THE *Official*

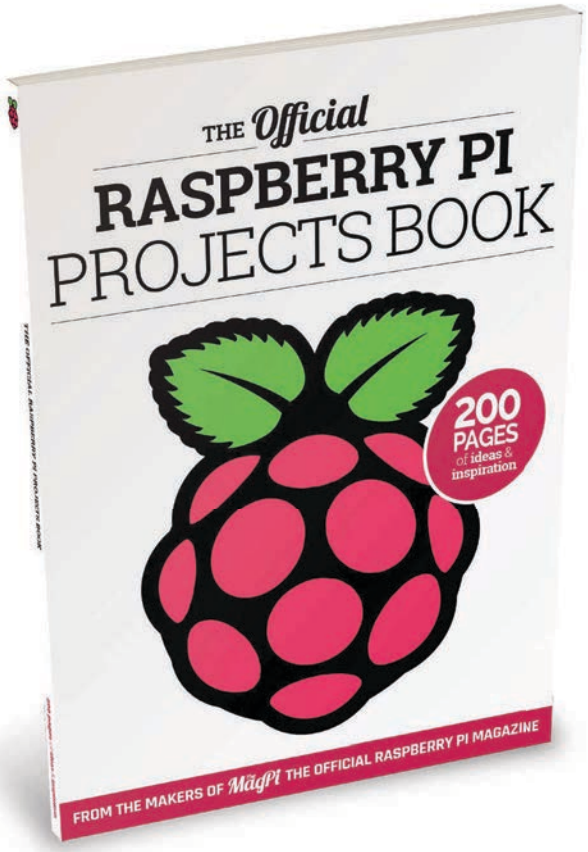
RASPBERRY PI PROJECTS BOOK

200 PAGES of **ideas** & **inspiration**

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

Raspberry Pi SWAG STORE

swag.raspberrypi.org



WELCOME

Just over seven million Raspberry Pis have been sold around the world. That makes this credit card-sized PC one of the most successful computers the UK has ever made. It's quite amazing really, that a tiny UK charity, dedicated to making computing and computer science affordable and available to people from all walks of life, should end up with such an incredible success story on their hands.

You can learn more about the Raspberry Pi Foundation's charitable and educational aims at **raspberrypi.org**, but regardless of what you want to do with your Raspberry Pi – be it for fun, for education, or as a proof of concept for a multimillion-dollar invention – The Official Raspberry Pi Project Book has something to inspire, help, and guide you on your journey. Whatever you decide to do with your Pi, I hope you have fun doing it.

Russell Barnes

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

The MagPi

Available on the App Store

Get it on Google play

EDITORIAL
Managing Editor: **Russell Barnes**
russell@raspberrypi.org +44 (0)7904 766523
Technical Editor: **David Whale**
Sub Editors: **Laura Clay, Phil King, Lorna Lynch**

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

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**

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

CONTRIBUTORS
Alex Eames, Sam Aaron, Gareth Halfacree, Lucy Hattersley, Richard Hayler & Son, Phil King, Simon Long, Martin O'Hanlon, Les Pounder, Richard Saville, Richard Smedley, Sean Tracey, Rob Zwetsloot & many more Pi-loving people!

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

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

Contents

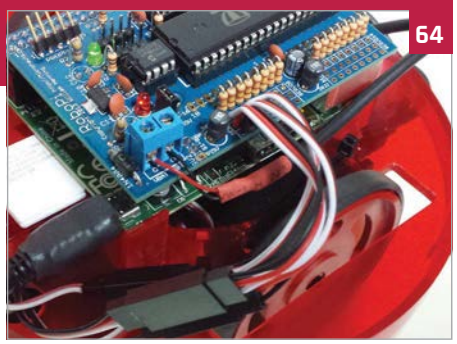
GET STARTED WITH RASPBERRY PI

Learn everything you need to know to become a pro with your Raspberry Pi

PAGE 06

Projects

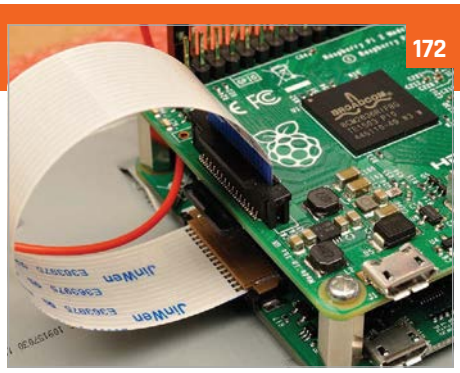
60

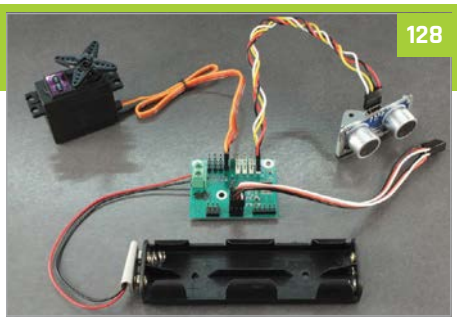


64

Tutorials

Reviews





Raspberry Pi

GET STARTED WITH RASPBERRY PI

Learn everything you need to know to become pro with your Pi

Congratulations! You've got yourself a brand new Raspberry Pi and you're ready to start using it for learning code, creating amazing projects, or just simply to power a home theatre. While the Raspberry Pi is generally very easy to use once you know how, it's that initial learning experience that can be a bit tricky for some.

Have no fear, though: we've put together the ultimate guide to getting started with your very own Raspberry Pi, from learning what all the ports and pins on your Raspberry Pi are for, to actually getting it up and running with your own monitor, mouse, and keyboard.

Whatever you want to use your Raspberry Pi for, you need to start here with the basics.

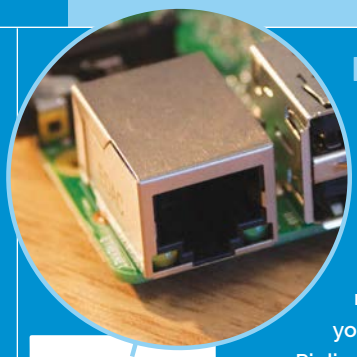
The little Model A+

As well as the 'standard' Raspberry Pi, the Raspberry Pi 2 Model B, there's also a smaller version of the Raspberry Pi that you can use. The Raspberry Pi Model A+ is a cut-down version of the original Raspberry Pi, with a little less power at its disposal and fewer connections on it. It's favoured by people who like to make big physical projects, due to its diminutive size and low power requirements. It also has only one USB port and no Ethernet port, making it slightly less useful to some.

THE INGREDIENTS FOR A RASPBERRY PI 2

USB ports

The Raspberry Pi 2 has four USB ports, allowing you connect it to keyboards, mice, WiFi dongles, and USB sticks containing all your files. Since the ports don't provide much power, if you want to add a USB hub to the Pi you'll need to find one that comes with an external power supply.

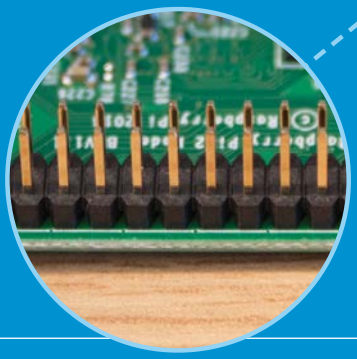


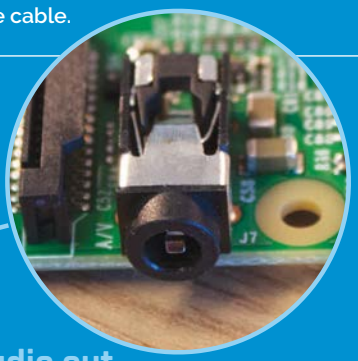
Ethernet port

The traditional way to connect to the internet is via a wire called an Ethernet cable. You'll find a few similar ports like this at the rear of your router at home that will let you connect the Raspberry Pi directly into it. This method is easier to set up than WiFi and may provide faster internet, but you're then limited by the length of the cable.

GPIO header

This comprises the general-purpose input/output (GPIO) pins. They're a set of connections that have various functions, but their main one is to allow you to connect to the Raspberry Pi with an electronic circuit. You can then program the Pi to control the circuit and do some amazing things with it.



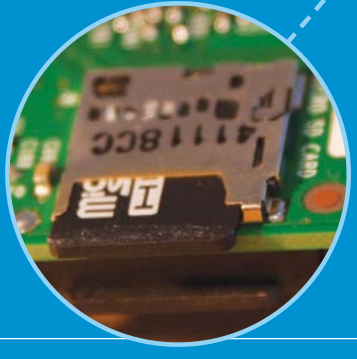


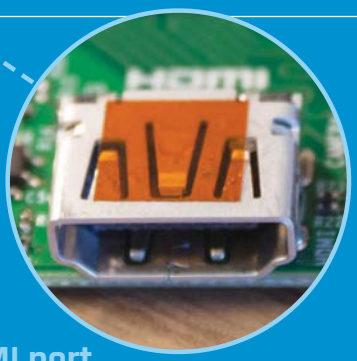
Audio out

This looks like a headphone socket because that's exactly what it is. A 3.5mm jack to be precise, this allows you to connect the Pi to computer speakers, or you could even plug in your favourite headphones and have a Raspberry jam.

MicroSD card slot

A little SD card is used as the Raspberry Pi's hard drive. This is where the operating system will live once you've put it on there. Most computers won't be able to directly connect to a microSD card, but you can get an adaptor that plugs into normal SD card slots.



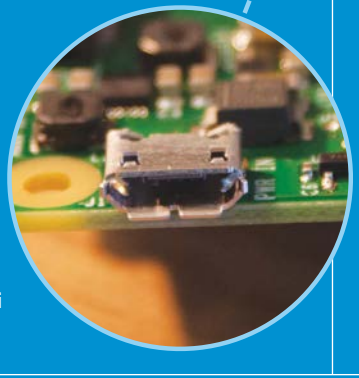


HDMI port

This is an HDMI port, the kind you'll find on the back of most modern TVs and computer monitors. Use a standard HDMI cable to connect your Raspberry Pi to your chosen screen, to see (and hear) whatever it's doing. You'll definitely need to plug it in to set up the Pi.

Power

This is the kind of small charging port you might find in your smartphone. This micro-USB port means you can power the Pi with the right kind of mobile phone charger or directly from your PC – however, it's best to use the official Raspberry Pi power supply to make sure the Pi is getting enough power.



CONFIGURATION TOOL TABS

The four tabs in Raspberry Pi Configuration allow you to set up various aspects of your Pi...

SYSTEM

Here, you can set whether your Pi boots to the desktop or the command line (CLI), set up autologin, and change password. There is also an overscan setting which is useful for getting rid of the black border around your desktop if there is one.

INTERFACES

Under this tab, you can enable or disable various interfaces on the Raspberry Pi, including the camera and other connectors.

PERFORMANCE

Here, you can set up overclocking and GPU memory allocation to optimise performance for the particular task you want to perform (e.g. HD video playback).

LOCALISATION

This tab enables you to set up the correct language, time zone, and keyboard layout.

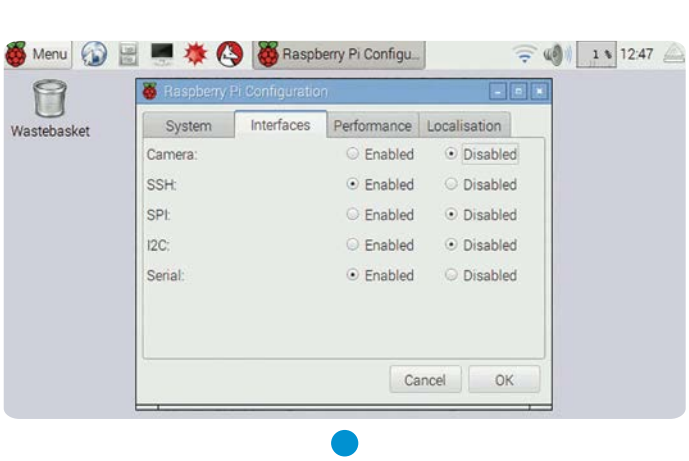
SOFTWARE CONFIGURATION TOOL

Once Raspbian has booted and is showing the desktop, click the Menu button at the top left of the taskbar to open the main menu. From there, select the Preferences option and then Raspberry Pi Configuration from the submenu.

This opens a tool which can be used to configure various options on your Raspberry Pi. Note that you don't need to change most of these options at this point, but one that is important is the 'Expand Filesystem' option.

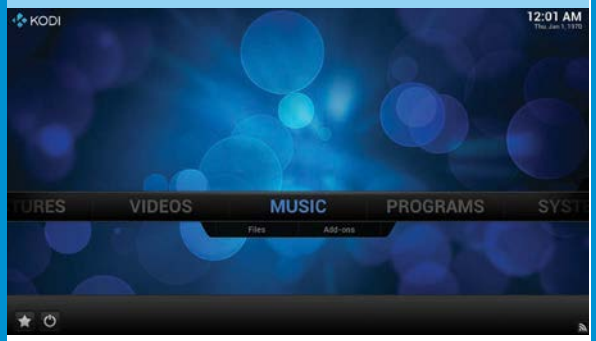
Since Raspbian doesn't take up much space to begin with, you need to tell it to use your entire SD card if you want to make use of all the free space on it to store files. So, click the Expand Filesystem button under the default System tab of the Raspberry Pi Configuration menu, and then reboot your Pi when prompted – choose Shutdown from the Menu, then select Reboot and hit OK.

Read more about the configuration tool's other options at the top-right of this page.



Various options are available under the four tabs, including enabling interfaces

Setting up a media centre

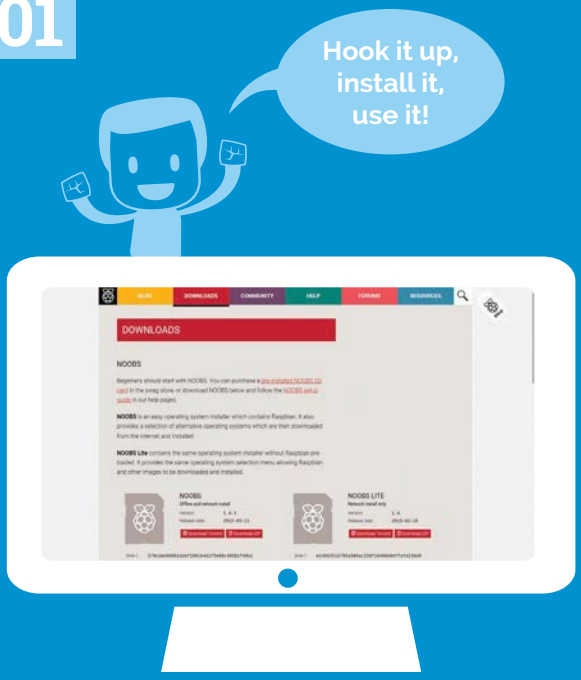


We touched on OpenELEC on the previous page, and how it can be used to make a home theatre PC. This is a PC that hooks up to your TV and powers all your media needs. You can find OpenELEC on NOOBS, and installing it is very similar to Raspbian: you select OpenELEC and hit Install!

OpenELEC runs on Kodi – software that lets you connect to your other computers over the network, as well as some online web services such as YouTube. It can play just about anything, but you need to show it where the files are. When adding folder locations to either Video or Music, you can find any shared folders via the SMB option, or you can simply plug in a USB hard drive full of videos and music and play them straight from the menus.

SET UP YOUR RASPBERRY PI

01



DOWNLOAD NOOBS

The Raspberry Pi comes with many operating systems you can use, which you could manually install yourself if you wish. There's a much easier way to install these OSes, though, and that's via the New Out Of Box Software, or NOOBS. It holds all the latest versions of the Raspberry Pi operating systems and you can grab it from the download page at: **raspberrypi.org/downloads**.

We prefer to use the full version of NOOBS, as it comes with Raspbian already downloaded, making the process slightly faster than with NOOBS Lite. However, all the other operating systems will be downloaded as they install, on both versions of NOOBS.

02

INSTALL SD CARD

While that's downloading, you'll need to get your SD card ready to work on your Raspberry Pi. This will require you to format it, so if there are any files on the card you want to keep, now's the time to take them off. You'll need to install the SD Card Formatter 4.0 tool to prepare the card, which can be downloaded from the SD Association: **bit.ly/1alC3Wp**.

Once you've formatted your SD card, extract the files from the NOOBS zip folder and put them all on the card. That's it: NOOBS is installed to your SD card and ready to use!

03

CONNECT THE CABLES

Take the SD card adaptor out, retrieve the microSD card, and slot it into the Raspberry Pi; this is very important, as the Raspberry Pi won't be able to turn on properly otherwise. To start with, you'll need to plug in an HDMI cable between the Raspberry Pi and your screen, an Ethernet cable for your router (or a USB WiFi dongle), along with a mouse and keyboard. Finally, when everything you need is plugged in, you can attach the power cable to the Raspberry Pi.

04

INSTALL RASPBIAN

The Raspberry Pi will turn on and display some text on the screen – you can ignore this until it gets to a menu which lists all the available operating systems. It allows you to select multiple OSes at once, but right now we just want to use the one that's called Raspbian. This is the main operating system for the Raspberry Pi, with all the official apps, software, and learning documents. Upon selecting Raspbian, click on Install and it will begin the Raspbian installation process, which may take a little while to complete.

Alternative operating systems

PIDORA

A bit like Raspbian, but based on a different core operating system. This is something people a bit techy can use for a slightly different Raspberry Pi.

ARCH

A very basic operating system that works entirely from a command line, no mouse and keyboard required. You'll really need to know your computers to start with this.

OPENELEC

An OS to turn the Raspberry Pi into a home theatre PC, complete with the Kodi software that plays music and videos as well as web video.

