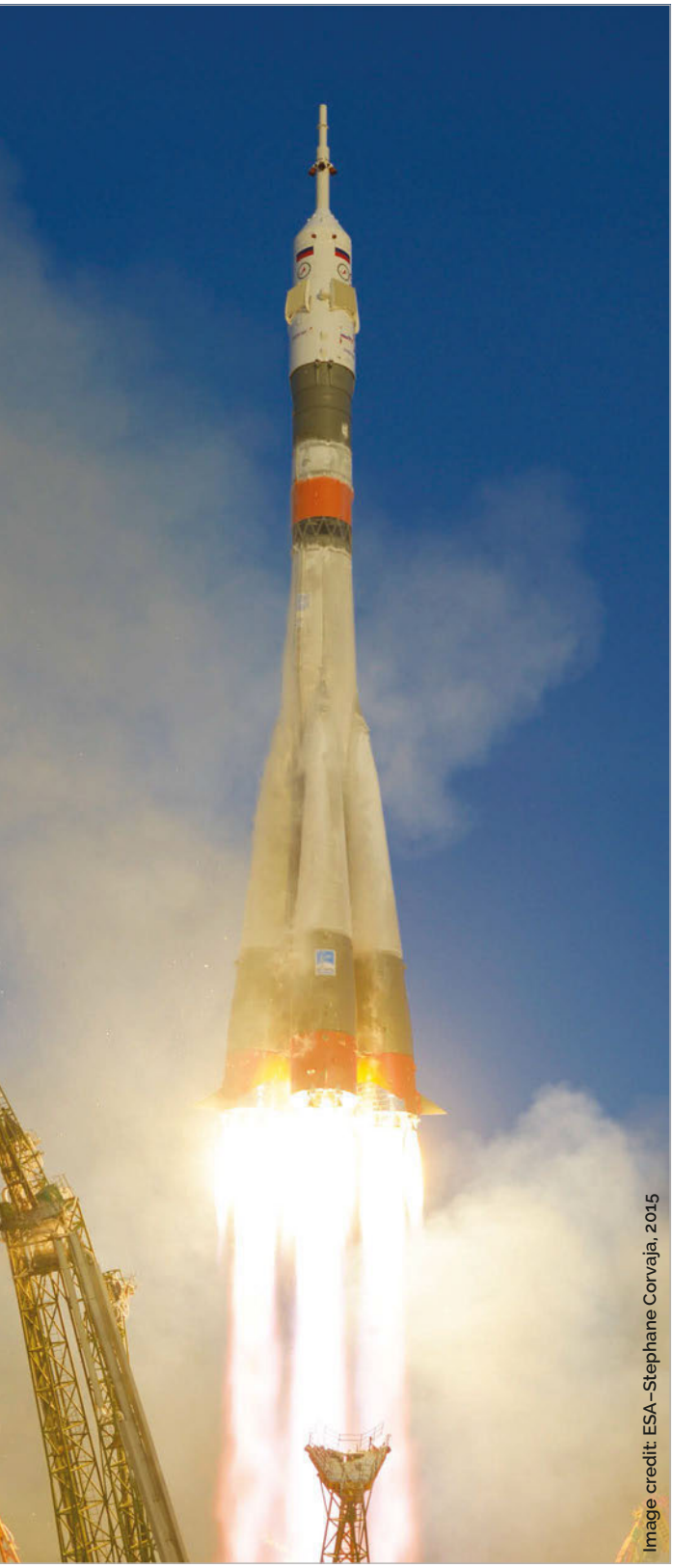


Image credit: ESA–Stephane Corvaja, 2015

Tim Peake and the Astro Pis have gone into space, making history in their wake...

If you watched the BBC coverage of Soyuz TMA-19M as it blasted off from Kazakhstan at 11:03am on Tuesday 15 December, it was difficult not to get caught up in the moment. Hundreds of schoolkids were counting down and cheering, while moustachioed space hero Chris Hadfield explained the process of the launch to Brian Cox and Dara Ó Briain. Everything went very smoothly and the launch happened on time, resulting in a thumbs-up from the spacebound Tim Peake that was tweeted around the world.

Just over a week earlier, things were different. The two Astro Pis were sitting in an unmanned Cygnus cargo freighter on the pad at Cape Canaveral in Florida, ready to be sent to the ISS themselves. With the launch originally set for Thursday 3 December, it wasn't until Sunday 6 December that the rocket finally went on its way. Bad weather delayed the launch until Friday, then, during three attempts, the countdown had begun before 'wind violations' forced a hold on the launch, as did high winds the following day. The suspense was torture at the time. However, the Cygnus finally departed on Sunday with no issues, berthing with the ISS a few days later.

Tim Peake makes history as the first British astronaut attached to the ESA to ever make it to space, let alone to the ISS itself. He's also the first Brit in space after more than a decade, and has inspired thousands in the UK to dream about space and its infinite possibilities once more. He now begins his six-month mission aboard the ISS performing several experiments, many of which are on the two Astro Pis aboard the space station with him.

Over the next six months, data from the Astro Pi experiments will be transmitted down to Earth for review as the crew members perform them. All sorts of information will be gathered by the seven projects created by the winning school teams. We'll have coverage of these experiments as they're performed over the next few months, as the real mission starts aboard the ISS.

ESA–Stephane Corvaja, 2015

The crew of expedition 46/47 wave goodbye for the last time as they board the Soyuz rocket

TIM'S TRIP: HOW DO YOU GET TO THE INTERNATIONAL SPACE STATION?

ESA–Stephane Corvaja, 2015

LAUNCH!

After several months of training, Tim and his fellow astronauts board the Soyuz rocket in Kazakhstan that will launch them into space. Within 9 minutes of the launch, all three rocket stages have separated from the Soyuz capsule, having propelled it to an initial altitude of 208km.

WAITING IN ORBIT

As Soyuz orbits the Earth, its solar panels and radio antennae now deployed, the crew spend time making sure it's ready to dock with the ISS once their paths align. Several engine burns are made to get the capsule to the correct altitude (around 400km) over the course of six hours.

DOCKING

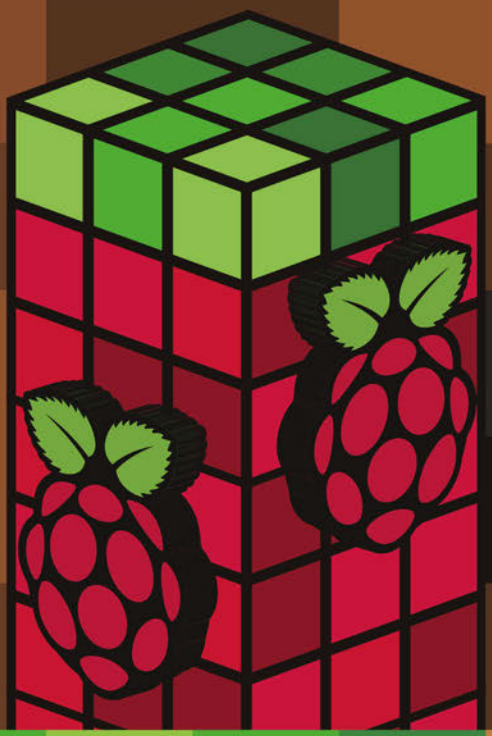
The Soyuz capsule carefully approaches the ISS and slowly docks, with contact and capture occurring at 5:33pm, six-and-a-half hours after launch. Hooks are attached and the crew spend a couple of hours making sure the seals are secure, before finally boarding the space station.

MINECRAFT MASHUPS

Use **MINECRAFT** with other software, or even have it control the real world…

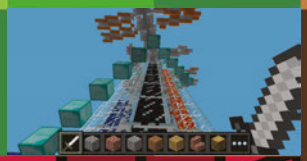
Minecraft on the Raspberry Pi is a little different to the version you might have played on another computer or games console. As well as there being no creepers to contend with, there's also the built-in ability to hack and modify *Minecraft*.

In the past, we've shown you some simple ways to interact with *Minecraft* using Python; however, you can connect up much more than a simple Python script. Whether it's other software, such as Sonic Pi, controlling the game with hardware like the Sense HAT, or even having *Minecraft* control things in the real world, there's a lot more to try out. So get your Raspberry Pi out and let's mash up some *Minecraft*!



MINECRAFT IN SPACE

Minecraft is also being mashed up on the International Space Station with Astro Pi! SpaceCRAFT represents environmental data in *Minecraft* as an experiment. Find out more here: **magpi.cc/21Ou3ux**

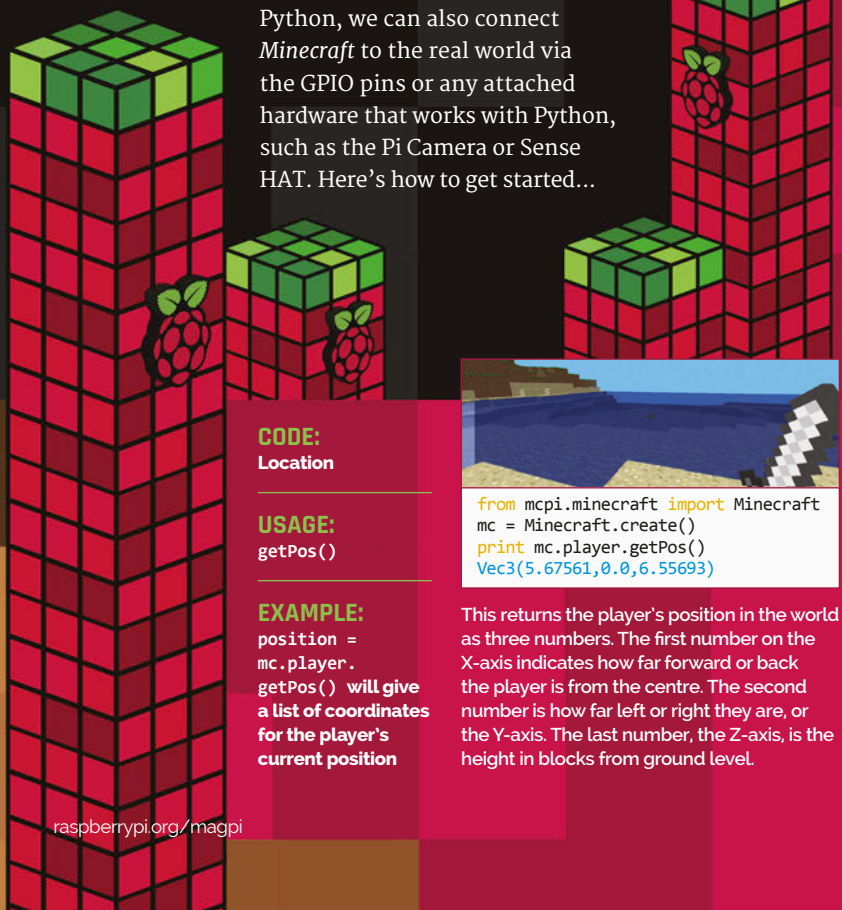


MINECRAFT PYTHON API

The first step in hacking MINECRAFT Pi is to program it in Python

There are a few ways to program *Minecraft* on Raspberry Pi, and we'll be talking about those in the rest of the feature. The most popular way to mod it, however, is by using the included Python library. If you've ever used Python, the syntax and logic of the code remains the same, only now there are functions that link directly into *Minecraft*!

As we're programming in Python, we can also connect *Minecraft* to the real world via the GPIO pins or any attached hardware that works with Python, such as the Pi Camera or Sense HAT. Here's how to get started...



CODE:
Location

USAGE:
`getPos()`

EXAMPLE:
`position = mc.player.getPos()` **will give a list of coordinates for the player's current position**

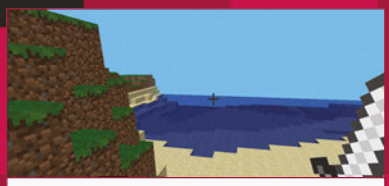

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
print mc.player.getPos()
Vec3(5.67561,0.0,6.55693)
```

This returns the player's position in the world as three numbers. The first number on the X-axis indicates how far forward or back the player is from the centre. The second number is how far left or right they are, or the Y-axis. The last number, the Z-axis, is the height in blocks from ground level.

CODE:
Teleport

USAGE:
`.setPos(x,y,z)`

EXAMPLE:
`mc.player.setPos(0.0, 0.0, 0.0)` **will teleport the player to the centre of the map**



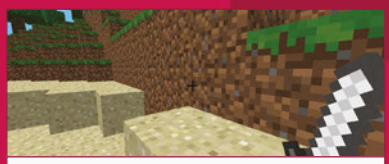
```
mc.player.setPos(10.0,12.0,5.0)
mc.player.setPos(10.0,12.0,-5.0)
```

Using the same coordinate system of X, Y and Z from `getPos()`, you can move the player around the map as you see fit. You'll have to figure out the coordinates you want to move to via testing, or by using `getPos()` at the desired spot.

CODE:
Block type

USAGE:
`getBlock(x,y,z)`

EXAMPLE:
`block = mc.getBlock(0,0,0)` **sets variable block as the type of block at the centre of the map**



```
print mc.getBlock(10.0,3.0,-5.0)
3
print mc.getBlack(5,-1,7)
12
```

Each block type has a name and identification number for it; for example, AIR has the ID of 0. Using the `getBlock` function, you can figure out what kind of block is at certain coordinates: this is useful for figuring out where a player is in your code if you need to trigger a certain event.

CODE:
Create a block

USAGE:
`.setBlock(x,y,z)`

EXAMPLE:
`mc.setBlock(0,0,0, block.OBSIDIAN.id)` **will create a block of obsidian in the centre of the map**

```
mc.setBlock(10.0,0.0,-1.0, block.OBSIDIAN.id)
mc.setBlock(10.0,0.0,-2.0, block.OBSIDIAN.id)
mc.setBlock(9.0,2.0,-2.0, block.ICE.id)
```

Create any kind of block wherever you want it in the world. You can see how this works in CrazySqueak's code on the page before this, as he has created lava, water, and empty spaces as part of the Natural Disasters program. You can also create a chunk of blocks with a range of coordinates.

CRAZYSQUEAK

Crazysqueak's favourite thing in *Minecraft* is his new Natural Disasters program. He says that it's particularly fun to make lots of things happen at once.
crazysqueak.wordpress.com
twitter.com/CrazySqueak

CREATE NATURAL DISASTERS IN MINECRAFT

Minecraft x Python

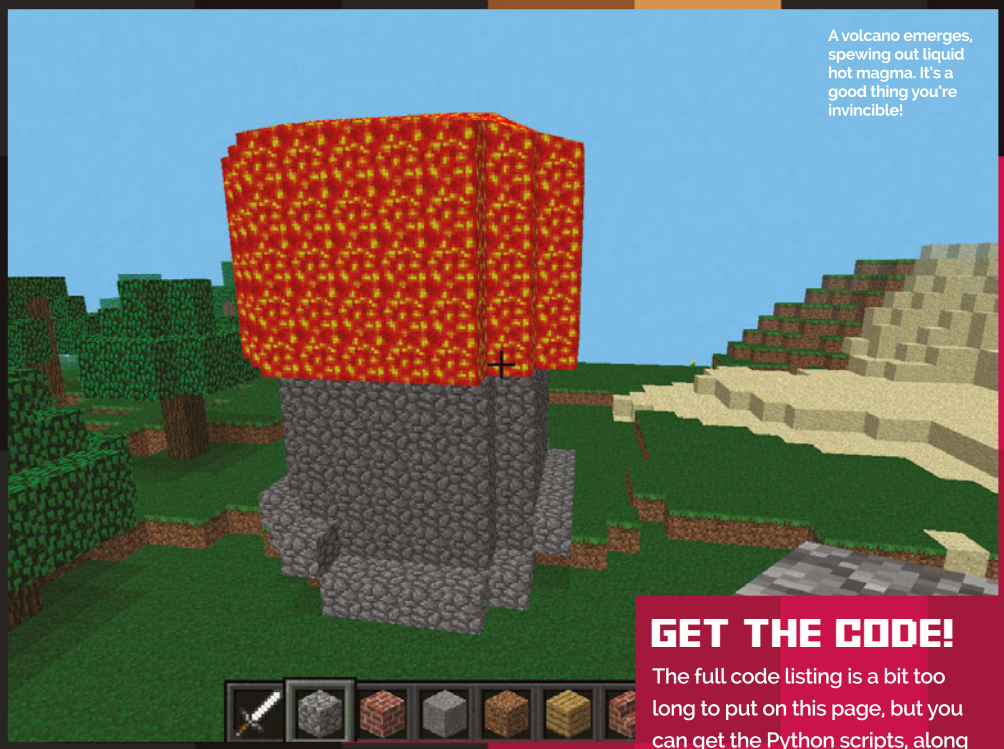
The popular programming language makes use of a special library that allows you to control *Minecraft* with it.

Cause peril in your MINECRAFT world by adding catastrophes such as meteors and earthquakes

"I have created natural disasters in *Minecraft* using Python," ten-year-old CrazySqueak writes on his blog. "It adds many disasters to your *Minecraft* that happen randomly, wherever you are in your world. The program randomly starts disasters on its own, so you should keep moving to avoid getting hit."

This excellent Python script for *Minecraft* does something very different from other Python hacks that require player interaction: it actually adds to the world in the way a normal PC game mod might. With earthquakes, sinkholes, meteors, geysers, and volcanic eruptions each acting differently and independently, a lot of work has gone into this program.

The code works by setting up the parameters of each disaster. Each type has individual timing for when it occurs, once triggered, and how long it works for. They all use the *Minecraft* Python API (discussed over the page) to create or remove blocks, such as creating lava for the eruption and meteor, or creating an empty space with the sinkhole and earthquake.



A volcano emerges, spewing out liquid hot magma. It's a good thing you're invincible!

All the disasters are triggered at random in the code, and are based around your location in the game – that's why CrazySqueak suggests staying on the move! You can also trigger each function individually to see how it works, and some even come with sound clips to further add to the effect of the mod.

As well as creating natural disasters, CrazySqueak received a Highly Commended award for a submission to Astro Pi. We can't wait to see what other mashups he creates for *Minecraft* in the future.

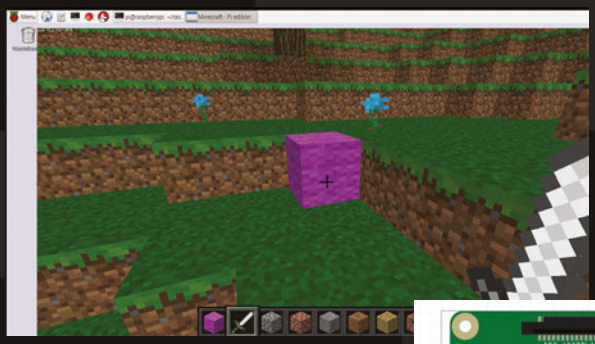
GET THE CODE!

The full code listing is a bit too long to put on this page, but you can get the Python scripts, along with incidental music and more tips on how to use the program, on CrazySqueak's blog:
magpi.cc/1NUZ8Zm

Holes form in the world, which can cause some serious trouble for navigation, and for any houses there

CONTROL AN LED WITH MINECRAFT

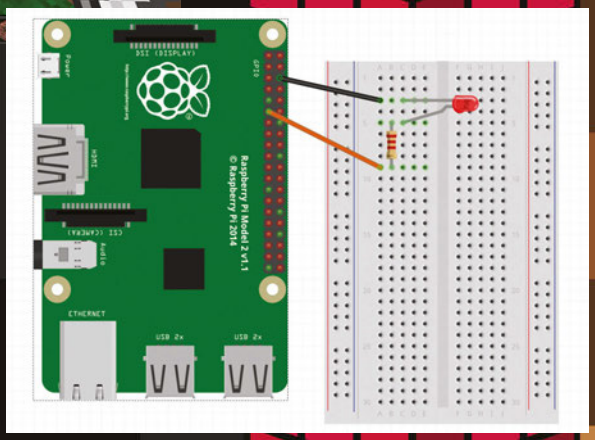
magpi.cc/1M6RXpj



Using GPIO Zero in conjunction with the *Minecraft* Python API, we can do some amazing things very simply. GPIO Zero is the new Python coding library that allows you to quickly and easily connect and control physical electronics in the real world with Python.

In this code, you need to hunt down a specific block; your only guide is an LED light. It blinks as you head in the right direction, blinking faster as you get close, before becoming a solid light when you're extremely close to it.

This code keeps track of the player's position, with a set block as the target. As the player gets nearer and the distance to the block shortens, the distance is used to alter the time delay in the LED flashing. There are four levels to the flashing: the slowest indicates you're generally heading in the right direction, the next step up is close, the following is near, and the final one is within ten blocks. The solid light occurs within four blocks of the target, and the game lets you know when you've found it by displaying a message on the screen.



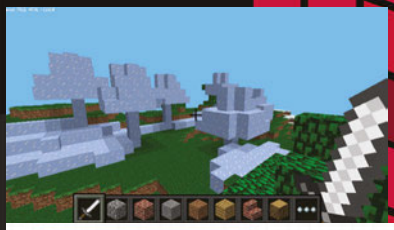
CONTROL MINECRAFT WITH A BUTTON

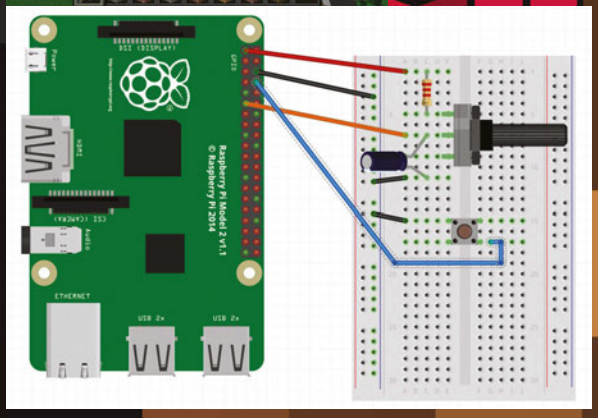
magpi.cc/1M6OjMe

We have an example of an output from *Minecraft* that controls an LED using GPIO Zero; now to talk about how an input can be used. Using a project designed by Richard Hayler, you can create the effect of having ice powers, freezing blocks in your path at the touch of a physical push button you've added to a breadboard. You can also change the range of the power's effect using a potentiometer, also known as a variable resistor.

The code uses the input reading of the GPIO pin and converts it to a number usable by the code. This then generates the range of blocks that will be affected by the player when the button is pressed. When it's pressed, the selection of blocks in the range that aren't made of air have their coordinates saved so they can all be set to ice blocks.

The code lets you know the power of your ice blast, so you can tweak the resistor and figure out how to get it configured perfectly.



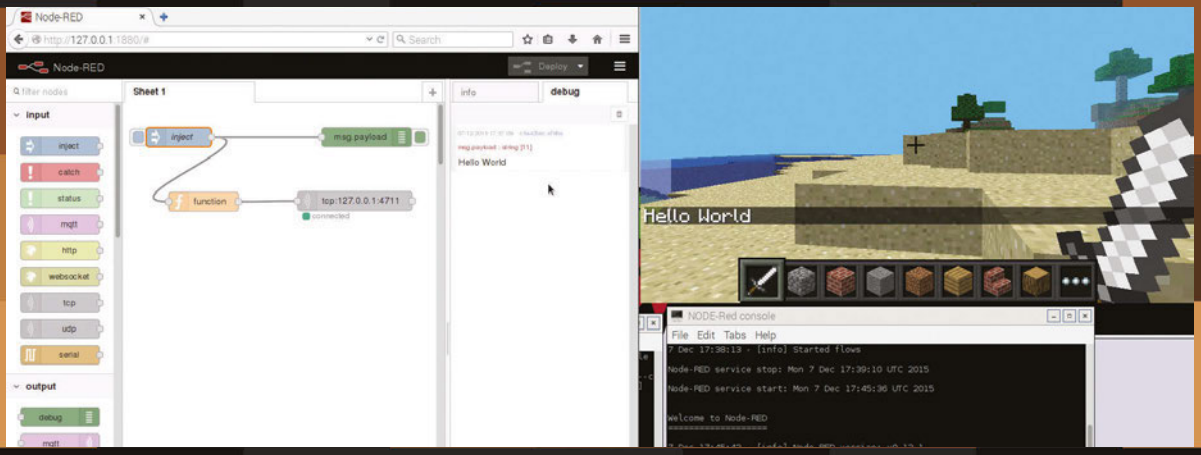


BORIS ADRYAN

Boris has particularly enjoyed using Node-RED to retrieve inventories of the blocks around him and plot histograms in R.
iot.ghost.io / twitter.com/BorisAdryan

NODE-RED AND CONTROLLING MINECRAFT WITH JAVASCRIPT

JavaScript is an interpreted programming language and one of the main components of interactive websites. With Node-RED, you can easily use it to control MINECRAFT



Right Node-RED and *Minecraft* on one screen can look a bit busy. Once you're comfortable, however, Steve's world is under your control

Minecraft x Node-RED

Node-RED is a visual tool for wiring the Internet of Things. It takes care of technicalities, like GPIO access or internet protocols, and lets you focus on your workflow.

This article should give you a taste of a workshop Boris taught at a recent CamJam. Here, we'll show you just how easy Node-RED is to use. You can refer to the original tutorial (**magpi.cc/1jLWFST**) for a lot more detail and theory.

Preparation

Start *Minecraft* and open a new world. Fire up Node-RED from the **Programming** submenu on the Raspbian desktop. After a few moments, Node-RED is going to run as a service in the background. Open a web browser (Iceweasel works best with Node-RED) and direct it to 127.0.01:1880 to see the programming environment.

On the left, there's a panel with nodes. These are visual components that you can use to build your flow. In the middle is the flow editor where you can simply drag and drop your flow together. On the right, you can switch between the help panel and the debug panel. Note that while 'debug' often has some negative connotation, it's your default text output in Node-RED.

Hello World in MINECRAFT

Drag and drop an inject node from the nodes panel into the flow editor. Once you've selected that inject node, you should see a general explanation about its functionality in the help panel – this is especially useful info when you select complex nodes with many different options.

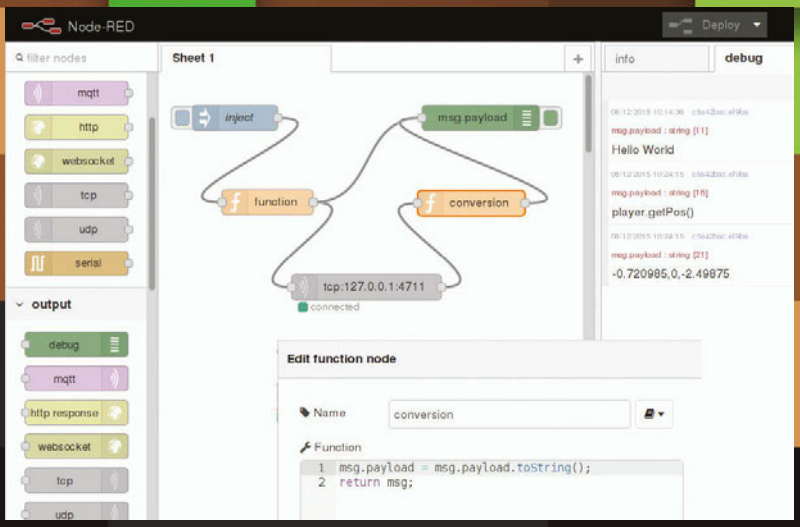
Double-click your inject node in the flow editor. Once the associated dialog opens, change the Payload to type 'string' and write 'Hello World' in the empty text field below.

The flow of information is modelled through the exchange of messages in Node-RED, which happens by passing along a variable called **msg**. It has two main properties: topic and payload. In simple terms, these could be interpreted like the subject and body of an email. Drag and drop a debug node

Language
>NODE-RED/
JAVASCRIPT

A TOOL FOR THE INTERNET OF THINGS

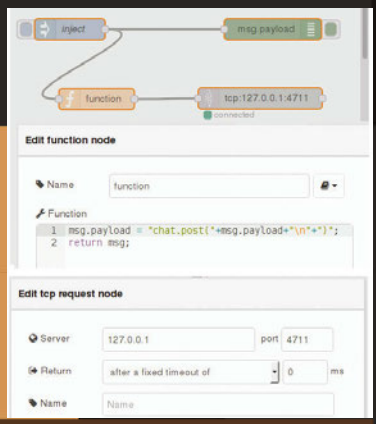
Node-RED encapsulates great functionality in already existing nodes. Let the Twitter output node feed into the Minecraft chat, or the GPIO input node trigger your flow with a physical button!



from the nodes library into the flow editor.

Drag and drop a function node into the editor. This is the node type that allows you to directly interact with the msg object in JavaScript. In your function node, write `msg.payload = "chat.post("+msg.payload+")\n";` before the line with 'return msg;'.

This line is going to take the incoming payload 'Hello World', and assign the new content `"chat.post(Hello World)\n"` to the variable. `chat.post` is a command from the *Minecraft* API, which you can read about in a file called **mcpi_protocol_spec.txt** in the *Minecraft* API directory. The end of our command is indicated by a line break, the Unix character `\n`.



Your first flow: the code you need to write into your function node, and configuration details for the TCP request node

Drag and drop a TCP request node from the function section of the node panel. The server is 127.0.0.1, and the port is 4711. As we're not expecting a return value, we'll Return 'after a fixed timeout' of oms. This connects Node-RED to the *Minecraft* TCP socket – if you're keen to understand what a socket is, have a look at the original workshop material: **magpi.cc/1jLWFST**.

Connect the nodes by drawing connections, as in the screenshots here. Start your flow by pressing the red Deploy button in the upper-right corner. Trigger your flow by pressing the rounded rectangle to the immediate left of your inject button. Do you see your message in *Minecraft*?

Retrieve values from MINECRAFT

So far, our interaction with *Minecraft* has been rather one-directional. We just sent a command string that had an effect on the *Minecraft* world. Now we're going to modify our flow so we can query values like our own position via the API.

In the function node, set `msg.payload = "player.getPos()\n";` before the line with 'return msg;'. Instead of leaving the TCP request node after some time, we expect our Return 'when character is received', namely `\n`.

By default, the TCP request node returns a buffer, and we need to convert the information from Node-RED using `msg.payload = msg.payload.toString();` in a new function node. The result of this new function node goes straight to our debug node.

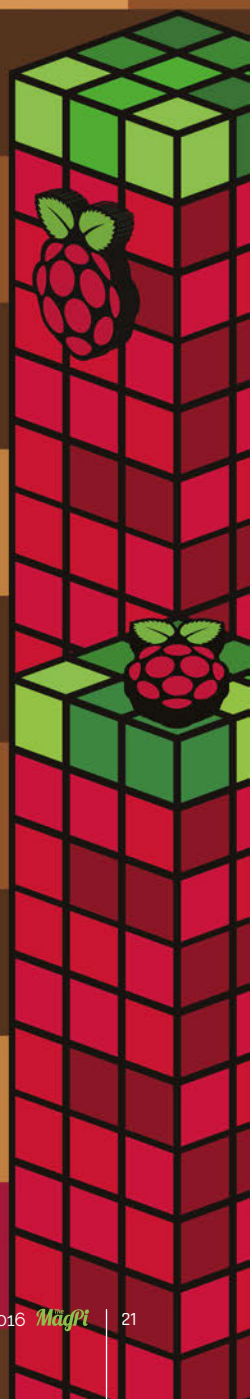
Deploy your Node-RED flow. Now have a walk around *Minecraft* and trigger your flow with the inject node. Do you see what you expected in the debug panel?

Look out for the second part of this tutorial, which will cover aspects of event-driven development with Node-RED, including the interaction of *Minecraft* with the physical world.

NODE-RED FLOWS DIRECTORY

Hello World
magpi.cc/1Qr4USK

Player Position
magpi.cc/1Qr4Xht



JASPER

Jasper's favourite *Minecraft* experience is when he 'accidentally' burnt down his dad's house because it had silverfish in the basement.

WALKING WITH STEVE

Tired of using your fingers to tap keys to move in MINECRAFT? Then why not use your wrist instead, and take advantage of the awesome power of the Sense HAT?

Stop wearing out the W, S, A, and D keys – use the Sense HAT instead

Minecraft x Sense HAT

The Sense HAT is an add-on board which attaches to the Pi's GPIO pins. It has lots of sensors, such as the humidity sensor and the magnetometer. The sensor we'll be using in this project is the accelerometer. It also has an 8×8 LED matrix.

One of the cool things about the console edition of *Minecraft* is that you can use a controller instead of a keyboard. The Pi edition might sometimes seem a little basic, but you can make your game more like the console edition by deploying your Sense HAT as a tiltable controller instead of using a keyboard. If you don't know which way to tilt it, the arrows appearing on the LED matrix will help you.

The first thing you need to do is to install all the necessary modules. An obvious one is the Sense HAT library: if you have Raspbian Jessie, this comes bundled with it; if not, you can install it by typing:

```
sudo pip install sense_hat
```

You'll also need another Python module, which in turn requires the Xlib library:

```
sudo apt-get install
python-xlib
```

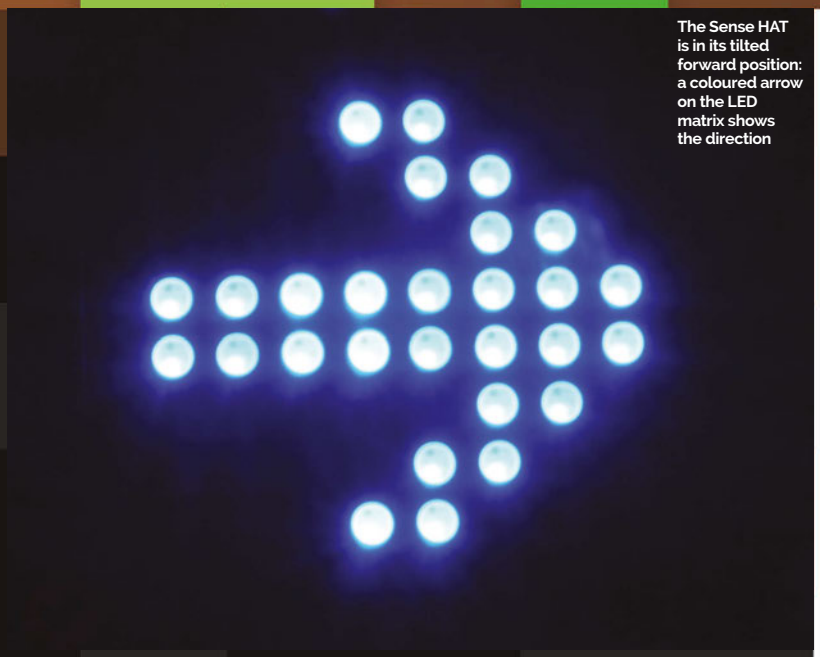
```
sudo pip install pyautogui
```

Using the pyautogui functions, you can simulate keys as if they were actually pressed. This is how you make Steve walk around his blocky world.

Instead of pressing keys, we'll use the Sense HAT's accelerometer to find out which direction the HAT is being tilted. Each time we measure, we get values representing the acceleration intensity of the x, y, and z axes (in Gs). These are sometimes called roll, pitch, and yaw, like on an aeroplane or a spaceship. We only need the x and y axes, as the

Language
>PYTHON2.7

DOWNLOAD:
magpi.cc/1Qr7wjw



The Sense HAT is in its tilted forward position: a coloured arrow on the LED matrix shows the direction

z axis is rotation and we're not using that.

How to use it

Download or type up the code from the listing (right) into IDLE, then press **F5** to run it. Make sure *Minecraft* is running and you've entered a world when you run it, otherwise lots of errors will appear! If you tilt the Sense HAT forward, the pyautogui module will trigger a W key and move Steve forward; make sure your mouse is clicked in the *Minecraft* window when this happens, otherwise it will just generate a 'w' in the Python shell. The same thing happens when you tilt it backwards, but it will generate an S. If you tilt it to the side, it will generate a D or an A, depending on which direction you've tilted. You still need to use the mouse to look around, and the E key to open your inventory. So, the idea is for you to get all the items you need in your hotbar, then hold the mouse in one hand and the Sense HAT in the other.

Using MINECRAFT

When the program is running, you'll only be able to walk (or fly) around using the keys when the Sense HAT is in the level position (all the LEDs will be red). Even then, you'll only move if you tap the key repeatedly instead of holding it down. So you're better off sticking to the Sense HAT!

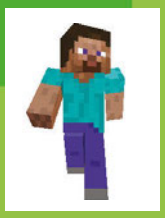
Once you have written and understood this program, you could try to improve it by making the Sense HAT's joystick open your inventory or something awesome like that.

LIGHT THE WAY

The arrows displayed on the LED matrix while you're moving look really cool, but are actually easy to generate. Using Python, we make a 64-element list containing the arrow shape - you can customise this to make your own shape of arrow - and then simply display it using a different RGB colour value and with a different rotation. The Sense HAT API makes this whole process very simple.

WalkingWithSteve.py