RISC OS

A throwback to your school days, RISC OS is what used to be on old Acorn computers. The Raspberry Pi is in many ways derived from them.

A TOUR OF RASPBIAN

Top right icons

WIRELESS INTERNET
This shows the state of your internet connection; solid blue lines means it's connected!

VOLUME
Control the volume of your Raspberry Pi from here. This will work whether you're getting sound from the HDMI port or via headphones.

PROCESSING POWER
The Raspberry Pi, while small, has a lot of processing power. This tells you how much is in use, so if it's running a little slow and this gauge is at 100%, you'll know why.

TIME
Set the time to be anywhere in the world! The Raspberry Pi relies on the internet to tell it what time it is.

Access the various menus, programs and settings for Raspbian; almost everything you do will start here

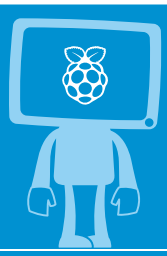
These icons let you quickly launch certain programs, such as the browser, the terminal, and the Mathematica programs for hardcore maths and graphing

The various open windows are listed on here; much like in other operating systems, you can click between them when you need to change location

This area offers quick access to tools such as WiFi to ensure your Raspberry Pi is running just fine

Programs and apps are categorised to make them easier to find – if you can't find the app you're looking for, you might need to go through all of them

Raspbian should feel familiar to most PC users



Windows here work just like any other kind of operating system: you can drag them, change their shape, and close them using your mouse

Raspbian looks and works very similarly to the kind of operating systems you're used to, except that the menu is now at top of the screen! Raspbian is based on something called Debian, which is a version of Linux, a highly customisable operating system that can be tweaked enough to run on the Raspberry Pi. It works extremely well, and even on the tiny Raspberry Pi it will almost feel like using a normal computer!

There are a few important icons on the top panel that you should make sure you're aware of.

The Menu is where all the programs and apps live; just like in any other operating system, you can access them from here and they'll open up in a new window. You'll find all the settings in here as well, in case you want to tweak the way Raspbian looks and works.

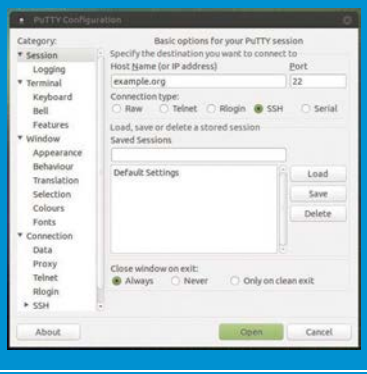
> "The cabinet represents the file system of Raspbian"

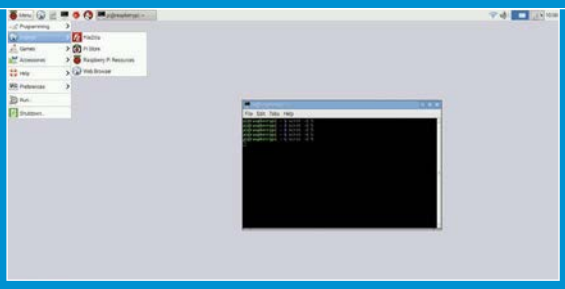
Next to the menu is a row of quick-start icons to quickly launch software. The globe picture is the Raspberry Pi browser, your access point to the internet. The cabinet represents the file system of Raspbian, allowing you to browse any documents or images you have saved onto your Raspberry Pi. The picture of the screen is the terminal, and it's what you use to run commands via text on the system,

REMOTELY CONNECT TO YOUR RASPBERRY PI

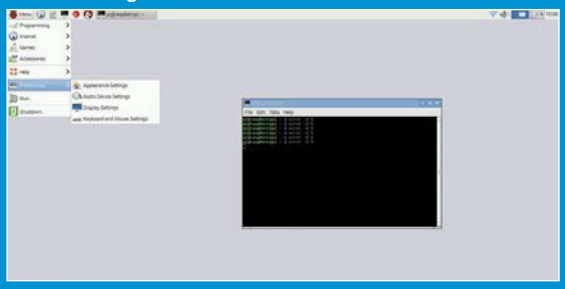
The Raspberry Pi is extremely flexible due to its design, and because of this it will let you connect to it from another computer via a system called SSH (Secure Shell). All you need to connect to it from another PC is the IP address of the Raspberry Pi and a way to access SSH. For the latter you can get PuTTY, a piece of software specifically made to let you easily connect to another system via SSH.

To find out the IP address of the Raspberry Pi, you simply open up a terminal window and type in **`ifconfig`**. It will list all the details of your network connections, including the IP address; this is listed as 'inetaddress'and may look something like 192.168.0.20. To connect to it, you need to use 'pi@ 192.168.0.20', give the password of 'raspberry', and then you can control your Pi from the command line.

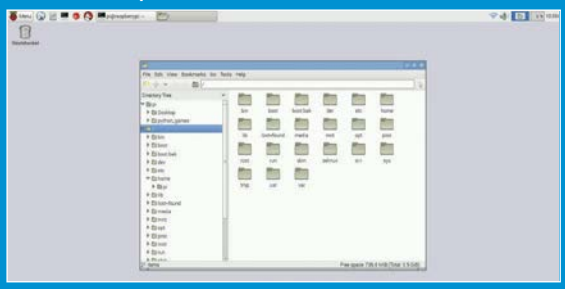




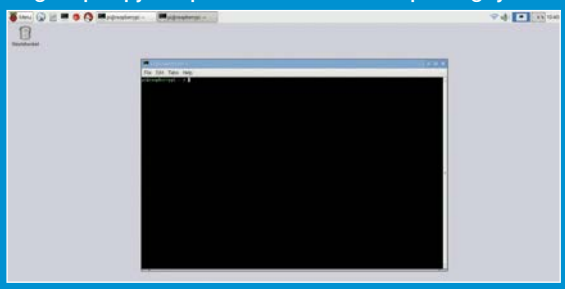
Look through the menu categories to find the software you want to use right now



Settings and preferences allow you to modify the look and feel of Raspbian



Browse files and folders on the Raspberry Pi, and use the same drag, drop, copy, and paste functions of other operating systems



Access the command line and control the Raspberry Pi with text commands. Almost like a hacker… almost

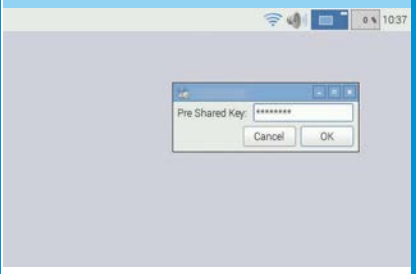
something you might have to do for more advanced projects.

The file system of Raspbian is a little different from something like Windows. Instead of having a C:/ drive with a My Documents folder and programs kept in Program Files, everything is spread out in multiple folders on the root, or top of, the file system. What could be considered My Documents is a folder called Pi in the Home folder on the root. You may see it referred to as the 'home directory', and that's why.

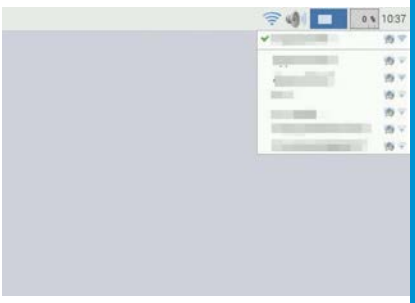
To turn off the Raspberry Pi when you're done for the day, you go to the Menu and press Shutdown. This makes sure everything is safely disconnected and turned off before the entire computer turns itself off. As the Raspberry Pi doesn't have a power switch, you'll have to manually unplug the Pi to fully turn it off standby, and you'll have to unplug it and plug it back in to turn it on again.

Raspbian is quite a simple interface, then, very similar to how you may have used computers in the past. You're now ready to start learning how to code and create your own excellent projects!

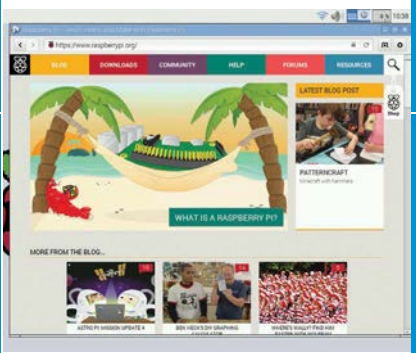
Connect to WiFi



In the right corner of the top panel, you'll find access to WiFi. If you have a compatible WiFi dongle for the Pi, clicking on this will drop down a menu that shows you all the available wireless networks you can connect to.



Click on the wireless network you want to use and you'll be shown a box that lets you put in your password. It will actually display what you're typing in, which should make it easier to type, but make sure your neighbour isn't peering through the window trying to copy it down!



The Raspberry Pi should now be connected to the internet! It will automatically set all its options from your router that will let it talk online. Open the browser and go to your favourite website to make sure it works. The Pi will remember your wireless details and connect whenever it's on.

INSTALLING AND UPDATING SOFTWARE

Expand and maintain Raspbian for a long-lasting Raspberry Pi experience

INSTALL NEW SOFTWARE

You're not limited to the software that's just on Raspbian when you install it. Raspbian has access to thousands of different programs that you can download and install, just as you would with smartphone apps. Raspbian doesn't have an app store, though, so you need to install them using the terminal.

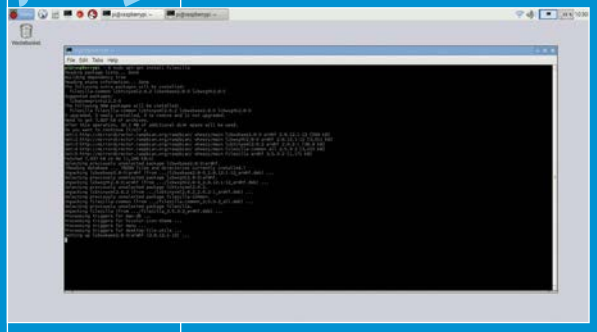
This does require you to already know what the software is called to install it in the terminal, since you can't browse the software in the same way as on your phone. If you're not sure of the exact name of the software you want, you may have to Google it. Otherwise, if you're looking for a specific kind of app, you can use a command like the following to search for it:

```
$ apt-cache search ftp
```

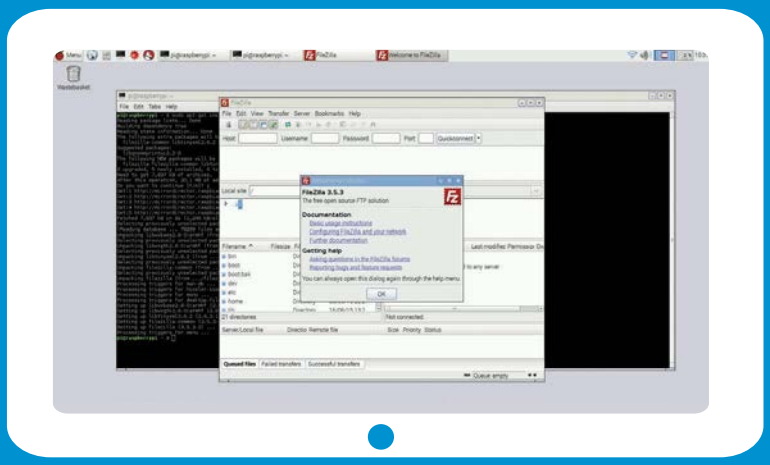
It will return a list of packages and their details. The package name is how you install the software; in our case, FileZilla comes back as an FTP client. Its package name is 'filezilla'. So, to install it, we use:

```
sudo apt-get install
filezilla
```

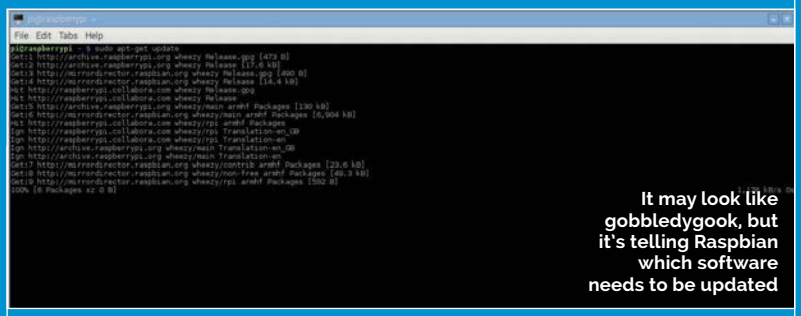
This will download the package and any other necessary software it needs to run, and install it to Raspbian.



Raspbian asks you to agree to an installation with a simple press of Y



When it's installed, you can immediately start using your new software, no restart required



It may look like gobbledygook, but it's telling Raspbian which software needs to be updated

UPDATE YOUR SOFTWARE AND OS

The software on Raspbian will be periodically updated online, bringing with it bug fixes and security updates. Those don't automatically sync with the Raspberry Pi, though, and you should regularly check to see if there are any updates for your system. This is handled entirely in the terminal again, much like the software installation.

The update process consists of two parts: first you need to update the repositories; this is the list of available software and their versions kept on your system. You do that by first entering the command:

```
sudo apt-get update
```

This will check online to see the state of the software repositories and report back to the Raspberry Pi, saving any changes. It will then determine what software can be and should be updated, but you then need to tell it to perform the update with this command:

```
sudo apt-get upgrade
```

Every now and then, there may be a major update to the Raspbian operating system, bringing with it big changes like a new interface or browser, etc. It's very rare, but when it happens, you can perform the upgrade with:

```
sudo apt dist-upgrade
```

USE THE GPIO

Make your first small project with a bit of code and the GPIO pins

The GPIO port is one of the most powerful tools at the Raspberry Pi's disposal, allowing you to connect directly to an electronic circuit to control it. In such a system, the Pi is referred to as a microcontroller. This is what makes the Raspberry Pi great for big projects, as you can use it to program a machine or circuit, and even have it connect to the internet via the other Raspberry Pi functions so that it can control contraptions with web data.

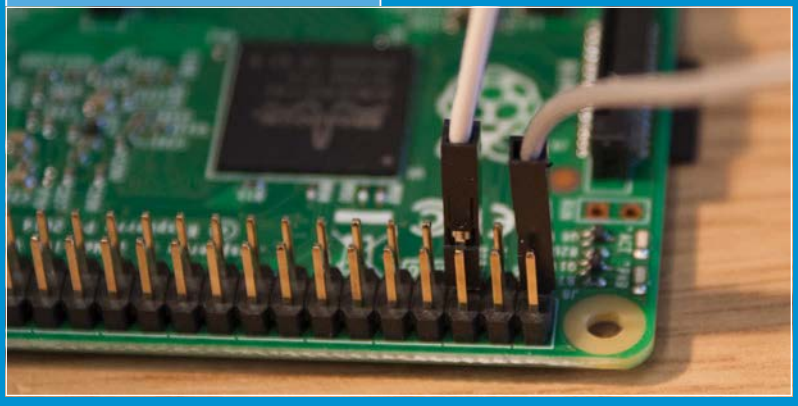
Each of the GPIO pins can do something different and very specific. At the basic core, though, you can have them provide power consistently to part of a circuit, program a power switch to one of the pins, and even have it sense a change over the pins (thanks to resistance). These three basic functions allow you to do a lot, and can be programmed with Python.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)

GPIO.output(7,True)
time.sleep(1)
GPIO.output(7,False)
time.sleep(1)
GPIO.output(7,True)
time.sleep(1)
GPIO.output(7,False)

print "Done"