```
from mcpi.minecraft import Minecraft
import pyautogui as pag
import time
from sense_hat import SenseHat
sh = SenseHat()

def unpress():# unpresses all the keys
    for key in ['s','w','a','d']:
        pag.keyUp(key)

def move(direction):# presses the correct key
    unpress()
    pag.keyDown(direction)

def displayArrow(c,rot):# the arrow
    arrow = [
    e,e,e,c,c,e,e,e,
    e,e,c,c,c,c,e,e,
    e,c,c,c,c,c,c,e,
    c,c,e,c,c,e,c,c,
    c,e,e,c,c,e,e,c,
    e,e,e,c,c,e,e,e,
    e,e,e,c,c,e,e,e,
    e,e,e,c,c,e,e,e]
    sh.set_rotation(rot)
    sh.set_pixels(arrow)

r = [255,0,0]# define the colours
e = [0,0,0]
g = [0,255,0]
b = [0,0,255]
stop = [# the stop sign
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r,
r,r,r,r,r,r,r,r]

mot = 'SSSS'
while True:# main loop
    x, y, z = sh.get_accelerometer_raw().values()
    x = round(x, 0)
    y = round(y, 0)
    if x  == -1  and abs(y) == 0 and mot != 'rrrr':
        displayArrow(b,0)
        move('d')# right
        mot = 'rrrr'
    elif  x == 1  and abs(y) == 0 and mot != 'llll':
        displayArrow(b,180)
        move('a')# left
        mot = 'llll'
    elif y == -1  and mot != 'wwww':
        displayArrow(g,270)
        move('w')# fwd
        mot = 'wwww'
    elif y == 1  and mot != 'bbbb':
        displayArrow(g,90)
        move('s')# back
        mot = 'bbbb'
    elif abs(x) == 0 and abs(y) == 0:
        unpress()# stop
        sh.set_pixels(stop)
        mot = 'SSSS'
```



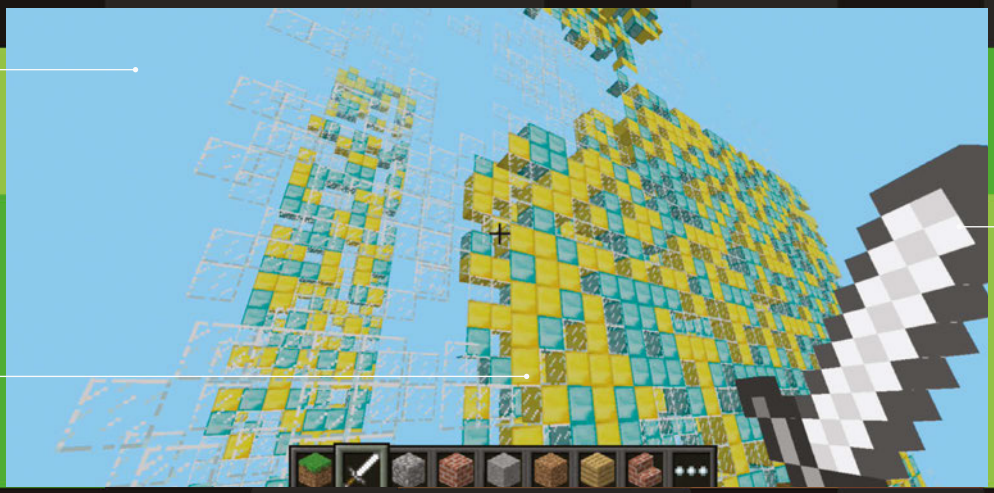
SAM AARON

Sam performed with Alan Blackwell at an Algorave. They live-coded both dance music and Minecraft visuals with Sonic Pi, and got the whole crowd dancing. That performance provided inspiration for this tutorial.

sonic-pi.net

BECOME A MINECRAFT VJ

Use Sonic Pi with MINECRAFT to create amazing visuals for your music as you perform it!



All of this is powered by Sonic Pi, allowing you to make your visualisations match the music

Visualisations as done in *Minecraft*, a step up from Windows Media Player

Move around your live set to see how the visualisation changes

Minecraft x Sonic Pi

Sonic Pi transforms your Raspberry Pi into a live-coding musical instrument. Create the beats, riffs, and drops of your dreams, all with simple code.

Everyone has built amazing structures, designed cunning traps, and even created elaborate cart tracks in *Minecraft*. How many of you have performed with *Minecraft*? We bet you didn't know that you could use *Minecraft* to create amazing visuals, just like a professional VJ.

As well as coding with Python, you can also program *Minecraft* with an app called Sonic Pi, which makes the coding not only easy but also incredibly fun. In this article, we'll be showing you some of the tips and tricks that we've used to create performances in nightclubs and music venues around the world.

Enter a new world in *Minecraft* and open Sonic Pi. When we're using *Minecraft* to create visuals, we try to think about what will both look interesting and also be easy to generate from code. One nice trick is to create a sandstorm by dropping sand blocks from the sky. For that, all we need are a few basic fns (Sonic Pi functions):

- `sleep` – for inserting a delay between actions
- `mc_location` – to find our current location
- `mc_set_block` – to place sand blocks at a specific location
- `rrand` – to allow us to generate random values within a range
- `live_loop` – to allow us to continually make it rain sand

Let's make it rain a little first, before unleashing the full power of the storm. Grab your current location and use it to create a few sand blocks up in the sky nearby:

```
x, y, z = mc_location
mc_set_block :sand, x, y + 20, z + 5
sleep 2
mc_set_block :sand, x, y + 20, z + 6
sleep 2
mc_set_block :sand, x, y + 20, z + 7
sleep 2
```

```
mc_set_block :sand, x, y +
20, z + 8
```

When you press Run, you might have to look around a little, as the blocks may start falling down behind you depending on which direction you're currently facing. Don't worry: if you missed them, just press Run again for another batch of sand rain – just make sure you're looking the right way!

Let's quickly review what's going on here. On the first line, we grabbed Steve's location as coordinates with the fn `mc_location` and placed them into the vars `x`, `y`, and `z`. Then, on the next lines, we used the `mc_set_block` fn to place some sand at the same coordinates as Steve, but with some modifications. We chose the same x coordinate, a y coordinate 20 blocks higher, and then successively larger z coordinates so the sand dropped in a line away from Steve.

Why don't you take that code and start playing around with it yourself? Try adding more lines, changing the sleep times, try mixing `:sand` with `:gravel`, and choose different coordinates. Just experiment and have fun!

Live loops unleashed

OK, it's time to get the storm raging by unleashing the full power of the `live_loop`, Sonic Pi's magical ability, which unleashes the full power of live-coding: changing code on the fly while it's running!