GPIO.cleanup()
```

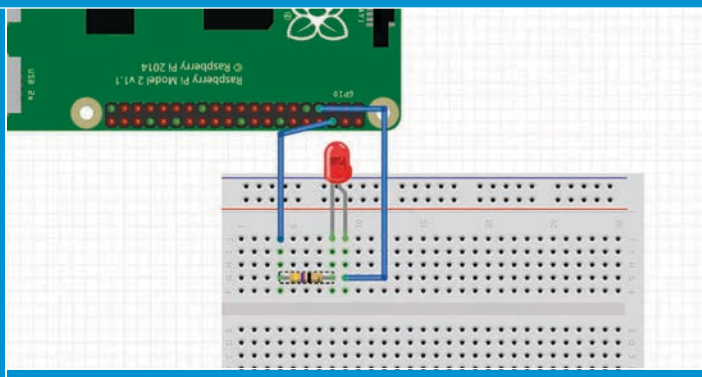
We're going to wire up an LED bulb to be programmable from the Raspberry Pi, to turn it on and off again a few times. For this, you will need a breadboard prototyping circuit board, an LED, a 50-ohm resistor, and some wires. Refer to our Fritzing diagram on the right, to see how it's wired up; the negative end of the LED goes to a ground pin on the Raspberry Pi (which is where the flow of electricity ends), and a programmable pin goes through the 50-ohm resistor to provide power to the LED when it's turned on.

Open up IDLE, the Python programming software, and create a New file. Save it as **led.py**, and input the code from the code listing. What the code does is first tell Python to use the GPIO module so we can connect to the GPIO pins, by **importing** the module. We then import the time module so we can create a delay between commands. We then tell the code to treat the GPIO pins as the number they are on the board, and to turn the seventh pin into an **output**. We alternate between **True** and **False** so that it turns the pin on and off. Once it's cycled a few times, it will **print** the message 'Done' into IDLE, and finally turn off the GPIO pins.

You can do a lot more with GPIO if you want to, and this is a good way to start before moving on to bigger projects.



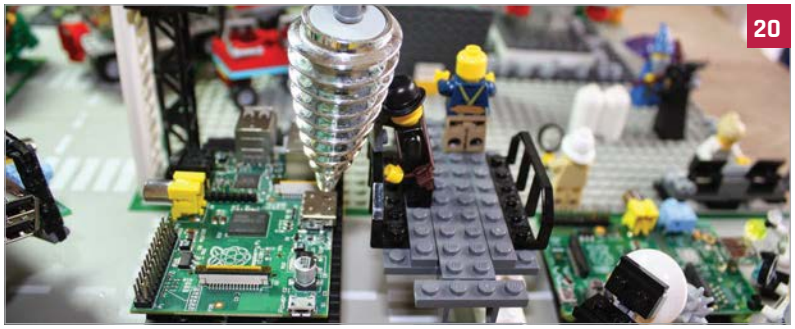
You can connect directly to the Raspberry Pi without needing any special slots over the pins

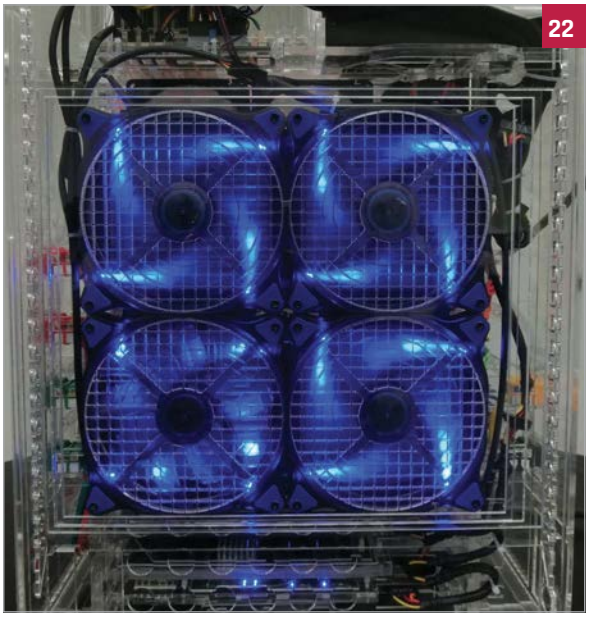


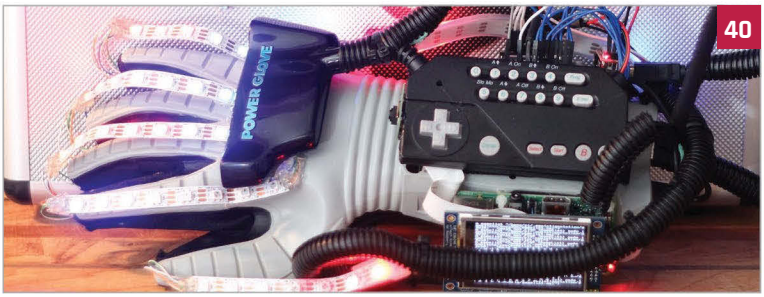
Wire the circuit up just like this

PROJECTS SHOWCASE

There's no better way to be inspired into action than to see what the rest of the Raspberry Pi community is making (and how)...







Projects

MAKERS: THE NEXT GENERATION

The Raspberry Pi is inspiring a new generation to learn how to hack and make amazing projects. We chat to four young makers about their impressive creations and achievements, and why they do them with the Raspberry Pi…

Zach

Name: Zachary Igielman
Age: 15
Location: London
Studying: 12 GCSEs
Twitter: @ZacharyIgielman

After teaching himself to code in Visual Basic at just 11 years old, Zach moved on to Objective-C and released several apps on the iOS App Store. Since discovering the Pi, he's learnt to code in Python and has built his own autonomous robots, enhancing them with a variety of sensors. He also helped 4tronix to develop the Pi2Go. A regular Raspberry Jam attendee, he has run many different workshops. Remarkably, Zach conducted a successful crowdfunding campaign on Indiegogo in early 2015 for his PiPiano musical add-on board (**bit.ly/1wsBmci**), raising nearly twice the original goal.

Tell us about some of the Pi projects you've worked on.

Along with my robots, I've done some cool work with the Camera Module: I programmed my Pi to do time-lapses. I made a tutorial for using an accelerometer with the Pi. I also ran a sensors workshop teaching line and distance sensing with my own materials.

Elsewhere I've been working on a self-balancing robot, which led to giving a talk on PID control theory for robotics at a Raspberry Jam… I have tested all sorts of Pi add-on boards, from LED to analogue, leading me to build my own.

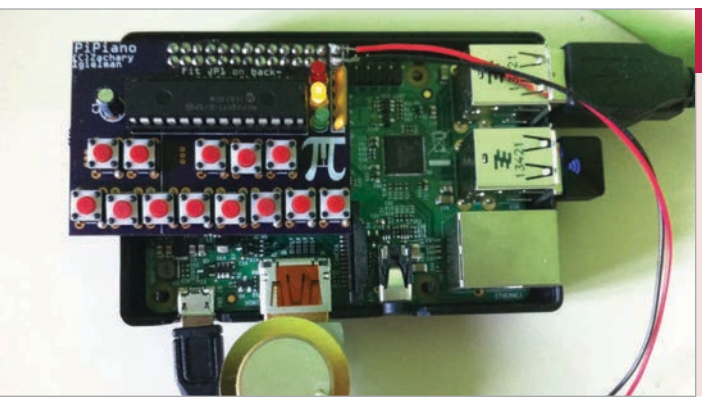
How did the idea for the PiPiano come about?

I wanted to create a piano with my Pi, but adding switches to a real piano would be a wiring nightmare… My solution was to create a simple, piano-style add-on for the Raspberry Pi, which includes buttons in a piano octave formation and a piezo transducer for sound output. After trying it on a breadboard, and being swamped with bundles of spaghetti wiring, I opted to lay out the idea on a PCB (printed circuit board).

PiPiano teaches programming, soldering and electronics, through reading the buttons, making traffic lights with the LEDs, playing a scale on the buzzer, and finally making a PiPiano. It uses a special chip so all 17 components connect to three [GPIO] pins.

What is it you love about the Raspberry Pi?

It's really cheap; I can afford a few of my own and I'm not always worrying about breaking an expensive computer. It's portable, easy to ferry from event to event. It's got an amazing community, which enables me to meet cool people, learn lots and have amazing opportunities. It has lots of documentation and a great forum for learning anything with it. It is very programmable [and] it can plug into electronics, unlike most computers.



BIG PROJECT: PIPIANO

Plugging directly into the top of a Pi, this musical add-on board features 13 buttons in a piano key formation, a piezo transducer for sound output, and three LEDs. Designed to be educational, it comes with documentation which takes you from the basics of soldering the board and setting up the software, to programming a fully working piano at the end. PiPiano comes soldered (ready-made) or as a kit, and with either a standard or stacking header. When not used as a piano, it's also a handy controller with an ample supply of buttons. It is also the basis for Pimoroni's Piano HAT. Learn more at **pipiano.com**.

Amy

Name: Amy Mather
Age: 16
Location: Manchester
Studying: 9 GCSEs (already has an A* in Computing)
Twitter: @minigirlgeek

Following an invite from Jimmy Wales, Amy presented a keynote speech at the Campus Party EU 2013

Already into electronics at the age of 12, Amy integrated an Arduino kit into a model volcano for a school homework project, which she was then asked to demonstrate at Manchester's first Mini Maker Faire. After getting hold of a Pi, she learnt Python and created her own version of Conway's Game of Life, even outputting the display to an LED matrix. Most notably, Amy teaches both adults and children to code and works closely with the STEM network to inspire other young people to get involved in computer science. This has led to her giving keynote speeches at many prestigious technology events. At ICT 2013 in Lithuania she received an award as the European Digital Girl of the Year.

How did you get started with programming?

I first got interested in coding when I was about 12, following a Manchester Girl Geeks workshop that I attended that was an introduction to JavaScript using Codecademy. I thought Codecademy was an amazing platform for learning how to code and I continued with their courses (you can find out more at **codecademy.com**).

What's so great about the Raspberry Pi?

It doesn't matter if you accidentally blow bits up on it – you can get another one! Or if the SD card corrupts, it's not the end of the world: you can reformat it. The Pi allows you to make all kinds of remarkable projects and there are so many awesome add-on boards. Also, the community's really welcoming and friendly, so they're open to any of your questions. I think the whole environment is just amazing.

Do you have any new Pi-based projects planned?

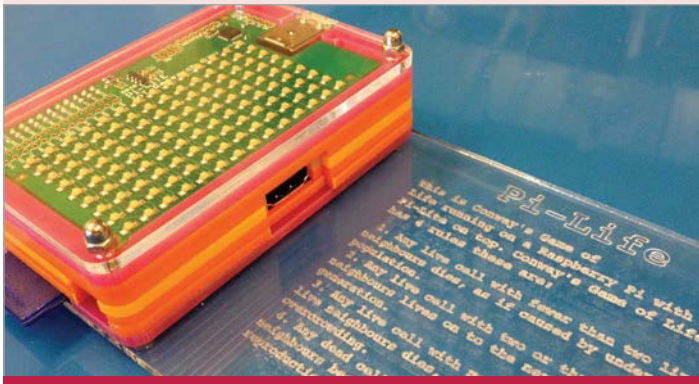
My school has just asked me if I have any ideas for ways that we can link the coding club and the STEM club. So I'm thinking about how to help them through the use of Raspberry Pi-based projects. I've helped out with teaching coding workshops and I'm currently leading a series of soft electronics workshops as the volunteering section of my Silver Duke of Edinburgh Award.

And you also make speeches at numerous events?

Yes, I speak at quite a lot of conferences, about how we can get more kids involved in STEM (Science, Technology, Engineering, and Mathematics) and why it's really important that we do so.

What advice would you give to other young coders?

Just get involved, find out where all the local events are, and get involved with the community; you'll definitely learn a lot more from talking with other people who are interested in similar things, rather than just sitting alone at home and doing it by yourself.



BIG PROJECT: PI-LIFE

Amy first came to the attention of the international Raspberry Pi community after giving an impressive presentation of her Python version of Conway's Game of Life – a zero-player game simulating cellular replication – at the 2013 Manchester Raspberry Jamboree (**raspberrypi.org/amys-game-of-life**). In it, she enthused about her love of coding and detailed how she developed various implementations of Life, including one with the Pi outputting the resulting patterns to an 8×8 LED matrix via a connected Arduino Mega. Since then, she's created a more compact version using a Pi-Lite LED add-on board.

Right Lauren created a portable Pi-powered system to sync the lights in Charles Peachock's juggling clubs to a music track

Lauren

Name: Lauren Egts
Age: 16
Location: Stow, Ohio
Studying: High School (Hathaway Brown)
Twitter: @laurenegts

When she was just nine, Lauren's father taught her how to write some Bash scripts. Using Scratch on the Pi, she later created The Great Guinea Pig Escape game and demonstrated it at a local Maker Faire. A long-time member of Akron Linux User Group, she has presented talks on GlusterFS and teaching kids to code on the Pi. She's an NCWIT Aspirations in Computing Ohio Affiliate award winner for 2014 and 2015, and is also an intern at the NASA Glenn Research Center G-VIS Lab. Recent projects include creating a Pi video wall (at NASA) and designing a portable LED lights system for professional juggler Charles Peachock.

What do you especially like about the Pi?

One thing that I really love is its versatility. I've used it in a variety of projects, and seen it used in even more! The Pi is so small that it can be used practically anywhere, which means it can be used in [so many different] projects.

What was it like being an intern at NASA? How did it happen?

It happened at the Cleveland Mini Maker Faire. I had my booth where I was presenting on Scratch and the Raspberry Pi. I was helping a friend take her booth out to her car, and my dad shows up with some other guy who I later found out was Herb Schilling, now my mentor at NASA. Turns out Herb had gone to my booth while I was away, and my dad had told him all about me. When Herb and I met, he was so impressed with what I had done with the Pi that he invited me to shadow him at NASA. After my shadow day, Herb invited me to come back for a few weeks over the summer! We figured out some dates, and that's how my internship happened!

Are you planning to do any more Pi projects when you return to NASA?

It depends on what projects I am assigned to but I would absolutely like to finish my work on the Pi video wall... Herb is very interested in showcasing the power of low-cost computing devices like the Pi. Finishing my work will require taking care of a few bugs in code, as well as fixing the aforementioned hardware issues. After the video wall project is done, hopefully it will be displayed outside the G-VIS Lab, playing a video that explains what the lab does when people walk by.

BIG PROJECT: PI VIDEO WALL

During her internship at NASA, Lauren – working with fellow intern Nick Patterson – did a proof-of-concept project to create a Pi-powered video wall (**go.nasa.gov/1DYEa2v**). Since they only had access to what was in the NASA G-CVIS Lab at the time, different-sized monitors were used, but the end result was still impressive. The setup involved connecting four Pis to a master computer via a router. The PiWall software package (**piwall.co.uk**) was used to split up the video display into four tiles, one for each monitor. Lauren hopes to improve the setup when she returns to NASA.

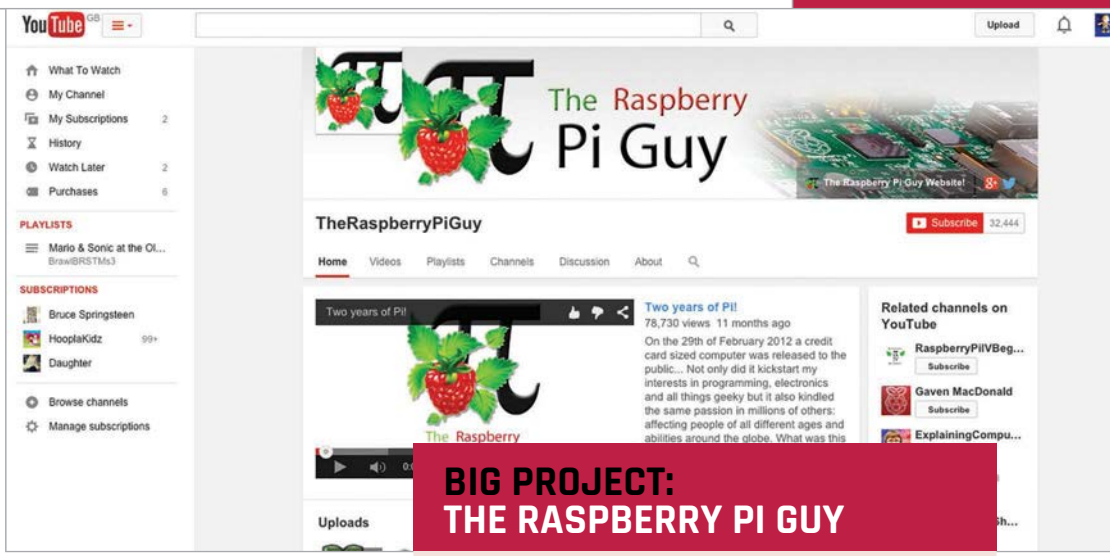
Matt

Name: Matt Timmons-Brown
Age: 16
Location: Bartlow, near Cambridge
Studying: 11 GCSEs
Twitter: @RaspberryPiGuy1

Better known as The Raspberry Pi Guy, Matt runs a YouTube channel dedicated to Raspberry Pi video tutorials. Amazingly, he only started coding and making around three years ago, upon discovering the Pi Foundation's credit card-sized PC. So far he's created projects of varying complexity, including a Pi-controlled model railway, but his real passion is robotics – in particular, making two-wheeled robots and pushing their abilities to the max. His latest project involves building an accessible robotics platform, with a complete set of learning materials, using 4tronix's Pi2Go-Lite. To this end, he has filmed a YouTube series called 'Raspberry Pi Robots'.

How did you first get involved with making?

I am the quintessential Raspberry Pi product: a schoolboy who now has a love of computer science because of the Pi... In the summer of 2012 I managed to get my hands on my first Pi and ever since then I have been in love with programming, making (robots!) and computer science. The Pi opened my eyes to computing and for that I am incredibly grateful!



BIG PROJECT: THE RASPBERRY PI GUY

Matt's YouTube channel (**youtube.com/user/TheRaspberryPiGuy**) has been running for over two years now and has proven immensely popular, amassing over 42,000 subscribers. "Dedicated to teaching the masses how to make the most of their Raspberry Pi computer", it provides a plethora of step-by-step video tutorials. These range from basic setup to attaching various add-ons and creating numerous projects – including, of course, robotics. The latest addition is the Raspberry Pi Robots series, which Matt hopes "will engage people in computer science through the most exciting medium: world-conquering robots."

Why did you decide to set up your own YouTube channel?

I am a very recent convert to the ways of Pi. As a result, I originally found computing a fairly hard subject to get into; there is all of this foreign jargon and sometimes it can be incredibly confusing... In September 2012 I realised that I had built up a considerable amount of knowledge on the subject and I thought it would be a great side project to teach people some of the stuff I'd had so much fun learning. I turned to YouTube as a way of doing this because I found the most easy way to learn something is by watching someone go through something step by step... On 1 September 2012, The Raspberry Pi Guy was born and I have been publishing videos ever since; I am just about to hit the 2 million view barrier on YouTube, something I never imagined!