```
live_loop :sand_storm do
  x, y, z = mc_location
  xd = rrand(-10, 10)
  zd = rrand(-10, 10)
  co = rrand(70, 130)
  synth :cnoise, attack: 0,
release: 0.125, cutoff: co
  mc_set_block :sand, x +
xd, y+20, z+zd
  sleep 0.125
end
```

What fun! We're looping round pretty quickly (eight times a second), and during each loop we're finding Steve's location like before but then generating three random values:

- `xd` – the difference for x, which will be between -10 and 10
- `zd` – the difference for z, also between -10 and 10
- `co` – a cutoff value for the low pass filter, between 70 and 130

We then use those random values in the fns `synth` and `mc_set_block`, giving us sand falling in random locations around Steve, along with a percussive rain-like sound from the `:cnoise` synth.

For those of you new to live loops, this is where the fun really starts with Sonic Pi. While the code is running and the sand is pouring down, try changing one of the values, perhaps the `sleep` time to `0.25` or the `:sand` block type to `:gravel`. Now press the Run button *again*. Hey presto! Things have changed without the code even stopping. This is your gateway to performing like a real VJ. Keep practising and changing things around. How different can you make the visuals without stopping the code?

Epic block patterns

Finally, another great way of creating interesting visuals is to generate huge patterned walls to fly towards and get close to. For this effect, we'll need to move from placing the blocks randomly to placing them in an ordered manner. We can do this by nesting two sets of iteration; press the Help button and navigate to section 5.2 of the tutorial, 'Iteration and Loops', for more background on iteration. The funny `|xd|` after the `do` means that `xd` will be set for each value of the iteration. So, the first time it will be 0, then 1, then 2 and so on. By nesting two lots of iteration together like this, we can generate all the coordinates for a square. We can then randomly choose block types from a ring of blocks for an interesting effect:

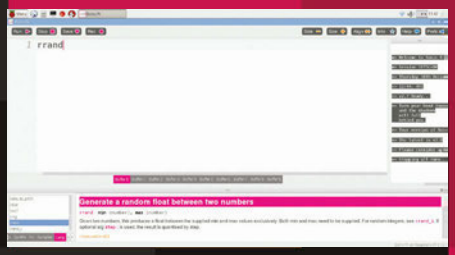
```
x, y, z = mc_location
bs = (ring :gold, :diamond,
:glass)
10.times do |xd|
  10.times do |yd|
    mc_set_block bs.choose, x
+ xd, y + yd, z
  end
end
```

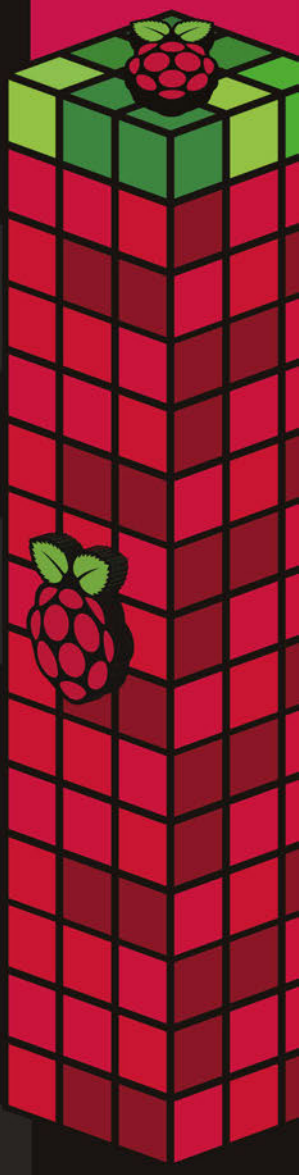
Pretty neat. Whilst we're having fun here, try changing `bs.choose` to `bs.tick` to move from a random pattern to a more regular one. Try changing the block types – the more adventurous of you might want to try sticking this within a `live_loop` so that the patterns keep changing automatically.

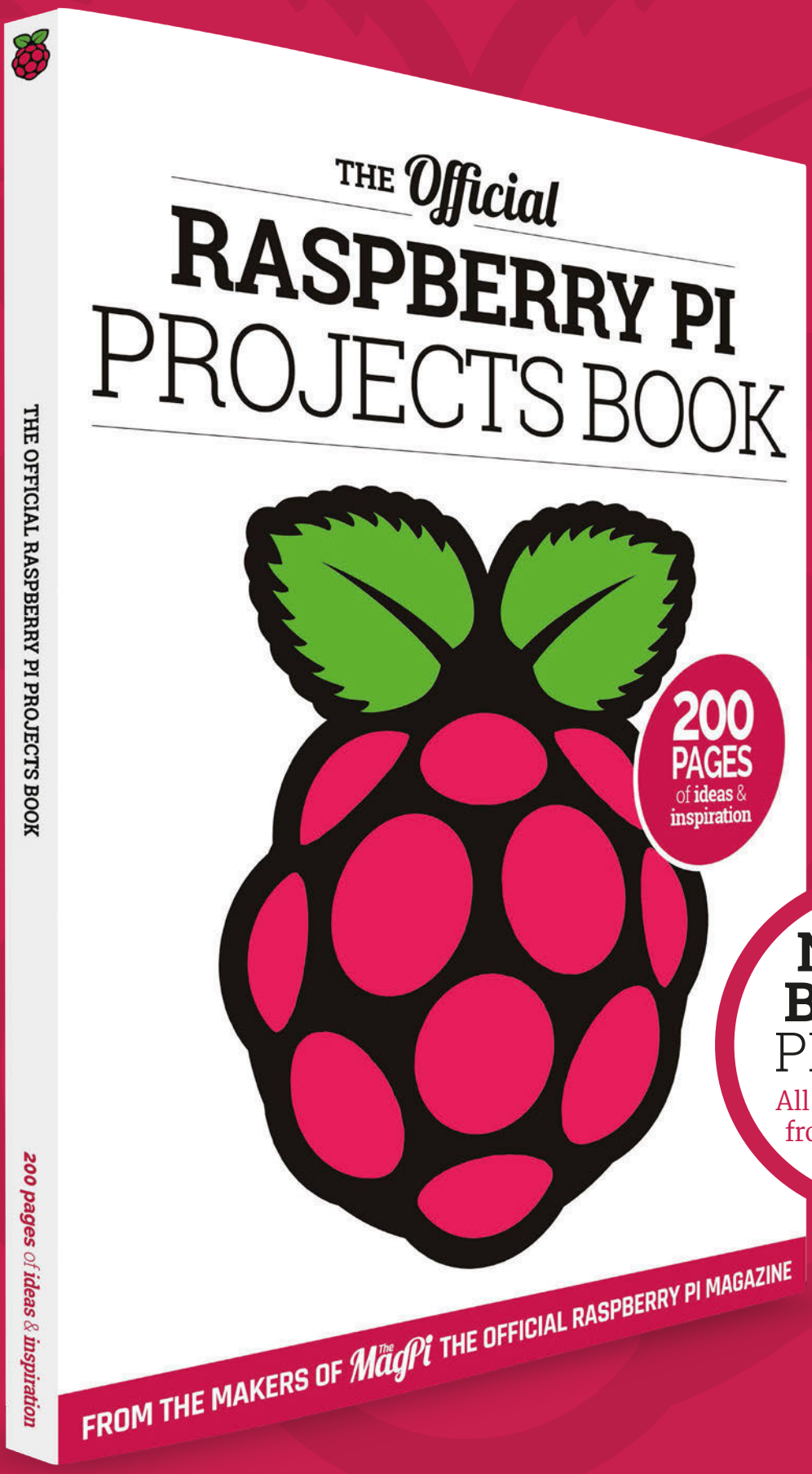
Now, for the VJ finale. Change the two `10.times` to `100.times` and press Run. Kaboom!... A gigantic wall of randomly placed bricks. Imagine how long it would take you to build that manually with your mouse! Double-tap **SPACE** to enter fly-mode and start swooping by for some great visual effects. Don't stop here, though – use your imagination to conjure up some cool ideas and then use the coding power of Sonic Pi to make it real. When you've practised enough, dim the lights and put on a VJ show for your friends!

HELP WITH FUNCTIONS

If you're unfamiliar with any of the built-in fns such as `rrand`, just type the word into your buffer, click on 'int', and then press the keyboard combo **CTRL+I** to bring up the built-in documentation. Alternatively, you can navigate to the 'lang' tab in the Help system and then look up the fns directly, along with all the other exciting things you can do.

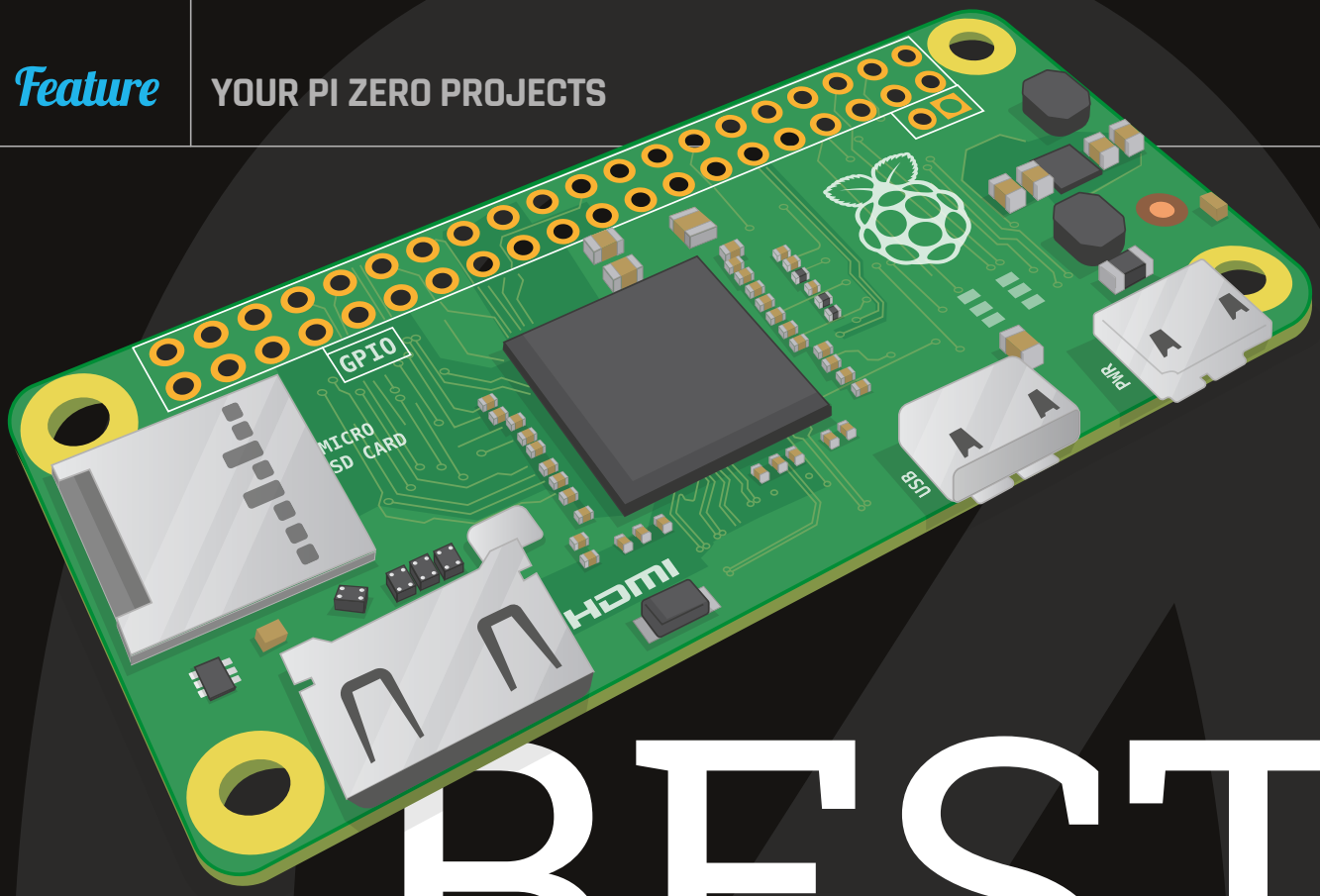






NEVER BEFORE PRINTED

All the best articles from issues **31–35**



THE BEST RASPBERRY PI ZERO PROJECTS SO FAR

It took no time at all for people to start putting their brand new Pi Zero to work to power smaller and cooler projects

Well, the launch of the Raspberry Pi Zero was exciting, wasn't it? We sold out of all our copies and had to do a second print run, meaning there's a lot of Raspberry Pi Zeros out there already for people to start using in projects. Within days, people had done some incredible things: new hacks, updated hacks, proofs of concept... There was a lot of excellent stuff flying around. You could spend a good couple of hours going back through the #pizero hashtag for them all, but here's some of the best ones that we found while writing the magazine.

SEND US YOUR PROJECTS

ON TWITTER **@THEMAGP1**

XBOX CONTROLLER PI ZERO

Bring new meaning to Xbox Media Centre by playing Doom built into a controller

TERENCE EDEN
Web developer
shkspr.mobi

There are many colloquial names for the original controller for the first Xbox. The Duke. The Bear. The Fire Hazard. Its size is attributed to the urban legend of being ergonomic for 'big American hands'; thanks to its size, however, it ended up being pretty perfect for Terence Eden's project.

"I wanted to play retro games on my TV," Terence tells us. "But I didn't want to buy yet another box to sit under there. I also didn't want to have to faff about with anything too complicated – wouldn't it be great, I thought, if I could fit an entire games console *inside* a controller?

Similar to our project idea (which ultimately resulted in a full tutorial this issue – see page 52), Terence

> "I wanted to play retro games on my TV"

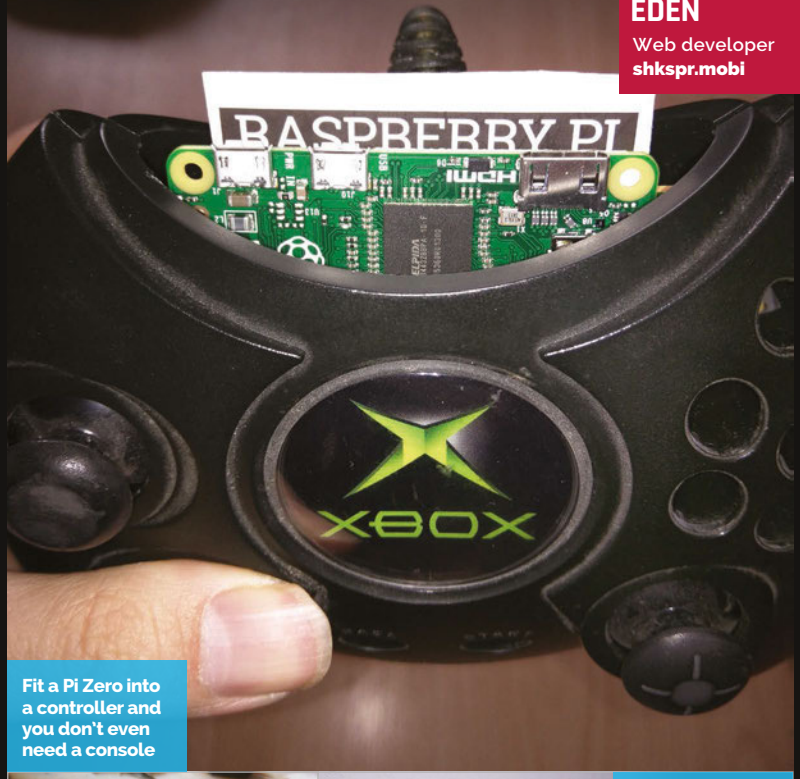
thought originally about using the iconic NES controller. However, his wife pointed out that the larger Xbox controller would be perfect for what he had planned.

"So, this project was about stuffing a Pi inside a controller and using it to play *Doom*!"

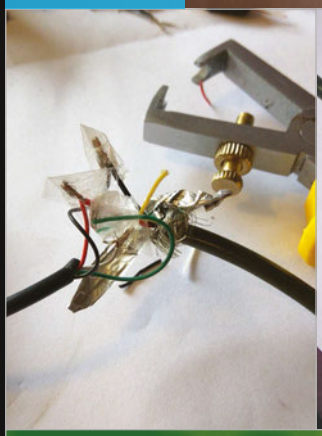
Apparently it was a tight fit even for the Raspberry Pi Zero, meaning that some of the interior of the controller housing had to be shaved off to squeeze it in. "It's great that all the ports are along one side; that makes fitting it into tight spots much easier," Terence adds.

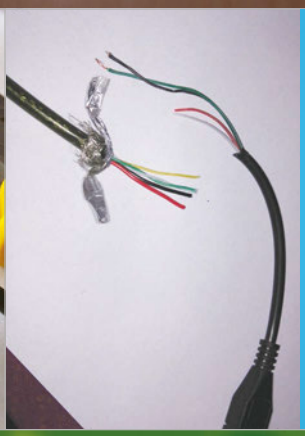
The component list was nothing fancy: just the standard cables needed to make the Zero work and the various assortment of tools needed.

Like everyone else, Terence has been inspired and is thinking how else he can make use of the Zero. "I'm curious as to whether it makes sense to use it as a door and window sensor. A Pi Zero is cheaper than a Z-Wave sensor so, as long as I can run power to it, I can place one on every door and window at home."

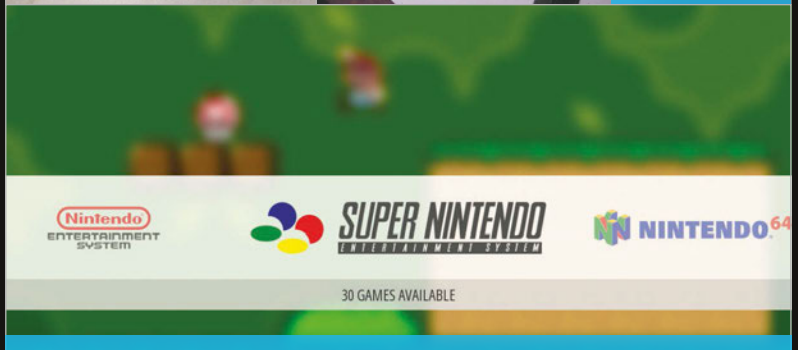


Fit a Pi Zero into a controller and you don't even need a console





USB OTG is a simple extension to the existing micro-USB spec. The connectors are really cheap: under £1 online. That said, it's pretty simple to make your own if you're confident with wiring and soldering. Terence spliced an old one directly into the controller lead



RetroPie is a brilliant resource for emulator fans, says Terence. He remembers the days when you had to scrabble around to get MAME to work, but this is a single disk image with everything built in. Magically, it even has kernel support for the Xbox controller via xpad – Terence thought he'd have to bodge that in for sure!

ENERGY MONITOR

The Internet of Things comes to the Zero with an energy consumption monitor

LAURA TREVAIL
Artist and tech explorer
twitter.com/lhtrevail

DANIELLE GRAYSTON
Software developer
twitter.com/Bladepanthera

"The project uses the Pi Zero as a tiny wee hub to connect my electronic devices at home to potentially anything," says Laura Trevail about the Pi Zero energy monitor. "It measures the device's energy consumption and publishes that data live, where it can be found and interacted with by Things elsewhere... While I'm brewing up [a drink], I can turn my kettle into something else somewhere completely different if I like, someone else can join me when I'm having a cuppa if they like, or use my inferred coffee consumption data (or whatever else I am sharing) to drive their own story in ways I may never have imagined."

So, a typical IoT device with a million uses from a simple concept! Apparently, Laura and Danielle spent the morning scouring two counties looking for a copy of *The MagPi*, and were surprised by how small the Pi Zero was when they finally got their hands on one.

The project makes use of Energenie, specifically an Energenie two-way PiMote and an Energenie MiHome monitor that uses David Whale's pyenergenie library (**magpi.cc/1jKpQ8L**).

Measure energy consumption using the Zero and a little extra hardware

RASPBERRY PI LAPTOP – ZER0 EDITION

A classic case for the Pi gets a revamp, thanks to the new size and spec of the Raspberry Pi Zero

PETER HOWKINS
Software engineer
marutan.net

It's basically a full laptop, albeit very easy to open and maintain

Lego cases have been a major part of Raspberry Pi since it very first launched – especially when there was very little available in the way of cases at the very start. Taking that to its logical conclusion in a very short time, a Lego Pi-powered laptop was created, and everyone loved it.

"It's a small portable Raspberry Pi system, ideal for carrying to events without having to carry around a separate monitor and keyboard," its creator Peter Howkins tells us. "It combines two of my hobbies: Raspberry Pi and Lego. This is the third variation of my laptop; it started using an original Model B, then I rebuilt it to fit the B+ and Pi 2, and now it's using the Pi Zero."

Peter could not believe the price of the Pi Zero when he found out about it, but it really does drive down the cost of the overall Lego laptop build. It contains converters, hacked USB ports, a screen, a USB power pack, mini keyboard, cables, and an awful lot of Lego.

MATCHBOT

A great little robot that really shows off how versatile the Pi Zero really is

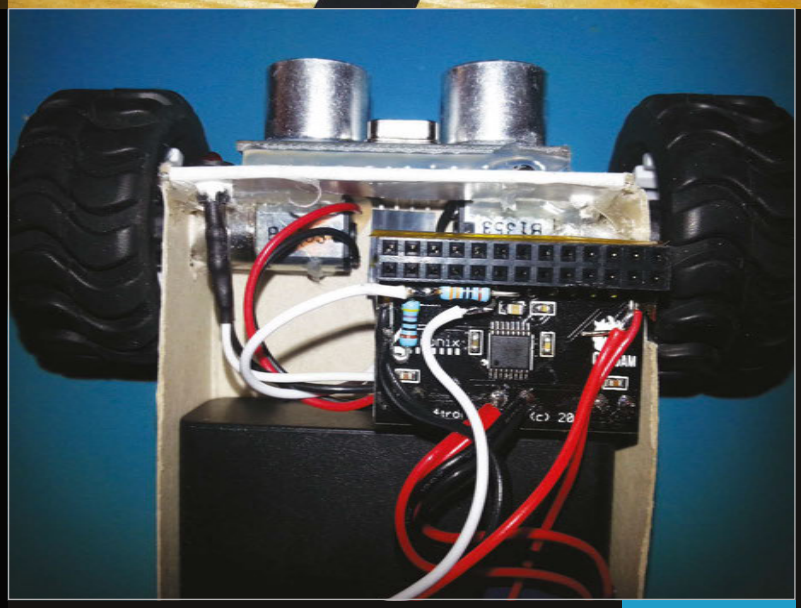
MARK CANTRILL
Electronics design engineer, Maker
astro-designs.com

In the last issue, we had an excellent 3D-printed Pi Zero robot, as made by Richard Hayler (**magpi.cc/1lLmeoi**). We didn't think we'd start seeing more robots until a few weeks down the line, but only a few days later, Mark's Zero-powered Matchbot crossed our Twitter feed.

"Matchbot, which was built in just a couple of days, is just for a bit of fun really," Mark tells us. "When the Pi Zero came out, I gave myself a challenge of building something really tiny. I really wanted a break from other things that I've been working on recently, and a chance to join in the fun of Pi Wars. Plus, as a nice simple robot, I hope to use it to help get our two girls interested in the Raspberry Pi and physical computing."

> "Matchbot was built in just a couple of days"

While it would have been amazing in itself to use spare robot kit he had lying around, Mark actually repurposed the parts of a CamJam EduKit 3 to build it. "It's built from a large matchbox, two tiny geared motors, plus wheels, and the rest (motor drivers, sensors, and battery box) is mostly from a CamJam EduKit 3. There's also a 5V regulator so that it can use the motor battery to power the Pi."



The LED flashes twice for the line-following program, three times for the proximity test, while four and five flashes will exit the script and shut down the Pi respectively

Matchbot fits in a matchbox and uses little more than the CamJam EduKit 3

The motors had to be tiny, geared just right, and reasonably powerful

As well as the motorised wheels, the Matchbot features ultrasonic distance sensors and a line-following sensor.

Fitting the robot inside a matchbox is a feat in itself. However, the control interface for the Matchbot is inspired – with no WiFi to talk to it remotely, Mark has programmed the distance sensors to change mode depending on how long you hold your hand out in front of it, with a flashing LED giving you an indication of what mode you've activated. On the day, at Pi Wars, it handled the line-following course very well!

SPANNER SPENCER

Spanner is the community manager for Element14, an online community for engineers.
magpi.cc/1m75kk9

PI ZERO RETRO GAMING SYSTEM

Spanner Spencer wasted no time hooking up the Pi Zero's RCA connection to a classic cathode ray tube television. The Pi Zero Retro Gaming system is the real deal...

Quick Facts

- It works like a dream with Xbox 360 controllers
- The whole project took only two hours to build
- The Pi Zero can be overclocked to 1GHz for gaming
- It uses RetroPie, part of the standard NOOBS installation
- The Pi Zero can even play some original PlayStation games

It's fair to say that the Pi Zero caused more than a bit of splash. Our new favourite smallest computer topped Twitter and caused chaos at British newsagents when it was cover-mounted on *The MagPi*. Even President Obama got a copy.

Those lucky enough to get a Zero quickly created some great projects and discovered its nuances. That the Pi Zero has RCA output was not lost on Spanner Spencer, community manager at Element14, who realised it could be inserted inside an old television.

"I picked up some crummy old Grundig for 20 quid at a charity shop in Huddersfield," says Spanner. "It's nice to think that the first ever Pi Zero games machine did its bit for charity.

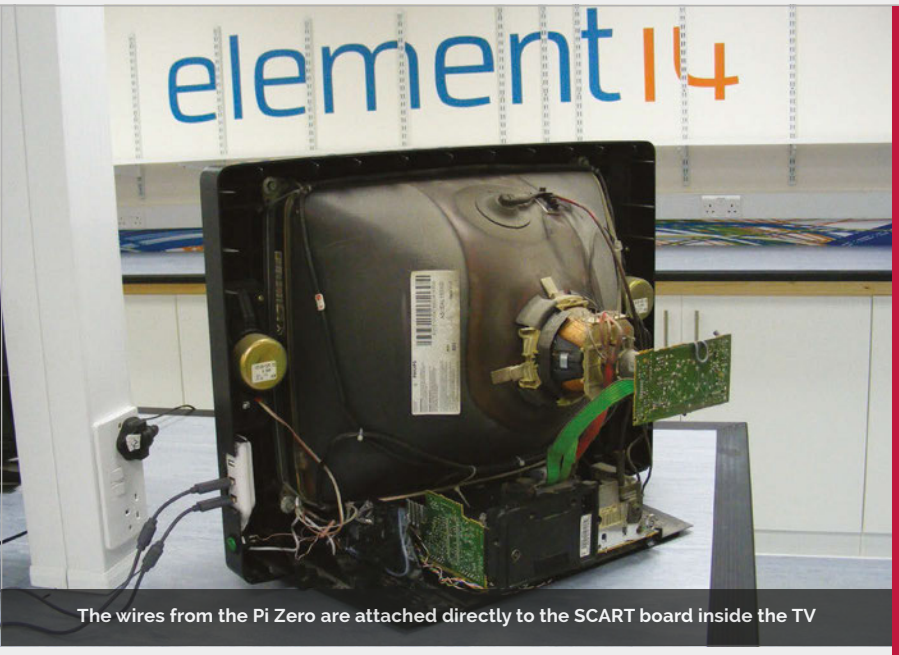
"Initially, I soldered a couple of flying leads to the back of the composite phono socket," he continues, "but there was something skewey going on, and I didn't get a picture. Instead, I switched to using the composite video in on the SCART socket and then soldering the wires onto the PCB underneath to avoid external wiring.

"The video signal came from the Pi Zero's composite output, which I soldered directly. It's a tad permanent, and it would have been better to add some pins

Using RetroPie on a classic CRT television is a great way to play old console games

Attaching a USB joypad (like this Xbox 360 controller) enables you to play games with minimal setup

A USB hub is placed on the side of the television. New games can be then added to RetroPie without opening the TV



The wires from the Pi Zero are attached directly to the SCART board inside the TV

> The Pi Zero is small enough to tuck inside the television

and a plug to make the board easier to remove, but there you go."

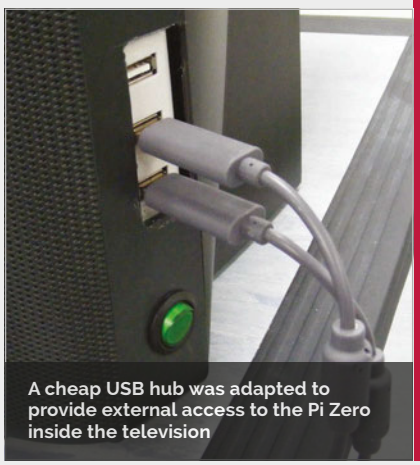
The Pi Zero is small enough to tuck inside the television, but that makes it hard to access. However, RetroPie enables you to transfer games by via a USB drive, so Spanner decided to use a USB hub to provide an external connection.

"We cobbled together a powered USB hub from a pound-shop hub," he tells us, "[and] soldered a socket onto the power lines on the PCB and replaced the cable with a butchered micro-USB one. We also ran a second micro-USB cable out of the hub to power the Pi Zero, so both the hub and the board run from a single 5V power supply.

"Other than having to change the composite output from the default NTSC to PAL in the Pi Zero's config file, it all worked right out of the box. We're able to play classic computer and console games like the Mega Drive, SNES, Amiga, ZX Spectrum, and loads of others."

The Pi Zero turns out to be great for retro gaming. "Given the extra RAM overhead and being able to crank the Pi Zero up to 1GHz, it's coped with RetroPie beautifully," says Spanner. "For the older systems, you don't even need to overclock it, so make it easy on yourself if you're planning on playing Atari 2600 games."

Safety announcement! Working inside old CRT televisions is dangerous (and not in a cool way). It is important to wear safety goggles and to discharge the electricity completely. Residual charge in an old CRT television can easily kill you. Please don't open an old TV set if you don't know what you're doing; it's wiser to connect the RCA from the Pi Zero to a SCART cable and plug this into the television, or use a newer HDMI monitor for your retro-gaming system.



A cheap USB hub was adapted to provide external access to the Pi Zero inside the television

PUTTING A PI ZERO INSIDE A TELEVISION

>STEP-01

Pi Zero RCA output

The Pi Zero comes with composite RCA output connectors on the board. Two wires are connected from the Pi Zero to pins 18 and 20 on a SCART socket.

>STEP-02

USB connection

Spanner took apart a USB hub and modified it to share power with the Pi Zero. A section of the TV is cut away to provide external access to the USB ports.

>STEP-03

NTSC to PAL

Pi Zero's config file needs to be changed to get colour on a UK PAL television. Edit the config file and uncomment the line `sdtv_mode=2`. The system is now ready to play games.

ADRIAN ATWOOD

Dad, engineer, tinkerer, woodworker, borker, and crazy hobbyist who is always building weird projects.
buildxyz.xyz

Quick Facts

- It took 90 hours to build
- This is Adrian's first Pi project
- Scratch is installed to encourage the kids to code
- He's adding wireless PS2 controllers very soon
- Adrian has more cool Pi stuff planned – keep an eye on his blog

RPIKIDS

Not a spin-off Saturday morning cartoon, RPiKids is a new entertainment system built for **Adrian Atwood**'s children

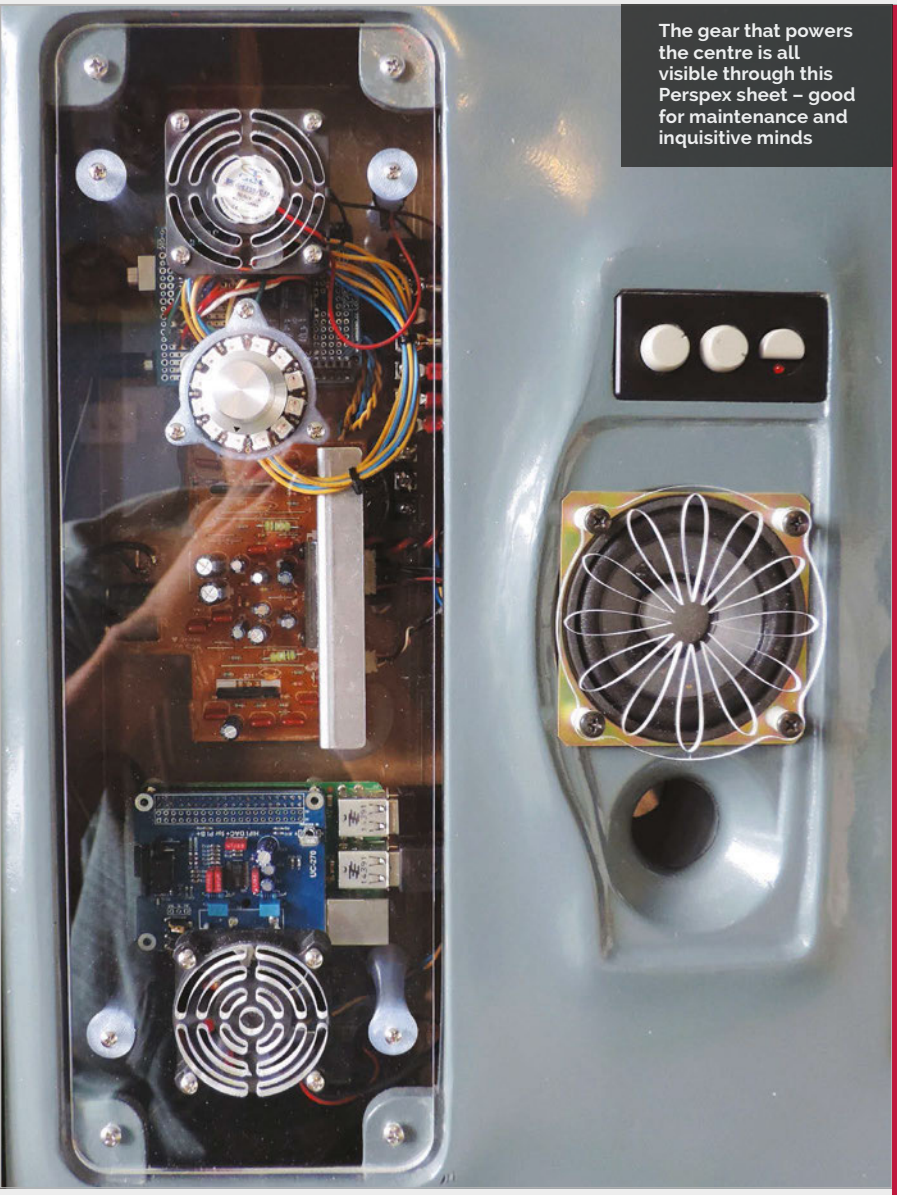
We've all seen wacky parental control methods for kids' entertainment. Our favourite is the exercise bike that powers a PlayStation, which may seem cruel but at least means there's a cut-off point for how much they're able to play. The best method is still the simplest: a key code and a timer so that the kids can then do something other than stare at a screen all day, freeing up the TV so you can binge-watch the entirety of *Once Upon a Time*'s third series. With the Raspberry Pi being a far more powerful option for operating an entertainment centre, it was only a matter of time before an intrepid hacker made the ultimate version for their kids. Meet RPiKids, created by Adrian Atwood.

"RPiKids is an all-in-one kids' entertainment centre with integrated parental control features," Adrian explains. "My kids can enjoy 16-bit games, watch movies and more, provided an adult has granted them access by entering a password and setting the timer."

The RPiKids is a mix of hardware; the system itself is powered by a Raspberry Pi, handling Kodi, Emulation Station and the like for the entertainment side. The passcode and timer are handled by an Arduino Uno using a rotary encoder (think bank safe), which can be pushed as a button to confirm selections. It's all built into a wonderful-looking, steampunky case that was handcrafted by Adrian himself from wood with 3D printed detailing.

There are USB access ports on the front for connecting controllers and portable media storage, and the hardware is visible behind a removable clear Perspex panel. While it all looks extremely impressive, Adrian claims it's an overall 'intermediate' level of build.

"I will say the Raspberry Pi configuration was very straightforward and a beginner can mirror my configuration," he tells us. "Coding the firmware for the Arduino parental controls, and having the Arduino and Pi 2 communicate with each other, was a little more complex. The case is no problem if you're into woodworking and auto body repair. I have access



The gear that powers the centre is all visible through this Perspex sheet – good for maintenance and inquisitive minds

> RPiKids is an all-in-one kids' entertainment centre with integrated parental control

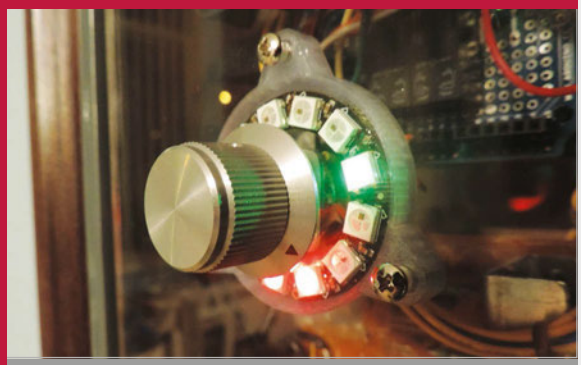
to a 3D printer and laser cutter and enjoy CAD modelling, but those parts could be contracted out or replaced."

Although he expected 'all sorts' of glitches, the system has worked extremely well for him and his children, requiring only minor adjustments to the brightness of the ring of NeoPixels used in conjunction with the encoder.

One thing Adrian recommends for anyone trying to replicate the project is an upgrade to the sound output: "The Pi definitely needed a DAC (I²S) sound card because there was a lot of hissing and artifacts with the onboard sound."

The kids have been very much enjoying it. Although it's an entertainment centre, he's also programmed an intentional security vulnerability into the lock mechanism, hoping his kids will learn something about security systems. We wish them the best of luck in figuring out to unlock 'unlimited mode'!

ENTERTAINING THE KIDS



>STEP-01

Dial it up

Using a digital encoder and a NeoPixel LED ring, the user (parent) enters a combination to unlock the system so the children can use it.

>STEP-02

Set a time

When unlocked, the encoder can then be used to set a timer on the system, from ten minutes to forever. Once selected, the Arduino lets the Raspberry Pi turn on.

>STEP-03

Shut it down

When time is up (or the off button is pressed), the Arduino part issues a shutdown command to the Pi. The Pi then shuts everything off safely, and the Arduino cuts power shortly afterwards.

STRATOS BOTSARIS

Stratos is a senior Java software engineer at Intrasoft International in Greece.

youtu.be/-Wjz6nY9r8c

Quick Facts

- The Spy Rover works for around 40 minutes
- At around 2mph, it won't break any speed records
- It took around three months to build
- The range is limited by the WiFi connection
- With port forwarding, it can be controlled over the internet

REMOTE CONTROL SPY ROVER

Fancy turning an old toy into a remote-control spy? **Lucy Hattersley** talks to Stratos Botsaris about his Spy Rover project

Java engineer Stratos Botsaris hacked a remote-control toy and turned it into a far cooler Pi-powered Spy Robot. If that wasn't excting enough, he now controls it from his Android phone while it bounces the video display to the screen.

A project like this deserves further investigation, so we caught up with Stratos to ask about the Spy Rover. "I did not want to build just another moving robot," he says. "At the same time, I was experimenting with the video recording capabilities of Raspberry Pi. So that was the moment that I came up with the idea of building a rover that could take real-time video.

"I wanted to use my Android programming skills to develop an application that could display live video to the user."

Rather than build a robot from scratch, Stratos took apart a Big Bargain King Force Excavator. With the top half removed, he slotted in an original Pi Model B hooked up to a breadboard, WiFi dongle, and USB webcam.

An L293D chip controls the motors. "The L293D is a motor driver integrated circuit that can simultaneously control two motors in either direction," says Stratos. "If I want to move the rover forward, I make both motors turn clockwise, and if I wish to get

the rover to reverse, I make both motors turn anticlockwise. In case the rover needs to turn right, I stop the right motor and make the left motor turn clockwise, and the same logic applies when the rover needs to turn left."

With the mechanics in place, Stratos turned his attention to the controller. It made sense to use a controller with a screen so that he could see through the webcam. Eventually, he decided to build a controller app for Android and control the Spy Rover directly from a phone.

"I have developed some Android applications in my spare time, so implementing an Android application for this project was not so difficult."

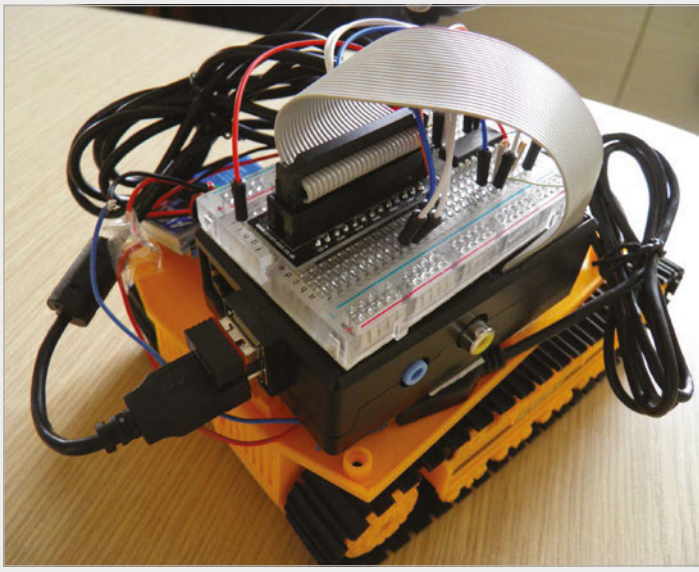
As Stratos discovered, streaming the video would prove a bigger challenge than controlling the Spy Rover. "After a lot of research, I found out that I had to use some libraries, written in C++, inside the Android app to accomplish my task. Fortunately, I found another project that had solved this problem, so by using parts of the source code, I was able to complete the implementation of the Android application."

> "I came up with the idea of cutting a USB cable and joining its power cables to those of the UBEC"

Aside from video streaming, the hardest part of the project was power. "The Raspberry Pi requires a constant power supply of 5V voltage and up to 3A current. After some research, I decided to buy a UBEC (Universal Battery Eliminator Circuit), which provides 5V from an input of 5.5V–20V and is capable of supplying up to 3A. Then I bought a battery box of six AA batteries to provide enough voltage (9V) to the UBEC.

"The next challenge I faced was connecting the UBEC to the Raspberry Pi. I had to find a way to connect the power output wires of the UBEC to the USB connector. Fortunately, I came up with the idea of cutting a USB cable and joining its two internal power cables to the output power cables of the UBEC; this was my real eureka moment."

Stratos aims to make the next project faster. "The plan is to transfer the rover to a plastic toy car with normal wheels."

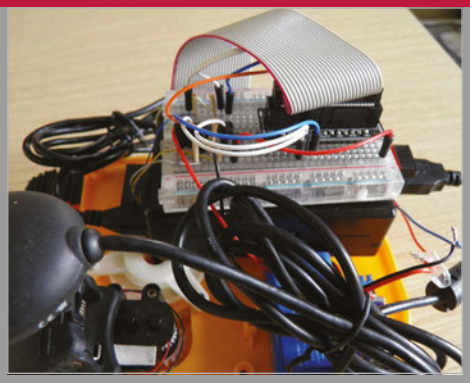


Above The parts are wired up using a breadboard. This enabled the Raspberry Pi to be tested with the L293D controller and motors

HACK YOUR SPY ROVER

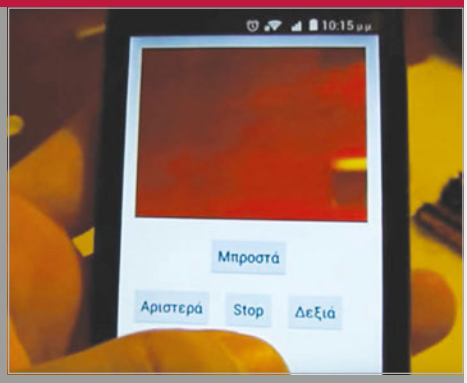
>STEP-01
Dismantle the toy

Rather than build a robot from scratch, Stratos decided to modify a remote-control toy. The Spy Rover started life as a Big Bargain King Force Excavator. This toy comes complete with tracks, as well as motors that can be modified to work with the Raspberry Pi.



>STEP-02
Adding the Raspberry Pi

The top half of the King Force toy is an excavator mechanism, which was removed to provide a base for the Raspberry Pi, WiFi dongle, webcam, and batteries. The L293D controller is slotted into a breadboard and wired to the original motors inside.



>STEP-03
The Android controller

The toy connects to a local WiFi network, and a custom-built Android app is used to send commands to the Raspberry Pi. Meanwhile, the view from the webcam is bounced back to the Android app. The result is a remote-controlled Spy Rover.

RON OSTAFICHUK

Ron is a tech enthusiast who has been playing on computers since the Z80. He is married with three kids and lives in Airdrie, Alberta, Canada.
ostafichuk.com

Quick Facts

- Ron used just a jigsaw, drill, and utility knife
- The whole project cost around £60 to build
- It uses a worn out battery but runs for three hours
- Ron plans to add ultrasonic and bump sensors to the Dalek
- The Dalek 2.0 will be based on a mobility scooter

RASPIMOWER DALEK

Ron Ostafichuk set out to build a robot lawnmower and ended up with a fully automated Dalek (you know how it is)

Daleks are the most feared race in the universe, so what could be more fun than building your own Raspberry Pi-powered Dalek?

That's what developer Ron Ostafichuk thought. "I've been tinkering with the Raspberry Pi ever since it came out," he tells us, "and I've seen many small robot projects, but decided that a big robot would be much more fun."

The Dalek didn't start out as a Dalek: Ron wanted to build a robotic lawnmower. "I have a huge lawn and I wanted to show my kids that you could build a large robot, using spare parts and scraps."

The wooden base contains two 12V motors recycled from a life spent adjusting the seats on cars. These motors are connected to the big off-road wheels using a V-belt system.

"Once I had the motorised base completed, my kids hopped onto it and proceeded to ride it around the neighbourhood. They loved it, and remarked how great it would be to ride it around for Halloween. Since I am a pretty big *Doctor Who* fan, the idea that came to me was to turn it into a Dalek!"

For power, Ron is using an old 12V deep-cycle camper battery. "[It] was no longer strong enough to use in my camper," says Ron, "[but] it lasted for around three hours of continuous operation this Halloween."

"I am running the standard Raspbian distro on the Pi," says Ron, and the Raspimower Dalek control code is written in C++ (available on Bitbucket – **magpi.cc/1HWNV8j**).

The Dalek is a body that sits on top of the Raspimower base. The frame was built from ¾-inch (19mm) chipboard, and the outside from ¼-inch (6mm) sheeting.

RASPIMOWER DALEK

>STEP-01
Raspimower base
The base of the Raspimower Dalek is a frame to house a 12V battery and motors connected to the wheels via a V-belt system. It's designed to work as a robotic lawnmower and is controlled by a Raspberry Pi.

>STEP-02
Dalek frame
Turning the Raspimower base into a Dalek required a chipboard frame. This was constructed from ¾-inch chipboard and ¼-inch sheeting left over from a shed-building project.

>STEP-03
The Dalek
A used bin, plunger, and aluminium pipe create the rest of the Dalek, and the whole thing is painted in glossy black paint.

"This was all left over from building a shed," reveals Ron. One arm is a toilet plunger and the other is an aluminium pipe wrapped with wire, and the whole thing is painted over with glossy black paint.

yard," jokes Ron. Speaking of extermination, no Dalek would be complete without its famous voice command. "It has some old computer speakers inside that are very loud; they actually

> "The old computer speakers inside are so loud, they actually vibrate the sides of the Dalek"

The Dalek was originally controlled with a wireless keyboard, but Ron has built a web interface to control the Raspimower Dalek. Ron has also bought an old Zappy mobility scooter for $60CND (around £40) and plans to use it to replace the 12V motors. He explains that it's much more powerful and mechanically sound.

"[The Dalek] is nowhere near completed and constantly changing, as I have lots of plans and want to make it autonomous so it will actually 'exterminate' any long blades of grass in my

vibrate the sides of the Dalek. It has about 15 sound clips from *Doctor Who*, but 'exterminate' is my favourite one!

"It was a huge hit this Halloween, and as the kids would approach, I would turn on all the lights and rotate the robot towards them while belting out 'exterminate!' or 'destroy!'. Watching the kids react was so much fun; my favourite reaction came from a little girl who turned to her dad and matter-of-factly said, 'Well, I guess we know which house we will never ever go to again!' "

After the Dalek has done its Halloween duty, the base returns to being a robot lawnmower

SUBSCRIBE TODAY!

Subscribe to the Official Raspberry Pi mag today for a whole host of benefits

Subscription benefits

- Save up to 25% on the price
- Free delivery to your door
- Never miss a single issue
- Get it first (before stores)

NEW US SUBS OFFER!
imsnews.com/magpi

SAVE UP TO **25%**

Pricing

Get six issues:

£30 (UK)
£45 (EU)
$69 (USA)
£50 (Rest of World)

Subscribe for a year:

£55 (UK)
£80 (EU)
$129 (USA)
£90 (Rest of World)

Direct Debit

£12.99 (UK) (quarterly)

How to subscribe:

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

Available on the App Store

Get it on Google play

SUBSCRIPTION FORM

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

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

Mag#41

YOUR DETAILS Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name ..... Surname .....

Address .....

.....

Postcode ..... Email .....

Daytime phone ..... Mobile .....

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

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

First name ..... Surname .....

Address .....

Postcode ..... Email .....

PAYMENT OPTIONS

1 DIRECT DEBIT PAYMENT £12.99 every 3 issues (UK only)
Instruction to your bank or building society to pay by Direct Debit

DIRECT Debit

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

Service user number 8 3 8 7 7 3

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

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

Address .....

.....

..... Postcode .....

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

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

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

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

Signature ..... Date ☐☐/☐☐/☐☐

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

SUBSCRIPTION PRICING WHEN PAYING BY CHEQUE OR CREDIT/DEBIT CARD

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

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

2 CHEQUE

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

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

Card number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

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

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

Signature ..... Date ☐☐/☐☐/☐☐

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

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

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

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

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

SIMON MONK

Simon Monk is the author of the *Raspberry Pi Cookbook* and *Programming the Raspberry Pi: Getting Started with Python*, among others.

simonmonk.org
monkmakes.com

PI MI-LIGHT

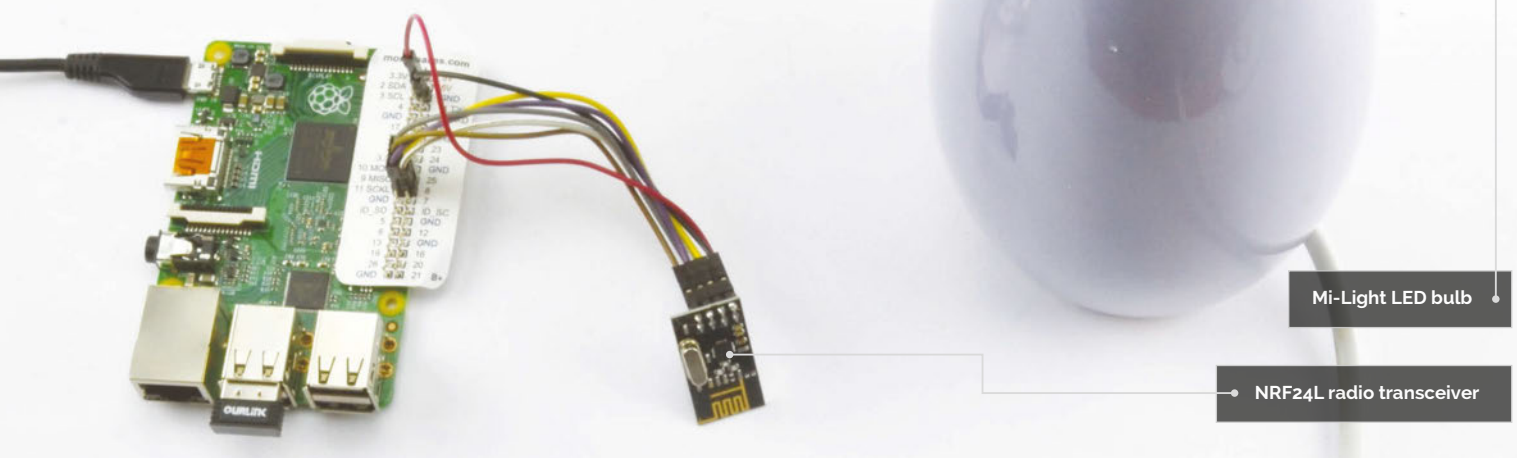
Solve real-world electronic and engineering problems with your Raspberry Pi and the help of renowned technology hacker and author, **Simon Monk**

You'll Need

- NRF24L01 2.4GHz wireless radio transceiver module (eBay)
- Mi-Light RGB LED light bulb
- Mi-Light 2.4GHz remote control
- 7 × female-to-female jumper wires

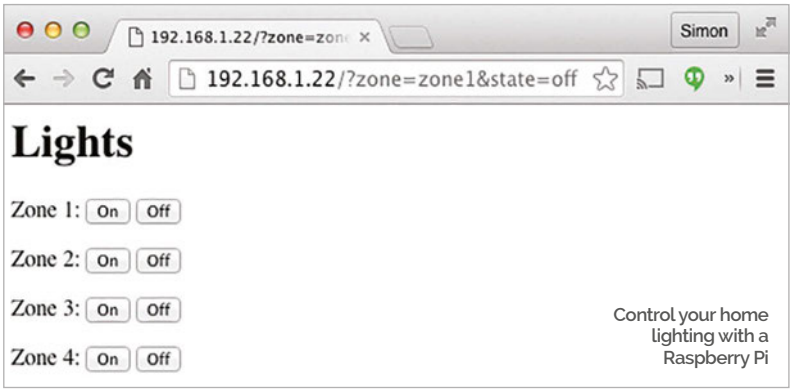
Mi-Light light bulbs look just like normal LED light bulbs and are available at a similar price, but they include a 2.4GHz RF radio link. This can be used with an RF remote control to switch lights on and off, with the lights grouped into four zones. The remote control also allows you to vary the brightness and colour of the light. Mi-Light produces a ready-made module that allows you to link the lamps to your WiFi router and then control the lighting with a smartphone app. However, by using a £3 ($5) radio module connected to a Raspberry Pi, you can let your Raspberry Pi take control of Mi-Light bulbs in your home, opening up all sorts of possibilities for home automation.

One possible use is to have the Raspberry Pi operate as a web server, providing you with a web interface that will let you turn lights on and off from the browser of any device connected to your network.



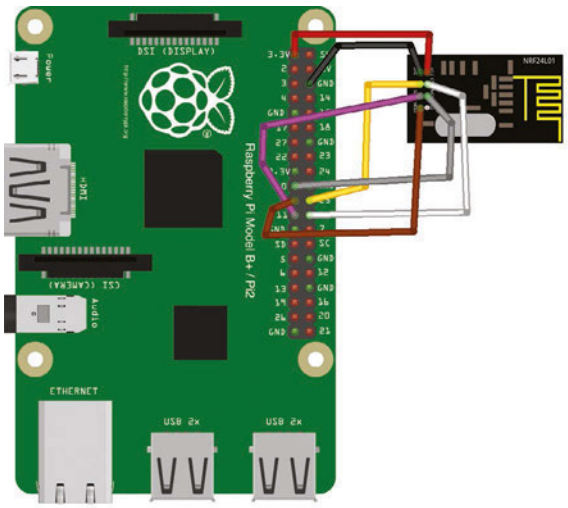
This project is based on an original blog post by Torsten Tränkner that you will find in German here: **torsten-traenkner.de**. The project has only been tested on a Raspberry Pi 2, and may need some adaptation for other models of Pi.

As you'll see from the list of required components, this project does not involve any soldering. The RF module is connected to the Raspberry Pi using female-to-female jumper wires. Even though the Raspberry Pi will eventually take over the operation of the lights, you do need a Mi-Light remote control to set things up in the first place.

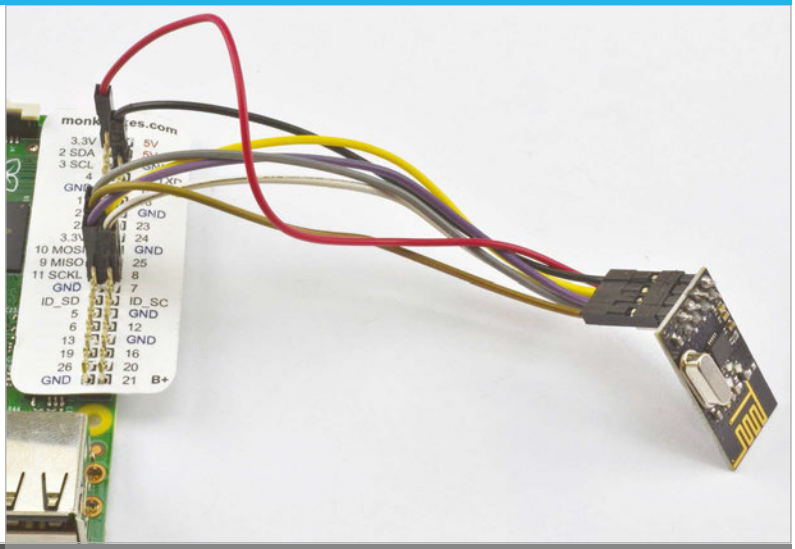


Control your home lighting with a Raspberry Pi

BUILDING YOUR PI-MI-LIGHT CONTROLLER



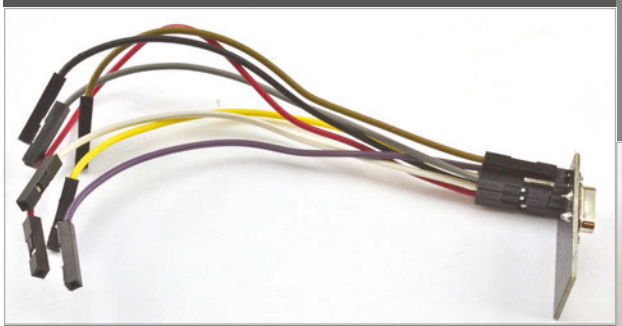
You should wire up the RF module with your Raspberry Pi powered down, and check very carefully that the connections are all correct before you power it back up again. Using a GPIO template like the Raspberry Leaf (by MonkMakes) will help you find the correct connections on the GPIO connector.



>STEP-01

Attach leads to the RF module

Start by connecting jumper wires to the RF module using the wiring diagram as a reference. You will need connections to every pin except the pin in the bottom right of the connector (pin 8). Using leads of the same colour as the wiring diagram will help you make the correct connections.



>STEP-02

Connect the RF module

Connect the leads from the RF module to the GPIO header as follows:

Pin 1 (GND) of the RF module to GND on the GPIO header.
Pin 2 (VCC) of the RF module to 3.3V on the GPIO header.
Pin 3 (CE) of the RF module to GPIO 25.
Pin 4 (CSN) of the RF module to GPIO 8.
Pin 5 (CLK) of the RF module to SCLK (GPIO 11) on the GPIO header.
Pin 6 (MOSI) of the RF module to MOSI (GPIO 10) on the GPIO header.
Pin 7 (MISO) of the RF module to MISO (GPIO 9) on the GPIO header.

>STEP-03

Pair the Mi-Light

For this step, you will need to plug your light bulb into an appropriate light socket. Then turn on the normal light switch and immediately press the On button for one of the four light zones on the Mi-Light remote. The lamp will blink three times as confirmation. Once paired, the remote will be able to turn the light on and off and make other adjustments.

This project requires quite a lot of software setup before you can go ahead and use it.

The radio module uses the Raspberry Pi's SPI interface, which needs to be enabled, so run raspi-config using the command:

```
sudo raspi-config
```

Scroll down to the **Advanced** option, select it, and then select **SPI**. Enable it, and when asked if you want the SPI module to load automatically, say Yes.

> This code is all in C++ but can be called from our Python program

Next, you need to install and build the NRF24 library for the RF module by issuing the following commands:

```
git clone https://github.com/TMRh20/RF24
cd RF24
make all
sudo make install
```

After that, you need to download a messaging library that provides a higher-level interface to the basic RF module using the following commands:

```
cd ..
git clone https://github.com/mysensors/Raspberry
cd Raspberry
make
sudo make install
```

Now download the code that Torsten Tränkner wrote using the following commands:

Below Turn light zones on/off from a phone browser

```
cd ..
wget http://torsten-traenkner.de/wissen/
smarthome/openmilight_raspberry_pi.tgz
tar xzvf openmilight_raspberry_pi.tgz
```

This code is all in C++ and so that we can make use of it in Python, your expert has written a C++ program called **send_cmd**, which is designed to be called from Python with a command message to be sent to the Mi-Light. Download this program and the Python code for the project from your Pi's command line, using:

```
git clone https://github.com/simonmonk/
pi_magazine.git
```

Copy all the files in the project folder **11_mi_light** into the **openmilight** folder, and then compile two of the programs by using the commands:

```
cp pi_magazine/11_mi_light/* openmilight
cd openmilight
sh ./comp.sh
```

You also need to install the Python Bottle library that will be used for the web server, using the command:

```
sudo apt-get install python-bottle
```

Before you can run the main program, you need to find the ID of the Mi-Light remote that you used to pair with the Mi-Light so that your Raspberry Pi can impersonate it.

Run the program **openmilight** using the command below, and then press the On button for Zone 1 on the remote a couple of times. Each time you press the button, you should see a stream of hexadecimal number like the ones in the example below.