What's so great about the Pi?

There is no other product out that there has the same ethos: to teach people about computing by introducing them to a [new] experience. After all, how many single-board computers has the average person seen? How many terminals have they programmed in? Scratch that, how many people have actually programmed?! The Pi is a gateway to the world of computers and [has] introduced me to a lifetime interest... It has inspired millions and continues to do so.

Get making!

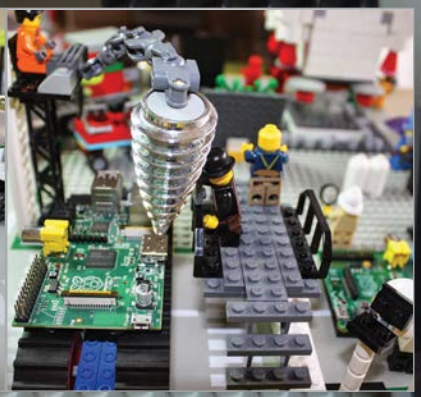
Our young experts offer a lot of good advice for how to get started with coding and making...

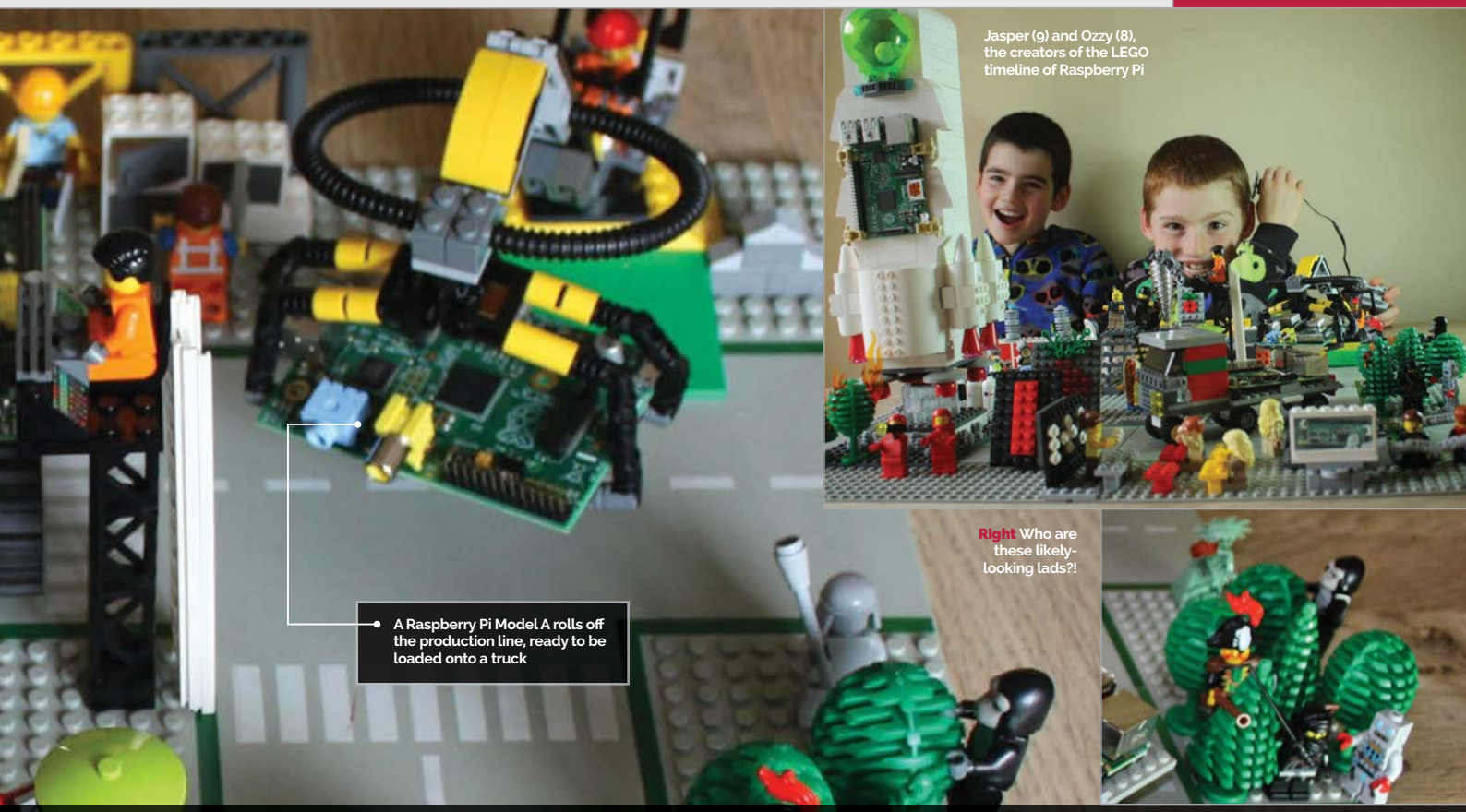
- There are lots of free online resources, such as Codecademy, to help you learn to code. Just Google 'coding courses'.
- Look out for local events and get involved – it's much more fun than trying to do it alone at home!
- Events such as Maker Faires and Raspberry Jams can give you ideas to try, and a place to exhibit your projects and make contacts.
- Join a robotics team if you're interested in making robots and entering them into competitions.
- Even if something seems hard to start with, keep trying: if you set your mind to it you'll achieve it. And you're never too young to start!
- However, remember to always have fun with what you're doing. If you don't like something, don't waste your time with it.

LEGO-LUTION OF PI

Mad scientists working on the original Model B Rev 1. A surprisingly accurate account

The Raspberry Pi 2 takes centre stage on this elaborate rocket. A brilliant nod to Astro Pi





A Raspberry Pi Model A rolls off the production line, ready to be loaded onto a truck

Jasper (9) and Ozzy (8), the creators of the LEGO timeline of Raspberry Pi

Right Who are these likely-looking lads?!

LEGO-LUTION OF PI

Ozzy, **Jasper** and **Richard Hayler** celebrate their collection of Raspberry Pis the only way they know how

When Richard Hayler isn't working for the Foreign Office, he's a Raspberry Pi enthusiast, CoderDojo mentor, and Code Club volunteer. "Pretty much everything else revolves around my sons, who love getting involved with all things Pi," he says.

Besides test-driving his educational material, Richard's sons Ozzy (aged 8) and Jasper (9) are often to be found hacking and making with the Pi. Their latest creation is this rather marvellous Lego scene designed to celebrate the evolution of everyone's favourite credit-card-sized PC.

"I recently liberated my Rev 1 Model B from the BrickPi robot and thought that it would be nice to take some photos of all the different versions I own," explains Richard. "I didn't get round to it straight away, and it languished on my list of 'things to do'. Then I was lucky enough to get a free Pi 2 on the day of launch by tracking down the Element14 PiCycle (**bit.ly/1DpL9Es**), which reinvigorated my interest in the idea.

"I asked the boys if they had any ideas of how to make the pictures more exciting than just a bunch of Pis on a desk, and they immediately suggested this."

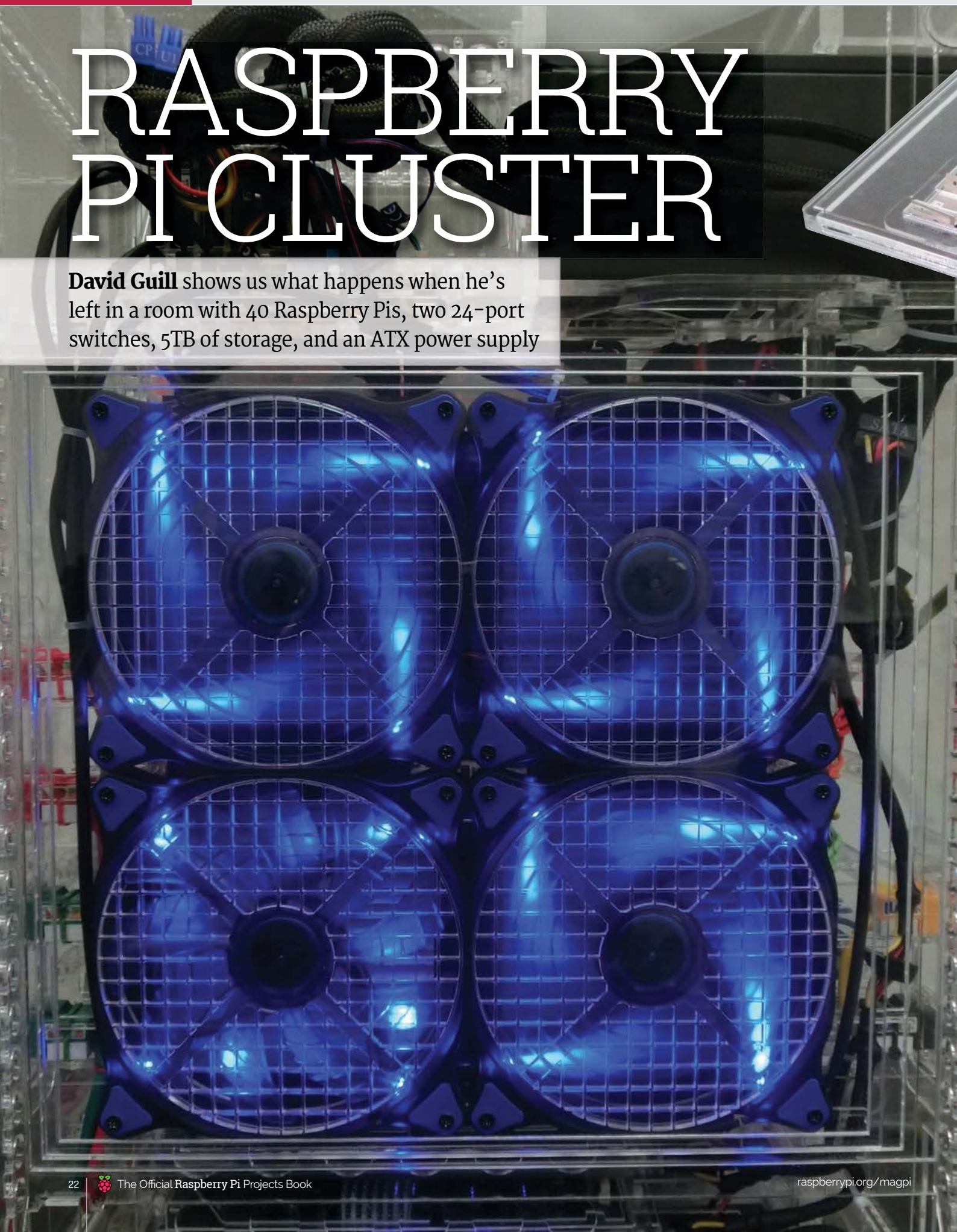
After discussing a few ideas, Richard's youngest, Ozzy, suggested creating a Lego timeline showing the Pis being used in different ways. "This morphed into a scene which follows the Pi from the design phase, through manufacture in a Pi factory, to being loaded onto a lorry. Then we have some children using it in a school, and finally, a Pi being strapped to a rocket, ready to launch up to the ISS to celebrate Astro Pi."

Each stage of the design boasts a more modern model of the Raspberry Pi, Richard explains, not to mention the addition of a couple of 'Easter eggs', including a rather suspicious-looking group consisting of a pirate, monkey, robot, and ninja.

What's next for the Haylers? "[We're] putting together some hardware based around the Model A+ for kite-mapping photography, that will record the altitude and orientation of the kite, and use it to have some intelligence about when (and when not) to capture an image." You can see more of the Haylers' Pi timeline, and learn more about the family's other projects, at **richardhayler.blogspot.co.uk**.

RASPBERRY PI CLUSTER

David Guill shows us what happens when he's left in a room with 40 Raspberry Pis, two 24-port switches, 5TB of storage, and an ATX power supply



A computer cluster is 'a set of connected computers that work together so that, in many respects, they can be viewed as a single system'. Clusters can be anything from a few cheap computers networked together to supercomputers made up of thousands of individual 'node' systems, designed to undertake complex tasks like modelling weather or trying to beat humans at chess.

Back in early 2014 David Guill, a recent MSc Computer Engineering graduate, showed the world his rather impressive project to create a computer cluster consisting of 40 Raspberry Pis.

He created his cluster entirely single-handedly, right down to the custom laser-cut acrylic case.

A new direction

A year on, we caught up with David to find he's still working hard on his pet project, and it seems it's taking him in new and exciting directions. "While it wasn't one of my original goals, the most important work I've done on it so far has been in porting software to ARM," says David. "I spent some time trying to get Apache Mesos working properly on it." It's a worthy distraction since ARM is fast becoming a real player in the server market, meaning David's work could have real value in the coming years.

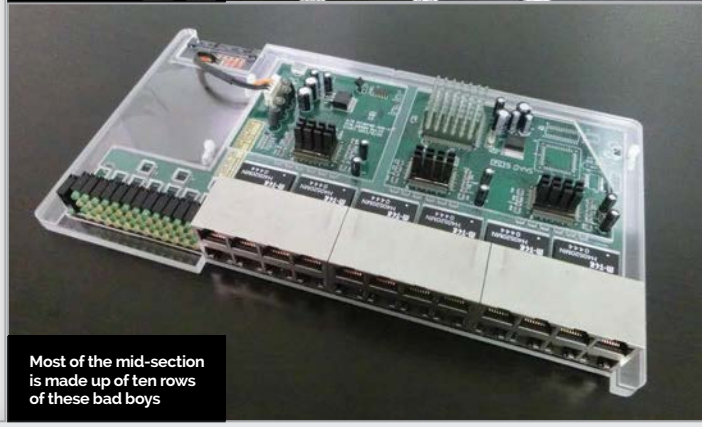
"While I've mostly been fixing supporting tools as I discover they aren't ready for ARM, I'll also be writing some of my own tools . My objective is to have a suite of tools with insignificant diminishment of returns for expansion, where the millionth node in a system would contribute nearly as much as the tenth did when it was new."

Virtual worlds

David's ultimate goal, though, is quite different – he wants to move into virtual reality. "My end goal is to develop detailed virtual reality simulations, like you might see in a hybrid of *Minecraft*, *LittleBigPlanet*, and role-playing games in general, with deformable planetary worlds. Of course, this is still hobby work – I have no guarantee that it'll ever get close to completion."

You can learn more about David and his work on the Raspberry Pi Cluster at **likemagicappears.com**.

The bottom section of David's cluster consists of two 24-port switches



Most of the mid-section is made up of ten rows of these bad boys

ANDY PROCTOR

A man with great entrepreneurial spirit, Andy was a tinkerer as a child and worked as an electrician as a young man. Now he drives his iData Truck.
idatatruck.co.uk

iDATA TRUCK

Quick Facts

- Andy learned to program for this project
- The community has already begun making his code better
- Most of the electronics come from a SunFounder starter kit
- Andy uses an iPhone to connect his Pi to the internet
- There will soon be a live camera stream from his lorry

A Raspberry Pi-powered lorry? It's not as strange as you think, as **Andy Proctor** shows us how he automates deliveries with Pi

Braving the often-congested motorways of Great Britain, we find our hero Andy Proctor – lorry driver and truck hacker extraordinaire – live-tweeting his schedule as he picks up, and delivers, the nation's shipping containers. There's much more to his tweets than meets the eye, though, and it all started with him finishing up his previous business and becoming a lorry driver.

"I was tweeting '#m25' and '#m12', and I noticed it was being retweeted automatically. I contacted the guy who was doing it and he told me it was powered by a Raspberry Pi and a bot. I looked up what a Raspberry Pi was and decided that I wanted to push a button and send a tweet."

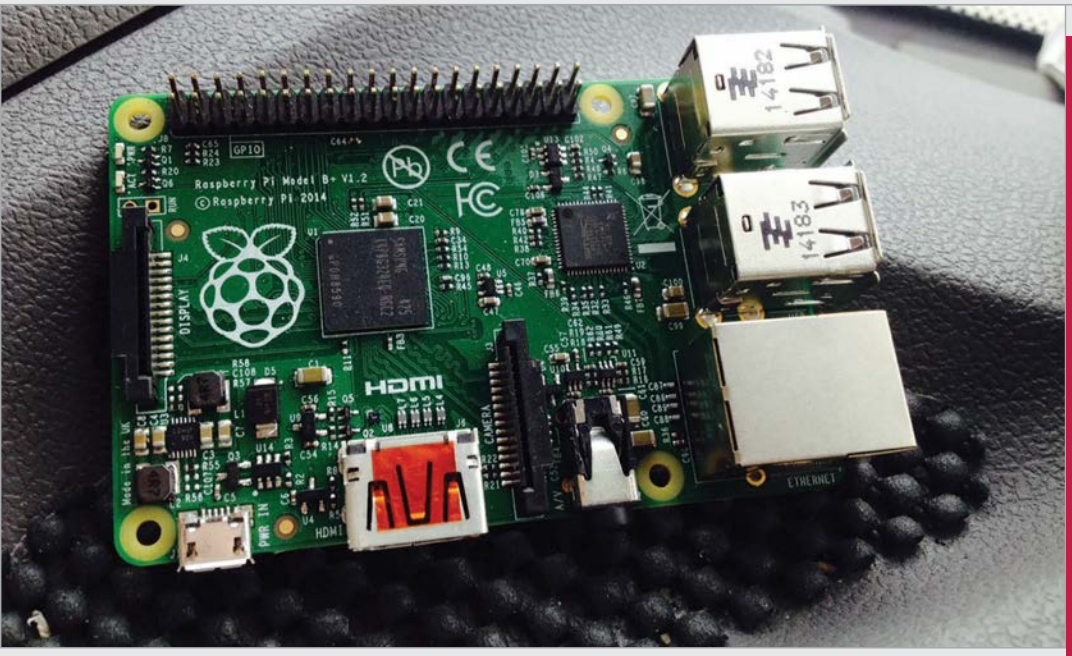
The transition from being a successful business owner to a lorry driver hadn't had the best impact on Andy, and his wife encouraged him to play about with his new Raspberry Pi on their honeymoon. With a background as an electrician, website builder and tinkering with computers and electronics as a kid, some of the Pi came naturally to him.

Humble beginnings

"I started off with Tweepy and Scratch to make some lights flash, and built a little box with a board for the lights, which made me happy! I then did the same in Python, learning along the way, and within six weeks I had created the box with the four buttons that you see now."

The iData Truck was born and not only did the buttons tweet out his current status, it emailed his office – a task he would have been doing manually anyway. He published a video on YouTube describing his setup, which got picked up by the Raspberry Pi Foundation and even the BBC. Andy isn't finished yet, though.

"I only have four things I can transmit," he laments. "So now what I'm doing is a barcode-scanning version. I've printed off loads of barcodes which I can stick to the back of my time sheet and



> "Not only did the buttons tweet his current status, it emailed his office"

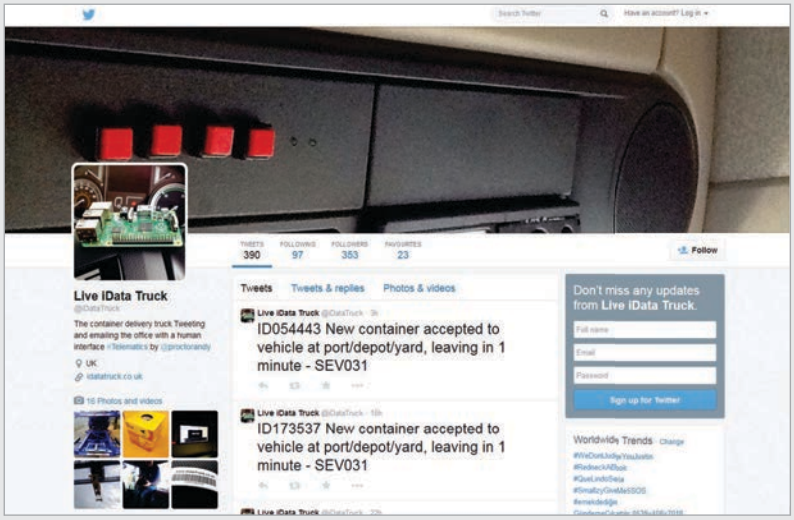
then I'll be able to scan them. If it's just scanning to say 'start of day', 'end of day' or 'on a break', it will tweet that but not email it. If it's 'running 30 minutes later' or 'box on'/'box off', it will still email it to the office. So there will be a split of what data gets sent where."

What's next?

Next on the list for Andy is a camera – Pi-powered, of course – in a blind spot of his lorry that will display on his dashboard and hopefully make it easier to manoeuvre while reducing the risk of accidents. He also has further plans for the iData Truck beyond his personal use of it.

"I've approached the people that make the software that everybody uses in the industry and they said if one of their customers wanted to use that, then that's fine, they'd support it... one person's been in touch that can make the hardware, a box to put it in, the switches in the panel, and the software, should I want to develop it further."

So next time you pass a container lorry on the M3, give it a wave and you might end up on iData Truck TV.



Left The information on the iData Truck stream is always expanding. Andy initially had trouble with duplicate tweets until he started using timestamps

HOW TO REPORT IN WITH iDATA TRUCK

> STEP 01

Press the button

When Andy loads or unloads, he presses one of the four preselected buttons on the iData Truck so he can let his company know what he's up to.

> STEP 02

Wait for the beep

You need to hold the button down for half a second: "It kept getting really hot, so I had to put a delay in to stop the processor working so hard from all the loops!"

> STEP 03

Email and tweet sent

An email is sent off to Andy's company to let them know of his status, and a tweet is sent to the @iDataTruck Twitter feed for everyone to see what he's up to.

MARK PARRISH

A .NET developer with a software consulting firm, who grew up playing Nintendo's ground-breaking consoles.
imgur.com/a/1RUwa

SNES PI CASE

Quick Facts

- It took two weekends to complete
- This is a US SNES, hence the purple and angles
- Some of the printed circuits are scratched off to make it work
- The original power light is fixed to a GPIO pin
- *F-Zero* is one of Mark's favourite SNES games

What happens when you turn a Super Nintendo into a Raspberry Pi? F-Zero becomes Raspbian, among other things...

You walk into a room and see a SNES. A classic, a legend, one of the greatest videogame consoles to ever be crafted by the hands of man. Beside it is a cartridge of the original *F-Zero*, perhaps not the best in the series but an excellent game nonetheless. You slam it in (gently though, they're both 25 years old), flick the power switch, and look for a controller. Suddenly, a Raspberry Pi logo shows up.

This isn't a Super Nintendo. It's a Raspberry Pi case that used to be a Super Nintendo.

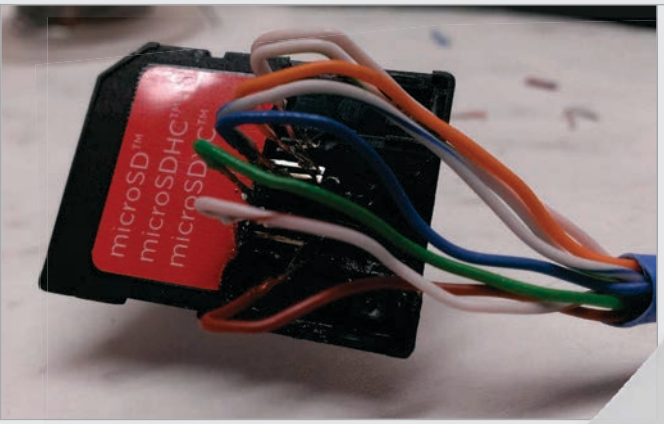
"Like most great ideas, [I got it] from watching others and seeing what they were building," Mark tells us. "[Also] how they were solving particular problems... then taking those ideas and improving on them in my own creative way."

His own creative way is frankly incredible. Instead of just fitting the Raspberry Pi into an empty case, he soldered parts of it directly to the original motherboard of the broken SNES he was working on. Most of the work on this project was the physical customisation part.

"The software side is easy since there are a numbers of solutions out there that have already been proven to be successful. The majority of the work I've done is with the physical part, and is easily 90-95% of the time invested."

Just about everything he could connect through the SNES has been done in that way, and just about everything uses the original port locations. USB and Ethernet are routed through the two front controller ports, the HDMI is in the old AV out, the power has been converted, and an on/off switch has been fitted into the aerial connector. That's not the best part, though:

"In my design, I've moved the SD card from the Raspberry Pi and connected it inside an actual game cartridge. I've noticed a few more failed boots than normally would be expected. Other than that, it works beautifully!"



Left Mark interfaced directly with a microSD card converter to allow booting from the cartridge

A common sight in '90s households – but this SNES holds a secret

A Nintendo console needs Nintendo controllers, surely? "I have two ideas," Mark tells us. "One is to take an original USB controller that works natively with the Pi and cram that into a controller housing. The result would look like the original controller. The other would be to map the current controller to the USB spec that the Pi expects. I haven't done a lot of research on this approach; however,

> "Figuring out these problems is the fun part of tinkering with gadgets"

figuring out these kinds of problems is the fun part of tinkering with gadgets."

While the internet likes to go a little bit mad whenever someone posts a new classic console mod like this, it sounds like this is nothing compared to the joy of actually doing it:

"I'm always amazed at the beginning of a project like this, that you have a workbench full of parts that by themselves do little or nothing. Then at some point while putting the parts together, something new and useful is created and essentially 'comes alive'."

POWERING UP THE SNES PI CASE

>STEP-01

Jack in

Plug your HDMI, power, Ethernet, and USB devices into the various ports – the controller connectors hold the latter two, in case you were wondering.

>STEP-02

Grab a game

Search through the game library for *F-Zero*. Make sure it's the right version of *F-Zero*, the one with an SD card of Raspbian on it. Slot it in the top of the SNES.

>STEP-03

Flick the switch

Unfortunately, the original power button won't help you here. Behind the SNES and next to the power cable is the on/off button – flick it to bring the SNES Pi to life.

SOUND FIGHTER

Cyril Chapellier and Eric Redon have brought a new dimension to the phrase 'duelling pianos' with an installation designed to turn two pianos into controllers for a game of Street Fighter Alpha 3

"We seamlessly transformed two classical upright pianos into PlayStation 2 controllers using custom analogue piezo triggers, a Pi B+, and Arduino Unos, and created a specific Python 3 firmware to map a classical playing style onto the *Street Fighter Alpha 3* gameplay, including combos and the like," explain the French duo.

The concept was pitched for the reopening of the Maison de la Radio, a historical radio building in the heart of Paris. "The building has been the home of the French public radio stations for more than 50 years and recently reopened to the public as a cultural space, offering a wide spectrum of entertainment choices, such as live concerts, workshops, and live radio shows," the pair explain on their blog.

To celebrate the reopening, Cyril and Eric worked on the overarching concept of bringing 'an alternative visual identity to music', which led to the idea of trying to bring the general public back in touch with classical music and classical instruments (like the piano) in this completely new and unique way.

While the concept was gladly accepted, the project itself was on a very tight schedule. "The go-ahead was given on 1 October [2014], the shooting of the teaser video was to take place on 12 November, and the live event on the weekend of 14 November."

The video shows classical pianists Alvise Sinivia and Léo Jassef, from the Conservatoire National de Paris, duking it out as *Street Fighter*'s principal characters, Ryu and Ken. You can see it at **youtu.be/7v2B71RUaqQ**.

Learn about the making of the 'Sound Fighter' installation in incredible detail on Cyril Chapellier and Eric Redon's blog, at **foobarflies.io/pianette**.

Above The project cleverly interfaces a piano to a PlayStation 2 controller so the game can be 'played'

Above The public were invited to test this novel take on 'duelling pianos'

Above Alvise Sinivia and Léo Jassef, from the Conservatoire National de Paris, demonstrate the installation

MARTIN MANDER

A database manager for Norfolk's Children's Centres who has a love of 'upcycling' vintage technology to incorporate modern components.
instructables.com/id/1981-Portable-VCR-Raspberry-PI-Media-Centre

RASPBERRY PI VCR

Quick Facts

- The project took six months to complete
- Martin has many similar broken devices ready to upcycle
- His next project is to upgrade a Seventies cassette player
- The Pi community helped a lot during construction
- This is Martin's first major Pi project

An Eighties video player or portable digital entertainment centre? Spoilers: it's the latter, and it looks truly amazing...

Portable video playback is a modern concept, right? Before magical internet-connected phones in our pockets, there were portable DVD players in cars. Laptops had DVD drives and video software a bit before that as well, so perhaps it's a little older than you might initially consider.

Meet the Sharp VC2300H and have your world turned upside down. This contraption from the ancient past of 1981 was able to play video stored on magnetic tape and housed in a big plastic rectangle – also known as a VHS. If you're not old enough to know what one of those is, ask your parents (or better yet, Siri). You did have to hook it up to a TV, but being able to lug it around was quite a novelty really, taking inspiration from portable stereos of the age.

This version is slightly different, though, as its creator Martin Mander explains very succinctly:

"I picked [this] up for 'spare or repair' on eBay for £6 – a top-loading VCR that unusually stands upright and has a carrying handle. I stripped out all of the internal circuits and replaced them with modern tech, with the Pi running the show, a powered USB hub housed in a pop-out VHS tape, an Arduino-powered clock, and a 15" HD TV panel integrated into the back of the unit."

The whole setup runs Raspbmc, the XBMC spin of Raspbian, and also allows you to stream from places like YouTube and the BBC iPlayer via WiFi. There's even a built-in IR sensor for media remotes.

"I'd made several other projects combining retro TVs and LCD panels before, but these were always tied to

Some of the physical buttons are used for fun lights and exposing the USB ports, and you can just about see the Pi through the side

a PC or video source," Martin tells us, when asked why he decided to put a Pi in it. "This time I wanted to make a much more interactive all-in-one device. The Raspberry Pi looked like the ideal solution on form factor alone and when I nosed around at the fan and support sites, I was impressed by the scale and enthusiasm of the community.

> "I stripped out all of the internal circuits and replaced them with modern tech, with the Pi running the show"

I'd experimented with APC (Android PC) boards in the past, but a lack of updates and support turned me off taking these any further."

Although all the physical buttons on the original machine still mostly functioned, Martin was unable to get them all working with the Raspberry Pi: "I spent several weeks looking into different ways of connecting up the hardware buttons to the Pi via USB, working my way through a series of stripped down gamepads and keyboards. One evening I wondered if I could maybe use an Arduino to mimic key presses, then the more I researched, it struck me that the Pi itself had GPIO ports and the buttons could possibly be connected directly... Ultimately, I think the 30+ year-old switch circuit was the reason I couldn't get this to work; in retrospect, I should have replaced the old push buttons with modern microswitches – next time! With the case already painted and the TV panel installed, I settled for just integrating the basic Play/Rewind/Fast-Forward buttons with the Pi, which I did using the circuit from a pound shop USB mouse."

While some more work may be required to perfect this fascinating project, overall it works really well, according to Martin. So if you ever find an old portable VCR at a car boot sale, maybe you can make yourself a cheap, portable media player with retro charm.

Below There's a custom paint job and a bit of Pi branding, but really you'd have to take a close look to realise it's different

GET WATCHING!

>STEP-01
Turn it on!

There are several buttons required to turn the full thing on. First the switch for the mains supply, then the switch to activate the lights and clock and other fun things, and finally the switch for the Pi.

>STEP-02
Add peripherals

Need to add some storage or a USB keyboard/mouse? Eject the cassette. That's where the USB ports are for the Pi – don't worry, you don't need to put it back in.

>STEP-03
Select a show

Raspbmc is running on the screen installed on the rear. Using an IR remote, you can select from local media or streamable videos on any supported service.

JOHAN TEN BROEKE

Software developer with a passion for good software, art, music, graphics, and print. Works with creative partner Jeroen van Goor.
fullscreennl.github.io/led-mirror

LED MIRROR

Quick Facts

- The panel is just 50mm thick
- The camera is smaller than a coin
- More than 272 connectors were clamped
- The mirror is displayed at Fullscreen.nl's Netherlands office
- Fullscreen.nl created BlackStripes, a Pi-based drawing bot

It's not every day that you are able to see yourself in a whole new light, but the LED mirror has managed to crack it

We wouldn't say Johan ten Broeke was a vain man, but he does spend an inordinate amount of time in front of a mirror. Catch his ghostly white, silhouetted reflection, though, and you will see exactly why – for this is no ordinary mirror, but a large and rather special panel powered by a Raspberry Pi.

Created as an art installation, the mirror consists of a staggering 2,048 LEDs. When someone stands in front of it, a camera picks up their movement and creates a series of snazzy effects. "Myself and Jeroen van Goor are imaging enthusiasts and we got the idea for the project when we were playing around with a Raspberry Pi Camera Module," Johan says. "We thought it would be an entertaining device."

Eating & sleeping LEDs

The project began on a small scale. "We built a prototype of several small 8×8 LED dot-matrix digits," Johan explains. It soon grew. Before long, the pair were connecting LED-strip PCBs together using ribbon cables, strapping them to an off-the-shelf reinforced floor grate using cable ties, and hand-soldering hundreds upon hundreds of LEDs at a pitch of 38mm apart, in a hugely repetitive process which took them both three days.

The effect is nothing less than stunning, however. "Although the displayed images are very abstract, spectators identify themselves instantaneously due to the feedback of their motions. This triggers people to move and wave in front of the mirror," says Johan.

Best of all, the mirror is adaptable. The horizontal spacing of the LED grid can be varied to compensate for narrow viewing angles, while the slim design of the LED modules allows the lights to be applied on curved surfaces. It gives the mirror wide-ranging potential.

"Some people have suggested the LED mirror belongs in a nightclub-like environment, close to a DJ set or something like that," enthuses Johan. "It certainly draws a lot of attention, so perhaps shopping

windows and interiors would be a good setting. It could be used for game-assisted physical therapy, for people who are recovering from injury and need exercise. Patients could 'hit' moving virtual sensors and the software app could produce progress reports to the therapist."

> "Some people have suggested the LED mirror belongs in a nightclub-like environment, close to a DJ set or something like that"

There's an app for that

Johan and Jeroen have written four main apps to accompany the mirror. The home app has 4×4 pixel sensors which are triggered when the change in pixel value exceeds a threshold; the recording app retains 10 seconds of display data at 15 frames per second and plays them back in reverse; the difference app displays a silhouette of spectators; and the drawing app only allows the brightest pixels to come through to the screen, allowing users to draw in white on a black canvas (or vice versa), with a flashlight or similar device.

"It's really adaptable and it could be made even bigger than the current screen size of 122cm by 244cm," Johan says. "The Maxim chips allow for a maximum of 256 cascaded devices on a single SPI bus and we use 16 chips right now. You could use a second SPI device in the Pi, allowing for a second 256-chip chain, making it 32 times as big." To which, on reflection, we can only stand in amazement.

USING THE LED MIRROR

> STEP-01
Plug in and play

The LED mirror is plugged into the mains and once it is switched on, you need to stand in front of it. There is a tiny camera mounted in the centre of the installation.

> STEP-02
Select a mode

The home app gives a generic view of the camera and accesses the other apps. Recording mode lets you create an animation. The difference mode lets you admire the view.

> STEP-03
Create a picture

In the drawing mode, grab a flashlight and create cool pictures on the mirror's 32×64 pixel canvas. After a while, the recording, difference, and drawing apps return to the home app.

Top Left As the user moves in front of the mirror, the LEDs power off to reflect the motion

Top Right A total of 2,048 LEDs were needed for the Raspberry Pi LED mirror project

JACK SMITH
A sales and IT assistant in the manufacturing industry, Jack knows his way around computers, but this was his first Pi project.

RASPBERRY PI ARCADE

Save money at the arcade and shoot down space invaders from the comfort of your own home

As kids, we all dreamed about having arcade machines at home so we wouldn't have to spend loads of money at the arcade to beat *X-Men* or figure out if Reptile was real in *Mortal Kombat*. While home consoles existed, it wasn't until the mid-Nineties that they came close to the quality of an arcade experience. The arcade was an experience, albeit one now lost to time; however, that hasn't stopped people trying to recreate it with modern tech. Enter the Raspberry Pi Arcade by Jack Smith.

"The idea was heavily inspired by the iCade by ION," Jack tells us, referring to the iPad dock that makes it look like an arcade cabinet. "I really liked the idea, but I don't have an iPad and I wanted something much more open. I had purchased a Raspberry Pi the previous week, which I had already been using to play games, and figured it would be the perfect match."