```
sudo ./openmilight
in listening mode
B0 A1 56 41 C1 03 72 .
B0 A1 56 41 C1 03 73 .
```

The first three digits (in this case B0, A1, and 56) will be different for your remote, so quit the program using **CTRL+C** and edit the file **lights.py**, changing the value of the **ID** variable at the top of the file to be the three hex digits for your remote.

Finally, you are ready to fire up the web server using the following command:

```
sudo python lights.py
```

Point a convenient web browser at the IP address of your Raspberry Pi and you will see a webpage with On and Off buttons for all four zones. Click the buttons for Zone 1 and you should be able to turn all the lights connected to that zone on and off.

Language
>PYTHON 2.7

DOWNLOAD:
magpi.cc/1NZhcLv

Lights.py

```
bottle import route, run, template, request
import os

ID = 'B0 A1 56'
MIDDLE = ' 06 C9 '

ZONES = {
    'zone1': {'on': '03', 'off': '04'},
    'zone2': {'on': '05', 'off': '06'},
    'zone3': {'on': '07', 'off': '08'},
    'zone4': {'on': '09', 'off': '0A'}}

def light_on(zone):
    print(zone + "on")
    send(ZONES[zone]['on'])


def light_off(zone):
    print(zone + "off")
    send(ZONES[zone]['off'])
```

```
from
        def send(code):
            global ID, MIDDLE
            for i in range(1, 5):
                message = ID + MIDDLE + code +
        ' 00'
                os.system('./send_cmd "' +
        message + '"')

        # Handler for the home page
        @route('/')
        def index():
            zone = request.GET.get('zone', 'zone1')
            state = request.GET.get('state', 'off')
            if state == 'on':
                light_on(zone)
            else:
                light_off(zone)
            return template('home.tpl')

        # Start the webserver running on port 80
        run(host="0.0.0.0", port=80)
```

How the code works

You will probably find it handy to have the code up in an editor while we go through it.

The program starts by importing the **bottle** and **os** libraries. The **os** library is needed to invoke the **send_cmd** C program from Python.

The constant **ID** needs to be set to the three hex digits for your remote, as described earlier. The codes in **MIDDLE** do not need to change. Note that the spaces in both these constants must be kept as they are, so that when the hex message is constructed, there are spaces between each hex digit.

The Mi-Light remote control allows up to four zones to be defined, and you can attach multiple lights to each zone. There are separate On and Off codes that control each of the zones independently. These are stored in the variable **ZONES**, which is a dictionary or dictionaries, making it easy to look up the appropriate hex code for the command you want.

Two functions (**light_on** and **light_off**) switch all the lights on or off for the zone name specified as their parameter. Both these functions use the **send** function to actually send the command to the RF module via the C program (**send_cmd**). The **send** function first constructs a message string by concatenating the **ID**, **MIDDLE** section, command **code**, and finally **00** into a message string. The C program is then called five times, passing the message as a parameter. Since it can be a little unreliable, sending the message five times makes it almost certain to get through to the light bulb.

Next, there is the web server part of the code, contained in the **index** function. This expects the webpage to provide two request parameters: the zone name (**zone**) and whether it is to be turned on or off (**state**). This information is then used to call either **light_on** or **light_off**. At the end of the index function, the contents of the template **home.tpl** are returned, to provide the browser with the HTML for the web interface.

The final line of code starts the web server running. Switch over to a browser on the Raspberry Pi, or another computer on your network, and type the IP address of your Pi into the address bar. To discover the IP address of your Raspberry Pi, type the command below into LXTerminal:

```
hostname -I
```

It will be at the start of the response as four numbers separated by dots; for example, 192.168.1.22.

Using your PiMi-Light Controller

As well as controlling the lights from the browser on your computer, you can just as easily use the browser on your smartphone, as long as it is connected to the same network as your Raspberry Pi.

You could also adapt the alarm clock program back in issue 33 of *The MagPi* to turn the lights on and off at certain times.

This project simply turns the lighting on and off. Considerable work has been done in reverse-engineering the Mi-Light protocol to figure out what all the codes do. So, if you want to extend this project to adjust the colour and brightness of the lights, then you might want to look at Henryk Plötz's work: **magpi.cc/1NZfcmt**

NEXT MONTH

In the next project in this series, you will learn how to make the ultimate in geek chic: a binary clock.

Sonic Pi PART 6

SAM AARON

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

BINARY BIZET

You'll Need

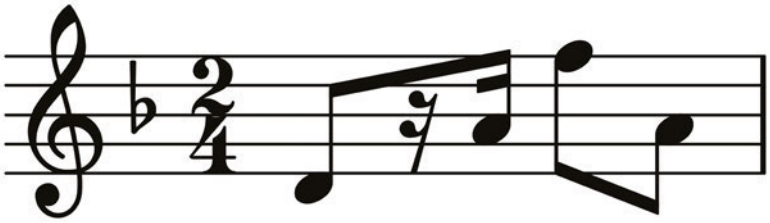
- Raspberry Pi running Raspbian
- Sonic Pi v2.7+
- Speakers or headphones with a 3.5mm jack
- Update Sonic Pi: `sudo apt-get update && sudo apt-get install sonic-pi`

In this month's guide we're going to bring a classical operatic dance piece straight into the 21st century, using the awesome power of code...

Let's jump into a time machine and head back to the year 1875. A composer called Bizet had just finished his latest opera, *Carmen*. Unfortunately, like many exciting and disruptive new pieces of music, people initially didn't like it at all because it was too outrageous and different. Sadly, Bizet died ten years before the opera gained huge international success and became one of the most famous and frequently performed operas of all time. In sympathy with this tragedy, let's take one of the main themes from *Carmen* and convert it to a modern format of music that is also too outrageous and different for most people in our time: live-coded music!

Decoding the Habanera

Trying to live-code the whole opera would be a bit of a challenge for this tutorial, so let's focus on one of the most famous parts: the bassline to the Habanera.



This may look extremely unreadable to you if you haven't yet studied music notation. However, as programmers we see music notation as just another form of code, only it represents instructions to a musician instead of a computer. We therefore need to figure out a way of decoding it.

Notes

The notes are arranged from left to right, like the words in this magazine, but also have different heights. *The height on the score represents the pitch of the note.* The higher the note on the score, the higher the pitch of the note.

In Sonic Pi, we already know how to change the pitch of a note: we either use high or low numbers such as `play 75` and `play 80`, or we use the note names such as `play :E` and `play :F`. Luckily, each of the vertical positions of the musical score represents a specific note name, as shown here...

Rests

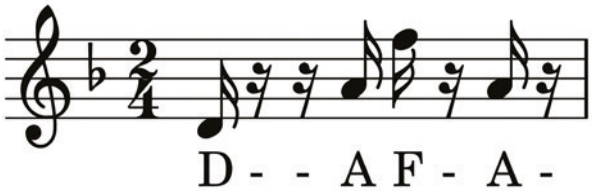
Music scores are an extremely rich and expressive kind of code, capable of communicating many things. It therefore shouldn't come as much of a surprise that musical scores can not only tell you what notes to play, but also when *not* to play notes. In programming, this is pretty much equivalent to the idea of 'nil' or 'null' – the absence of a value. In other words, not playing a note is like the absence of a note.

If you look closely at the score, you'll see that it's actually a combination of black dots with lines which represent notes to play, and squiggly things which represent the rests. Luckily, Sonic Pi (v2.7+) has a very handy representation for a rest – `:r`. So if we run `play :r`, it actually plays silence! We could also write `play :rest`, `play nil`, or `play false`, which are all equivalent ways of representing rests.

Rhythm

Finally, there's one last thing to learn how to decode in the notation: the timings of the notes. In the original notation, you'll see that the notes are connected with thick lines called beams. The second note has two of these beams, which means it lasts for a 16th of a beat. The other notes have a single beam, which means they last for an 8th of a beat. The rest have two squiggly beams, which means they also represent a 16th of the beat.

When we decode and try to understand new things, a handy trick is to try to make everything as similar as possible to attempt to spot any relationships or patterns. When we rewrite our notation purely in 16ths, you can see that our notation just turns into a nice sequence of notes and rests...



Recoding the Habanera

We're now in a position to start translating this bassline to Sonic Pi. Let's encode these notes and rests in a ring:

```
(ring :d, :r, :r, :a, :f5, :r, :a, :r)
```

Let's see what this sounds like. Throw it in a live loop and tick through it:

```
live_loop :habanera do
  play (ring :d, :r, :r, :a, :f5, :r, :a,
:r).tick
  sleep 0.25
end
```

Fabulous: that instantly recognisable riff springs to life through your speakers. It took a lot of effort to get here, but it was worth it. High-five!

Moody synths

Now we have the bassline, let's recreate some of the ambience of the operatic scene. One synth to try out is **:blade**, which is a moody 1980s-style synth lead. Let's try it with the starting note **:d**, passed through a slicer and reverb:

```
live_loop :habanera do
  use_synth :fm
  use_transpose -12
  play (ring :d, :r, :r, :a, :f5, :r, :a,
:r).tick
  sleep 0.25
end

with_fx :reverb do
  live_loop :space_light do
    with_fx :slicer, phase: 0.25 do
      synth :blade, note: :d, release: 8,
cutoff: 100, amp: 2
    end
    sleep 8
  end
end
```

Now, try the other notes in the bassline: **:a** and **:f5**. Remember, you don't need to hit Stop; just modify the code while the music is playing and hit Run again. Also, try different values for the slicer's **phase:**, such as **0.5**, **0.75**, and **1**.

Bringing it all together

Finally, let's combine all the ideas so far into a new remix of the Habanera. You might notice that I've included another part of the bassline as a comment. Once you've typed it all into a fresh buffer, hit Run to hear the composition. Now, without hitting Stop, *uncomment* the second line by removing the **#** and hit Run again; how marvellous is that? Now, start mashing it around yourself and have fun!

```
use_debug false
bizet_bass = (ring :d, :r, :r, :a, :f5, :r, :a, :r)
#bizet_bass = (ring :d, :r, :r, :Bb, :g5, :r, :Bb, :r)

with_fx :reverb, room: 1, mix: 0.3 do
  live_loop :bizet do
    with_fx :slicer, phase: 0.125 do
      synth :blade, note: :d4, release: 8,
        cutoff: 100, amp: 1.5
    end
    16.times do
      tick
      play bizet_bass.look, release: 0.1
      play bizet_bass.look - 12, release: 0.3
      sleep 0.125
    end
  end
end

live_loop :ind do
  sample :loop_industrial, beat_stretch: 1,
    cutoff: 100, rate: 1
  sleep 1
end

live_loop :drums do
  sample :bd_haus, cutoff: 110
  synth :beep, note: 49, attack: 0,
    release: 0.1
  sleep 0.5
end
```

BBC Ten Pieces

If you've enjoyed this creative mashup of a classical piece, you should definitely get yourself and your school involved with the BBC Ten Pieces project. Check out the website to see all the other fantastically creative responses people are making to classical music: **bbc.co.uk/tenpieces**. Last year, one of the schools coded their way to the finals with Sonic Pi; let's see what you can do this year!

TONY GOODHEW

Taught programming since 1967 (IBM 1130 with FORTRAN IV). BBC Micro teacher trainer, now using Raspberry Pi and Arduino.

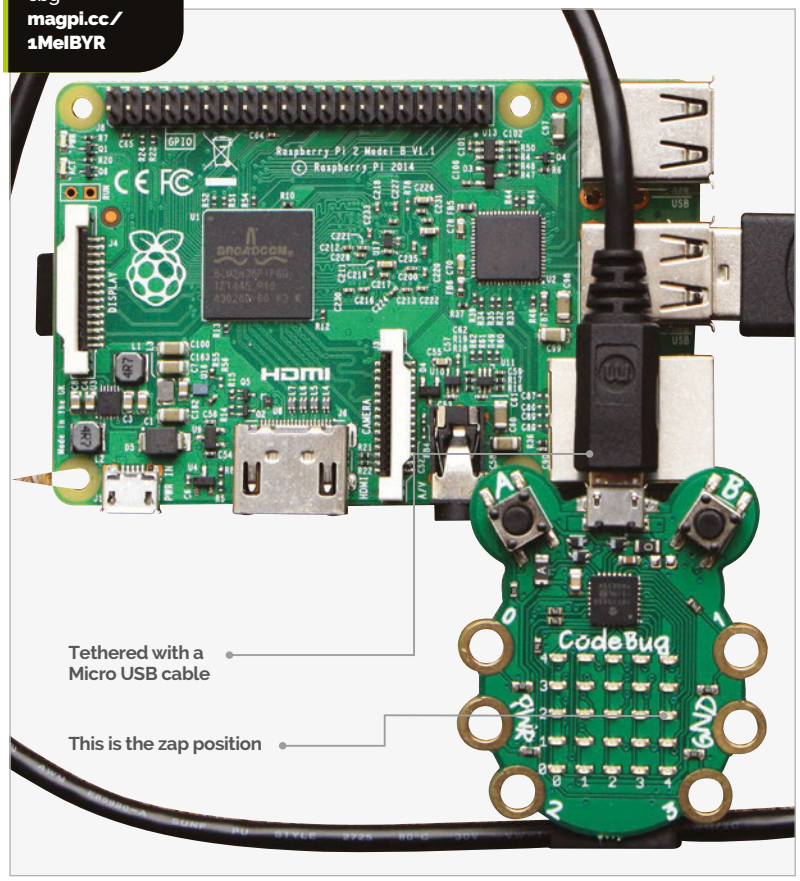
PROGRAM CODEBUG WITH PYTHON 3

You'll Need

- CodeBug **magpi.cc/1MelyfA**
- USB Micro B cable (included)
- Python libraries: python3-serial & python3-codebug-tether
- CodeBug script: codebug-tether.cbg **magpi.cc/1MelBYR**

Move on from Blocky programming and control your CodeBug with all the power of Python...

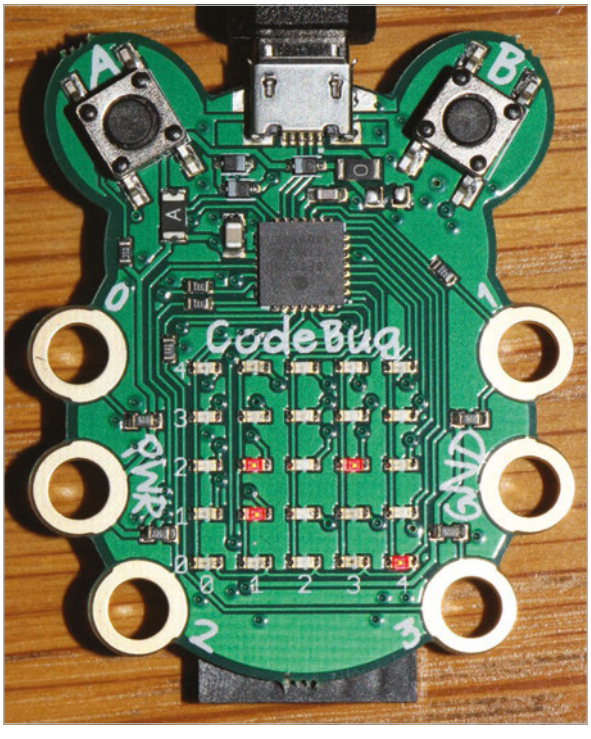
The CodeBug is simple to set up and start programming with the Scratch-like Blocky interface on the website. Once you have mastered this, you'll probably want to move on to the next stage and take full control with a more powerful language like Python 3. This allows you to include more ambitious data structures, such as lists and tuples, and build larger projects with procedures. Your expert started to wonder how good a game you could build with just a square of 25 LEDs and a switch. This is his first attempt.



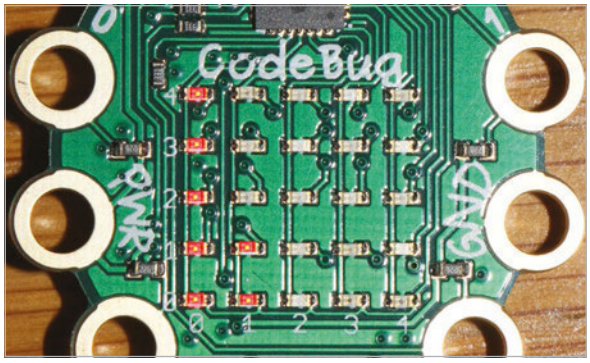
>STEP-01

Game rules

We need a simple game in which you press a button when a target is in a certain position. Back in the late 1970s, your expert used to set a 'Zap the Rat' game as homework while teaching Commodore PET or Ohio Scientific Basic. You have to click a button as a moving 'rat' target passes a certain 'zapping point' on a circuit. In this version, the speed increases each time you successfully hit a rat, and you need to hit three rats with as few attempts as possible. It would be nice to display the hits while the game is running, and a final score to show how many times you missed the target. Can all of this be fitted on a 5×5 LED display?



Above This shows the rat at (4,0), the zap position marker, and hits counter at two zaps



Above The final score: seven missed opportunities to zap the rats

>STEP-02

Planning the display

The circuit will be all the outside edge LEDs, with the rats starting their run from the bottom-left corner: (0,0). The zap cell will be at (4,2), with a permanently lit LED at (3,2) to mark it. Hits can be indicated by a column of lit LEDs rising from (1,1) to (1,3). Button 'B' will be used to zap the rat. The final score can be displayed with either a character 'O' to indicate no missed zap opportunities – quite difficult to achieve – or lit LEDs (0 –25) for each miss. For a 'hit' to register, the button must be pressed while the zap position LED is lit; holding the button down won't work.

>STEP-03

Setting up the CodeBug

To run the CodeBug in tethered mode, you'll need to download and install a program called **codebug_tether.cbg**. Once installed in the normal way, the CodeBug temporarily becomes a slave device to your Raspberry Pi and obeys instructions sent from Python 3. (This is rather like using Nanpy to control an Arduino from a Pi.)

Two additional Python libraries, **python3-serial** and **python3-codebug-tether**, need to be installed on the Pi. Point your browser to **magpi.cc/1MeIWdR** and follow the excellent instructions from Thomas Macpherson-Pope.

Open IDLE 3 and type in the **Zap_the_rat.py** Python script. Save it and run it.

>STEP-04

Things to do

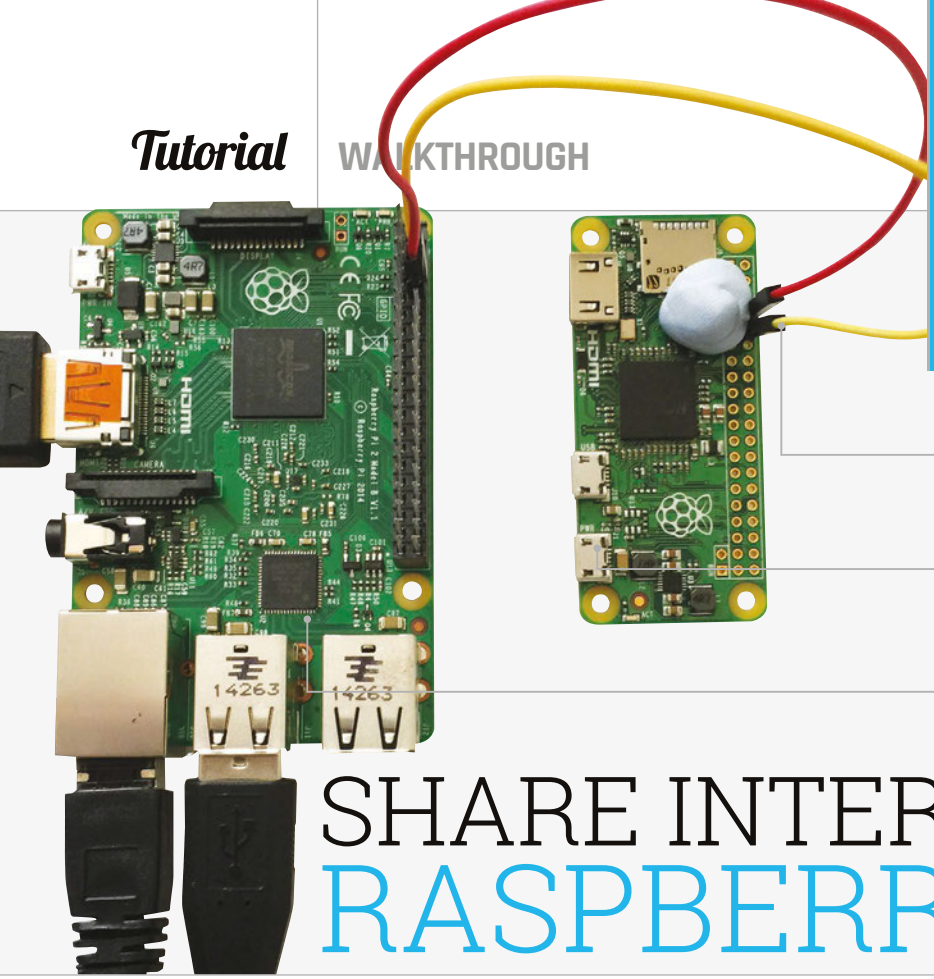
There are several modifications and additions you could make to the script:

- Randomise the rat's starting position
- Randomise the direction in which the rat runs around the circuit
- Improve the zero final score to a square, expanding and contracting from position (2,2) to the edge and back again five times as a celebration.

Zap_the_rat.py

Language >PYTHON

```
# Zap the Rat - Python3 Game for Tethered CodeBug
# Tony Goodhew - 5 Oct 2015
import codebug_tether
import time
cb = codebug_tether.CodeBug()
cb.clear()

# Circuit LED co-ordinates - round the edge from bottom-left
LEDx =(0,0,0,0,0,1,2,3,4,4,4,4,3,2,1)
LEDy =(0,1,2,3,4,4,4,4,4,3,2,1,0,0,0,0)
for i in range(0,-66,-1):  # Display game title
    cb.write_text(i,0,'Zap the Rat', direction="right")
    time.sleep(0.1)

score = 0             # Rats zapped!
opps = 0              # Zap opportunities
p = 0                 # Position of rat on circuit (0-15)
delay = 0.05          # Initial delay between rat moves
old_sw = 0                    # Last value of switch B
running = True                # Loop control variable
cb.clear()
cb.set_pixel(3,2,1)           # Show Zap position

while running:
    cb.set_pixel(LEDx[p],LEDy[p],1)   # Show rat
    if p == 10: opps = opps +1        # Incr opps - in (4,2)
    time.sleep(delay - score * 0.007) # Wait - gets faster
    sw = cb.get_input('B')            # Read switch B
    pixel = cb.get_pixel(4,2)         # Rat in Zap position?
    cb.set_pixel(LEDx[p],LEDy[p],0)   # Hide rat
    if old_sw == 0 and sw == 1 and pixel == 1: # Hit?
        score = score +1              # Increment hits
        cb.set_pixel(1,score,1)       # Show hits
        if score == 3:                # Game over?
            running = False           # Stop looping
        p = -1              # Position of next rat
    old_sw = sw             # Store switch value
    p = p + 1               # Update rat position
    if p > 15:              # Rat pointer out of range?
        p = 0               # Reset to start position

cb.clear()  # Game over - display opportunities
opps = opps - 3  # Remove successful opportunities
print("Missed Opportunities were ",opps)
if opps > 25:  # Reduce missed opps to 25 max
    opps = 25
if opps == 0:
    cb.write_text(0,0,'O') # Zero missed opps
else:
    for x in range(0,5):   # Display missed opps
        for y in range(0,5):
            if opps > 0:
                cb.set_pixel(x,y,1)
            else:
                cb.set_pixel(x,y,0)
            opps = opps - 1
time.sleep(5)
cb.clear()  # Tidy display
```



DAVE HONESS

All-round space geek and employee of the Raspberry Pi Foundation, currently heading up the Astro Pi mission, because SPAAACE!

raspberrypi.org
twitter.com/dave_spice

SHARE INTERNET TO THE RASPBERRY PI ZERO

You'll Need

- A Raspberry Pi A, B, A+, B+ or 2
- A Raspberry Pi Zero
- 2× SD cards, both with a fresh install of Raspbian Jessie
- 2× male-to-female jumper cables
- A blob of Blu-Tack or plasticine

Use one Raspberry Pi to share its internet with a Pi Zero through the GPIO ports, using Point-to-Point Protocol

Point-to-Point Protocol (PPP) is how everyone used to access the internet, back in the days of dial-up. You can use it with your Pi Zero to piggyback on the internet connection of another Pi which is already online. This saves the single micro-USB data connector to use with something else, if you don't want to hook up a USB hub and need to connect more than just a WiFi dongle.

For the following steps, the host Raspberry Pi needs to have internet access. First, enable IPv4 forwarding by going into the terminal and using:

```
sudo nano /etc/sysctl.conf
```

Find the line **net.ipv4.ip_forward=1** and remove the **#** symbol at the start of the line. Save and exit using **CTRL+X** and then pressing **Y**. Now run this command to reload the config:

```
sudo sysctl -p
```

Next install the ppp package:

```
sudo apt-get install ppp -y
```

The plan is to use the UART GPIO pins to link the host Pi to the Zero. We need to use physical GPIO pin numbers 8 and 10; these pins are otherwise known as TXD (transmit) and RXD (receive).

By default, there is a login prompt running on the UART, so we first need to disable it:

```
cd /boot
sudo cp cmdline.txt old_cmdline
sudo nano cmdline.txt
```

You should see this:

```
dwc_otg.lpm_enable=0
console=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline fsc
```

Remove **console=ttyAMA0,115200**, then save and exit. For the speed of the data link to be acceptable, we need to speed up the UART internal clock:

```
sudo nano config.txt
```

Append **init_uart_clock=16000000** to the bottom of the file, then save and exit. Then we need to disable the login prompt service that runs on the UART:

```
sudo systemctl disable serial-getty@
ttyAMA0.service
```

Reboot the Pi. We now need to create an options file that will configure the PPP service when it starts:

```
sudo nano /etc/ppp/options.ttyAMA0
```

Copy the following lines into the file:

```
noauth
nocrtscts
xonxoff
passive
local
maxfail 0
nodetach
192.168.0.100:192.168.0.101
persist
proxyarp
```

Save and exit. The two IP addresses need to be available on your LAN, and should be different to the one allocated to the host Pi's Ethernet adaptor. Next, create a config file that tells the PPP service to start automatically on boot:

```
cd /etc/systemd/system/
sudo nano rpippp.service
```

Copy the lines below into the file:

```
[Unit]
Description=PPP
[Service]
Type=idle
ExecStart=/usr/sbin/pppd -d /dev/ttyAMA0
1000000
Restart=always
[Install]
WantedBy=multi-user.target
Alias=ppp.service
```

Now run the following commands to enable the service:

```
sudo systemctl daemon-reload
sudo systemctl enable rpippp.service
```

Shut down the Raspberry Pi.

On the Zero

It's a good idea to change the host name on the Zero in order to avoid confusion when you're SSHed into it:

```
sudo nano /etc/hostname
```

Change **raspberrypi** to **zero**, then:

```
sudo nano /etc/hosts
```

Do the same again: **raspberrypi** to **zero**. Save and exit. Next, install the ppp package:

```
sudo apt-get install ppp -y
```

As with the host Raspberry Pi, you'll need to disable the login prompt in the same way as before. When you get to creating an options file, use this:

```
noauth
nocrtscts
xonxoff
passive
local
maxfail 0
defaultroute
persist
nodetach
192.168.0.101:192.168.0.100
```

Next, we need to preconfigure a DNS server, otherwise the Zero will not be able to perform DNS queries.

```
sudo nano /etc/resolvconf.conf
```

Find the line **#name_servers=127.0.0.1** and insert the line **name_servers=192.168.0.1** below it. Save and exit.

Now run the following commands to enable the service:

```
sudo systemctl daemon-reload
sudo systemctl enable rpippp.service
```

Shut down the Pi Zero.

Put it all together

Connect the UART 'pins' of the Pi Zero as shown in the main image on the opposite page. Use a blob of Blu-Tack or plasticine to wedge the male ends of the jumper cables so that they're in contact with the metal holes on the Zero; remember, it's physical GPIO pin numbers 8 and 10.

Boot up both the host Pi and the Zero. The following command can be used to monitor the status of the PPP service:

```
sudo systemctl status rpippp.service
```

This will give you the log file of the PPP service. The following line indicates that the connection has been established (the pid number for you will be different):

```
Script /etc/ppp/ip-up finished (pid ###),
status = 0x0
```

If you have any issues or see error messages, you can restart the PPP service using this command:

```
sudo systemctl restart rpippp.service
```

You should now be able to SSH into the Zero using this command:

```
ssh pi@192.168.0.101
```

WIRED INTERNET

You can get USB Ethernet adaptors if you want to get the most out of your internet connection with your Raspberry Pi Zero.

NEED FOR SPEED

The internet on the Zero won't be as fast as it is on the host; however, it won't be massively slower.

ROB ZWETSLOOT

Tech writer, avid coder, and Raspberry Pi enthusiast with a history of building many things with Raspberry Pi.
raspberrypi.org/magpi

NES ZERO

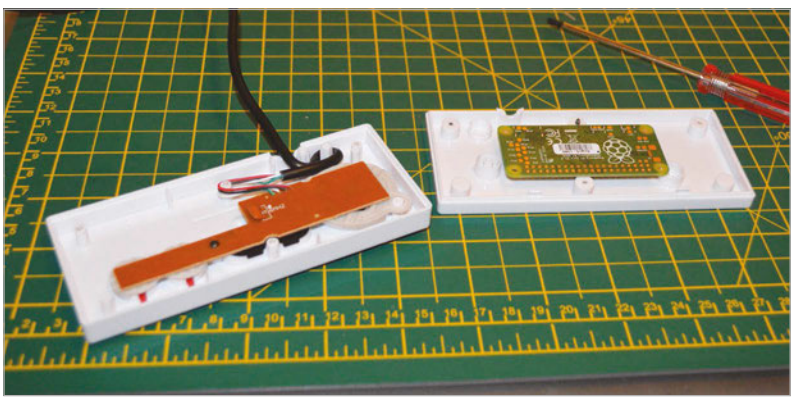
We make last month's project idea into reality with a NES controller that's been Zero-charged to be able to run NES games...

You'll Need

- USB NES controller (check eBay!)
- A Raspberry Pi Zero - **raspberrypi.org**
- RetroPie - **magpi.cc/1HVgNba**
- A rotary tool (like a Dremel)
- An old micro USB cable
- A screwdriver, soldering iron and a way to strip wires

Ever since first seeing the Raspberry Pi Zero a few months ago at Pi HQ, we've spent a lot of time since then thinking about the cool ways that the Pi Zero can be used. One of our first ideas was slipping a Pi Zero into a NES controller, taking a bit of inspiration from Ben Heck's great inventions to further miniaturise the NES to its most core component.

We wrote about how you could do this last issue but we have now finally had time to make a proof-of-concept, so we're going to show you how to exactly put a Pi Zero inside an NES controller.



>STEP-01

Set up RetroPie

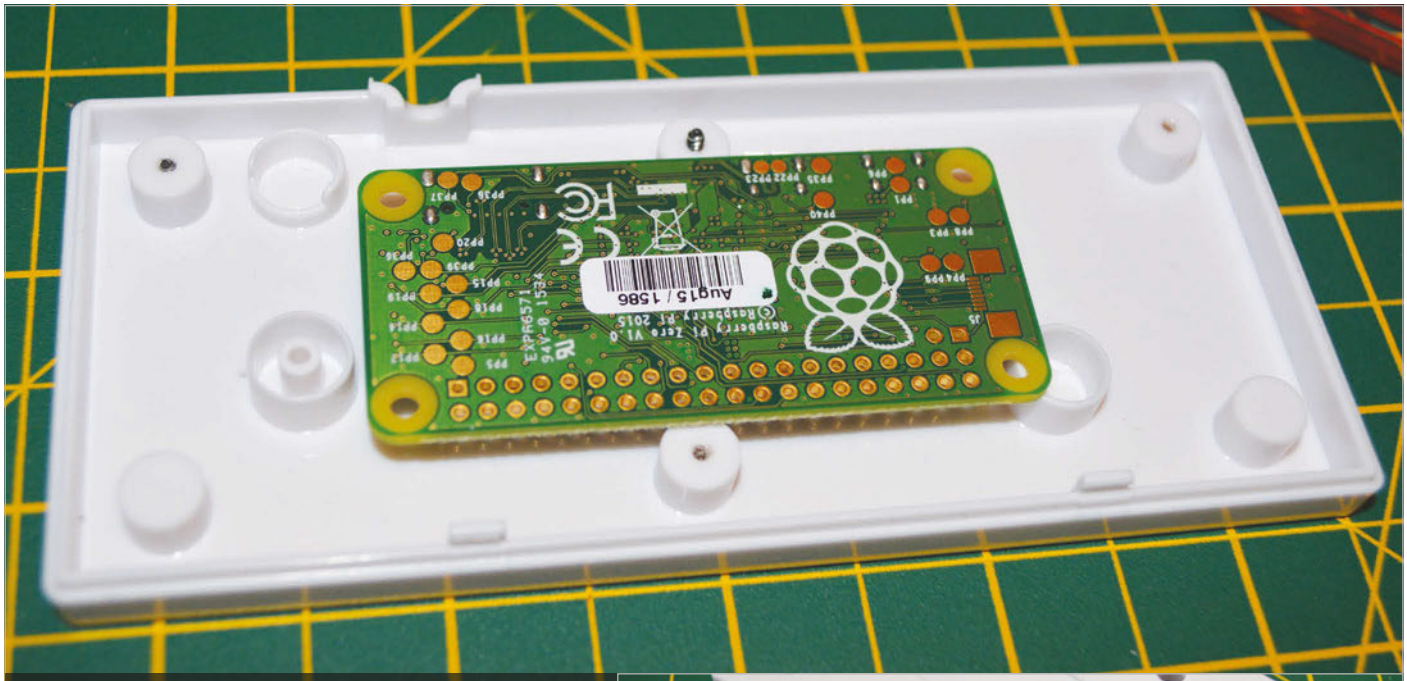
This is really more for convenience than anything else. Download the RetroPie image and write it to a microSD card using your preferred method (check the Raspberry Pi guide if you've not done this before: **magpi.cc/1XTmymk**). Once it's written, open it up in your file explorer and navigate to home, pi, RetroPie and put your NES ROMs into the nes folder.

Pop the micro SD card into the Pi Zero and hook the USB NES controller up to it before turning it on. You can then do the initial setup and configure the controller.

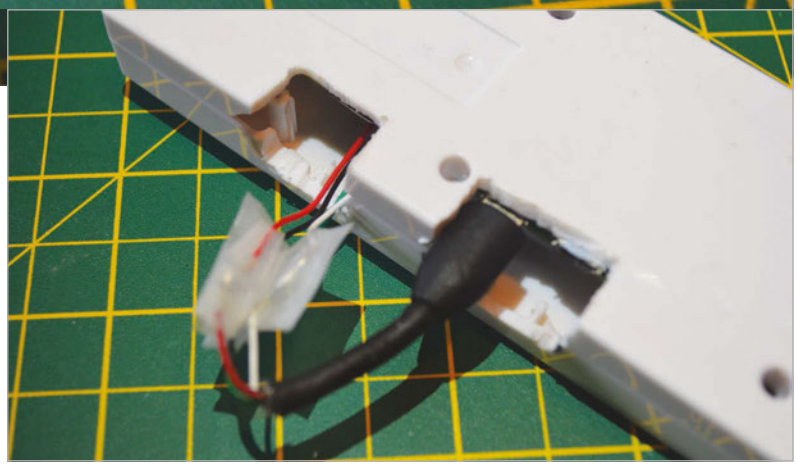
>STEP-02

Make some space

It seems that many USB NES controllers have roughly the same internal structure (they may be using a mould of the NES controllers, or just a generic one) and it is therefore likely that you'll need to make some space. We used a rotary tool to trim the screw holes and a ring on the edge. We wanted to use the original USB wire port for our HDMI cable, so we suggest that you lay the Pi Zero (with SD card inserted!) inside the back plate and figure out where needs to be trimmed.



We used a Sharpie to mark out the cuts we needed to make by periodically laying the Zero over the backplate as we worked



>STEP-03

Cable holes

If you're doing what we did, not only will you need to widen the USB hole to allow for plugging in a mini HDMI cable and a new hole for the power port to slot in, you'll also need a space for the micro USB port to jut out.

Mark the amount of space you'll need to cut out for the mini HDMI cable to fit in; you can even consider cutting a square out of the back plate to help it to slot in more easily. Test it as you go, but be careful of plastic shrapnel as you file it up.

> All that's left is to plug it into a TV and power supply...

>STEP-04

Wire up the controller

USB cables are wired up very simply, with four wires providing power, ground and data. Luckily, the colouring of the wires inside a cable is standard. Snip off the end of the micro USB cable you want to sacrifice and strip off the rubber, the braided wire and the foil surrounding the four individual wires. Strip them to reveal the core. Do the same with the USB cable on the NES board: on ours the wires were accessible before the cable insulated them.

Solder the corresponding wires together and individually insulate them with electrical tape or heat shrink tubing.

>STEP-05

Assemble it

We used a bit of Blu Tack to secure our Pi Zero to the inside of the base plate while carefully making sure the micro USB connector was plugged in and not obscuring any of the screws or clips connecting the controller together. We then carefully screwed it in and checked that the connectors fitted. All that's left is to plug it into a TV and power supply and make sure the whole system works.

Above Our cutting is a little rough in places; you may also need to open up the bottom of the controller to fit the cables in

>STEP-06

Game on

As we have noted, our version is simply a proof of concept. It did, however, survive a solid six hours of young kids getting very frustrated over old games at Pi Wars, so clearly the controller and Pi Zero are robust enough to handle extended play sessions. You will need to dismantle the device to get to the SD card and load or change ROMs on it. You could, however, probably get it to connect over WiFi and transfer files over SSH.

OTHER CONTROLLERS

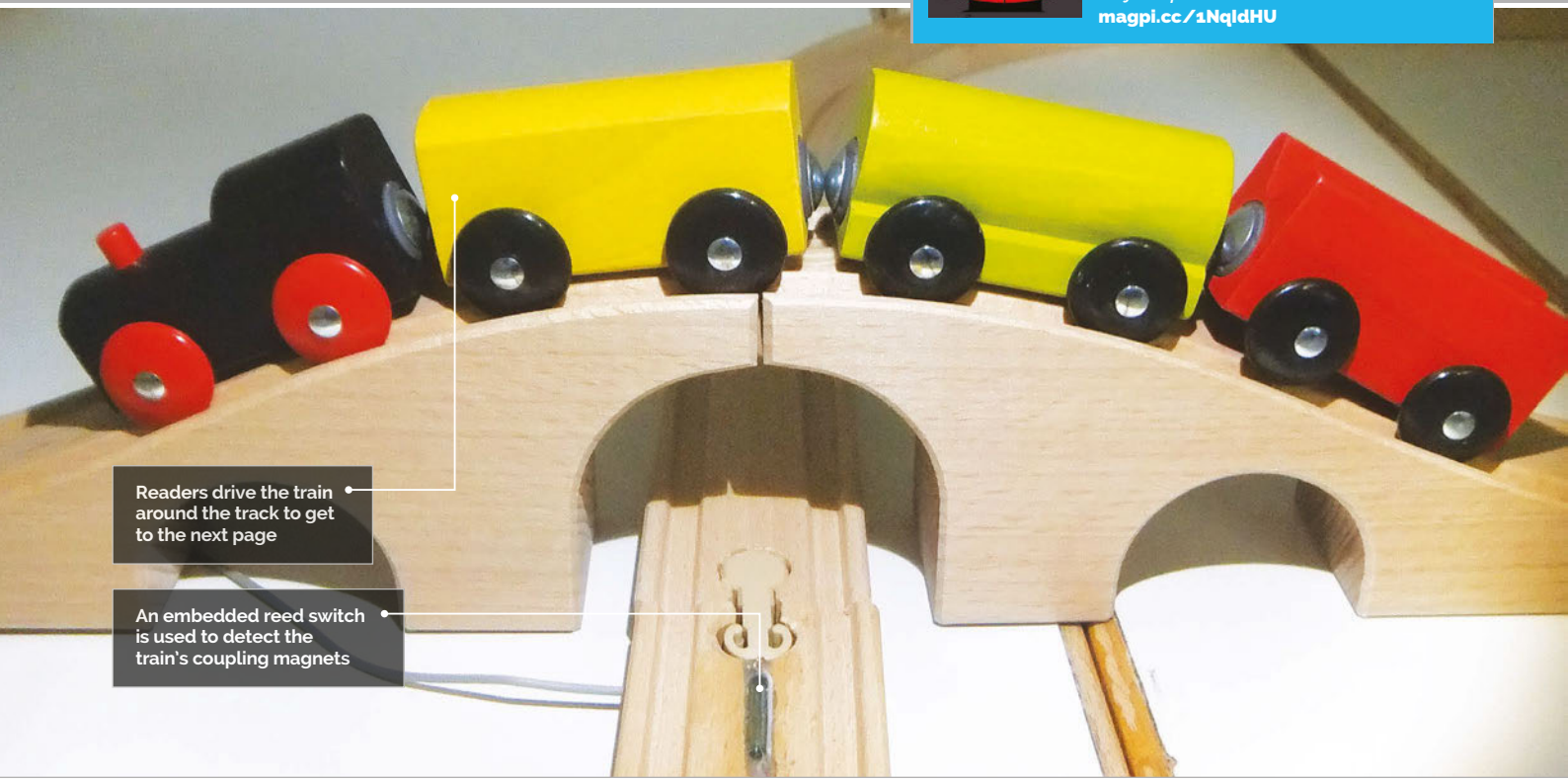
You just have to look at the Xbox controller hack from earlier in the magazine to see how this could be done with other controllers

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/1NqIdHU



THE STORY TRAIN

You'll Need

- Ikea Lillibo wooden train set (or similar)
- Optional Ikea Lillibo track extension pack
- 2× reed switches
- Connecting wire
- Silicon sealant

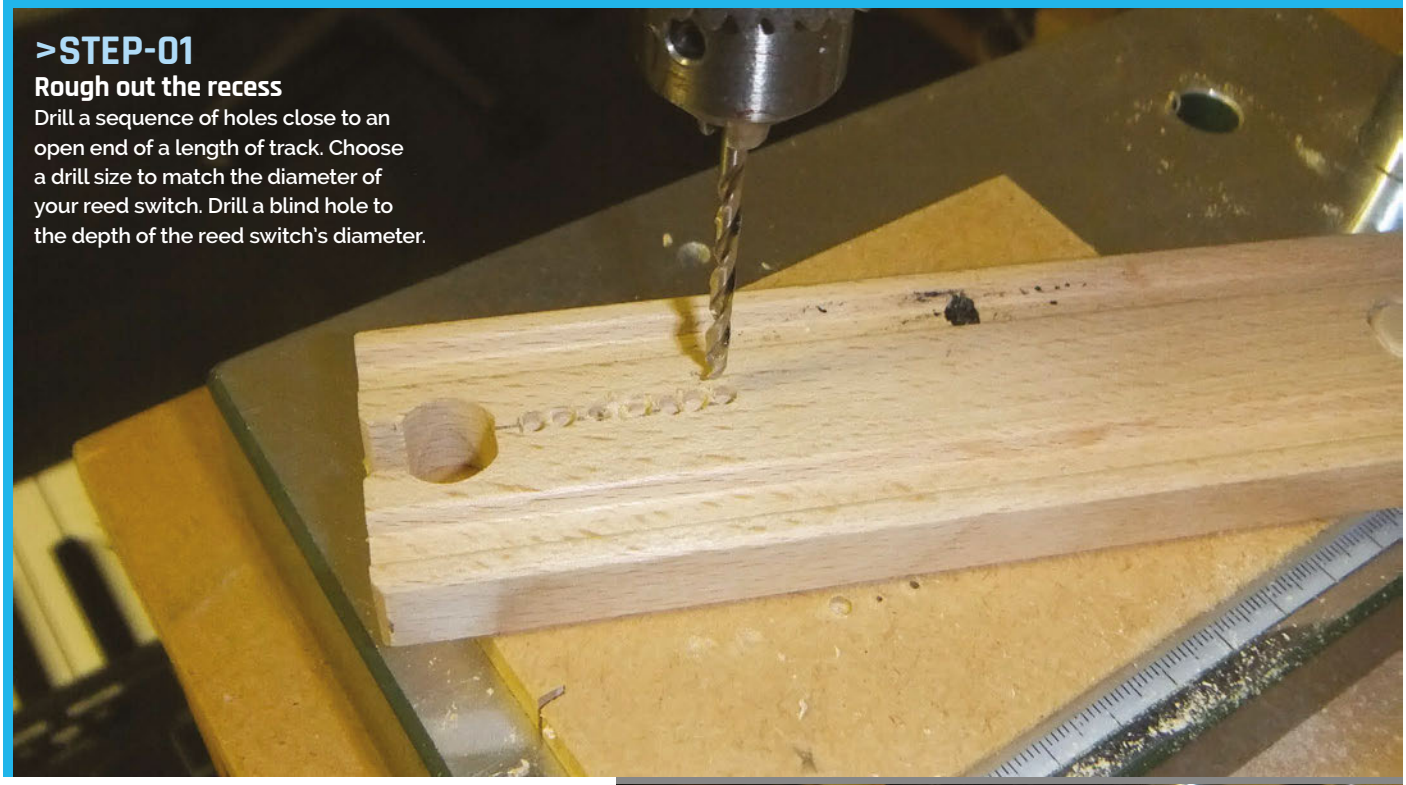
Motivate children to read by using the Story Train to turn the pages of an electronic book

Persuading a child to read can be easy, but when they've had a bad early experience, it can be difficult to get them started again. Adding motivation to get them reading once more can kick-start the learning process if it has stalled. This is where the Story Train can be helpful: it provides a fun distraction and incentive to get to the next page. The Story Train can present a written page, or it can simply read a page out loud; what gets you from one page to the other is taking the Story Train around the track. But it's not only using the Story Train that can be helpful – older children can write stories for the Story Train to tell, and that in itself is another motivation for learning and creativity.

The project

This project uses the low-cost Ikea Lillibo wooden train set, although there are other, similar products available that might work just as well. The train and carriages are joined together with magnets, and it's these magnets that are the key to detecting the train as it passes a point on the track. The sensor is a simple reed switch. Reed switches come in many sizes and consist of two metal plates in a sealed glass tube. The idea is that the presence of a magnetic field close to them attracts the two plates together and they make contact. They are often used for door monitors in electronic security systems, and you can even get change-over switches. The ones we used on this project were 20mm long, although you can get both shorter and longer ones. Actually, what we had planned to do was to drill a blind hole into the underside of the train and push a small round neodymium magnet in the hole. That's when we discovered that the coupling magnets themselves were enough to trigger the reed switch, so no extra magnets were needed. We recessed the switch flush with the track – see the step-by-step guide for details – then simply wired each switch between a GPIO pin and ground. The project requires two switches like this to work, but you could incorporate more. Our code uses GPIO pins 2 and 3.

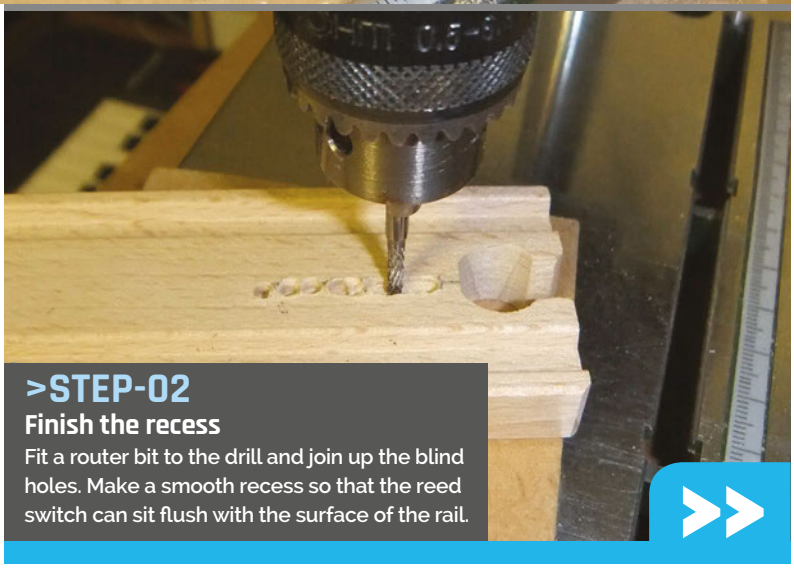
EMBEDDING THE SENSOR SWITCHES



>STEP-01
Rough out the recess
Drill a sequence of holes close to an open end of a length of track. Choose a drill size to match the diameter of your reed switch. Drill a blind hole to the depth of the reed switch's diameter.

The track layout

The wooden train track can be laid out in many different ways, although there are only two basic topologies: a loop or single-ended. We favour the loop because it's obvious what you need to do to drive the story on. At the end of each page of the story, you just drive the train around the track to move on to the next page. At least two switches are required: one at point A, and the other at point B. However, this is not critical at all, as long as they are separated so that only one switch can be triggered at a time. You could use more switches at other parts of the track to trigger extra sound effects if you like. We found that someone had written a program to generate all the possible track layouts for the basic Lillibo set; the code is online here: **magpi.cc/1NM4vUh**.



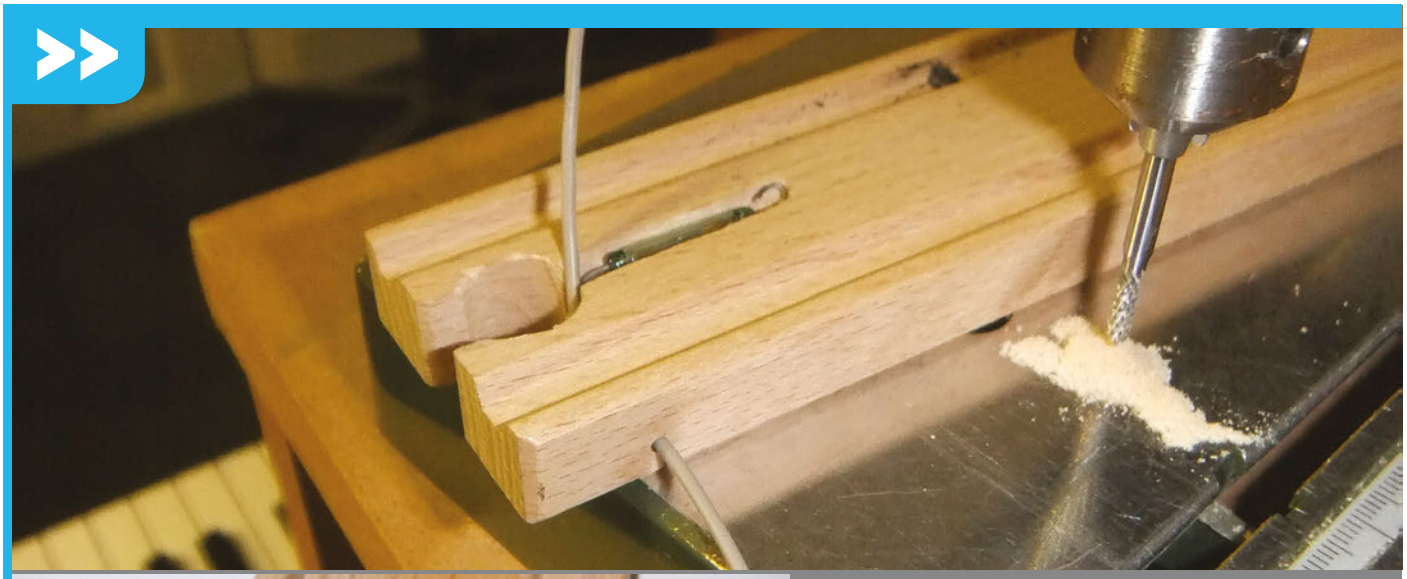
>STEP-02
Finish the recess
Fit a router bit to the drill and join up the blind holes. Make a smooth recess so that the reed switch can sit flush with the surface of the rail.

The software

The idea is that when the train pulls into the station, a page is produced and a sound sample is played. This page can be text, just pictures, or a mixture of both. When the page has been read and the sound sample played, the software looks for the other switch being triggered. When it sees that, a second sound effect is played and the program then looks for the station switch to be triggered before producing the next page. In this way, there's no problem with contact bounce on the switches, nor from multiple triggering due to the carriage's coupling magnets.

The media

The electronic book consists of a number of pages, each comprising a JPEG image file and a WAV sound file. These have to be prepared in advance, and these pages are what the Story Train is presenting as the story unfolds. Older children can write their own stories and draw accompanying pictures; youngsters or a teacher can write a story or borrow one from a book. The image files can be produced in any package; for keeping it all on the Pi, however, the Inkscape application is hard to beat, especially on the faster Model 2 Pi.



>STEP-03
Fit the wire

Drill in from the side to allow you to access the switch with the wire. You'll have to cut a channel in the open end of the coupler, so that the wire doesn't interfere with the next joining rail.

>STEP-04
Fit the reed switch

Bend the wire from the reed switch into a loop. Make sure you have the pliers between the glass end and the bend, to avoid chipping the glass. Cut off the excess lead. Solder the screen to the end close to the join, and the core to the far end, for mechanical convenience.

>STEP-05
Seal the switch

Test to see the next rail can be fitted, and then fill the recess with silicon sealant and smooth it flush with the top of the rail.

Each story is contained in its own directory, and this must conform to a fixed structure so that the software knows where things are. In each story directory, you need a **Read_Me**, which is just a blank file, although you can use it to put the story credits in: it's just a target for selecting the story. You need **nextComing.jpg** and **nextComing.wav** files to indicate the next page is on its way. Finally, you need **pages** and **sounds** directories; into these you put the sound samples and the pictures that make up the story. Each needs a file called **begin** and one named **the_end**, along with files called **page01**, **page02**... and so on. To clear up any questions, please look at the example story on GitHub (**magpi.cc/1NqJjmV**), where you'll find the program to drive this.

The code

The code listing drives this structure. It uses GPIO 2 and 3 as the switch inputs, but this can be changed to any two pins by changing the `pinList` list. The right cursor key acts as an 'advance to next switch' key – useful when testing a book before you hand it over.

Taking it further

This is for a simple linear story. If you fit more than two switches to the track, however, you can get involved in creating interactive stories. The child then chooses one of two (or more) paths by driving the train round a different loop. You'll need the extra track pack for this; it contains points, thus giving the choice of two routes.

We hope the Story Train can inspire new adventures in learning and creativity. Good luck.

StoryTrain.py

```
import pygame, time, os
import wiringpi2 as io
from tkFileDialog import askopenfilename

pygame.init()                    # initialise graphics
interface
pygame.mixer.quit()
pygame.mixer.init(frequency=22050, size=-16, channels=2,
buffer=512)

os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
pygame.display.set_caption("The Story Train")
pygame.event.set_allowed(None)
pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])
screen = pygame.display.set_mode([600,400],0,32)

try :
   io.wiringPiSetupGpio()
except :
   print"start IDLE with 'gksudo idle' from command line"
   os._exit(1)

pinList= [2,3] # GPIO track sensing switches

def main():
   global pageList, pagesInStory
   initGPIO()
   print"The Story Train"
   while True:
      notFinished = True
      page = 1
      getStory() # get the story to show
      showPage(0) # beginning picture
      while notFinished:
        holdUntilSwitch(1)
        # trigger halfway round display
        showPage(-1)
        holdUntilSwitch(0)
        # display page sound and picture
        showPage(page)
        page += 1
        if page >= pagesInStory :
           notFinished = False
      showPage(pagesInStory) # the end
      time.sleep(2.0)

def initGPIO():
   for pin in range (0,len(pinList) ):
      io.pinMode(pinList[pin],0)
      io.pullUpDnControl(pinList[pin],2) # input enable
pull-up

def holdUntilSwitch(pin):
   global advance
   advance = False
   while (io.digitalRead(pinList[pin]) == 1) and (advance
== False):
         checkForEvent()
         time.sleep(0.2) # let other apps have a look in
   return

def showPage(page):
   pygame.draw.rect(screen, (0,0,0),
(0,0,600,400),0)
   if page ==-1 :
      screen.blit(nextPicture,[0,0])
      nextSound.play()
   else :
      screen.
blit(pagePictures[page],[0,0])
      pageSounds[page].play()
   pygame.display.update()
   time.sleep(0.4)

def getStory():  # get the list of pages to show
   global pagePictures, pageSounds, pagesInStory,
nextSound, nextPicture
   pathP = "nothing"
   pathS = "nothing"
   while not(os.path.exists(pathP) and (os.path.
exists(pathS)) ):
      checkForEvent()
      filename = askopenfilename()
      path = os.path.dirname(filename)
      pathP = path + "/pages/"
      pathS = path + "/sounds/"
   pageList = [os.path.join(pathP,fn) for fn in next(os.
walk(pathP))[2]]
   list.sort(pageList) # put in alphabetical order
   pagePictures = [ pygame.image.load(pageList[frame]).
convert_alpha()
                for frame in range(1,len(pageList))]
   soundList = [os.path.join(pathS,fn) for fn in next(os.
walk(pathS))[2]]
   list.sort(soundList) # put in alphabetical order
   pageSounds = [ pygame.mixer.Sound(soundList[frame])
            for frame in range(1,len(soundList))]
   pagesInStory = len(pagePictures)-1
   nextSound = pygame.mixer.Sound(path+"/nextComing.wav")
   nextPicture = pygame.image.load(path+"/nextComing.
jpg").convert_alpha()

def terminate(): # close down the program
    print ("Closing down please wait")
    pygame.mixer.quit()
    pygame.quit() # close pygame
    os._exit(1)

def checkForEvent(): # see if we need to quit
    global advance
    event = pygame.event.poll()
    if event.type == pygame.QUIT :
         terminate()
    if event.type == pygame.KEYDOWN :
       if event.key == pygame.K_ESCAPE :
          terminate()
       if event.key == pygame.K_RIGHT :
          advance = True

# Main program logic:
if __name__ == '__main__':
    main()
```

Language

>PYTHON 2.7

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS

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

LUCY ROGERS

Maker. Tamer of Robot Dinosaurs. Hacker of the IoT. Author. Fellow of IMechE. Positively affecting the lives of a billion people...
lucyrogers.com / @DrLucyRogers

CHEERLIGHTS ORB

Build a colour-changing, tweet-reacting orb that requires no previous code or electronic skills!

You'll Need

- White ping-pong ball
- Network-connected Raspberry Pi
- LedBorg **magpi.cc/1lH1XL5**
- Alternatively: various resistors (100R, 220R, and 270R), various LEDs, breadboard, and cables

DEBUGS

Add debugs to trace what is happening throughout the flow.

The spiky pins sticking up along the edge of the Pi (or the holes on the Pi Zero) are the gateway to interacting with things in the real world.

Follow these simple instructions to use those GPIO pins with Node-RED drag-and-drop visual programming to make something magical you can show off with pride.

Once you have mastered this, there is no limit to what your colour-changing orb can display: have it light up when you have a Twitter mention, use it as a traffic light system for your home energy usage, or even to highlight if you need to run for your bus.

>STEP-01

Getting set up

Before turning on your Pi, connect your devices. If using a LedBorg, connect it as per the instructions on the site: **magpi.cc/21LV9SU**. If you're using LEDs, connect it up as in the main image at the top of the page.

You'll need to boot into Raspbian Jessie and have your Pi connected to the network. Open Node-RED – click on **Menu**, **Programming**, **Node-RED** – and a console will open. At the top of the console, there are instructions such as:

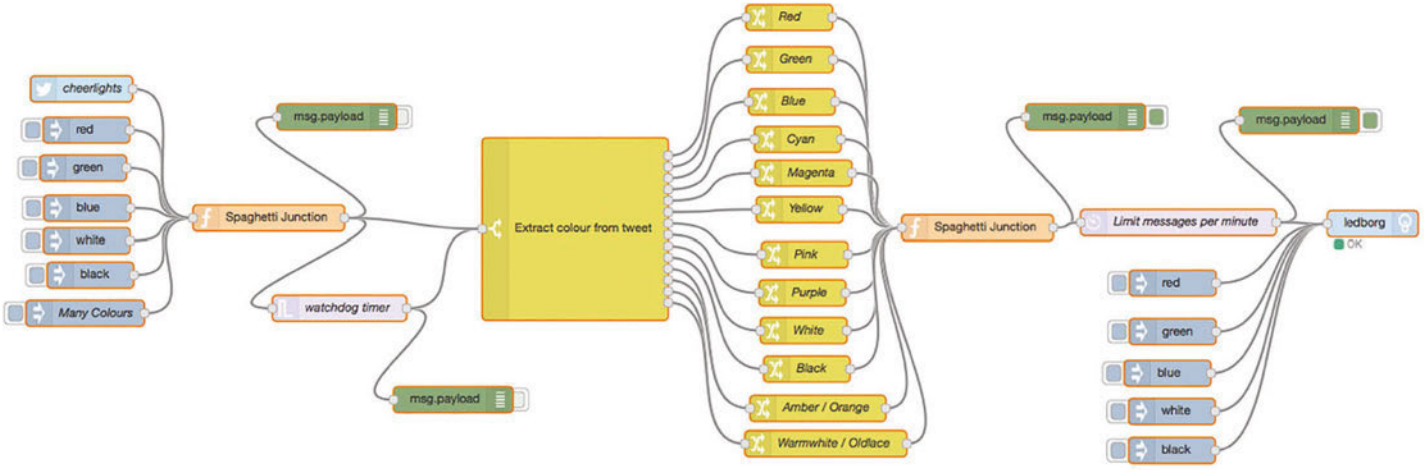
'Once Node-RED has started, point a browser at **http://192.168.X.XXX:1880**'

If using a Pi 2, you can do this in the browser. The address **http://127.0.0.1:1880** will also work, as will **localhost:1880** – but only if you have already connected to a network. If you're using another type of Pi, it can be easier to connect from another computer's browser using the address provided.

>STEP-02

Write a flow

Drag an **`inject`** node from the Input section of the left menu into the middle pane. Its name should change to **`timestamp`**. Repeat with a green **`debug`** node from Output, its name changing to **`msg.payload`**. Join the two: click the small grey pimple on the right of the **`timestamp`** and drag it to the pimple on the left of **`msg.payload`**. A wire will connect the two. Click the debug tab on the top right. Blue circles on top of each node show the flow has not been saved/deployed; click the red Deploy button, top right, and the blue circles disappear. Activate the inject node by clicking the square to the left of **`timestamp`**; the resulting message ('payload') appears in the debug tab.



>STEP-03

Change the flow

Double-click the inject **timestamp** node. A dialog box appears. Click the dropdown for Payload and choose 'string'. Under 'string', type 'red'; this will be the message payload. In the 'name' box, rename the node e.g. to 'red', then click OK. Deploy and then inject by clicking the square on the left of the 'red' node. The message payload (red) appears in the debug tab.

Add four more inject nodes and change the message payloads to green, blue, white, and black. Connect them all to the debug **msg.payload** node and deploy. Inject each colour – the message payloads appear in the debug tab.

>STEP-04

Add lights and orb

For both the LedBorg and separate LEDs, scroll down the left-hand pane and drag a **ledborg** node onto the flow. Connect the inject nodes to the **ledborg**. Deploy and click some inject nodes; the LEDs or LedBorg will light up with the corresponding colour.

Slice the bottom off the ping-pong ball and place over the LedBorg to act as a diffuser. If using LEDs, cut a small hole in the bottom of the ping-pong ball and push the LEDs into it. Keep the LEDs as low as possible inside the ball; use Blu-Tack to hold the ball if necessary.

>STEP-05

Connect to Twitter

Add a **twitter** input node from the social section – it has a pimple on the right. Double-click and add Twitter credentials; you'll need to be logged into Twitter. In the 'for' box, add 'cheerlights'. Add a **function** node; this is used to keep the wires tidy, so double-click and call it 'spaghetti junction'. The message will pass through and not be changed. This node can be used for adding JavaScript code in future projects.

Delete all the wires, and rewire all the colour inputs and **twitter** node to the input of the **spaghetti junction** node. Connect a **debug** node to the output of the **spaghetti junction**.

>STEP-06

Extract the colour

The LedBorg only accepts certain colours; extra words such as 'cheerlights' means it won't work. Connect a yellow **switch** node to the output of **spaghetti junction**. In its dropdown box, select 'matches regex'; next to that, type 'red' and tick the 'ignore case' button.

Click '+ rule' and repeat for all the cheerlights colours; they're shown in the info tab for the **ledborg** node. Connect a yellow **change** node to each of the outputs of the **switch** node. Double-click each one, and set the name and the payload to the required colour.

Connect all the outputs of the **change** nodes to another **spaghetti junction** node, and then to an **ledborg** node before deploying. You can add more sheets on your Pi, but make sure there is only one **ledborg** node, and pins 11,13, and 15 aren't used in other flows.

The code

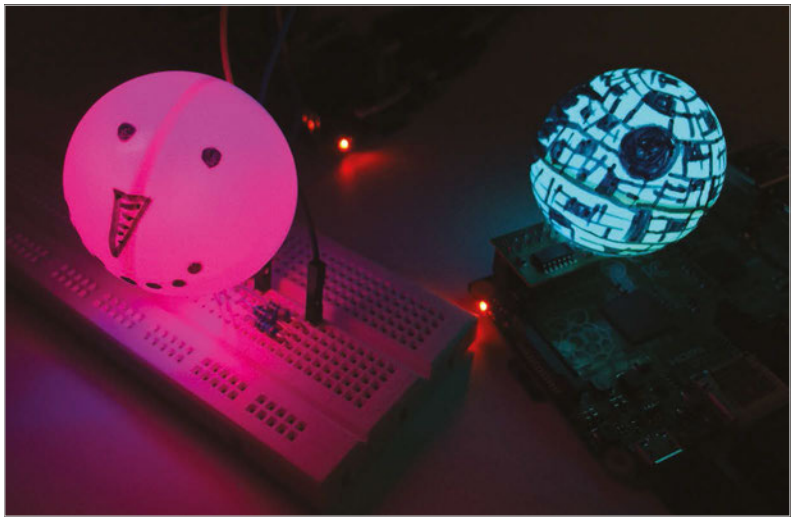
The flow can be found at **magpi.cc/1IH4rt1** or on GitHub at **magpi.cc/1IH4vc7**. Copy the flow to the clipboard, then import it into Node-RED. Click on the NodeRED menu (the hamburger icon), then **Import**, **Clipboard**, paste the flow, and click OK.

WATCHDOG TIMER

Add a 'trigger' node in parallel: send nothing, then wait 30 seconds, then turn the string payload black. Tick 'extend delay if new message arrives'.

DELAY

Add a delay node to limit the number of messages per minute (limit rate to four messages per minute).



FIND THIS & OTHER RESOURCES: raspberrypi.org/resources

MAKE A DIGITAL MAGIC 8 BALL

Bring the time-honoured tradition of shaken-not-stirred fortunes to the Sense HAT, making use of those motion sensors

You'll Need

- Sense HAT **magpi.cc/SenseHAT**
- Sense HAT Python library **magpi.cc/1RKRoqcc** (pre-installed with Raspbian Jessie)

In this activity you will build your own Magic 8 Ball using your Raspberry Pi, a Sense HAT, and some Python code. A Magic 8 Ball is a toy which you verbally ask a closed question to, then shake, and it will give you a prediction.

In this tutorial you will use IDLE 3, the Python development environment, to write some code for the Sense HAT. This means you can test your code and fix it as you write it as well. To open IDLE 3, click on the main Raspbian **Menu** (top-left), followed by **Programming** and then **Python 3 (IDLE)**.

Fig 1 Test the program first in a Python shell window

Once the Python shell window has loaded, click on **File** and **New File**. This will open a text editor window in which you can write, save and test your code. Save the blank file as **magic8ball.py** by clicking on **File** and **Save As.**

Printing replies to the screen

A good way to start your Magic 8 Ball program is to first create a text version of how it works. Let's think about what a Magic 8 Ball does. First, you ask it a question, before shaking the ball, turning it over, and then reading a reply that it has randomly selected. Therefore, you will need a list of replies and a way of randomly choosing one from the list and displaying that answer on the screen.

To start, you need to import the **random** and **time** libraries. Type the following into your **magic8ball.py** text file:

```
import random
import time
```

Using the **print** function, you can print text to the screen for the person using your program. Type:

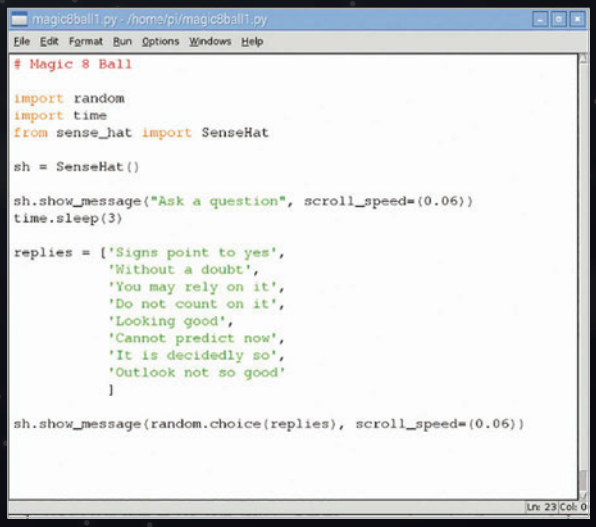


Fig 2 You can alter the speed of the scrolling text

```
print("Ask a question")
```

Then there needs to be a pause before the program responds, so that the user can ask a question verbally or mentally. You can use the **time** library to ask the program to sleep for a set amount of time, like this:

```
time.sleep(3)
```

The program will pause for three seconds. You can change this value to make the pause longer or shorter. Now, create a list of replies that the program could give to the question. Lists can be named in much the same way as variables; for example, **number = [1, 2, 3, 4]**. This list called 'number' has four items in it. Your list will contain strings of text that will be displayed on the screen; these strings will be quite long. To create your list, type:

```
replies = ['Signs point to yes', 'Without a doubt', 'You may rely on it']
```

Add as many replies to your list as you like. Make sure that you separate each reply with a comma. You can break up your list onto multiple lines, like this, to make it easier to read; however, this is not required for your program to work:

```
replies = ['Signs point to yes',
        'Without a doubt',
        'You may rely on it',
        'Do not count on it',
        'Looking good',
        'Cannot predict now',
        'It is decidedly so',
        'Outlook not so good'
        ]
```

Finally, an instruction is needed to select an item from the list at random and then display it on the screen. You can use the **random** library to do this, by typing:

```
print(random.choice(replies))
```

Save your code by clicking on **File** and **Save**, then run your program to test it works by clicking on **Run** and **Run Module**. Your code should look similar to that in **Fig 1**.

Display text on the Sense HAT

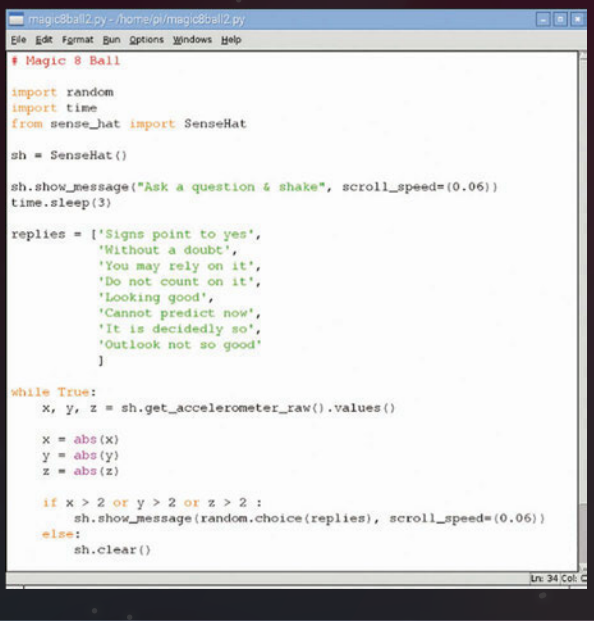
Now that you have text outputting to the Python 3 shell window on your screen, let's change the code so that the text scrolls across the LED matrix on your Sense HAT. To do this, you will need to use the **SenseHat** library and replace the print functions with a 'show message' function. Underneath the import modules section of your code, add the following lines:

```
from sense_hat import SenseHat
sh = SenseHat()
```

Next, replace **print** with **sh.show_message** in your code. There are two places where you will need to do this. To test the code, save your program by pressing **CTRL+S** on your keyboard, then run it by pressing **F5** to check that it works on the Sense HAT.

You may find that the text is slow to scroll across the LED matrix on your Sense HAT. To speed up the text, you can add **scroll_speed=(0.06)** to your text strings, as in **Fig 2**.

Normal Magic 8 Balls activate by being shaken up after asking a question. How do you think you could make that happen using the Sense HAT's motion sensors? That's your next programming challenge!



Left A little hint for how to use the Sense HAT's motion sensors so you can shake your Magic 8 Ball

FIND THIS & OTHER RESOURCES: raspberrypi.org/resources

INTERACTIVE PIXEL PET

Create a pint-sized pocket monster living in the digital world of a Raspberry Pi Sense Hat

You'll Need

- Sense HAT **magpi.cc/SenseHAT**
- Sense HAT Python library **magpi.cc/1RKRoqc** (pre-installed with Raspbian Jessie)
- Pypng **magpi.cc/1lj5ijm**
- 8×8GridDraw **magpi.cc/1lj5oYo**

Using sensors and output devices is a great way to make your computer programs more interactive. The Raspberry Pi Sense HAT contains a whole set of sensors that can be used to detect movement, which will be used in this activity to take a digital pet for a walk.

The first thing that needs to be done is to install the extra software required for this tutorial. Install the Python PNG library – an image library for Python that uses the PNG file type – by opening a Terminal with **Menu**, **Accessories**, **Terminal**, and typing:

```
sudo pip3 install pypng
```

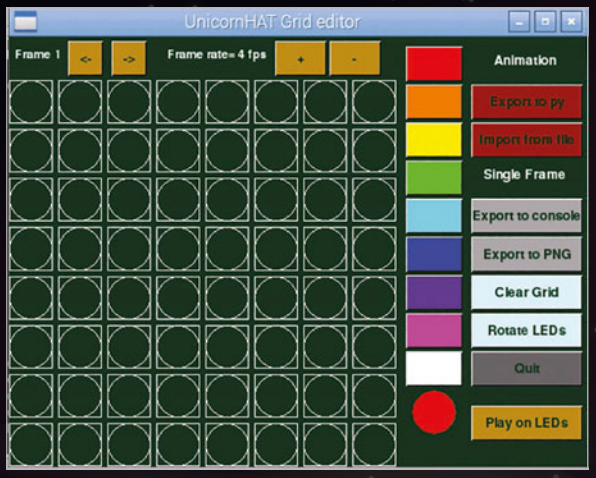
Press **ENTER**. After that has been installed, remain in the Terminal and type:

```
git clone https://github.com/jrobinson-uk/RPi_8x8GridDraw
```

You'll need to design your pet avatar before you program any actions. There are examples of some famous characters you can make on an excellent sprite sheet created by Johan Vinet, which can be found at **magpi.cc/1SrqGDm**.

Open a Terminal by clicking on **Menu**, **Accessories**, and **Terminal**. Enter the line `cd RPi_8x8GridDraw`, followed by `python3 sense_grid.py`. This will run an application which you can use to draw your space pet avatar, as seen in **Fig 1** below.

Fig 1 Using 8×8GridDraw



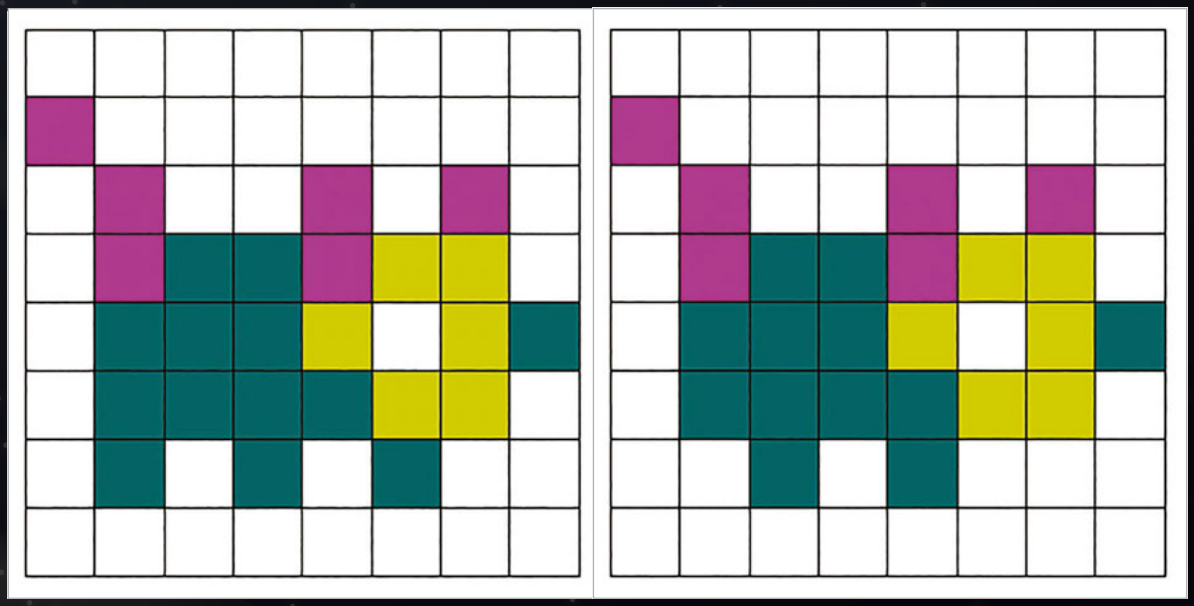


Fig 2 Two very simple frames can give you enough variation to create an animation, even if the difference is a few pixels!

Simply select the colour you wish to use from the grid with your mouse pointer, and then select the circle in the grid to change it to that colour.

Alternatively, you may wish to draw your picture out on squared paper with coloured pencils. You'll need two pet designs, with the second preferably very similar to the first, so that we can animate your pet. In **Fig 2** above, you can see that our two images are almost identical to each other, but the feet are in a different place.

Later, when you code your animation, you will create the illusion that the pet is walking.

Labelling each pixel

Think of a letter from the alphabet to represent each colour in your pixel pet image, for example, **w** for white or **r** for red. If using squared paper for your design, you can write the letters on top – see **Fig 3** on the next page. Note that **e** stands for **empty**.

If you're using the 8x8GridDraw editor, then you can write out your squares on paper, representing each colour with a letter and separating them with a comma. Alternatively, you could type them into a text editor like Leafpad, which you can find by clicking on **Menu**, **Accessories**, and **Text Editor**. You'll end up with something that looks like this:

```
e, e, e, e, e, e, e, e,
p, e, e, e, e, e, e, e,
e, p, e, e, p, e, p, e,
e, p, g, g, p, y, y, e,
e, g, g, g, y, w, y, g,
e, g, g, g, g, y, y, e,
e, g, e, g, e, g, e, e,
e, e, e, e, e, e, e, e
```

You'll notice that we have eight rows and eight columns of letters, each separated by a comma, to make up the LED matrix on the Sense HAT. Repeat this step for your second pet design, so that you end up with two grids of letters. Can you think of any problems that might arise when only using one letter to label different colours? How might you solve this issue?

> When you code your animation, you will create the illusion of walking

Code your pet

Now that you have your designs represented as letters in a grid or array, you can start to code them in Python. Click on **Menu** then **Programming**, followed by **Python 3**. This will open the Python 3 shell window. Next, click on **File** and **New File** to open an empty text editor window. Save this empty file as **space-pet.py**.

First, you'll need to import all the modules and libraries required for this project in your code, by typing:

```
from sense_hat import SenseHat
import time
```

Underneath that, type:

```
sense = SenseHat()
```

Note that capital letters, full stops, and commas are very important in Python. Your code might not work if you don't include these. Next, create a **variable** for each colour label in your pet design, like this...

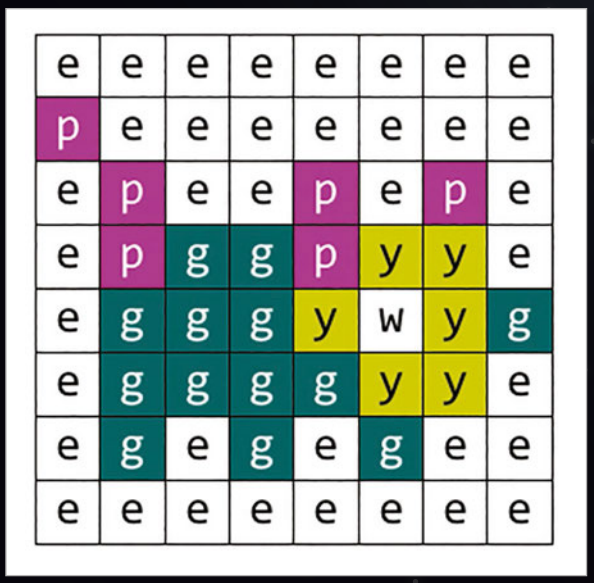


Fig 3 Thinking of the colours as letters instead of numbers makes it slightly easier to understand and code it into the program

```
p = (204, 0, 204) # Pink
g = (0, 102, 102) # Dark Green
w = (200, 200, 200) # White
y = (204, 204, 0) # Yellow
e = (0, 0, 0) # Empty
```

The numbers used here inside the brackets are **RGB** values, or **Red**, **Green**, and **Blue** values. Mixtures of these primary colours make different shades. The higher the number, the more of that colour it will contain. For example, **(255, 0, 0)** would make a solid red, whereas **(0, 255, 0)** would create a vivid green. You can change these numbers in your code to get the colours that you want.

Next, use a **list** to store your pixel pet design, like this:

```
pet1 = [
    e, e, e, e, e, e, e, e,
    p, e, e, e, e, e, e, e,
    e, p, e, e, p, e, p, e,
    e, p, g, g, p, y, y, e,
    e, g, g, g, y, w, y, g,
    e, g, g, g, g, y, y, e,
    e, g, e, g, e, g, e, e,
    e, e, e, e, e, e, e, e
    ]
```

Here you have created a variable called **pet1** and stored a list of labels for each colour by using **[** at the start of each letter and **]** at the end. Repeat for the second pixel pet design, using a different variable name like **pet2**. Your code should start looking something like **Fig 4**.

If you ran your code now, nothing would happen, because so far you have only told the program to store information. To make something happen, you will need to write a command to call on that data and display your colours in the correct order on the Sense HAT's LED matrix. Type this command underneath your lists:

```
sense.set_pixels(pet1)
```

Save your code by pressing **CTRL+S** on the keyboard, followed by **F5**. Note what happens. Why did only one of your pet designs display? It's because you have only called **pet1** in your command.

Add a delay using the **time.sleep** function, and then call the second picture using the same command as before, like this:

```
time.sleep(0.5)
sense.set_pixels(pet2)
```

Save and run your code to see your pet.

Animate your pet

So far, your pixel pet only changes once. To animate it fully, you will need to switch repeatedly between the pictures with a time delay. You could write the commands out over and over again, but it makes more sense to put them into a loop.

Move to the end of your program and locate the **sense.set_pixels(pet1)** part. Change it to look like this:

```
for i in range(10):
    sense.set_pixels(pet1)
    time.sleep(0.5)
    sense.set_pixels(pet2)
    time.sleep(0.5)
```

Don't forget to add the extra **time.sleep(0.5)** on the last line, and remember to indent the lines after **for i in range(10):** as this means they are inside the **for** loop. This **for** loop with the **range** function will repeat the indented code ten times and then stop.

Save and run your code to watch the animation.You will notice that after the animation has completed, you

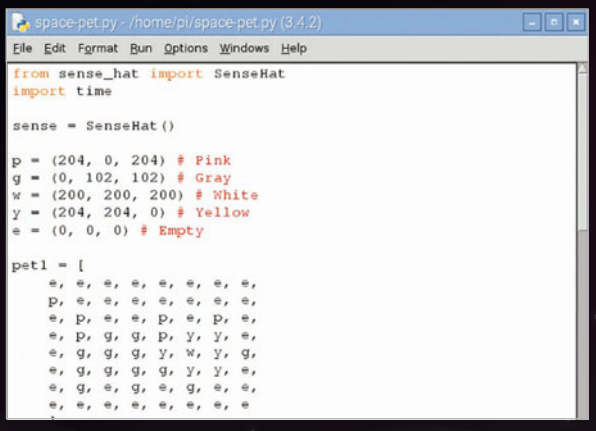


Fig 4 At this point, your code should look a bit like this

Language
>PYTHON 2.7

DOWNLOAD:
magpi.cc/1mcpSli

are left with the same image still displayed on the LED matrix. There is a great function that you can use that will clear the LEDs. Add this line above your new loop to clear the LEDs when you first run your program:

```
sense.clear()
```

Create a walking function

A **function** is a piece of code that you can use over and over. As the goal is to trigger the walking animation later on, it makes sense for us to put the animation code into a function that can be called when an action has been sensed by the hardware.

To put your code into a function, you simply need to add this line above your **for** loop and indent the lines beneath, like this:

```
def walking():
    for i in range(10):
        sense.set_pixels(pet1)
        time.sleep(0.5)
        sense.set_pixels(pet2)
        time.sleep(0.5)
```

The use of **def** here means that you are **defining** a function which you have called **walking**.

Now you need to call the function. So, at the bottom of your code, type:

```
walking()
```

Shake for more

It's time to use the Sense HAT's movement sensors, in particular its **accelerometer**, to trigger the walking function to make the project more interactive. Underneath your walking function, but above the function call line of **walking()**, type:

```
x, y, z = sense.get_accelerometer_raw.
values()

while x<2 and y<2 and z<2:
    x, y, z = sense.get_accelerometer_raw.
values()

walking()
```

The first line will obtain current movement readings from the Sense HAT on its x, y, and z coordinates. As your Raspberry Pi is presumably sitting still on a desk, those readings will have a very low value.

A **while** loop is then introduced to continually check the accelerometer values, to see if they have changed to above or equal to the value **2**. You can help the Sense HAT have an accelerometer reading of above **2** by shaking it! Save your code and then run it. Nothing should happen until you shake your Raspberry Pi.

Space-Pet.py

```
from sense_hat import SenseHat
import time

sense = SenseHat()

p = (204, 0, 204) # Pink
g = (0, 102, 102) # Gray
w = (200, 200, 200) # White
y = (204, 204, 0) # Yellow
e = (0, 0, 0) # Empty

pet1 = [
    e, e, e, e, e, e, e, e,
    p, e, e, e, e, e, e, e,
    e, p, e, e, p, e, p, e,
    e, p, g, g, p, w, w, e,
    e, g, g, g, w, y, w, y,
    e, g, g, g, g, w, w, e,
    e, g, e, g, e, g, e, e,
    e, e, e, e, e, e, e, e
    ]

pet2 = [
    e, e, e, e, e, e, e, e,
    p, e, e, e, e, e, e, e,
    e, p, e, e, p, e, p, e,
    e, p, g, g, p, w, w, e,
    e, g, g, g, w, y, w, y,
    e, g, g, g, g, w, w, e,
    e, e, g, e, g, e, e, e,
    e, e, e, e, e, e, e, e
    ]

sense.clear(0, 0, 0)
x, y, z = sense.get_accelerometer_raw().
values()

def walking():
    for i in range(10):
        sense.set_pixels(pet1)
        time.sleep(0.5)
        sense.set_pixels(pet2)
        time.sleep(0.5)

while x<2 and y<2 and z<2:
    x, y, z = sense.get_accelerometer_
raw().values()

walking()
```

FIND THIS & OTHER RESOURCES: raspberrypi.org/resources

SENSE HAT DATA LOGGER

Use your Sense HAT to take environmental readings, so you can walk around the house saying "sensors indicate…"

You'll Need

- Sense HAT **magpi.cc/SenseHAT**
- Sense HAT Python library **magpi.cc/1RKRoqc** (pre-installed with Raspbian Jessie)
- Python 3 Pillow **magpi.cc/1Srsr3C**
- Python 3 Evdev **magpi.cc/1Srsso7**

During the Astro Pi mission, a pair of Raspberry Pis with Sense HATs attached will be capturing and logging a range of data about life on board the International Space Station.

In this activity you will use a Raspberry Pi, a Sense HAT, and some Python code to create your own data-logging tool, which you can use to capture interesting data and perform experiments.

Before we start, we need to install the relevant software. Open a Terminal by going to **Menu**, **Accessories**, and **Terminal**. Type in the following and then press **ENTER:**

```
sudo pip-3.2 install pillow
sudo pip-3.2 install evdev
```

First, we'll write a short script to get data from the Sense HAT and output it to the screen. Using the sensors, we can capture the following data:

Temperature (This can be read by two different sensors)
Humidity
Pressure
Movement (This is actually made up of twelve different sensor readings)

To begin your script, you need to boot your Raspberry Pi into desktop mode and run IDLE for Python 3 from the **Programming** section of the **Menu – Python 3 (IDLE)**. Once IDLE has loaded, you'll need to select **File** and then **New File**, which will load a separate window in which you can write your code.

In your blank window (which should be on the right-hand side), add the lines of Python code from **Fig 1**. Note that the lines starting with a **#** symbol are comments and are ignored by the computer. You should use comments here to break your code into four sections, which will make it easier to build your program as it becomes more complex.

The first section, **Libraries**, is where you'll import code that will give your program extra abilities. The line `from sense_hat import SenseHat` allows your program to use the Sense HAT hardware.

The section headed **Logging Settings** is where you'll be able to control different features of your logger program.

The third section, **Functions**, will contain short 'chunks' of reusable code which do a specific job, such as writing the current data to a file.

The final section, **Main Program**, is the part of your code which uses each of the functions in the correct sequence to run the whole program.

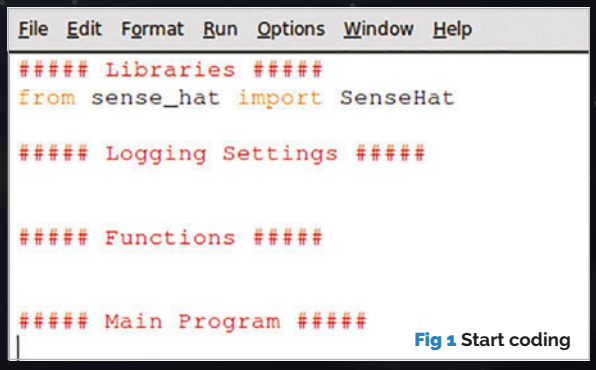


Fig 1 Start coding

To get data from the Sense HAT, you'll need to write a function called **get_sense_data**, which will check each sensor in turn and store the sensor data in a list. The function should be added to the **Functions** section.

```
def get_sense_data():
  sense_data=[]
  sense_data.append(
sense.get_temperature_from_humidity())
  sense_data.append(
sense.get_temperature_from_pressure())
  sense_data.append(sense.get_humidity())
  sense_data.append(sense.get_pressure())
```

The first line defines your function name. The second sets up an empty **list** structure, into which you'll add data. The next four lines obtain data from some sensors and add (or append) them to the **sense_data** list.

The rest of the sensors are a bit more complex, as they each give three values back. In the code below, you are asking the Sense HAT for each sensor's values and then extending the **sense_data** list by them.

```
  o = sense.get_orientation()
  yaw = o["yaw"]
  pitch = o["pitch"]
  roll = o["roll"]
  sense_data.extend([pitch,roll,yaw])

  mag = sense.get_compass_raw()
  mag_x = mag["x"]
  mag_y = mag["y"]
  mag_z = mag["z"]
  sense_data.extend([mag_x,mag_y,mag_z])

  acc = sense.get_accelerometer_raw()
  x = acc["x"]
  y = acc["y"]
  z = acc["z"]
  sense_data.extend([x,y,z])

  gyro = sense.get_gyroscope_raw()
  gyro_x = gyro["x"]
  gyro_y = gyro["y"]
  gyro_z = gyro["z"]
  sense_data.extend([gyro_x,gyro_y,gyro_z])
```

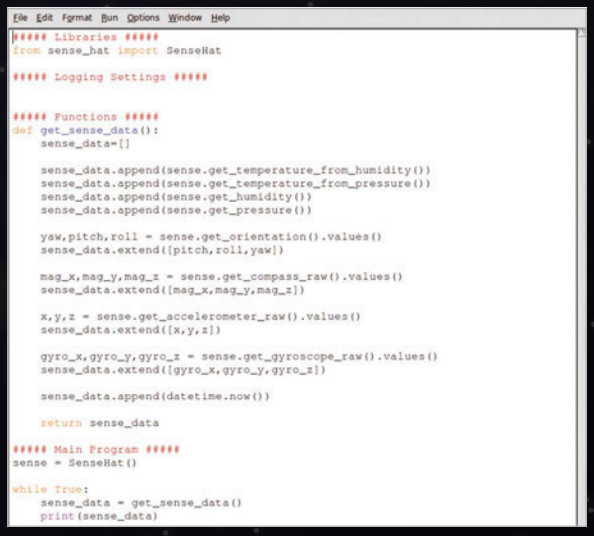


Fig 2 Right now we're getting basic data from the Sense HAT, but not using it for anything

```
  sense_data.append(datetime.now())
  return sense_data
```

The current time is also appended. The final line of the function sends the **sense_data** list to where the main program will ask for it.

Next, you'll need to add the following lines to your **Main Program** section. This will do two things: create a **sense** object, which represents the Sense HAT, and repeatedly **get_sense_data** from the sensors and display it.

```
sense = SenseHat()

while True:
  sense_data = get_sense_data()
  print(sense_data)
```

Your program should now look like **Fig 2**.

You can now test your logger. First, you should save it: press **CTRL+S** and choose a name, such as **Sense_Logger_v1.py**. Once the program is saved, you can run it by pressing **F5**. You should see lots of text scrolling past, as shown in **Fig 3** (overleaf).

The highlighted section shows a single line of data bundled together into a list; you should be able to tell which sensor data is which. To stop the program running, you can press **CTRL+C** to cancel the execution.

Writing the data to file

The program you have produced so far is able to continually check the Sense HAT sensors and write this data to the screen. Unless you're a very fast reader, however, this is not very helpful.

What would be more useful would be to write this data to a CSV (comma-separated values) file, which you can examine once your logging program has finished. To create this file, you'll need to specify the file name for it, add a header row to the start of the file, convert each list of data into a line of text for the file, and

Fig 3 The raw data provided by the Sense HAT is a little tricky to understand without parsing

periodically write a batch of data out to the file. The first thing you need to do is to add two lines to your **Settings** section. These are:

```
FILENAME = ""
WRITE_FREQUENCY = 50
```

The first line here will be used to choose a file name for the data output, and the second will set how often the program will write data out to the file. In this case, it will collect 50 lines of data and then add these to the file in one go.

Next, you'll need to add a **log_data** function, which will convert the **sense_data** to a line of comma-separated values ready to be written to the file. The line of text will be added to a list called **batch_data**, which will store the data until it's written to the file.

Add the following code after the **Functions** heading and before the **get_sense_data** function.

```
def log_data():
    output_string = ",".join(str(value) for value in sense_data)
    batch_data.append(output_string)
```

The first line of this section takes each value in the **sense_data** list, converts them to **strings** (text), and

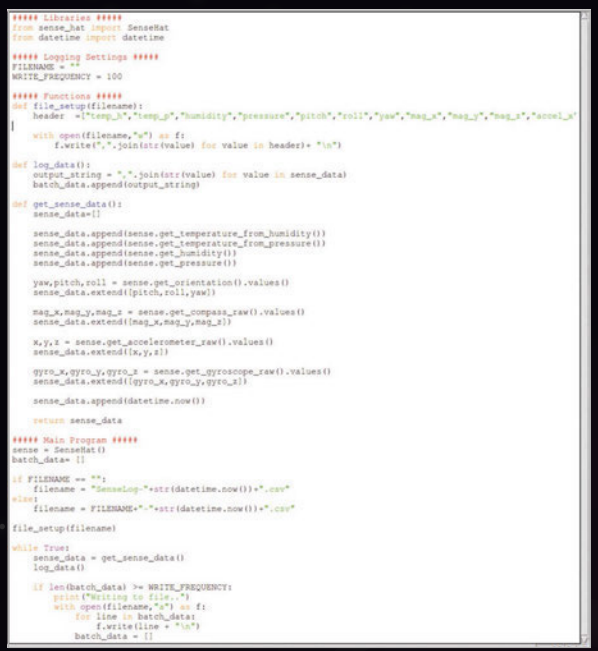


Fig 4 The code will now output something that is much more readable than the raw data

then joins them together with the **,** symbol. Therefore, a line like this:

```
[26.7224178314209, 25.068750381469727,
53.77205276489258, 1014.18017578125,
3.8002126669234286, 306.1720338870328,
0.3019065275890227, 71.13333892822266,
59.19926834106445, 39.75812911987305,
0.9896639585494995, 0.12468399852514267,
-0.004147999919950962,
-0.0013064055237919092,
-0.0006561130285263062,
-0.0011542239226400852, datetime.
datetime(2015, 9, 23, 11, 53, 9, 267584)]
```

...gets converted to:

```
26.7224178314209,25.068750381469727,
53.77205276489258,1014.18017578125,
3.8002126669234286,306.1720338870328,
0.3019065275890227,71.13333892822266,
59.19926834106445,39.75812911987305,
0.9896639585494995,0.12468399852514267,
-0.004147999919950962,-0.0013064055237919092,
-0.0006561130285263062,-0.0011542239226400852,
2015-09-23 11:53:09.267584
```

Another function you'll need is **file_setup**, which will create a list of headings that will be written to the file before any data. This function is shown below, and needs to be added to the **Functions** section of your program.

```
def file_setup(filename):
    header  =["temp_h","temp_p","humidity","
pressure",
    "pitch","roll","yaw",
    "mag_x","mag_y","mag_z",
    "accel_x","accel_y","accel_z",
    "gyro_x","gyro_y","gyro_z",
    "timestamp"]

    with open(filename,"w") as f:
        f.write(",".join(str(value) for value
in header)+ "\n")
```

This function is slightly different to the previous one, as it needs an input (or **parameter**) in order to work; in this case, the input has been called **filename**. When the main program calls this function, it must also give the function the name of the file to write to. If it were called as **file_setup("output.csv")**, the function would create the file **output.csv**.

The function itself creates a list of header names, called **header**. It then opens a file in **write** mode (which will overwrite any previous data) and refers to that file as **f**. While the file is open, it joins all the list

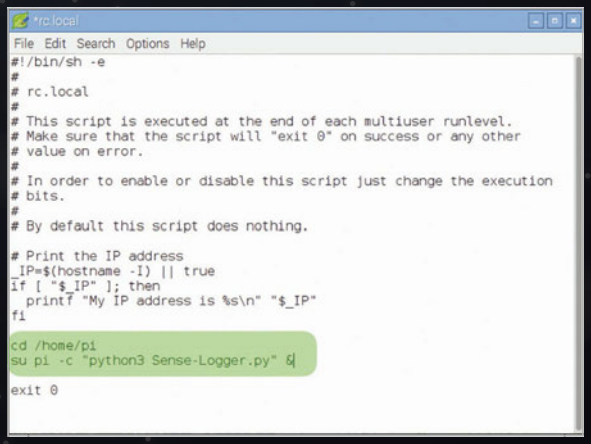


Fig 5 Add lines to rc.local to start the logger upon bootup

headings together using commas, and writes that line to the file.

The two functions and the settings you have added now need to be used in the main program.

Straight after the lines that read:

```
##### Main Program #####
sense = SenseHat()
```

...add the following:

```
batch_data= []

if FILENAME == "":
    filename = "SenseLog-"+str(datetime.
now())+".csv"
else:
    filename = FILENAME+"-"+str(datetime.
now())+".csv"

file_setup(filename)
```

The first line here creates an empty list that the program will keep adding **sense_data** lines to until it reaches 50 (or whatever value is set by **WRITE_FREQUENCY**). The **if / else** block checks whether a **FILENAME** has been set; if it hasn't, then the default of 'SenseLog' is used. The current date and time is also added to the file name.

Finally, the **file_setup** function is called and given the file name that was decided upon in the previous **if / else** block.

The last step is to change some of the logic inside the **while True:** loop. You need to make it collect **sense_data**, then use the **log_data** function to convert the **sense_data** into CSV form and add the the current **batch_data**. Once the data is logged, the program checks whether the size of **batch_data** exceeds the **WRITE_FREQUENCY** setting; if so, the data is written to a file and **batch_data** is reset.

Your **while True:** loop should be updated to look like this...

```
while True:
    sense_data = get_sense_data()
    log_data()