The Pi Arcade itself uses a Raspberry Pi running Raspbian with EmulationStation over the popular RetroPie, and has a 7" TFT display along with an authentic joystick and four buttons. It also

Quick Facts

- The project took two weeks to complete
- *OutRun 2* is Jack's favourite arcade game
- The original plan was to gut a Game Boy or SNES
- His next project is a light-gun arcade cabinet
- Jack's favourite game to play on the Pi Arcade is *Contra III*

> I really wanted that retro look, similar to that of an Atari 2600 or a Moog synthesizer

connects wirelessly for Jack to maintain it and add games. As for the cabinet itself, Jack made it himself:

"I designed the cabinet from scratch. I used SketchUp for the plan, mainly because it's quick and easy to use and also gave me the exact measurements required to make it out of wood. I'm not the best craftsman, so a friend of mine who makes children's toys from wood was a big help. The cabinet is made from MDF, but I really wanted that retro look, similar to that of an Atari 2600 or a Moog synthesizer, so I used black paint and some rolls of Fablon to achieve that.

"It works surprisingly well," continues Jack. "The one issue, as many people have said, is that it only has four buttons. I intended it to have six, but I had miscalculated the size of the buttons and could only fit four. The controls that are on it, however, feel responsive and very similar to the real thing. Most of the emulators work very well; however, it does struggle with PS1 and some MAME games, but hopefully the newer model of the Pi will give it the extra muscle it needs."

While it may not have quite enough buttons on the cabinet, you can plug in an Xbox 360 controller to have a little more control over the games – although it may ruin the experience a little.

"It was a great feeling [finishing the Pi Arcade]. The finished product really did take me back to the old bartop arcade machines I remember playing on holiday when I was a kid."

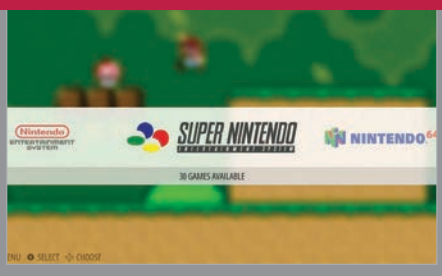
Above A hollowed-out interior houses the Raspberry Pi and the wires for the authentic arcade parts

Above Small but mighty – the Pi Arcade can play games from over a 25-year time span

INSERT COIN TO PLAY

>STEP-01
Power it up

The machine is powered by a portable battery – plug it into the Pi and activate it for the arcade machine to start turning on. It should run fine for a few hours.



> STEP-02
Choose your emulator

After loading EmulationStation, you can choose what to play on as long as you have some appropriate games to load on that particular emulator.

> STEP-03
Just have fun

Play using the controls or by plugging an Xbox 360 controller into it, and just enjoy the retro arcade experience on your sofa with no sweaty men to jostle with.

#OZWALL

For his flagship art installation in Nashville, Joseph Hazelwood brings old and new video technology together with the Raspberry Pi

The #OZWall video installation, the brainchild of Joseph Hazelwood, sits in the 'Escaparate', the focal point of Nashville Tennessee's centre for world-class contemporary art, OZ Arts (**ozartsnashville.org**).

Before becoming a hotbed for creatives, the space was actually a cigar warehouse that held over 100,000 of the owner's private cigar blends.

"From what I'm told, it was originally one of the largest cigar humidors in the world," quips Hazelwood, before getting to the crux of the project.

"A visitor to OZ will walk in the Escaparate and be drawn into an interactive multimedia experience. We like to think of this installation as a canvas for other artists to build upon, and that's the beauty of open source and platforms like the Raspberry Pi."

Hazelwood has effectively retrofitted six vintage TVs with modern LCD panels, though he plans to double this to 12. "Each TV is outfitted with its own Raspberry Pi 2. We used the code from the CCFE Pi Wall project (**ccfe.ac.uk/computing_projects.aspx**) and tailored it to our needs.

"Right now we have it displaying one large image and switching content via video editing, but in the long run we plan to use the Raspberry Pis to switch content and to make the wall more dynamic and interactive. We may also add cameras, motion detectors, and other sensors to the room/building for all manner of interactivity."

You can learn more about the #OZWall video installation and Joseph Hazelwood's fascinating penchant for juxtaposing old and new technologies at **hazelwoodlaboratories.com**.

Image: Sam Frawley, www.samfrawley.com

Below As shown from the rear, each TV in the #OZWall video installation is powered by its own Raspberry Pi 2

Below The programming aspects of the project were taken care of by developer Phillip Lehner

Below Joseph Hazelwood retrofitted and lovingly restored old TVs with modern flat screens

H.A.L. 9000

Willem Koopman opens the pod bay doors on his latest Raspberry Pi project: a fully functioning H.A.L. 9000

Inspired by a BBC Radio 4 documentary celebrating the theatrical rerelease of Stanley Kubrik's 1968 epic *2001: A Space Odyssey* (**bbc.in/1vc3NG8**), Willem Koopman decided to build his own homage to this particular piece of cinematic history – a working model of H.A.L. 9000, the ship's computer.

Much like the original H.A.L. who so famously lost his digital marbles part way through the film, Willem's effort still needs code to be added and polished: "The eventual plan is to have it as a work bot, so that when the website breaks or someone make a loud noise, we can take a picture for prosperity and/or automatically shame the perpetrator." But it already performs some suitably clever – and fitting – parlour tricks.

"At the moment, it runs a very lightly modified version of Jasper (**jasperproject.github.io**), which

Sci-fi fans will instantly recognise the design, which is near identical to the one in the film

allows a basic Siri-like behaviour. You can ask it to do various things like tell the time, check emails and Facebook – basically, all the marvellous things that Jasper allows you to do. You can also ask it to open the pod bay doors."

Besides the Raspberry Pi-powered elements, there was an awful lot of work involved in getting the case to look just right – not least the eerie fisheye with its chilling red backlighting.

"I was researching how the original was made and found out that the 'eye' was a standard Nikkor 8mm fisheye. Many years ago, I bought a wide-angle lens converter (the sort used to make skateboard videos before many readers were born). I dug it out, and the plan snowballed from there." The lighting is catered for by two well-placed LEDs.

The case itself was created using laser-cut MDF. "Cheap laser cutting is the most marvellous innovation of modern times," enthuses Willem. "Just create an SVG, send it off, and a few days later precisely machined parts arrive in the post. I can design and build straight dovetail joints to within 0.2mm, all without a chisel!"

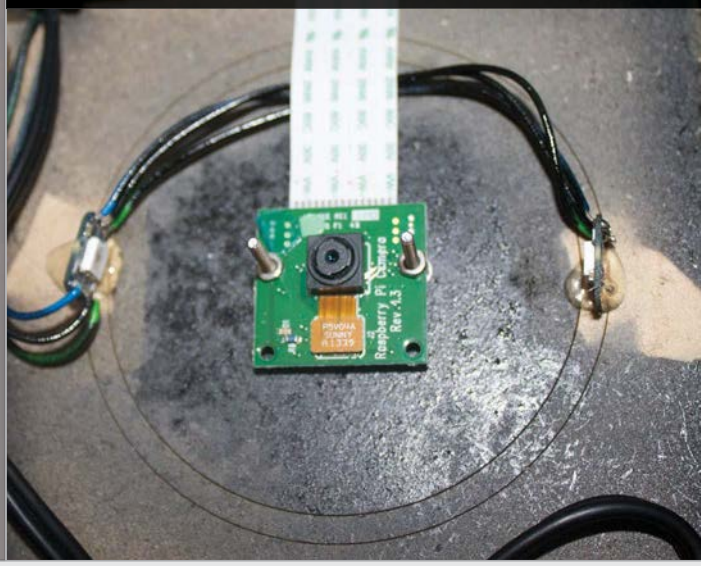
While the cutting was pain-free, the painting process has been anything but. "I spent about three weeks trying different techniques to get the glossy spray paint to actually stick." Willem says that the moral of the story is to always use a decent undercoat, but stay away from spray primers. "Lightly sand to get rid of the brush strokes. Repeat."

You could potentially set up Jasper to create your own personalised Siri with the Pi, but until then you can learn more about Willem Koopman from his blog: **secretbatcave.co.uk**.

Below The business end of the project comprises the Pi, speakers, a Camera Module, and a USB microphone

Below Willem used an old wide-angle lens converter to create H.A.L.'s iconic 'eye'

Below Hiding behind the lens is the Camera Module and two LED lights, one on either side



CORY KENNEDY

A family man and a cyber defence professional, who is also the co-founder of Kansas City's premier information security meetup, SecKC.
imgur.com/a/9JXbT

PWNGLOVE

Quick Facts

- It took seven days from conception to presentation
- Cory's son helped with the soldering
- The Power Glove was an official Nintendo licensed product
- It accepts two Wii Remotes as controllers
- It was created as part of a 'show-off' meetup

The infamous NES peripheral, the Power Glove, has been the butt of many jokes and centre of a lot of nostalgia. What happens when you don't need the rose-tinted glasses any more?

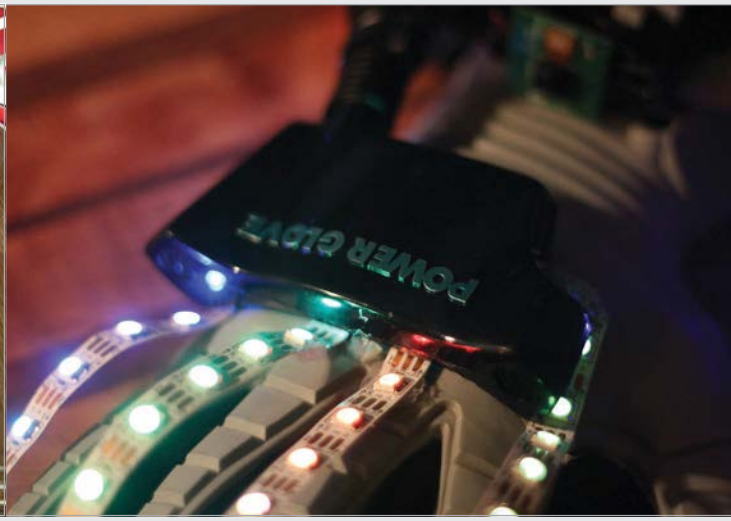
Let's face it, once you take off your rose-tinted glasses and forget the quotes and memes surrounding cult 1980s adver-film *The Wizard*, you're left with a barely functional peripheral that is very much a product of its time. The Power Glove was a novelty and, while it captured the minds of many young children, it never quite lived up to its promise at the time. Apparently, it's this exact reason that got Cory Kennedy motivated.

"The PwnGlove was a project rooted in the desire to build the memory my 12-year-old self was deprived of, thanks to the marketing genius behind Power Glove's capabilities," Cory explains to us. "I saw other people doing all sorts of projects with Power Gloves and I thought it would be a perfect match to pair it with retro NES gameplay."

The PwnGlove was originally created for Cory's information security meetup's 'Hacker Show-off Contest', where he had 15 minutes to wow the crowd with his invention. Although the focus was information security, Cory wanted to try something else:

"I wanted to do something different, mainly because where I am today professionally is rooted in tinkering – not only with computers at a young age, but also videogames… I wanted to be the kid from the 'now you're playing with power' ad. Which is exactly why I added the NeoPixel array."

As well as being a pretty good fit for the inside of the Power Glove, the Raspberry Pi came in very handy for enabling



> As well as being a pretty good fit for the inside of the Power Glove, the Raspberry Pi came in very handy for enabling Bluetooth and Arduino interaction

Above The lights on the glove are supposed to pay homage to advertising material for the Power Glove

Above Left An Arduino can be found in the original circuit compartment, plus a screen and Raspberry Pi on the wrist

Bluetooth, Arduino interaction, and access to the actual games.

"The bodywork was tough, mainly trying to resist the urge to take away from the original look too much," Cory says, regarding actually getting everything into and around the glove. "I say that, but then I take a look at the monstrosity it has become and I have to say... It now reminds me of the *Back to the Future* DeLorean."

He did try to use as much of the original parts as possible, though.

"It leverages all the original components, modified physically a bit to cut [out] the non-essential PCB, to allow for room for new components. There are four original bend sensors (thumb, index, middle, and ring), which connect to an analogue multiplexer living in the palm housing, which sends that data back to the Arduino. This is all piped over Bluetooth back to the Raspberry Pi. The wrist pad buttons are mostly intact (programming buttons aren't working), and they are set up now to do things like allowing the Konami code and switching the colour and pattern sequence on the NeoPixel array."

Cory is forever making tweaks to his PwnGlove, improving it in minor ways to solve what he sees as problems, but admits that most people don't even notice.

"To be honest, the reactions it gets from people, especially kids, certainly makes those imperfections disappear."

START PLAYING WITH POWER

>STEP-01
Konami coding

To get into the PwnGlove, you need to use the password. This is entered using a special sequence of keys: Up, Up, Down, Down, Left, Right, Left, Right, B, A, Start. 30 extra lives await.

>STEP-02
Choose your game

The RetroPie interface comes up, enabling you to select any compatible game from your library. It's best to use the buttons on the pad to select one.

>STEP-03
Team up with a bad dude

Throw a Wii Remote to a friend and you can then rescue the president, save the princess, fight the Red Falcons, or swear at Battletoads together.

TOM SHERLOCK

Our intrepid astronomer is a software developer at Wolfram Research, and has a background in optical and atomic physics, with a long-time interest in photography.
bit.ly/1HSxF4U

Quick Facts

- The whole thing only took a few hours to make
- You don't need a very expensive telescope for this project
- You can see the camera preview on a remote desktop
- Tom wants to have Wolfram control the motors next
- The whole Pi setup weighs less than a normal CCD camera

MATHEMATICA TELESCOPE

The Raspberry Pi has access to a huge amount of mathematical power and knowledge, thanks to Wolfram's Mathematica – here's how one man uses it for stargazing...

Space. The final frontier. This is the adventure of Tom Sherlock. His mission: to explore the stars from his backyard, using a telescope to photograph stunning cosmic vistas. And he's managed to make the process much better and easier for himself by using a Raspberry Pi.

He needs more than a Pi, though – it's good, but it's not good enough to photograph close-ups of the Moon. In this particular instance, the Raspberry Pi is connected to a telescope using a custom-made mount to slot it into the eyepiece, so that the Raspberry Pi Camera Module can take these excellent photos. On its own, it's impressive enough, but Tom has added a few more tricks to make it a truly unique Pi project – namely, he can control it with Wolfram Language and Mathematica.

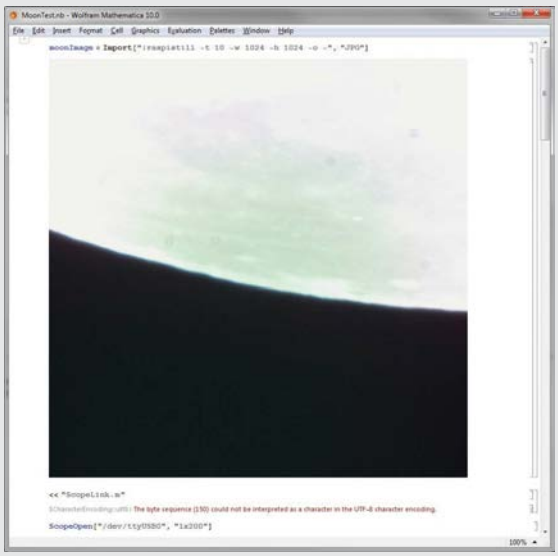
"Mathematica combines a powerful language, device control, and image processing facilities with a great deal of built-in data," Tom explains to us. "This includes astronomical data, so it was not hard to see the advantage of having one piece of

software which could manage all these tasks in a unified, scriptable way."

Tom actually works at Wolfram on Mathematica, so is well versed in the ways of the language and Mathematica itself. He's also an amateur astronomer, which is where the idea came from in the first place:

"Over the past decades and especially in recent years, amateur astronomy has become more and more dependent on computers for various tasks, like planning observations, pointing and driving the telescope, capturing and processing images and data, and finally sharing images and data with other astronomers.

"Typically, you have to run several different packages on several different computers to accomplish these tasks. For example, you might plan observations with one package, control the telescope with another package, capture images with still another package, and then process the images with one or more additional packages, sometimes involving specialised plug-ins."



"It has to be controllable from a computer," Tom elaborates. "Many scopes these days have computerised mounts, and the computer built into the mount will let you select any object in the mount's internal database and slew the scope to that object. Generally, these mounts will also allow you to connect them to an external computer and then, using the correct protocol, specify an arbitrary set of coordinates. Since Mathematica has the coordinates of hundreds of thousands of different objects, you can easily look things up and have the scope slew to that object."

Above The Pi camera might not be of the highest fidelity, but it can take stunning shots when used correctly

> "On its own, it's impressive enough, but Tom has added a few more tricks to make it a truly unique Pi project"

Tom was able to solve all these complications, thanks solely to the Pi having Wolfram available with Raspbian. Just having a Pi and some coding skills is only one part of the challenge, however, as telescope selection is also important.

Above Just a Raspberry Pi Model B+ is needed to perform the complex calculations needed for the telescope

It all sounds good in theory, but just how does it work in practice?

"Remarkably well, considering how much the Pi is doing. The main problems I found were more details than anything else. It takes a bit of time to look up astronomical coordinates on-the-fly, so I found it was better if I did that ahead of time instead of 'in the field'."

SEEING STARS

>STEP-01
The setup, according to Tom

Set up the scope and mount, and perform the mount's alignment procedures so that its computer knows how it is oriented, with respect to the Earth and the celestial sphere.

>STEP-02
Attaching the Raspberry Pi

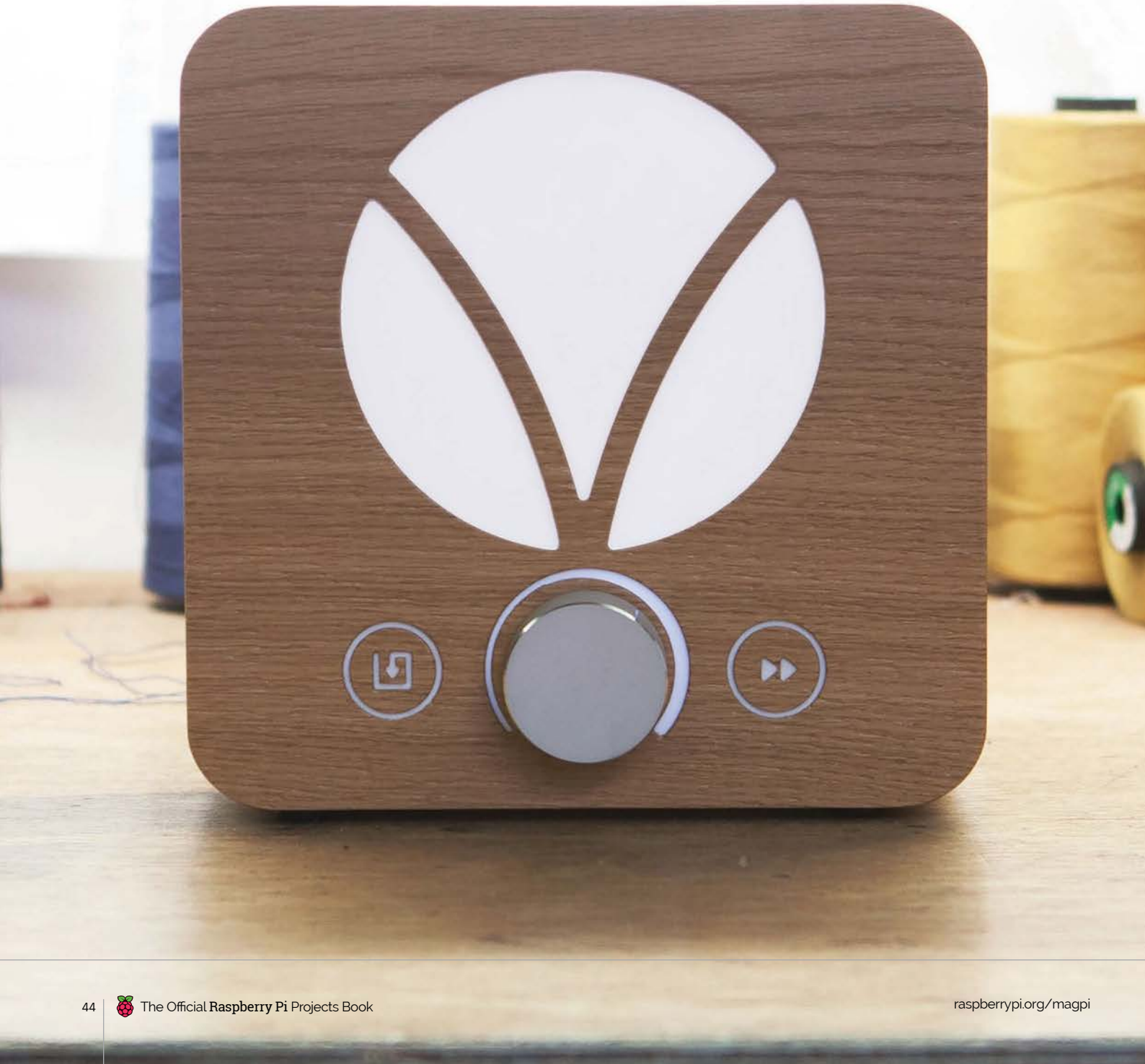
Once the telescope is sorted, then you just need to attach the Raspberry Pi. Tom uses a serial-to-USB adaptor to do this, then boots up the Pi.

>STEP-03
Do the maths

Once the Pi is booted up and in Mathematica, you can start issuing commands using the Wolfram Language to control the telescope – this will change depending on what mount you're using.

#HIUTMUSIC

Creative technology agency Knit built an internet radio with a difference to entertain Hiut Denim's workforce...



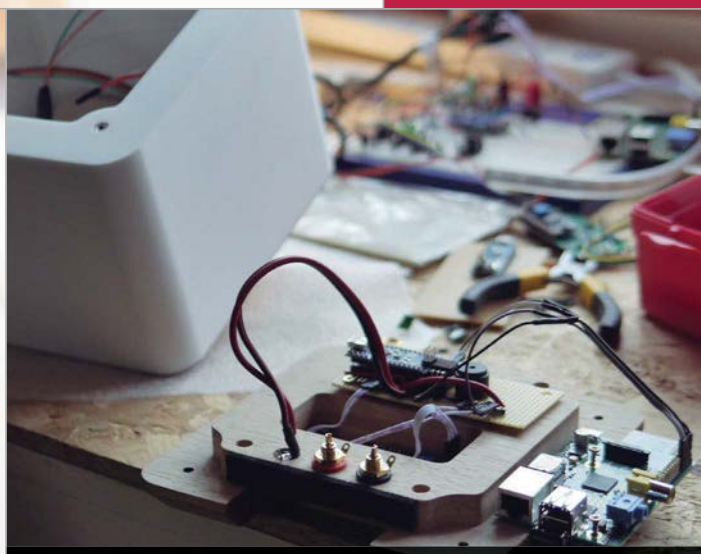
The #HiutMusic jukebox is a rather beautiful Twitter-powered music player that takes pride of place in the Hiut Denim Factory on the west coast of Wales, where 'the music is loud and the coffee is strong'. It was created by Knit, a creative technology agency that approached Hiut Denim with an idea to help customers connect with the boutique jeans company.

"We wanted to facilitate a dialogue between Hiut and their fans through the emotion of music, creating an opportunity for customers to have an impact on the people that make their jeans," explains Jack Chalkley, head creative technologist at Knit.

It's powered by an internet-connected Raspberry Pi, which uses the Spotify and Twitter APIs in a rather novel way. "It plugs into the existing sound system on the factory floor and fans can request a track by posting a tweet that includes #HiutMusic, the artist, and track title," says Jack. The tweet is detected and the song is queued up and played, but the project doesn't end there.

"Hiut's jean makers can skip, save, and adjust the volume of tracks on the face of the radio," continues Jack. "The 'skip' button and volume knob do exactly what you would expect, but the 'save' button saves the current track to a favourites playlist and a tweet is sent from the **@HiutMusic** Twitter account, sharing the request." What's more, the backlit display on the front of the #HiutMusic jukebox changes colour based on how far away the track request was sent. The further the sender, the warmer the colour displayed. "For example, a request from Wales would turn the display light yellow, whilst a request from New Zealand would illuminate deep red," explains Jack.

You can learn more about the #HiutMusic jukebox at **weareknit.co.uk**.



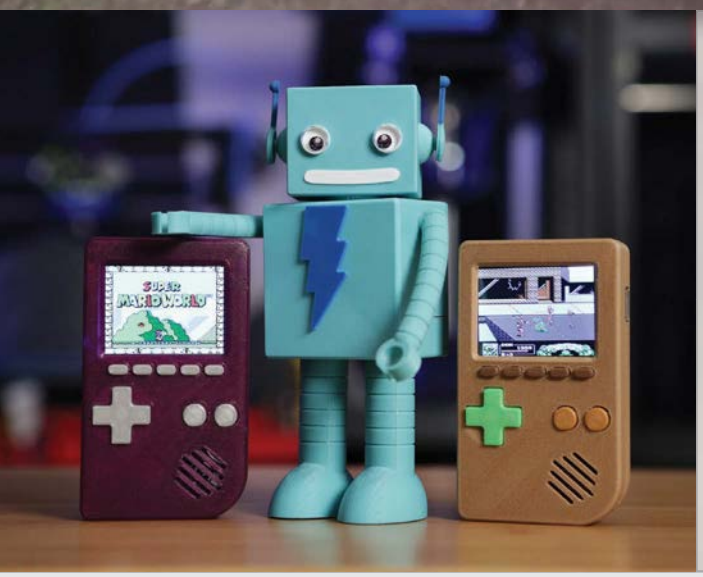
Above The project also uses an Arduino Nano, partly to help reduce the CPU load on the Raspberry Pi, which uses up to 70% on decrypting Spotify tracks

Above A ring of NeoPixels is used to create the geographically controlled light effect

Above Capacitive touch sensors are used for the 'skip' and 'save' buttons

POCKET PIGRRL

Adafruit's Ruiz brothers are back with a 2015 refresh of their brilliant Nintendo Game Boy project that's half the size and twice the fun of its predecessor…



Last year it was the 25th anniversary of the legendary Nintendo Game Boy handheld console; to celebrate, **Adafruit.com** came up with a great project for Raspberry Pi emulation fans, called the PiGRRL.

Suffice it to say that the project was a resounding success and **Adafruit.com** has returned with this 2015 refresh, named the Pocket PiGRRL. It uses the Raspberry Pi Model A+ and a 2.4" PiTFT HAT (with a resolution of 320×240 pixels), making for a much smaller and lighter project than its predecessor. According to its makers, it's about half the size overall, measuring 118mm tall and 69mm wide.

To create the controls, instead of using a SNES controller as before, the makers have opted to use cheap and easily sourced tactile switches soldered to a cut-down Perma-Proto PCB, which is wired to a ribbon cable and connected to the Raspberry Pi's GPIO pins. While some readers might be wondering how you can play *Super Mario World* with just two buttons, you'll also find instructions and 3D printing

files for a four-button version of the Pocket PiGRRL.

To power the project, the Ruiz brothers are using the PowerBoost 1000C, which features a built-in load-sharing battery charger circuit, meaning you can power your Raspberry Pi while it charges the project's ample 2,000mAh Li-Po battery. The Pocket PiGRRL even features an audio amplifier and a tiny 1W mono speaker.

While you're free to set up the software side of the project in any way you like, the Ruiz brothers have opted to use RetroPie (**bit.ly/1lYZkDg**), a great emulation package for the Raspberry Pi that enables users to play games from all sorts of classic systems.

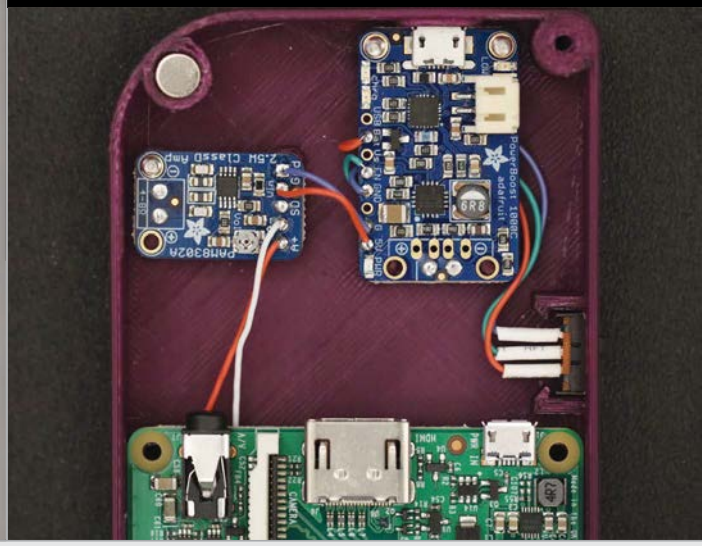
It's fair to say the project isn't particularly taxing to build, though you will at least come out of it an expert solderer. You're definitely going to need helping hands with a large magnifier – some of those joins need precision work!

You can learn more about the project, and find out how to build your own Pocket PiGRRL, on the Adafruit Learning System via **learn.adafruit.com/pocket-pigrrl**.

Below There aren't a great deal of parts for the Pocket PiGRRL, making it a great introduction to more advanced Pi projects

Below The files you need to 3D-print the chassis are available for download on the project's webpage. If you don't have a printer, you can order 3D printed parts online

Below Here you can see the audio amp (top left) and the power charging unit (top right). The three wires leading to the side of the chassis control the power switch



Right The Pi mascot, Babbage Bear, beats Felix Baumgartner by reaching 39km

SPACED OUT

Dave Akerman's frequent and eye-catching forays to the edge of space have made him the Raspberry Pi's foremost near-spaceman

It may have proven to be one small step for man, but space travel has certainly been one giant leap for the tiny Raspberry Pi. When the makers of this bare-bones computer came up with the device, no one in their wildest dreams would have thought it would boldly go where few other machines have travelled before: to the stratosphere and back. But thanks to the Pi community's very own 'spaceman' Dave Akerman, that is exactly where it has been, and the results have been nothing less than stunning.

Below Dave Akerman and Heston Blumenthal during filming for the celebrity chef's UK TV series

Dave is a high-altitude ballooning enthusiast who has been tethering Raspberry Pis to helium balloons and sending them to the edge of space since 2012. His hobby and choice of computer have been attracting much attention since, leading to a rather hectic life for the software programmer, and helping to show further evidence of the adaptability of the Pi.

In the Pi's lifetime, Dave has appeared on television, become something of a YouTube sensation, and worked with a celebrity chef, making him one of the most well-known users in the ever-growing Pi community. Not that he's complaining about becoming a Pi celeb. "The Raspberry Pi had two big effects [on my high-altitude ballooning]: the addition of live images and all of the media attention," Dave tell us. "I expected the former, but not the latter. It's all been good, though."

High-profile performances

One of his most high-profile recent performances came on the BBC 2 show *Stargazing Live* on 20 March 2015. Dave was invited to rub shoulders with the likes of the European Space Agency astronaut Paolo Nespoli and Paul Franklin, the visual effects supervisor of the Hollywood movie *Interstellar*.

Stationed at Leicester racecourse, Dave was asked to launch a 434MHz (RF link) balloon equipped with 'Pi in the Sky' telemetry boards to capture stills and video from above the clouds of the solar eclipse taking place that day. The Raspberry Pi soared to a maximum altitude of 30km, taking in images, before popping and landing in a field just south of Leighton Buzzard – a huge success which delighted viewers and the BBC. "It was just a shame that there was so little time to explain the flight," says Dave. "The images shown weren't near our

best ones either, but these are the limitations of live TV."

Even so, the flight proved to be a thoroughly enjoyable experience for the radio amateur. "There was so much rehearsal that by the time the cameras were live, I was pretty relaxed," says Dave about his long, busy and exhausting day. Unfortunately, he did not get to meet the main presenters, Prof Brian Cox and Dara Ó Briain, since they were based at Jodrell Bank, just a few miles from Manchester Airport – "I'd have been unlikely to get permission from the CAA to launch – and yes, I did ask!" Dave laughs. Nevertheless, he is fired up about the total solar eclipse that will take place on 21 August 2017 in the USA. "The whole thing was good practice and I can't wait for that one."

While thinking up and taking part in numerous headline-grabbing events, Dave has been able to refine his methods, regularly coming up with new and more efficient ways of 'exploring' life beyond our planet. "The first Pi I sent into the air used a webcam, and that's been replaced by a Pi camera," he tells us.

"Back then, I also handmade the radio boards, but now I use PCBs, which makes things simpler and more reliable. The very first Pi flight used linear regulators with more batteries than was strictly necessary, meaning that the inside of the payload box got very, very warm. Nowadays the tracker board and Pi all use switching regulators, so fewer batteries are needed."

The LoRa module

To make the Pi more useful for his hobby in the future, Dave has been working on refining a long-range radio (LoRa) module for the Pi. It was on board the balloon used in *Stargazing Live*, but the media activities surrounding the launch meant Dave didn't get a chance to set up the tracking. Had he done so, he would have been able to eliminate the need for a PC, since LoRa devices are transceivers – they receive as well as transmit. The hard work of demodulating the signal is done

> "The Pi is an ideal device, allowing the telemetry to be automatically uploaded"

GET INVOLVED

David Akerman discusses what it takes to be a Pi spaceman or woman...

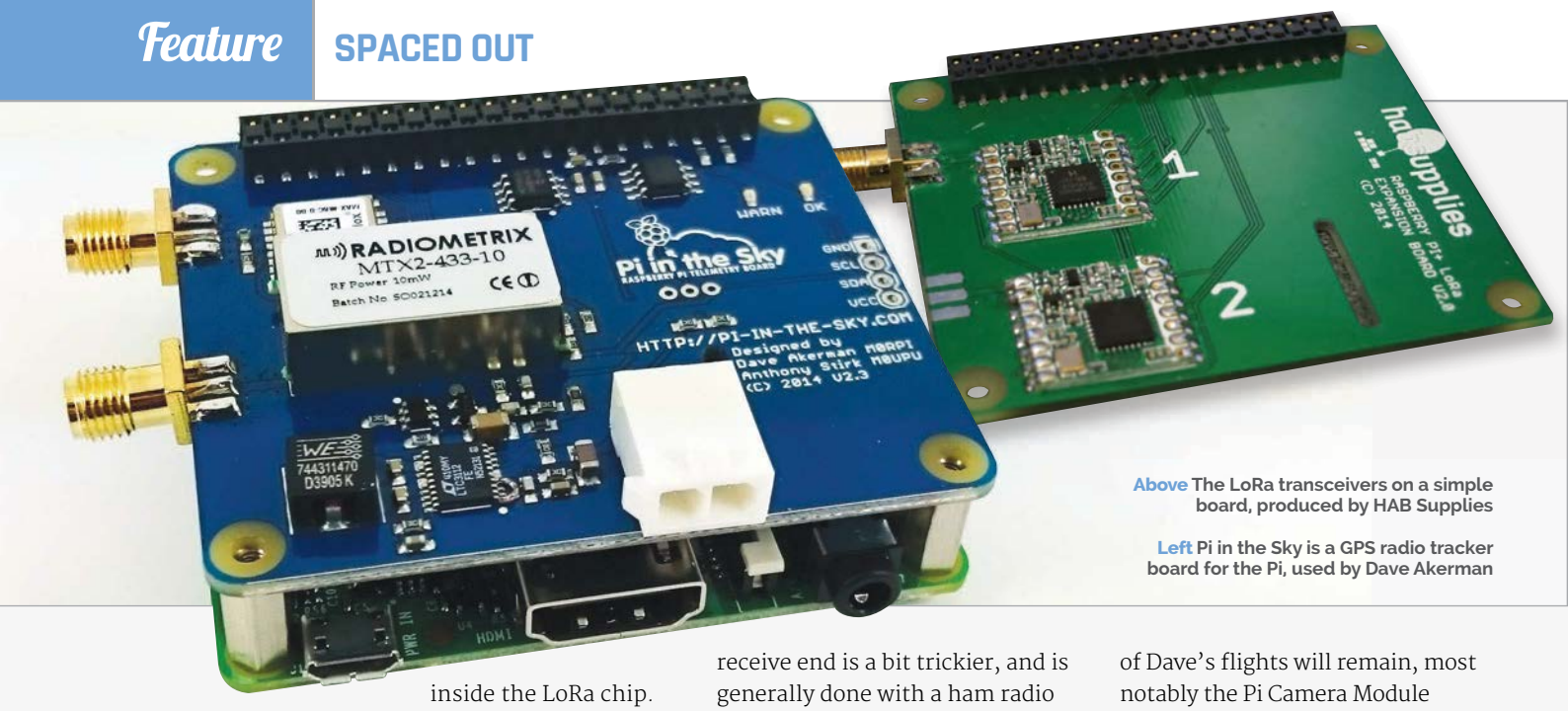
MagPi: What should amateur Pi fans be doing if they want to get involved?
Dave Akerman: Research! There's a lot of information on the web, with the UKHAS wiki (**ukhas.org.uk**) being the best single resource. Also, I encourage everyone to design, build and code their own tracker, as it's much more rewarding to fly something of your own design than to just fly an 'off the shelf' tracker. That said, if you are short of time or skills, then you can buy a Pi tracker board rather than make one, and/or you can use open-source software rather than write your own. But, if you can DIY, please do.

MagPi: What will readers learn along the way?
Dave: Lots. Whilst high-altitude ballooning isn't complicated (pause as I wheel out an old joke) – it's not rocket science – it does encompass many fields, such as physics, weather prediction, how the atmosphere works, electronics, radio, and software.

MagPi: Which Pi model would you recommend?
Dave: These days the A+ is the obvious choice. What you need from a tracker computer is light weight and low power consumption; you don't need higher power consumption, extra USB sockets, and a network interface!