    if len(batch_data) >= WRITE_FREQUENCY:
        print("Writing to file..")
        with open(filename,"a") as f:
            for line in batch_data:
                f.write(line + "\n")
            batch_data = []
```

The line **print("Writing to file..")** is optional, but it will show whenever data is being written. The line **with open(filename,"a") as f:** opens the file in **append** mode, which adds the data at the end point of the file rather than overwriting. Your code should start looking like **Fig 4**.

When running the program, you should simply see the messages saying 'Writing to file...' every so often. You can stop logging by pressing **CTRL+C**.

Starting on boot

It's quite likely that you'll not want to have a screen, keyboard, and mouse handy every time you want to log data. A handy way to avoid this is to have your program run whenever your Raspberry Pi boots up. To do this, you will first need to open a Terminal window and enter the command **sudo leafpad /etc/rc.local**. The **rc.local** script is the last startup script to load as the Raspberry Pi boots. Anything you add to this script will load on boot. Once Leafpad has loaded, you should add two lines like the ones shown in **Fig 5**.

The first line changes to the directory where your data logger script is stored.

The second line changes to the **pi** user and runs the command **python3 Sense-Logger.py**; the **&** symbol makes this command run as a background task and allows the Raspberry Pi to continue with other tasks.

You'll need to update these lines to reflect the name and location of your program. The next time your Raspberry Pi boots, it should automatically start logging data.

Collect your data

Conduct an experiment involving a change in one condition, in order to measure and collect data. You could:

- Place your Raspberry Pi in the fridge and record the ambient temperature. What happens when you open the door? How quickly does the temperature inside the fridge return to normal?
- Drop your Raspberry Pi from a height and track the changes in orientation and acceleration (ensure you protect your Raspberry Pi carefully before dropping it).
- Send your Raspberry Pi high into the atmosphere using a high-altitude balloon and explore the changes in temperature, pressure, and humidity throughout the flight.

Language
>PYTHON 2.7

DOWNLOAD:
magpi.cc/1M4UQXR

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved...

NEED A PROBLEM SOLVED?

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

RASPBERRY PI ZERO

ISSUES BOOTING YOUR #PIZERO

>STEP-01

SD card

If you're using an old SD card that's been in use by a Raspberry Pi for a while, you may need to reinstall the operating system onto the card. Only Raspbian Jessie works on the Pi Zero, which you can install manually or via NOOBS.

>STEP-02

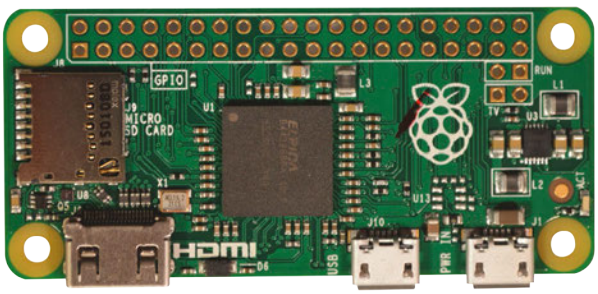
Power supply

While the Raspberry Pi Zero is very small and barely uses any power, you still need to make sure you have enough for the job. The official Pi power supply will provide enough power, along with most recent Android phone chargers.

>STEP-03

HDMI cable

Make sure the cable for the screen is plugged in properly, as the green light will stay on even if the HDMI cable is not correctly displaying an image. Check the cable or adaptor to make sure it's not faulty.



USB HUB PROBLEMS WITH #PIZERO

>STEP-01

Cycle

It may sound like a massive cliché, but turning the Raspberry Pi Zero off and on again while the USB hub is plugged in can work, especially if everything you want to use via the hub is plugged in as well.

>STEP-02

Powered hub

While the Raspberry Pi Zero has a lot more leeway in terms of what kind of hubs it will allow, making sure you can power the hub with an external power supply may help you to get it working.

>STEP-03

Check compatibility

Some USB hubs may just not be compatible for whatever reason. The Raspberry Pi forums (**magpi.cc/1NlH5rQ**) have plenty of questions and answers on using the Raspberry Pi, so you should be able to find more info.

WHAT ARE THE HOLES & PADS FOR?

GPIO

The main 40 pins lining the top of the Pi Zero are the standard 40 GPIO pins of a Raspberry Pi. You can either attach wires to the holes, or solder a header to it. It has the same lay-out as the other Pi's.

Other pins

Next to the GPIO there's a couple of holes that allow you to create a reset switch, which actually exist on other Raspberry Pi's as well. The other two are for component video out, allowing you to solder it up to an old TV.

Pads beneath

There are many pads on the underside. The ones underneath the USB ports are direct connections to the USB, while others are for testing purposes during creation and manufacturing

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

What is the username and password for the Raspberry Pi?
The default username for Raspbian is "pi" (without any quotation marks) and the default password is "raspberry" (again, don't include the quotation marks). If this doesn't work, check the information about your specific distro on the downloads page.

Why does nothing happen when I type in my password? Did my Raspberry Pi freeze?
To protect your information, Linux doesn't display anything when typing in passwords in the Bash prompt or the terminal. As long as you were able to see the username being typed in, your keyboard is working correctly.

What are the differences between the non-Zero models of Raspberry Pi?
These are the other models of the Raspberry Pi available: the Pi 2 Model B, and the Pi 1 Model B+ and A+. The Model A+ is the low-cost variant of the Raspberry Pi. It has 256MB RAM, one USB port, 40 GPIO pins and no Ethernet port. The Model B+ is the final revision of the original Raspberry Pi. It has 512MB RAM (twice as much as the A+), four USB ports, 40 GPIO pins, and an Ethernet port. In February 2015, it was superseded by the Pi 2 Model B, the second generation of the Raspberry Pi. The Pi 2 shares many specs with the Pi 1 B+, but it uses a 900MHz quad-core ARM Cortex-A7 CPU and has 1GB RAM. The Pi 2 is completely compatible with first generation boards, and is the model we recommend for use in schools, due to its flexibility for the learner. You can check our products pages for more details on current boards. There are also some models of Raspberry Pi which are no longer in production, but which may be available second-hand or from resellers. The Model A was the initial low-cost variant of the Pi. It was replaced by the smaller, neater Model A+ in November 2014; it shares the same specs as the A+, but has only 26 GPIO pins. The Model B was the previous incarnation of the B+; again, it shares most of the same specs, but has only 2 USB ports and 26 GPIO pins.

THE MAGPI APP

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

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

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

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

Get it on Google play

Available on the App Store

01 MOTIVATIONAL BATHROOM SCALE

DOT SILVERMAN

Dot Silverman is currently a junior research scientist in Autodesk's Bio/Nano Research Team. When not in the lab, she enjoys long runs, African dance, and crochet.

magpi.cc/1RekKjz

A digital bathroom scale is hacked to measure your weight

10 AMAZING NEW YEAR'S PROJECTS

It's the start of a new year, and (hopefully) the start of a new you. Start 2016 as you mean to go on with these amazing projects...

We all like making New Year's resolutions. Sticking to them is another matter, and there's only so many times you can promise to give up cake and fail after three days.

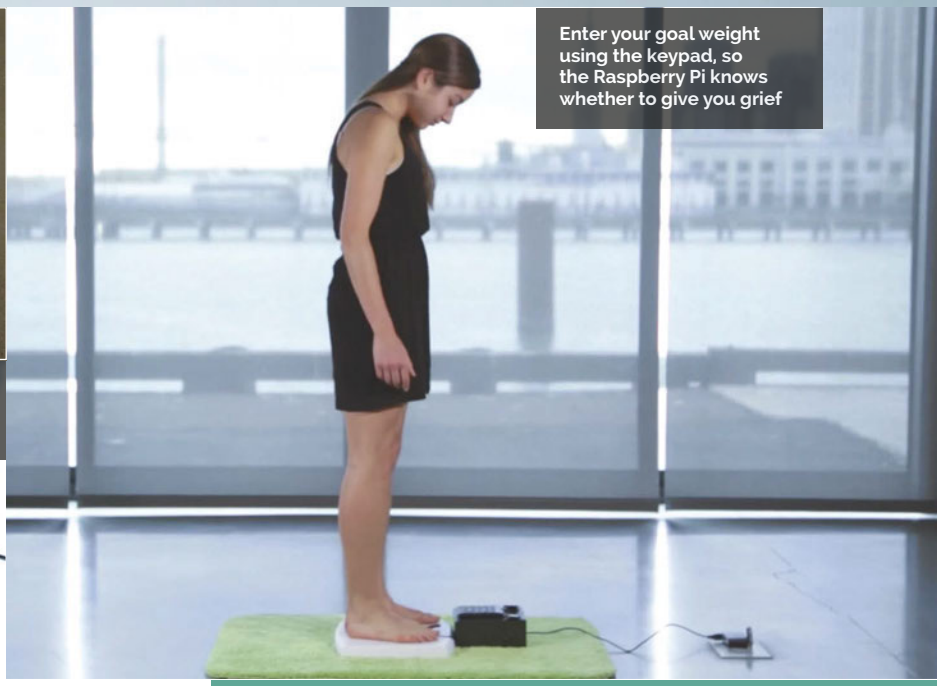
So, rather than resolving not to do something fun, we thought it would be a great idea to find the greatest Raspberry Pi-powered tech projects with a healthy, feel-good vibe. The makers of these projects have done us proud. What we have here are some of the world's coolest projects to see you into the new year.

These tech projects will help you live a little healthier, be a better citizen, and save some money to boot. Not bad, eh? More importantly, you can do all this while having fun.

Living a little healthier is a great place to start, so our first project is Dot Silverman's Motivational Bathroom Scale. "You type your goal weight into the scales using the keypad," explains Dot, "and the device compares your weight to your goal weight.

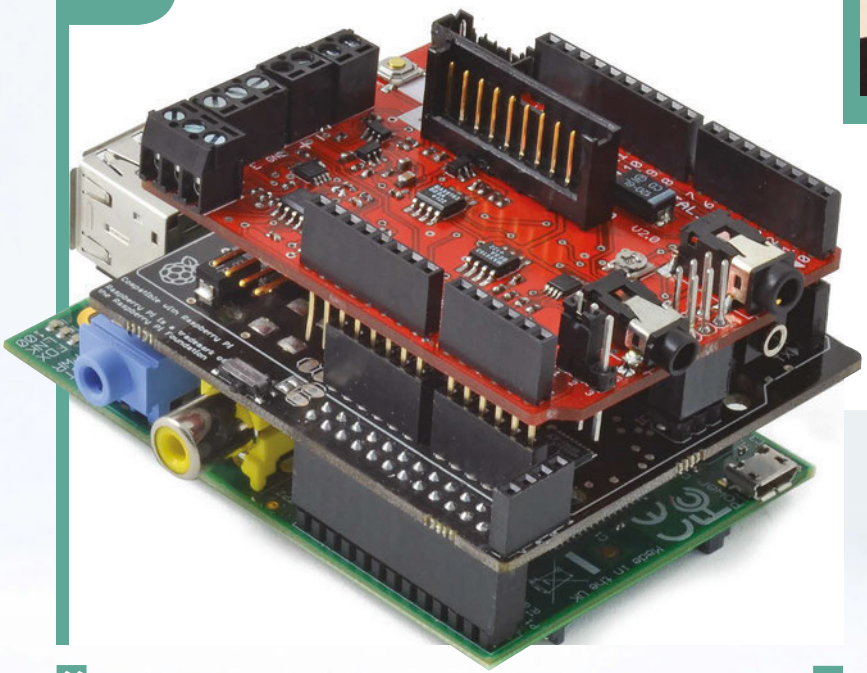
"If you're on target, it'll give you a flattering compliment. Otherwise, it'll sass you to get back to the gym. Remember that you're beautiful, no matter what the scale might tell you."

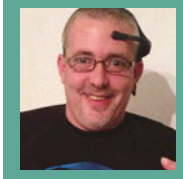
A Raspberry Pi is hooked up to the scale and speakers to provide audio feedback



Enter your goal weight using the keypad, so the Raspberry Pi knows whether to give you grief

02 E-WHEELCHAIR





PHIL CASE

Phil Case is an English maker who was disabled after injuring his C spine in an accident.
magpi.cc/1lU6z6f

The E-Wheelchair incorporates a Cooking Hacks E-Health v2 Sensor to monitor vital body parameters (**cooking-hacks.com**). The E-Health v2 and sensor attachments enable the Raspberry Pi to monitor body levels, such as blood glucose.

> "What we have here are some of the world's coolest projects to see you into the new year"

If you want to get a little more serious about monitoring health, Cooking Hacks' E-Health v2 sensor is worth investigating. It enables you to connect nine sensors to the Pi to measure blood pressure, oxygen levels, airflow, body temperature, and glucose levels, along with electrocardiogram and electromyography results.

One Raspberry Pi project maker making use of the E-Health sensor is Preston-based Phil Case, whose life goal is to build an affordable smart wheelchair. Phil's 'E-wheelchair' is mind-controlled, using Neurosky MindWave Mobile and Mindflex EEG products to measure the user's brain activity. The E-Health system is used to monitor electrocardiogram levels and blood sugars, and he plans to implement a body position sensor.

Phil is raising money towards the development of his smart wheelchair project, and a GoFundMe page can be found for those looking to be more charitible in 2016 (**gofundme.com/hfymdo**).

03 TECHFUGEES

MIKE BUTCHER

Mike is the founder of TechHub and editor at large at TechCrunch. He initially proposed the concept of Techfugees. **techfugees.com**

Techfugees is a community response to the European refugee crisis. It runs conferences and hackathons, and uses a global network of collaborators to address the crisis in technological ways.

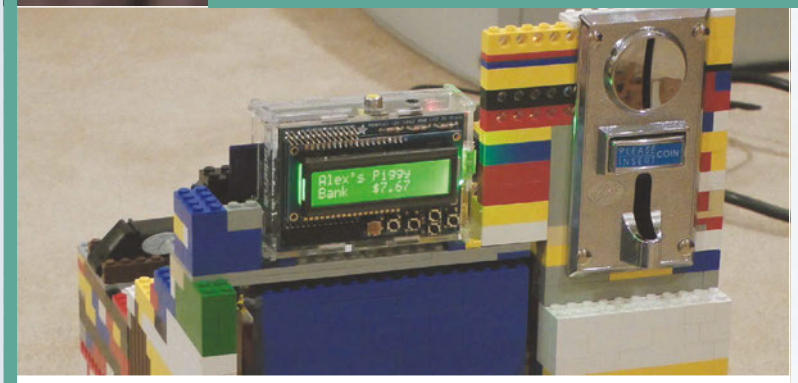
TECH FOR CHANGE

The power of geeks to organise for a good cause was highlighted this year thanks to Techfugees (**techfugees.com**), a UK tech community response to the European refugee crisis. The initiative brings together hackers in the UK to crowdsource ideas. So far, Techfugees has organised two conferences and a hackathon, and raised £5,000 to provide free WiFi to the Calais refugee camp. So how about for the New Year you invest some time in providing tech skills to charities?

04 RASPBERRY PI PIGGY BANK

ALEX STRANDBERG

Alex Strandberg is a student at Cornell University College of Engineering. **youtu.be/XanRVZgY6ow**



The Raspberry Pi Piggy Bank uses a coin-sorting mechanism to place coins in the correct slots. It counts, sorts and stores coins using a Lego mechanism that rotates for each coin. A fingerprint sensor is used to provide access to the bank.

> So how about for the New Year you invest some time in providing tech skills to charities?

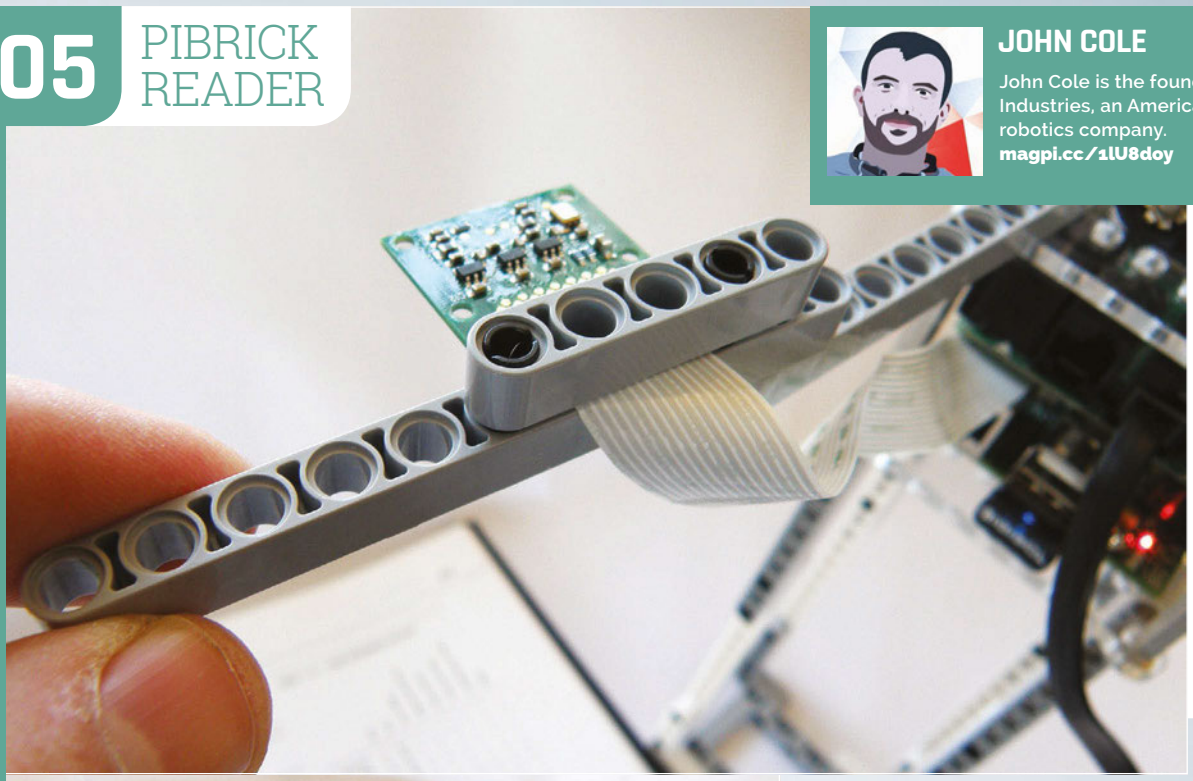
MONEY MACHINE

If your New Year's resolution is to save more money, then take a look at Alex Strandberg's amazing Raspberry Pi Piggy Bank. "After a coin is inserted, it's placed into a stack with the same type of coin," explains Alex. "The LCD screen displays the total amount of money in the bank. If I want to access the coin from inside, I hold my finger on the sensor to unlock the bank."

Another Lego Mindstorms-based project to investigate is the fabulous BrickPi Bookreader, built by Dexter Industries founder John Cole. This robot manually flips pages, photographs each page, and then reads it out loud. "We wanted to build a digitizer that could read books out loud," John tells us. "We were fascinated by the Google Books project and thought 'why couldn't we build this at home?'"

You'll need a Dexter Industries BrickPi device and Lego Mindstorms equipment. The BrickPi turns a Raspberry Pi into the brains for Lego Mindstorms robots. It controls two Lego EV3 motors: one pushes the pages up, and another acts as an arm and flips it over. The Pi Camera Module snaps an image of the page, and OCR software turns it

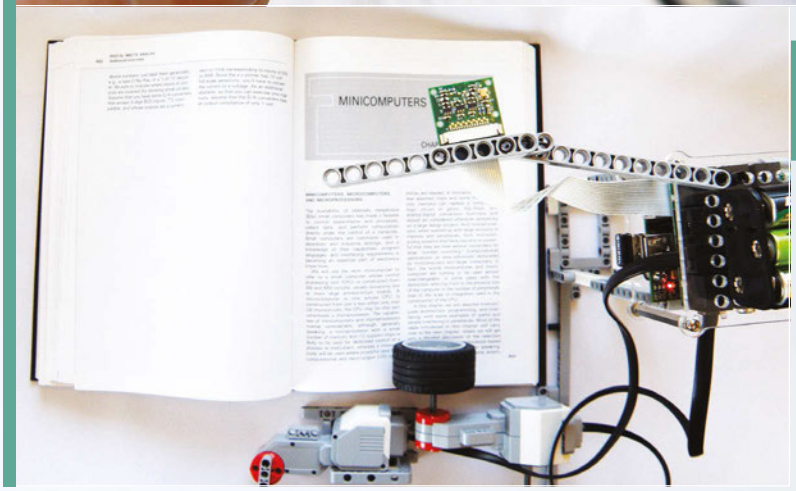
05 PIBRICK READER



JOHN COLE

John Cole is the founder of Dexter Industries, an American educational robotics company.
magpi.cc/1lU8doy

This project uses a PiBrick and Lego Mindstorms kit to recreate the robots Google built for the Google Books project. It flips pages, scans them, and reads each page out using voice reader software.



into text. "Just for fun, we used some free text-to-speech software so the Raspberry Pi reads the book out loud," says John.

NATURE CALLING

If your New Year's resolution is to get closer to nature, then one project you should look at is Sam Webster's Tweety Pi Bird Box (**magpi.cc/1lU7cN9**). To be fair, there are lots of Raspberry Pi bird box projects out in the wild. We really like Sam's, though, because it takes things a bit further than most. Sam has fitted his bird box with a PIR sensor, and hooked it up to Twitter. Whenever a bird flits into the box, it automatically starts sharing updates to **@tweetypibirdbox**.

We're big fans of eco projects here at *The MagPi*, and learning how to use solar power in your Pi projects is a noble aim for the upcoming year. Solar panels can be picked up for around £20, and can be used to provide direct power or charge up an attached battery. Veteran maker Koff has a great tutorial over on Instructables (**magpi.cc/1lU7fZh**).

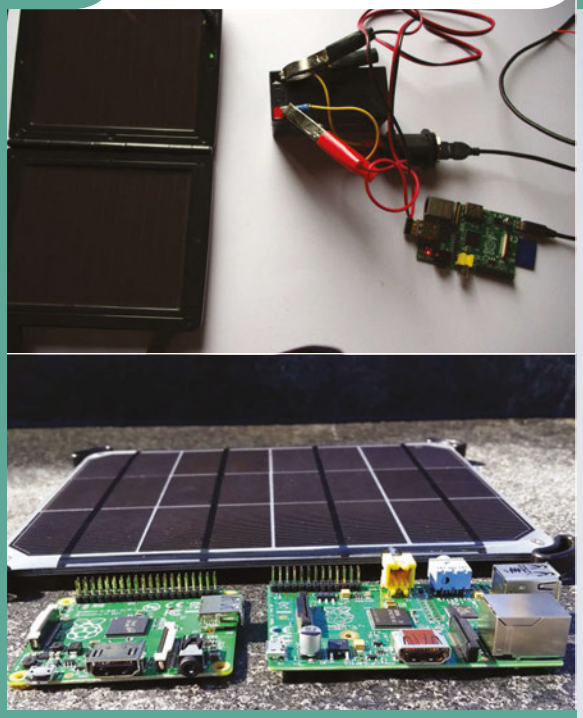
06 TWEETY PI BIRD BOX

SAM WEBSTER

Dr Samuel Webster is an anatomy and embryology lecturer at Swansea University.
magpi.cc/1lU7cN9

The Tweety Pi Bird Box uses a PIR sensor to detect movement, and the Pi Camera Module takes photos. A Python script then tweets each image from @**tweetypibirdbox**. It's a fun project that lets you watch the birds in your garden.

07 SOLAR POWERED RASPBERRY PI



KOFF

Koff is an international man of mystery, but he's been making stuff since the 1980s and has some great projects on his YouTube and Instructables pages.
magpi.cc/1lU7fZh

Running a Raspberry Pi from solar power is a great project. You get to learn how solar power works, and you can create devices that run outdoors under their own steam.

08 RASPBERRY PI WALL-MOUNTED GOOGLE CALENDAR

ALEX PINE

Alex Pine is an IT service engineer from Adelaide, South Australia. His projects have been viewed over 800,000 times.
magpi.cc/1Y94RtZ

This popular project turns an old HDMI monitor into a wall-mounted display that shows information from Google Calendar. The Raspberry Pi is placed on the rear of the monitor, and it is bracket-mounted to the wall.

> "Email is a modern curse, but practising inbox zero can bring a lot of relief"

GET ORGANISED

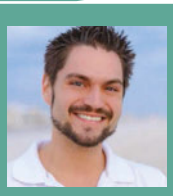
If one of your New Year's resolutions is to be a bit more organised (ours usually is), then we have three great projects for you to take a look at. The first is a fantastic wall-mounted Google Calendar by Alex Pine (**magpi.cc/1Y94RtZ**). You can use it to create a shared calendar for yourself and your family, so everyone can see exactly what's going on.

Our second productivity project is Michael Mitchell's Inbox Zero Taunter (**magpi.cc/1Y953cF**). Email is a modern curse, but practising inbox zero can bring a lot of relief from mail overload. The bright light of a Raspberry Pi LED reminds you that you've got email in your inbox that needs to be dealt with. It's a super-simple project – just one LED and around ten lines of code – but once configured, it'll hassle you into clearing out an email account.

Finally, if you're in serious need of organisational help, then why not turn your Raspberry Pi into a voice-activated digital assistant? We love the Raspberri Personal Assistant by Jan Wante. He's turned a 1950s Bakelite Televox intercom into a voice-controlled digital assistant (think Siri inside an old American radio). Inside is a Raspberry Pi with a USB audio sound card, running Steven Hickson's Voice Command software (**magpi.cc/1Y95Pqc**).

You will undoubtedly learn something new from building any of these projects, and learning something new is the best New Year's resolution you can make in any case. Whatever project you choose to build in 2016, make sure you have fun.

09 INBOX ZERO TAUNTER



MICHAEL MITCHELL

Michael Mitchell lives in Tallahassee, Florida and runs the MitchTech website. He creates projects for Raspberry Pi, Arduino, Android, and Ubuntu.
magpi.cc/1Y953cF

The Inbox Zero Taunter is a simple project that uses an LED connected to your Raspberry Pi. A Python script is used to detect emails in your Gmail inbox, and light up the LED.

10 RASPBERRI PERSONAL ASSISTANT

JAN WANTE

Jan Wante is a maker from the Netherlands. He's a member of the J^3 Associates, who run a project-making lab.
j3associates.be

The Raspberri Personal Assistant project transforms a 1950s Bakelite speaker into a voice-activated personal assistant. Like Siri, it can be programmed to answer requests about the weather and news, or to take messages.

Maker Says

"All you need to start transforming the world around you"

Bare Conductive

BARE CONDUCTIVE TOUCH BOARD STARTER KIT

With slick presentation, is the Touch Board the conductive marvel Bare Conductive has promised?

The first thing to discuss about the Bare Conductive Touch Board Starter Kit has to be its price. At just a shade below £100 in the UK and $150 in the US, it's certainly no impulse buy for the maker on a budget. At the same time, though, it's hard to call it bad value for money when you see the care and attention that has gone into its creation.

The box, a sizeable affair that has no hope of slipping through a letter box while you're out, contains an impressive selection of parts. The highlight, naturally, is the Touch Board itself, which is joined by a generous pot of conductive paint with a thick brush, a tube of the same paint with a fine-tipped nozzle for more precise application, a self-powered rechargeable speaker, USB cables, a microSD card reader, and banana clip cables. This is in addition to a rolled-up stencil set (about which more later) and an impressive, oversize, full-colour guidebook covering the kit's three primary projects.

The guidebook is far from an afterthought, but it's hardly required to get started. The packaging of the Touch Board itself, a box within a box, doubles as a quickstart guide and reveals a very clever feature: pre-recorded instructional messages already loaded onto the bundled 128MB microSD card.

The Touch Board, for those unfamiliar, is an Arduino-compatible device designed to make working with conductive paint, thread, and even Play-Doh as simple as possible. It arrives preconfigured to play MP3 files from the SD card each time one of its 12 electrode inputs is touched. This is then used to drive the user's introduction: connect the micro-USB cable and the bundled speaker or a pair of headphones, touch input E0 with your finger, and you'll hear a congratulatory message; press E1 and you'll learn about the board's inputs. This process continues through to E11, which sends you off to the Bare Conductive website to learn more.

With the Starter Kit, though, you have offline support too. The aforementioned guidebook does a fantastic job of walking you through three example projects in a step-by-step fashion. With full-colour pictures at every step, it's hard to get lost or confused.

The first suggested project is the biggest: learning how to make graphical sensors. This involves the rolled-up stencil and overlays, which come with handy sticky tabs.

Related

MAKEY MAKEY

The device that brought the banana piano to the masses, the Makey Makey is a cheaper way to get started with conductive materials.

£40 / $50

makeymakey.com

bareconductive.com

£95 / $150

> "The guidebook does a fantastic job of walking you through three example projects"

Attach the stencil to a wall – or, if you're not sure about the somewhat permanent nature of the paint becoming a fixture of your house, a large sheet of paper or card – and brush on the paint. Decorate with the coloured overlays, use the thinner paint tube to draw wires leading to the Touch Board, copy the MP3s to the microSD, and voila: an interactive house.

The remaining two projects are variations on a theme. The first adds touch-sensitivity to everyday objects, such as a pill bottle which reminds you of dosages, or a photo frame that describes its own scene. The final project has you brushing the paint onto the floor to create an intruder alarm, albeit one which only works if said intruder is barefoot.

The three projects detail some, but not all, of the Touch Board's capabilities. A page of extended projects at the back of the guidebook points you towards the website to learn about the board's more advanced capabilities, such as reprogramming it to operate as a non-contact distance sensor or a Makey Makey-style musical instrument.

At this point, the Touch Board is likely to have entranced children but potentially turned more advanced hobbyists off. Towards the back of the guide, it's revealed that the Touch Board is more powerful than it would first appear: it's a fully functional Arduino, compatible with various Shields and add-ons to provide everything from Bluetooth connectivity to motor control.

A glance at the board itself reveals the familiar Arduino headers, albeit without pins, with full access to these for more advanced creations. This, combined with an Atmel ATmega microcontroller, gives the board the ability to do anything an Arduino can do, including interfacing with external hardware, such as motors, servos, sensors, or even the Raspberry Pi itself. Combined with the flexibility of Bare Conductive's clever paint, this gives the kit considerable legs when the three bundled projects are finished.

Last word

It's not cheap, but for anyone looking to get started with conductive paint or touch-based projects, this kit is easy to recommend and very well put together, with considerable attention to detail.

★★★★★

magpi.cc/1Ue0ugK

£12.99 / $20

Maker Says

"The security camera solution for the Raspberry Pi"

SB Components

SPI-BOX

A very simple solution to creating a motion-detecting security device that won't break the bank

It's one of those classic tutorials in tech mags with small PCs and microcontrollers: the motion-detecting camera. There are many ways you can do it – some easier than others – and the Raspberry Pi offers many tools to make a project like this quite simple in general.

Enter SB Components with a dedicated motion-detecting camera kit for the Raspberry Pi. Designed as a cheap security system, the features that make up the kit are quite standard: a PIR motion detector, a specialised case for the whole setup, and even a custom version of Raspbian with software for sending security alerts. All for just over £10; throw in the cost of a Raspberry Pi B+ or 2 with a Pi Camera and – on paper – you've got yourself a pretty good security camera for about £50.

Assembly is dead simple. The case splits in two and the Raspberry Pi clips into the bottom part, with full access to all the major ports around the edges. The PIR and Pi Camera can be screwed to the lid, while a set of three wires is used to connect the PIR directly to the GPIO ports. Pop in the supplied SD card and you're all ready to get started.

The case has pretty standard mounting holes on the back for fitting over a couple of screws attached to the wall. You'll need to set up the emailing system on the device, accessed through the config files on the SD card, which you can probably do from another computer; however, all it then needs is a WiFi dongle and power to do its job.

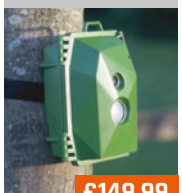
As the code is written in Python, it's quite simple to modify if you need to tweak the sensitivity, or the way it captures photos or video. It all works pretty well, especially for the price, although we wouldn't rely on it for any serious security work.

Really, this is as good as any other Raspberry Pi security camera. It'll be a lot better than one you try to put together yourself, and probably cheaper and neater too, so give it a look if you're interested in a bit of added security.

Related

NATUREBYTES

Although designed for a different use, the Naturebytes Wildlife Cam Kit could be configured for security. It's also weatherproof.



£149.99

magpi.cc/1QnR4k8

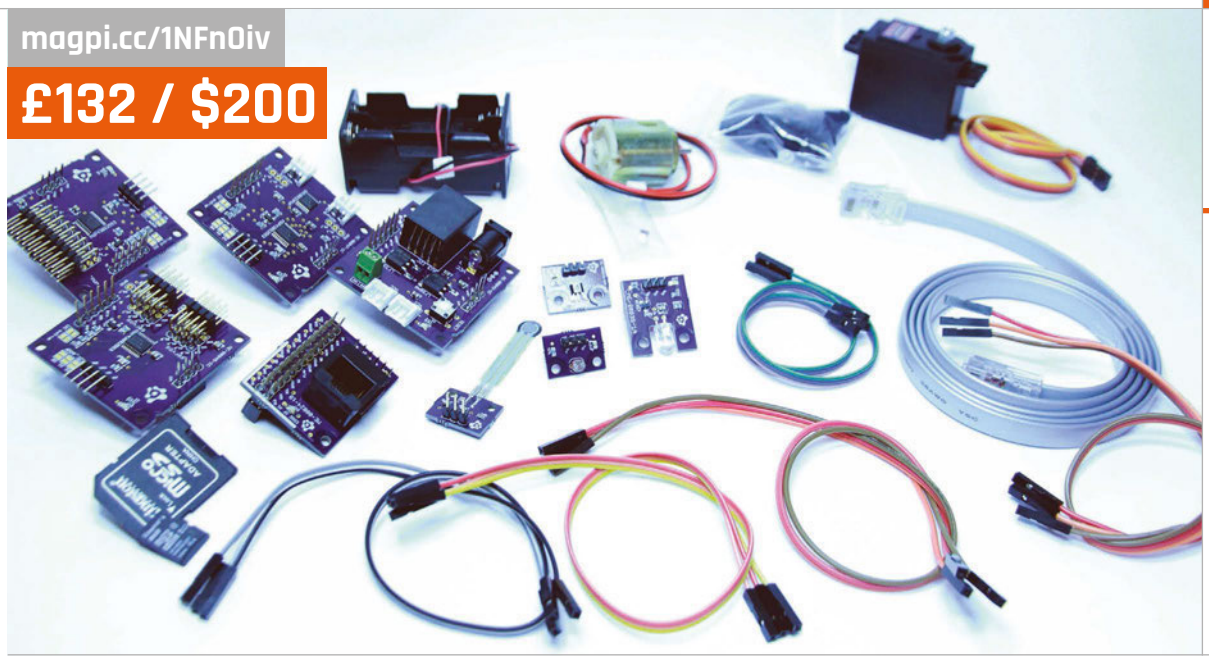
Last word

A great little kit that very simply creates a serviceable Pi-powered security camera that's great for any indoor application. It also barely breaks £20 (with camera), making it a cheap way to keep a better eye on something.

★★★★☆

magpi.cc/1NFnOiv

£132 / $200



Maker Says

"Great for experienced users and novices alike

Magzor

MAGZOR MECHATRONICS STARTER PACK

A modular robot that can be built to your specification, without even having to write any code. Is it too good to be true?

At a time when Raspberry Pi robotics kits are branching out in all directions – whether they be simple, easy and cheap, or more advanced – it's nice to see something that is trying to bridge the gap from beginner to intermediate level.

The Magzor system starts with a kit of parts that can interact with each other in any combination, and hook up to the Raspberry Pi to create your own robot or mechatronic project.

The kit works by having you input what you want to use on a web interface, a bit like the blocks of a puzzle. The interface then creates a custom recipe of how the parts should fit together, and even offers some basic code to get them all working. It's quite intuitive, although we didn't get one of the more universal parts in the starter kit we received. This meant that we had to figure out what other bits of information it would need before giving us a build that was possible with our equipment.

The instructions themselves are good, but we did find ourselves slightly confused at times; at one point we were looking for a port which wasn't really labelled on the PCB in the way the instructions described. By the time we had finished an initial build, though, we had basically figured out the language that the instructions were trying to use.

In a large departure from how Raspberry Pi-centric devices work, the kit is programmed in C++. A basic script is generated that gets everything activated, which you can modify yourself to get it to your exact specifications. It's a weird jump in skill level from the construction phase to the programming phase, though, requiring you to manually compile your code rather than being able to run it live and make changes as you go.

While this level of programming is a good thing to learn, it does seem at slight odds with the nature of the quickly iterable and rebuildable robot. With the tutorials the way they are now, there's no easy step-by-step guide for creating the code and moving it to a Pi to then compile and run it; a bit more intuition and knowledge is required than some robotics beginners might possess. With some more focus on that side of the system, it will be a lot better, but for now it's maybe outside the range of the beginner-level folks.

Last word

There are some great ideas here, but some parts of the building and programming are a bit obtuse. It's quite good for people doing slightly more advanced mechatronics than simple robots, though.

★★★☆☆

Related

CAMJAM EDUKIT #3

Also reviewed in this issue, the CamJam kit allows you to build a robot and then fairly simply program it to perform various tasks. Much simpler and aimed towards a very beginner level of robotics, though.

£17/$26

magpi.cc/105UE0h

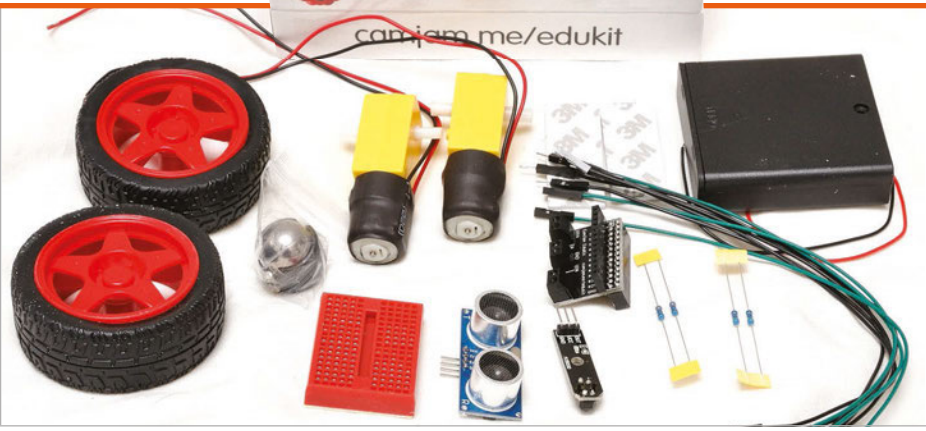
magpi.cc/1SWEq9s

£17 / $26

Maker Says

"Create your very own Raspberry Pi-powered robot"

CamJam



CAMJAM EDUKIT #3 ROBOTICS

A low-cost robot kit that can teach you how to put together a Raspberry Pi robot, as well as how to program it

If you cast your mind back a few issues, you may remember that we did a cover feature on how to create a £50 Pi-powered robot. The only real problem with the robot was that there were a lot of different components you'd have to buy from many different places to keep the price down. We didn't quite have the time to put together a kit for people to buy; it doesn't look like that matters too much any more, however, as the latest CamJam EduKit is incredibly close to what we would have created.

A box full of very standard robotics components – and a custom board – make up the third CamJam kit. While it does have these standard components, it's not like a Lego or IKEA kit that some of these robot kits resemble. As they're components without a chassis (although the box it comes in will do in a pinch), the CamJam EduKit #3 is designed to give you more freedom in what you create and actually make you think about what you're constructing.

A breadboard is also supplied with the kit. While on its own you might just think it's another breadboard and reduces the soldering you need to do, it also gives you lots of ways to actually expand beyond the limits of the original robot, with extra sensors, components, and suchlike. This makes it very good for extracurricular learning beyond the scope of the original robot.

The online worksheets for the robot are very comprehensive, teaching you how to put it together and code it. As the release of the EduKit #3 is somewhat tied to Pi Wars, there are also tips on how to program it for some of the styles of challenges that were involved in that event. Indeed, we saw a few robots powered by this kit trundling around at Pi Wars, and one of them did extremely well on the line-follower course that others were failing with.

It's a great beginner's kit then, or at least a kit for people wanting to take a step further than some of the very simple kits. It teaches you a bit more about how the Raspberry Pi can actually control a robot, and gives you room to grow.

Last word

A great kit for beginners looking to be a bit more hands-on with the robots they create. There's plenty of excellent documentation to get them properly started, as well.

★★★★☆

Related

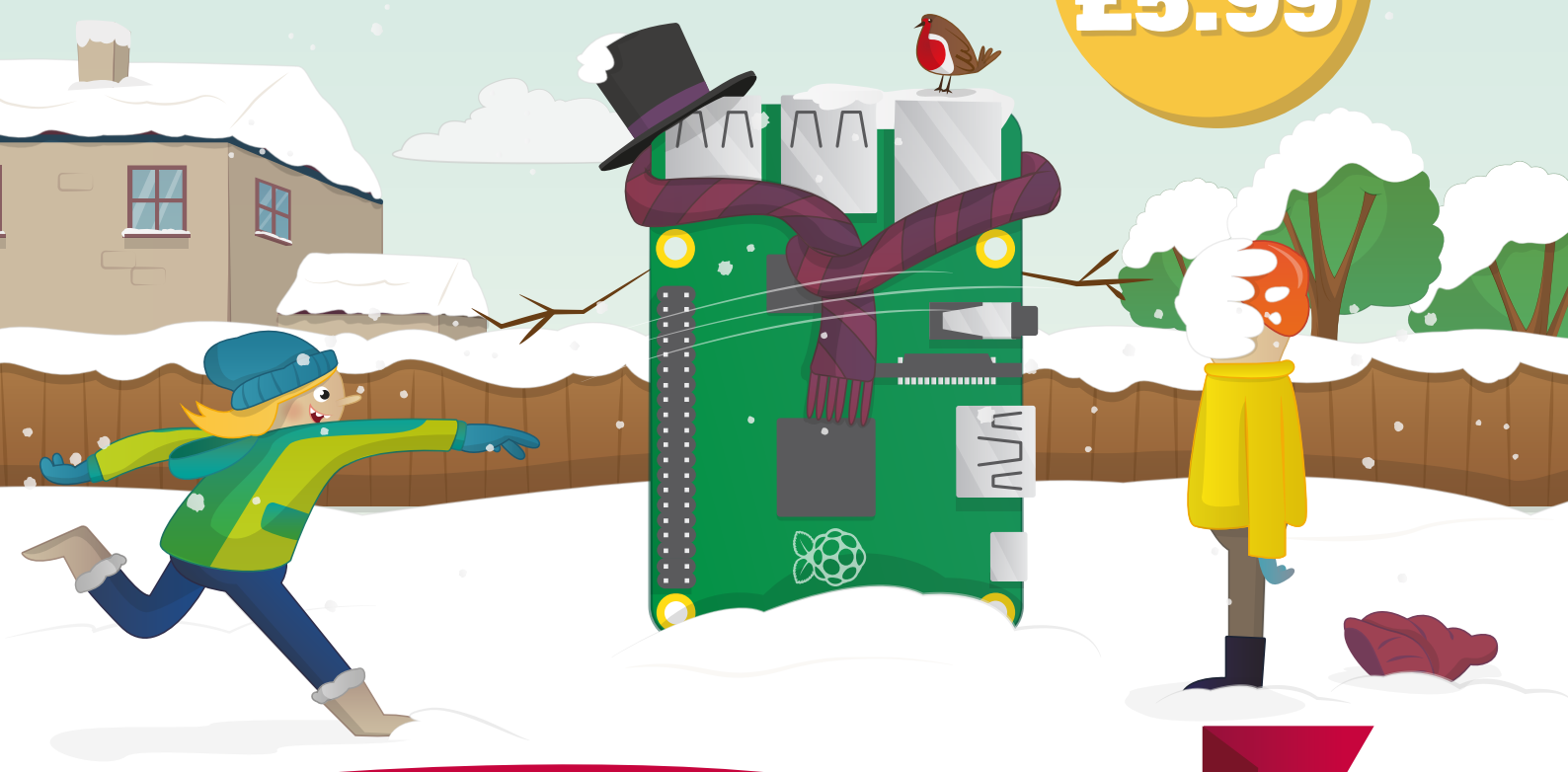
PI2GO

A little more pricey. However, this is also a great little starter kit that can teach you about robotics with the Pi.

£55 / $83

magpi.cc/1SWFHxi

2016 RASPBERRY PI CALENDAR



ONLY £5.99

FREE BONUS DAY!

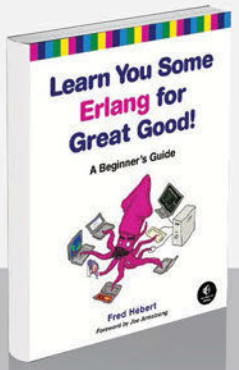
Raspberry Pi Birthday: 29th Feb 2016

swag.raspberrypi.org

RASPBERRY PI BESTSELLERS ERLANG

17 years after the source was opened, it's time you embraced the robust, functional language which shines at concurrency

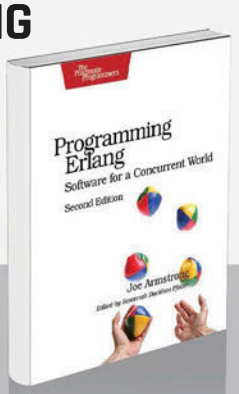
LEARN YOU SOME ERLANG FOR GREAT GOOD!



Author: Fred Hébert
Publisher: No Starch
Price: £33.50
ISBN: 978-1593274351
magpi.cc/1NuXisx

Accessible introduction full of real-world examples, with an honest look at the good and not so good of using Erlang. Comprehensive, and a good introduction to functional programming.

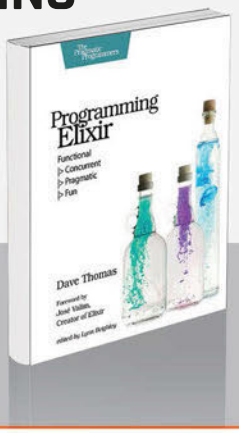
PROGRAMMING ERLANG



Authors: Joe Armstrong
Publisher: Pragmatic
Price: £27.99
ISBN: 978-1937785536
magpi.cc/1NuXoAm

From one of Erlang's creators, containing real insights into the thought process of programming in Erlang. The 2nd edition contains much for beginners, but finishes with an open-ended problem, 'Sherlock's Last Case'.

PROGRAMMING ELIXIR



Authors: Dave Thomas
Publisher: Pragmatic
Price: £23.99
ISBN: 978-1937785581
magpi.cc/1NuXzM4

Elixir runs on the Erlang VM, and the underlying Erlang/OTP architecture, but with an extendable, Ruby-like syntax that's quick and powerful in use. Full review in *MagPi* #33.

CLOJURE FOR THE BRAVE AND TRUE

Author: Daniel Higginbotham
Publisher: No Starch
Price: £23.50
ISBN: 978-1593275914
magpi.cc/1NuXHeC

Books with humour can be hit-or-miss affairs, but Higginbotham's introduction to the powerful JVM Lisp is such a good language tutorial that even those who usually prefer a drier approach will be bowled along.

Early on, readers are introduced to Emacs as REPL and editor: downloading the book's Emacs configuration package will actually give you one of the best starts in using Emacs; you'll soon be appreciating Emacs features like the kill ring, and discovering why it remains a top choice for editing and interacting with Lisp family languages after 40 years. After setting up your tools comes a crash course in Clojure fundamentals, with comparisons with how you would do similar things in languages like Ruby along the way, to highlight key differences such as immutable data.

"Clojure is so elegant that it's difficult to tell anyone anything about it without somehow improving them," writes Alan Dipert in the introduction. Improvement here involves fun examples – particularly if your idea of fun is violence towards hobbits – that can make the learning less onerous than some contrived 'real world' examples. Every topic is introduced in the best order for growing real understanding of Clojure, in a journey that will change the way you code.

Score ★★★★★

GIT FOR TEAMS

Author: Emma Jane Hogbin Westby
Publisher: O'Reilly
Price: £33.50
ISBN: 978-1491911181
magpi.cc/224fsew

"You can do more with Git than just build software," says Hogbin Westby of this people-first approach to version control. *Git for Teams* combines a practical look at best practices in Git usage with a look at the 'why' of Git – where it fits into workflow, and even how to build the right team for a project – and concludes with a useful section on Git hosting.

The chapter 'Workflows that work' covers the neglected essentials of documenting your process – something often left until after your project has got into a confusing mess. The practical chapters start with 'Teams of One', giving an excellent introduction to Git use, then the 'Rrrrgh!' chapter – rollbacks, reverts, resets, and rebasing – shows the best way to make amendments without losing code or having a negative impact on the workflow of team members. Reviews and bug-fixing get a chapter each, and the poorly named blame command receives a fair examination.

The section on various Git hosting services may help you find the right public or private repository. This book is a most useful single guide for those needing the technical and social side of managing a project, whether in traditional employment, an IoT startup, or an open source project.

Score ★★★★☆

PROGRAMMING THE RASPBERRY PI

Author: Simon Monk
Publisher: Tab Electronics
Price: £9.89
ISBN: 978-1259587405
magpi.cc/224f0lh

This could be the perfect introductory programming book to give to someone who's just got a Raspberry Pi for Christmas. With no wasted words, Monk introduces the Pi and its operating system, then teaches both Python and using Python with the Pi, in a direct and easily absorbed text that harks back to the best beginner guides of the 8-bit era.

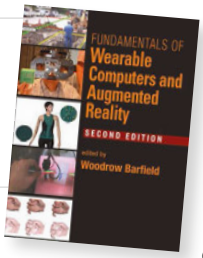
Skip the first two dozen pages if you are not new to the Pi, and dive into Python with an introduction that – through well-chosen examples, such as a dice rolling simulation – will have you learning conditionals, control flow, and comparison operators before you know it. Next, Hangman introduces functions, as well as strings, lists, and dictionaries. OOP is touched on with a temperature converter, then file handling and GUI programming (with Tkinter) through building on the earlier code examples. The Pygame chapter makes use of many of the techniques learned, then moves on to refactoring.

The same concise style is used to cover the Pi hardware for the rest of the book: GPIO pins, breadboard prototyping, connecting an Arduino, then a range of sample projects culminating in a Raspberry Pi robot. Unreservedly recommended for confident beginners of all ages.

Score ★★★★★

FUNDAMENTALS OF WEARABLE COMPUTERS AND AUGMENTED REALITY

Author: Woodrow Barfield
Publisher: CRC Press
Price: £95.00
ISBN: 978-1482243505
magpi.cc/224g3wG



As computers get small enough to mount on the cover of a magazine, and as sensors get better, wearable computing – data where you need it – is within reach of all. From Google Glass to Google Cardboard, ubiquitous computing is getting nearer and cheaper. Augmented reality is used across many industries, and children solder together wearable computers at MakeFests. If you're a maker or programmer involved in an AR project, this collection of essays will broaden your appreciation of the field considerably. Barfield presents 25 academic papers, over 700 pages, split over four sections: Introduction (with some shorter, more philosophical, overviews of the topic); The Technology (mostly displays and haptics, but also tracking); Augmented Reality (with some fascinating and varied cases); and Wearables. The last section particularly highlights integration into textiles and clothing, as well as presenting useful research on haptic rendering.

This is a collection of academic papers: extensive references are given, and the language is academic, though mostly quite approachable. The research crosses many disciplines, and non-mathematical readers will only be given pause by a couple of chapters. An expensive purchase, but perhaps a worthwhile one for the local makerspace library, particularly if you have an augmented reality project in development.

Score ★★★★☆

ESSENTIAL READING: NEW YEAR RESOLUTIONS

Kick start your resolution to learn something new, or even boost your IT career!

Resolution: Invent that Pi add-on!

Fritzing for Inventors

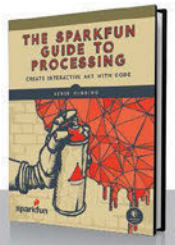
Author: Simon Monk
Publisher: Tab Electronics
Price: £21.99
ISBN: 978-0071844635
magpi.cc/224guXY

Develop, prototype, test, produce, and fund an electronics project. Full of useful tips.

Resolution: Get arty!

The SparkFun Guide to Processing

Author: Derek Runberg
Publisher: No Starch
Price: £19.99
ISBN: 978-1593276126
magpi.cc/224gyHd

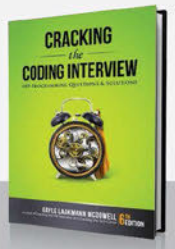


Lovingly produced guide to art and code with Processing, which now runs on the Raspberry Pi.

Resolution: Get a new job!

Cracking the Coding Interview

Author: Gayle Laakmann McDowell
Publisher: Career Cup
Price: £25.36
ISBN: 978-0984782857
magpi.cc/224gPd8

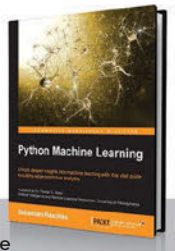


Start the New Year with a confident crack at a major coding job interview, with McDowell's peerless guide.

Resolution: Machine learning!

Python Machine Learning

Author: Sebastian Raschka
Publisher: Packt
Price: £28.99
ISBN: 978-1783555130
magpi.cc/224gXJw

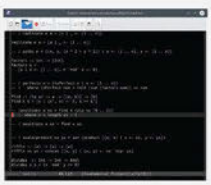


Process, learn from, and draw actionable insights out of the otherwise impenetrable walls of big data.

Resolution: Functional programming!

fp101

Author: Delft University
Publisher: EdX – Online MOOC
Price: Free – Online MOOC
ISBN: N/A
magpi.cc/224h3Rp



Get to grips with the what, why, and how of functional programming while seamlessly absorbing Haskell, then apply in the real world.

THE MONTH IN RASPBERRY PI

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

PI WARS: THE ROBOTS STRIKE BACK

The MagPi team attended Pi Wars on 5 December, to see the toughest and most innovative robots the country had to offer battling it out in challenges meant to test the skills of the machines and their makers. The day consisted of several events for the robots, including an obstacle course, bowling, line-following, a test of their stopping speed, and various other little challenges dotted around. The robots were also scored by judges on a variety of criteria, such as size, features, and aesthetics.

We also had a stand for the show, with a few copies of the very limited *MagPi #40*; as you might expect, people were queuing down the hall to grab a copy of the much sought-after magazine with the Pi Zero on the cover. We also had the NES controller from this month's tutorial on show, with many kids playing *Super Mario Bros 3* and illustrating why modern games consoles no longer have a Start button.

Here are a few photos from the day to show just how much fun 500 people can have at a Pi-powered community event.

The obstacle course had a big surprise of a rotating platform to make things trickier for the competing robots

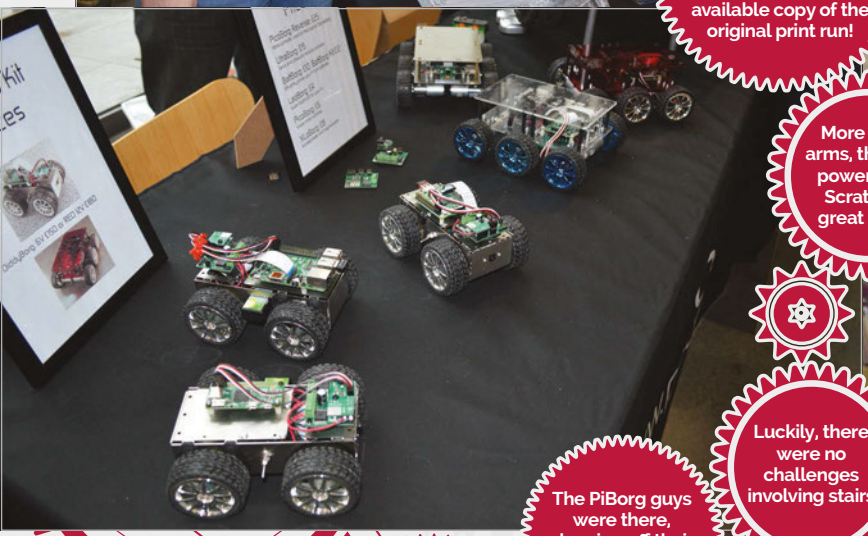
Different stalls had different robots to show off, including this great robot arm

Liz Upton was briefly seen firing about in the BigHak before racing out the door to go and fight crime

3D printers and 3D printed robots were on display

We were selling a few *MagPi*-related bits on the stall, as well as showing off the NES controller from this month's tutorial

We had a few copies of the lesser-spotted *MagPi #40* to sell, and this lucky woman bought the last available copy of the original print run!



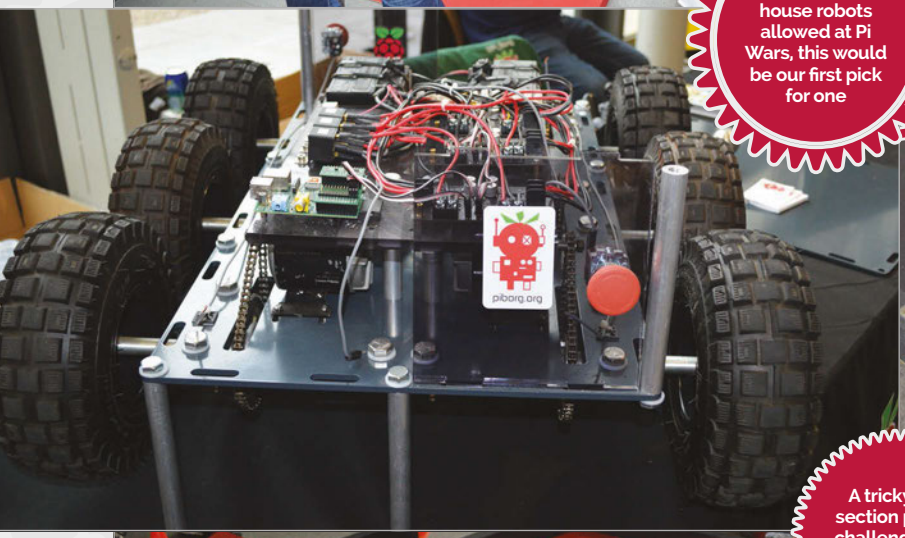
More robot arms, this time powered by Scratch to great effect

Luckily, there were no challenges involving stairs

The PiBorg guys were there, showing off their excellent range of robots

The bowling event brought out many unique robotic techniques

3D printing parts for robots can be a great way to build unique-looking machines

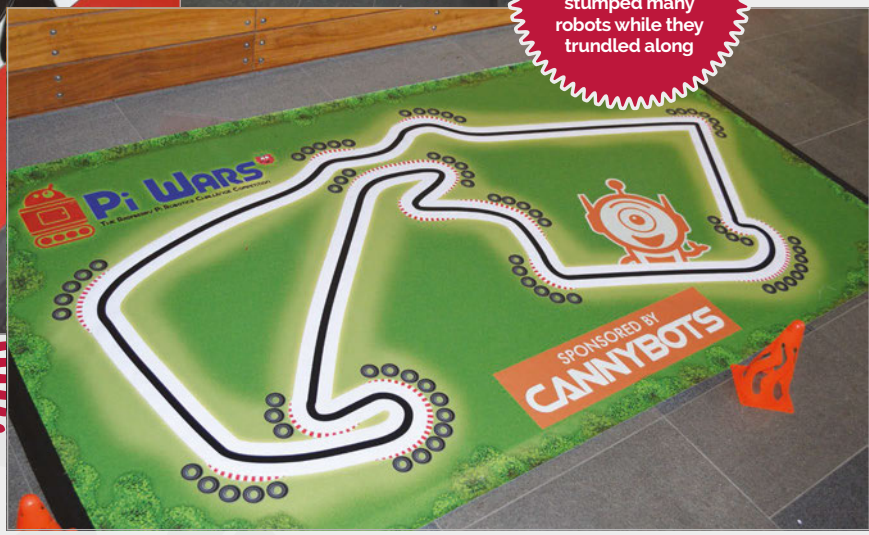


If there were house robots allowed at Pi Wars, this would be our first pick for one

A contender tackles the obstacle course – the ramp here was a real killer for the smaller ones

A tricky hilly section proved challenging for robots with a long wheel base

The turntable, built by PiBorg, created an interesting challenge to overcome



Looking deceptively simple, the line-following course stumped many robots while they trundled along

CROWDFUND THIS!

The best crowdfunding hits this month that you should check out...

PI-TOPCEED

igg.me/at/getCEED

The desktop-based sibling to the Pi-Top laptop we reviewed in the previous issue, this version comes in a lot cheaper at only $99. It's still roughly the same piece of kit, albeit without a folding clamshell, keyboard, touchpad or battery inside it. As the software is still the same, you can share user accounts between this stationary Pi-TopCEED and the portable Pi-Top, making best use of the strengths of both. At the time of writing, Pi-Top has hit its goal, so give it a look when you can and maybe pre-order a CEED for when it becomes available to everyone.

POCO MICRO COMPUTER

igg.me/at/pocopi

While there seems to have been a little confusion over why the Poco was called a supercomputer (our guess is that it's multipurpose and very small, not a thousand-core processing monster), the actual idea of the Poco is quite neat. Using the Raspberry Pi Compute Module, the creators plan to produce a tiny little handheld that can do everything the Pi can do, albeit with even greater portability than even the tiniest home-brew portable Pi. We wonder if there's any way to improve on the idea by using the Pi Zero, though.

DRAGONFRUIT

kck.st/1l5q5Ng

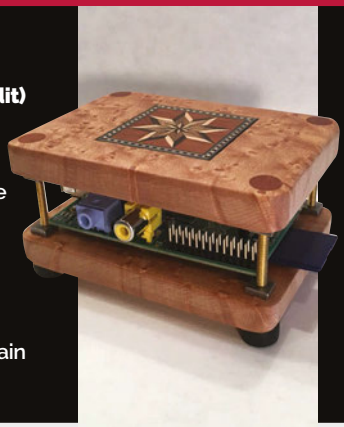
An interesting solution to the portable Raspberry Pi problem perplexing many people. Using a case with a built-in keyboard not unlike the Microsoft Surface, the DragonFruit makes use of the official touchscreen and other equipment to create a portable kit for the Raspberry Pi. It also has its own version of Raspbian called DragonFruit OS, which the maker believes better suits the kit. Did we mention it was designed by a 13-year-old? Funding should be still going as you read this, so give it a look!

BEST OF THE REST Here are some other great projects we saw this month...

A WOODEN CASE

magpi.cc/1SMbi4G (via Reddit)

A wonderfully made, classy wooden case for an equally classic Raspberry Pi (in relative computing terms) by Reddit user ChadBroChill_17. It's probably as solid as the Astro Pi flight cases, as well, and blends in a bit better with certain home decor.



TV CONVERSION

magpi.cc/1Qd4WNZ (via Reddit)

A relic of days gone by is the 'portable' TV, usually weighing a good few kilos and needing a proper plug socket to work. This conversion gets it powered by Pi without losing the retro styling of the original set. Reddit user KennethRay has a full gallery with more info.

RASPBERRY JAM EVENT CALENDAR

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

PUT YOUR EVENT ON THE MAP

Want to add your get-together? List it here:
raspberrypi.org/jam/add

1 JAMMING TIL THE END

When: Sunday 27 December
Where: Connected Community Hacker Space, Melbourne, Australia
magpi.cc/1Y9sXtL
The last of the current Raspberry Jams held at the CCHS, going out with a bang!

2 RASPBERRY JAM LEEDS

When: Wednesday 6 January
Where: Swallow Hill Community College, Leeds, UK
magpi.cc/1O6SITu
The monthly Leeds Raspberry Jam continues in the new year, allowing people to spend an evening with Pi.

3 NORTHERN IRELAND RASPBERRY JAM

When: Saturday 9 January
Where: Farset Labs, Belfast, UK
magpi.cc/1UobRCV
A monthly event for kids and adults of all ages to come together and learn a bit more about the Raspberry Pi.

4 10TH EGHAM RASPBERRY JAM - GAMIFICATION

When: Sunday 17 January
Where: Gartner UK HQ , Staines-upon-Thames, UK
magpi.cc/1SWHqmr
Gamify your Pi experience with great projects, and explore the RetroPie media centre.

5 MALVERN RASPBERRY JAM

When: Wednesday 20 January
Where: Wyche Innovation Centre, Upper Colwall, UK
magpi.cc/1Uocg8n
There are two Malvern Jams: one is for students, and the one held later in the evening is for everyone else.

6 WARWICKSHIRE RASPBERRY JAM WITH CODE CLUB

When: Thursday 21 January
Where: Nicholas Chamberlaine Technology College, Bedworth, UK
magpi.cc/1Uod2Cg
Share ideas, resources and contacts to help make coding less scary.

7 CODING EVENING FOR TEACHERS

When: Wednesday 27 January
Where: Anchor Hotel, Horsham, UK
magpi.cc/1SWHFoB
An opportunity for teachers and hobbyists to discuss the new computing curriculum and how it relates to Pi.

8 TWICKENHAM CODING EVENING

When: Thursday 28 January
Where: Stokes and Moncreiff, London, UK
magpi.cc/1Y9ulwv
The first birthday of the coding evening will have the usual Raspberry Pi stuff, along with cake!

DON'T MISS: 10TH EGHAM RASPBERRY JAM GAMIFICATION

When: Sunday 17 January **Where:** Gartner UK HQ , Staines-upon-Thames, UK

The 10th Egham Raspberry Jam is mixing it up a bit again, this time with a gamification theme. Bring along your gaming centres, your game controllers, your own games (either on screen or made physically with a Raspberry Pi) and show them what you have got. Maybe you made something amazing with your Christmas present and you want to show it off at an event that's fun for all the family and full of Pi enthusiasts. For more details, visit **magpi.cc/1SWHqmr**.

YOUR LETTERS

Pi pricing

Why are the Raspberry Pis sold in US dollars and not British pounds? I thought you were a British company, or are you secretly an American one in disguise? Not that I'm complaining – I'm American myself so it makes it easier for me.
Yours confusedly,
Phil C

Because of the economics of the way the Raspberry Pis are made, it makes more sense for the Pi to be sold in US dollars. All the parts are negotiated and bought in dollar amounts and stay fixed at that price, whereas the currency exchange rate is continuously changing from day to day. If the price of the Pis were in pounds, the price itself would fluctuate to accommodate this exchange rate. This way, the Pi is always at a fixed price and it also makes it a little more accessible around the rest of the world.

The MagPi around the world

Hi, I'm from Canada. Are there any newsagents or stores that you know of that stock *The MagPi* #40 here? If not, how can I buy the magazine? I know it's sold in the US at least, but that's a little far for me to go to pick up a copy.
Thanks,
Rebecka Robbins

We've had a lot of messages, emails, comments, tweets, and posts about this since issue 40 was released last month, asking where *The MagPi* can be bought in other countries. We didn't quite expect the reaction to the free Raspberry Pi Zero on the front cover to be as extreme as it was, even in the countries in which we currently publish!

We can tell you and everyone else that we're looking to expand into other countries in 2016 where we can. Canada, Australia, mainland Europe etc. are all on our list – the main thing stopping us is that we can't force issues of the magazine into stores! They need to place orders so copies can be sent out to them. For the moment, a subscription to the magazine is the most sure-fire way of getting it, and you can find out more on subscriptions here: **magpi.cc/Subs1**

If you really want to see *The MagPi* in your local store, make a noise with them about it. If they hear enough interest, they'll come knocking and ask to stock the magazine.

Buying a classic

Is it still possible to get a hold of a Raspberry Pi 1 Model A? I had one when it first came out but lost it, and I want to replace it if possible so I can complete my collection of Raspberry Pis!
Anthony Henderson

The original Model A is not on sale any more as it's been replaced by the Model A+, as you've probably discovered. The most immediate method is to keep an eye on eBay to see if anyone is selling theirs.

Otherwise, put a post on the Raspberry Pi Forum (**raspberrypi.org/forums**) and see if anyone has one to sell! Hopefully, you can use this to complete your collection.

Map to success

Thought I'd share a more pleasant story about issue 40. After visiting and calling loads of shops on Thursday, yesterday evening I saw your link for the map (GREAT resource!). Had a scout around and found that a small 'Premier' newsagent in Waltham Cross were selling it. That's about 30 minutes' drive from me. Called them and they had one copy, which they kindly put aside for me to collect this morning. Gave the guy a tenner and put the change in his NSPCC charity box on the counter, by way of indirect thanks. I'm now the happy owner of the mag. It's a real shame that people have been buying them up to try and sell on, but persevere and you may just get lucky.
Steve Green

Thank you, Steve! We're glad you managed to find a copy of the magazine, especially after it became so hard to find in the days after it came out. The people selling it on eBay were a bit silly (to put it lightly), but luckily by our count it was only a tiny fraction of a percent of copies that appeared there.

The map is a bit of a work in progress. We're going to try to get it updated every month as our stockists change, but as we have absolutely no way of knowing what shops have stock left, it's only a rough guide to finding copies. We're recommending that people ring ahead before making the trek so that they can avoid disappointment. For a current version of the map, you can take a look here: **magpi.cc/1lNKIOo**

FROM THE FORUM: ESSENTIAL BOOKS

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

Hi – I've just bought this month's copy of *The MagPi* and while downloading it, I came across your 'The MagPi Essentials' guide to the command line.

I was wondering if you are planning on compiling a series of these? A guide on Python, C, C++, Mathematica, and Java would be really helpful to your readers, and help us develop our coding skills.

Also, as there's a reported shortage of programmers within the IT industry, and the Raspberry Pi Foundation was created to address this, are there any plans to create a program of adult education for adults who'd like a career change?
Woll

We're planning to release more and, in fact, we've already released the *Make Games in Python* e-book. This makes use of Pygame and several tutorials to learn how to make a shoot-'em-up in Python. We have many more series that will turn into Essentials e-books over the next year, covering a wide variety of subjects, so watch this space!

As for adult education, a lot of the resources already available for teaching can be used by everyone, but perhaps that's something the magazine can look into in the future.

WRITE TO US

Have you got something you'd like to say?
Get in touch via **magpi@raspberrypi.org** or on The MagPi section of the forum at **raspberrypi.org/forums**

TAKE US **ANYWHERE**

FREE: DOWNLOAD ALL 30 ORIGINAL ISSUES

The MagPi Magazine

Available now

for smartphones & tablets

Available on the App Store

Get it on Google play

Subscribe from

£2.29 rolling subscription or **£26.99** full year subscription

Download it today – it's free!

- Get all 30 legacy issues free
- Instant downloads every month
- Fast rendering performance
- Live links & interactivity

In association with:

FUZE
Teaching kids to code

WIN A FUZE T2-SE WORTH £230!

WHAT'S YOUR EARLIEST COMPUTING MEMORY?

Tell us by 18 January for your chance to win!

THE FUZE SPECIAL EDITION

Paying tribute to the home computers of the 1980s, everything you need to get coding is bundled, including the Raspberry Pi 2 & microSD card.

How to enter:

Simply email **competition@raspberrypi.org** with your earliest computing memory (100 words max).

Terms & Conditions

Competition closes 18 January 2016. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don't like spam. Participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

A MEANINGFUL MISSION

Matt Richardson on Raspberry Pi's success: guided by a mission and supported by a community...

I recently had the incredible opportunity to tour NASA's John F. Kennedy Space Center right before the launch of two Raspberry Pis to the International Space Station for our Astro Pi competition. While there, I attended a briefing by Robert Cabana, the Center's director. Before taking that position, Mr Cabana was an astronaut, logging 38 days in space over four different Shuttle missions, including the first International Space Station assembly mission in 1998.

When he spoke to us, Mr Cabana talked about why NASA is so successful as an organisation. One of the main reasons he gave is that they have a meaningful mission and single common vision: "We reach for new heights and reveal the unknown for the benefit of humankind." According to Mr Cabana, everyone at NASA is doing their part to make a difference for humanity.

I had an amazing experience seeing all the work underway at Kennedy Space Center, but I especially liked hearing Mr Cabana talk about the importance of a meaningful mission. It really resonated with me and got me thinking about the Raspberry Pi Foundation's mission. A major part of the success of Raspberry Pi as a computer is because of Raspberry Pi's mission as a charity.

Making inexpensive computers like the Raspberry Pi is part of *what* our Foundation does, but the *why* is even more important and is what makes us so unique. We've set out to use computers to inspire and empower the next generation. We want to see all people – especially the young – learn about how computers work, learn how to code, and learn how to make things with computers. Creating an affordable computer is just one of many ways that we succeed in that mission.

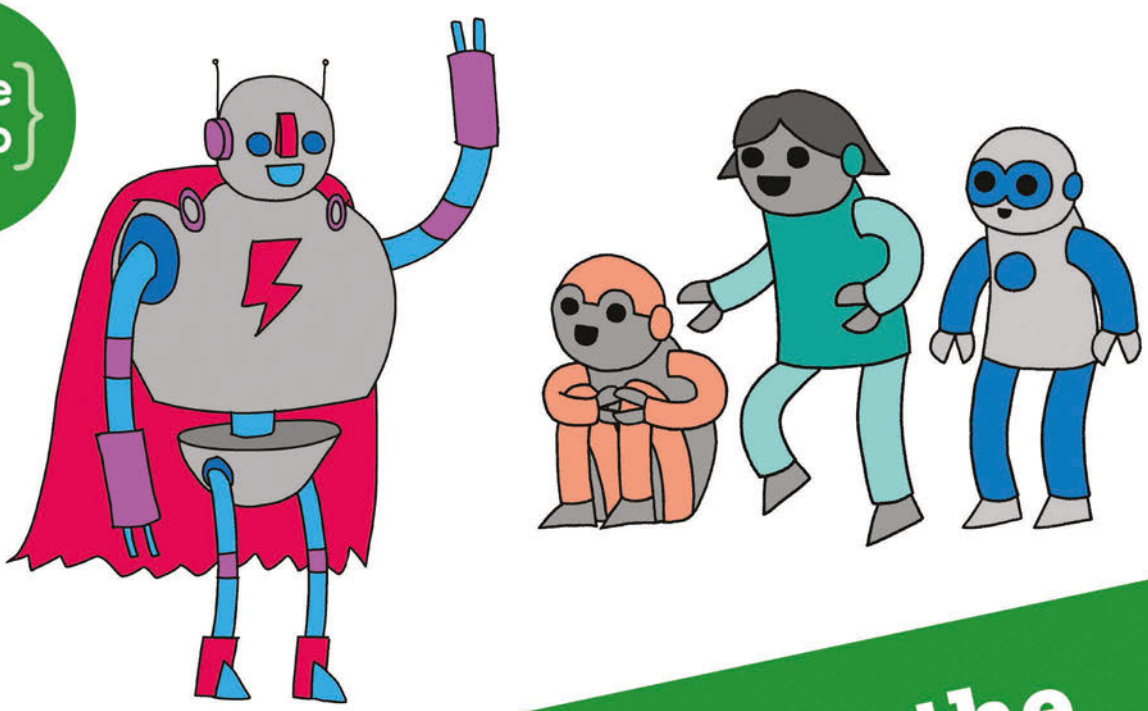
For those of us who work at Raspberry Pi, this mission is critical to us. It's our North Star, guiding us and ensuring that we're all working towards the same goal. It informs our choices, from the littlest details to the big strategic decisions.

For what I do, a meaningful mission is especially helpful. I work in San Francisco, in a time zone that's eight hours behind my colleagues at Pi Towers in Cambridge, UK. Although we have some incredible communication tools at our disposal, it's helpful for me to be able to make decisions independently after my colleagues across the pond have gone to bed. Those decisions that I make are very much guided by our mission.

It's not just the people that work for Raspberry Pi who help to advance this mission: it's also you, the members of our community. If you've bought a Raspberry Pi, you've helped to advance the Foundation's mission. If you've ordered a subscription to *The MagPi*, you've helped to advance the Foundation's mission. If you tell someone about Raspberry Pi, if you answer a question in the forum, if you share what you've made with Pi, if you get a young person excited about computers, you've helped to advance the Foundation's mission. We wouldn't have such a strong impact without such passionate support from our community members, many of whom go to great lengths to put the power of computing in the hands of others.

I'm sure that most of you already knew that when you buy a Raspberry Pi, you're not just buying a product like any other. You're endorsing and supporting our charitable mission. All of us at the Raspberry Pi Foundation deeply appreciate that support. While we may not be doing anything quite on NASA's scale, all of us, like them, are working together to make a difference for humanity.

{code club}



Can you help inspire the next generation of coders?

Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!

So to find out more, join us at **www.codeclub.org.uk**

MAKE GAMES WITH PYTHON

Get started today for **just £2.99 / $3.99**

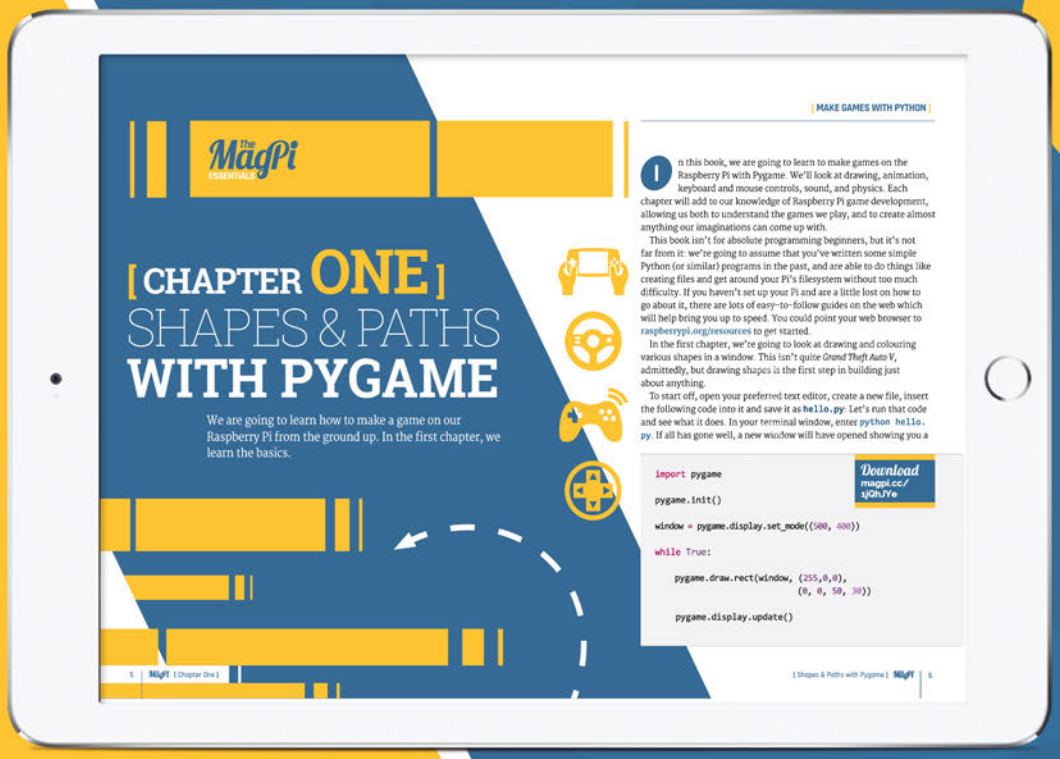
The MagPi ESSENTIALS

From the makers of the official Raspberry Pi magazine

Available on the App Store

Get it on Google play

Find it on The MagPi digital app



magpi.cc/make-games-v1