Behind A view over Cornwall taken from a high-altitude balloon carrying a Raspberry Pi

Below Dave Akerman being interviewed on TV by Dr Lucie Green



Above The LoRa transceivers on a simple board, produced by HAB Supplies

Left Pi in the Sky is a GPS radio tracker board for the Pi, used by Dave Akerman

inside the LoRa chip. "The Pi is an ideal device, allowing the telemetry to be automatically uploaded to the internet for display on a map," says Dave. "It also becomes possible to upload messages to the balloon tracker."

LoRa will replace the traditional RTTY (radioteletype) balloon tracker for Dave, even though the older system is very easy to program and has been used for around 95 percent of amateur high-altitude balloon launches in the UK. "It's essentially the same as RS232 serial communications, with the ones and zeroes being denoted by two slightly different radio frequencies," explains Dave. "On the Pi, all that's needed is to connect the serial port to a small radio transmitter via a few resistors, and then send the data out of the serial port. The receive end is a bit trickier, and is generally done with a ham radio receiver and a PC."

Dave has carried out four LoRa test flights so far, the first of which landed on a golf course during a competition. "The payload was collected by one of the golfers, who wrapped it round his trolley, which explained the fact that the landing position kept moving!" Another test flight had two LoRa trackers, one receiving data from the first and then relaying to the ground over an RTTY link. "Most impressive, though, was a trial of high-data-rate images, where the incoming packets managed to saturate the uplink on my admittedly slow ADSL internet," Dave adds. "More work is needed, but this does look promising."

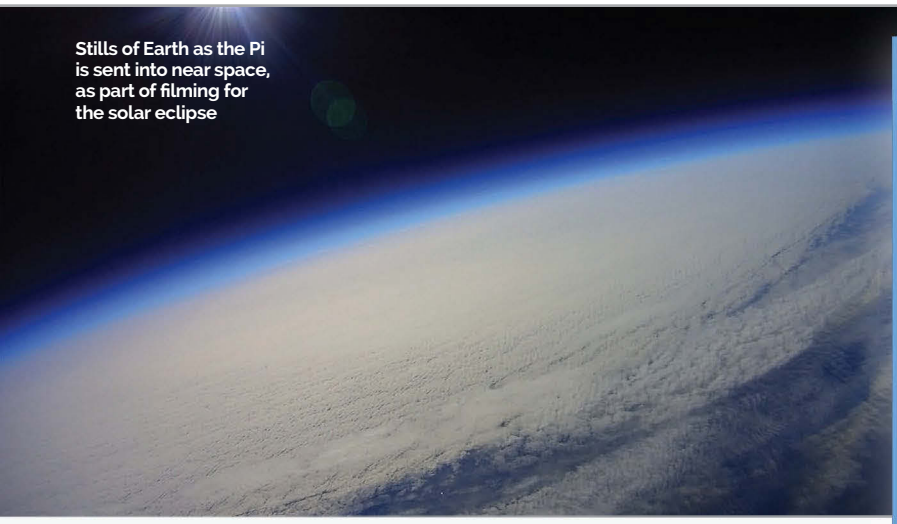
Yet even though LoRa modules are being introduced, some aspects of Dave's flights will remain, most notably the Pi Camera Module which replaced the original webcam on the balloon's payload. "The Pi camera quality was a big improvement on the webcam," he tells us. "It was lighter too, which helps. Previously, I used Canon compact cameras for stills, or a Kodak camcorder for video."

Below The potato in space, as filmed from a GoPro

Video capture

Such video-taking capabilities have proven to be very useful, especially during some of the publicity-generating stunts that Dave has pulled. His favourite involved taking a 20cm-tall teddy bear – the Raspberry Pi mascot, Babbage – to an altitude of 39 kilometres and dropping it from the sky. He wanted the toy to rival human daredevil Felix Baumgartner, who had set the world record by skydiving from a height of 38,969 metres in October 2012.

For the 2013 Babbage flight, the teddy dropped at speeds of up to 200 miles per hour as a fitted camera filmed what the toy 'saw'. Landing four hours later in a field near Shaftesbury, the teddy – which contained a Pi and a tracking device in its stomach – was intact, and its endeavours eventually came to be watched by more than 160,000 YouTube users. "I wanted that downward video of the [Baumgartner] jump, but with a slightly lower budget and rather less seriousness,"



Stills of Earth as the Pi is sent into near space, as part of filming for the solar eclipse

explains Dave. "I'd bought a Babbage soon after he went on sale, with the hope that I'd be able to fit a Pi Model A inside him, and was pleased to see that it just fitted. The hardest part was replacing Babbage's eye with a Pi camera – those eyes are very, very well fixed in." Babbage beat the world record by 31 metres.

This 'mission' showed Dave's eye for detail, even though there were problems. At one stage he was going to build a replica of Baumgartner's capsule, until it appeared it would be too heavy. Tests had also shown a reluctance for Babbage to jump. Even when it transpired all had gone to plan, Dave's attempts to track the toy were hampered by the SIM card in his MiFi (mobile WiFi hotspot) running out of credit. "I also had a call from the BBC asking about the progress, just as I was trying to find Babbage in a field. I managed to walk straight past him."

> "For the 2013 Babbage flight, the teddy dropped at speeds of up to 200 miles per hour"

However, as with Dave's other exploits, publicity followed. Two journalists from Austria even accused him of faking the flight as a publicity stunt for the Raspberry Pi Foundation. "Well, like the moon landings, it was easier to do it than fake it," laughs Dave. Not that the launches are actually simple. He says anyone wanting to replicate Pi in the Sky flights would need to spend a good few months researching first, and even he has had his fair share of mishaps. "I did manage to lose a couple of payloads to the sea: one early on, when I was unaware of the effect of wind on gauging the amount of gas in a balloon, and one later on, when the predicted flight path was completely wrong because the Met Office didn't fly its own balloons the day before."

But with the top four flight records for live images at around 40 kilometres altitude, Dave's hobby is, he adds, a thrill that never leaves him. "On the very first flight, the thrill was to hold something in my hand that had, to all intents and purposes, been to space," he says. "Prior to that flight, that was all I wanted, plus images that I had taken from near space. For many people this is enough, but like a few others in this hobby, I like to find new things that I've not done before."

CHEF ENJOYS A SPUD LAUNCH

In October 1995, the potato made history when top scientists working for NASA developed a special kind of spud: one that could be grown in space to feed hungry astronauts. This amazing fact was not lost on English celebrity chef Heston Blumenthal when he was creating his Channel 4 series, *Heston's Great British Food*.

For the 'Pie' episode, Heston decided it would be a good idea to use Dave Akerman's high-altitude Pi in the Sky balloon to launch a potato. "My first question was 'are you expecting anything physical to happen to the potato?', thinking that Heston probably wanted it freeze-dried or something," says Dave. "But no, it was just to put it into near space and get it back again. No problem then."

To achieve this, Dave built a payload with a Raspberry Pi doing the tracking and live imaging, plus three GoPro cameras – one up, one down, one sideways – for video. There was a second AVR tracker in a separate payload as a backup. Dave then had to find a day when the wind predictions were suitable and Heston was available. "That was not an easy task," he adds.

But how receptive was Mr Blumenthal? "Heston, as you might imagine, is basically a ten-year-old adult, so he and I got on just great," Dave replies. "We started chatting and the producer stopped us, explaining that he wanted cameras to record Heston's reaction when he learned things for the first time. We then had to drive up and down the road for some footage, during which Heston kept starting sentences like 'So, when the balloon bursts...' and I'd reply with 'Sorry, can't tell you yet!' It was obvious that he'd been looking forward to the day for some time."

The flight was straightforward, with the crew following the live images in the chase car. Quite fittingly, the landing was in a muddy field and lots of coverage was gathered for the TV programme. "It was great to see that when it eventually got broadcast, and it was a lot of fun to then see the flight ridiculed a few days later by [English comedian] Alan Carr."

Even so, it piqued a lot of interest. Following the show, Dave had a noticeable increase in the number of emails received from people who wanted to follow in his footsteps. "Flights like these do encourage others to get involved," he says.

SKYCADEMY

Aided by Dave, the Raspberry Pi Foundation's Skycademy event introduced 24 teachers to high-altitude ballooning, enabling them to help schoolchildren launch Pi-equipped balloons.

LAURI HAKKARAINEN

Lauri and his friends founded Esmes Digital, a web and mobile application business that sometimes has spare time to make crazy projects.
sneek.co/blog/candypi

CANDYPI

Use your Raspberry Pi to satisfy your sweet tooth with an old-school candy dispenser and new-school technology

Quick Facts

- The project uses a Raspberry Pi Model A+
- It took the team a day to build the whole thing
- The USB port was removed and attached via wires
- The project's components are wrapped in duct tape to protect against power surges
- Their next project is a Pi-powered 3D printer

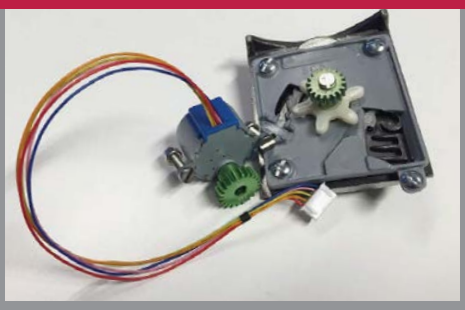
We remember going to the shops as kids and looking at the sweet-dispensing machines. We only needed ten pence to try to get some! Unfortunately, our mother had our health in mind (and probably better sweets in a hidden location anyway), so it was a very rare occasion to actually get any sweets from one of these dispensers. While these machines seem to have all but disappeared, mini versions of them are now popular, offering gifts and trinkets. Inserting coins, though, is old-hat, which is where Lauri and his team at Esmes come in:

"We are huge fans of Jelly Bellys, and a while back we ordered ourselves a small candy machine. We needed to use coins to give us candy, and putting the coin in the slot became boring after a while. So we decided to modify the machine so that we could use a mobile phone to trigger the mechanism, since using coins is so 2014. Complete overkill, but why not! With Raspberry Pi, we could host the mobile front-end on the device itself and interface with a stepper motor controller."

With that, the CandyPi was born: the mobile-phone controlled candy dispenser, with no need to nag your mum. They thought about the concept for a while, but it wasn't until they came across the right gears and screws and other components needed that they decided to actually give it a go.

HOW TO GET A JELLY BEAN

>STEP-01
Find Candy Pi
With CandyPi set up and connected, all you need to do is navigate to the browser interface for the actual machine itself: it makes sense to do this via your smartphone.



>STEP-02
Push the button
No complicated controls or levers to push: all the machine can do is dispense random jelly beans when asked. So there's a single button to press to get to your sweets.

>STEP-03
Eat your sweets
The Raspberry Pi and technological part is done. Before you lies a bounty of sugared and/or sweetened goods. Grasp them with your hand and feast on your victory.

> "When we run the motor, it makes the original dial rotate. Rotating that dial moves the mechanism, which drops the candies to the 'chute'"

Lauri explains how the CandyPi works: "The motor is attached to the rotary dial via gears. When we run the motor, it makes the original dial rotate. Rotating that dial moves the mechanism, which drops the candies to the 'chute'. We decided not to reinvent the wheel and it seems that it was a good idea, since the mechanism is quite stable."

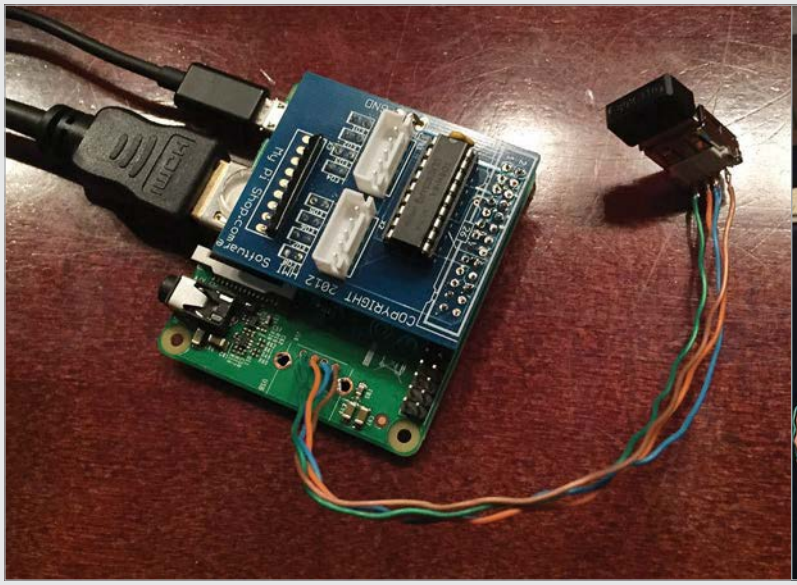
The setup includes the original machine, a stepper motor, some specific gears to get the mechanism working, and a connected Pi (with Wi-Fi) to power the whole thing and provide the web interface. Why, specifically, did they use a Pi?

"Raspberry Pi is the favourite embedded platform for us," Lauri says. "Many available tutorials, GPIO, small form factor, cheap price, [and the] possibility to run Linux are huge pros. With Linux, we could host the web front-end easily on the device itself (nginx), and the stepper motor interface was easy to do in Python."

While Esmes won't be selling the CandyPi, pre-assembled or as a kit (it was just a fun little project, after all), the build process is well documented on their website (**sneek.co/blog/candypi**) if you want to give it a go. As for the future of the project, they might add a more powerful motor to make the CandyPi work a little better, but otherwise they'll probably just use it to get sweets whenever they're a bit peckish in the office.

Below left Everything is such a tight fit that the protruding USB port had to be removed and reattached via wire

Below A slight tear-down of the device reveals that it's mainly the candy dispenser and the Pi



MCMASTER FORMULA HYBRID

Members of the McMaster Formula Hybrid team worked together for over 18 months to build a racing car capable of competing in the competition. **formulahybrid.ca**

- Instead of using traditional dials, McMaster's Pi-powered racing car uses an LED display to give the driver information about his speed, revs, and lap time
- The Raspberry Pi 2 is housed behind the dashboard, where it is sealed in to protect it from the elements. Telemetric data is sent wirelessly to the team in the pits
- The front wheels are powered by 15kW electric motors; the rear wheels by a 250cc petrol engine. Sensors throughout the car provide data on how well the hybrid engine is performing

Quick Facts

- The car is quarter the size of a regular Formula 1 car
- The front wheels are powered by an in-hub 15kW electric motor
- The rear wheels are powered by a 250cc KTM SFX motorbike engine
- It can do 0-100 km/h in three seconds flat
- It has a top speed of 150km/h

MCMASTER FORMULA HYBRID

McMaster University needed a smart telemetric system to get its hybrid racing car onto the winner's podium, and the Raspberry Pi 2 provided everything the team could ask for...

When the engineering students from McMaster University in Canada started working on their entry for the 2015 Formula Hybrid and EcoCAR 3 competitions, they knew they'd need more than raw muscle to get a podium finish. Not that their race car lacks muscle: it packs a 15KW in-hub motor for the front wheels, and a 250cc motorcycle engine for the rear wheels.

The challenge was bringing all the technology together and keeping the car going. McMaster's secret weapon was a Raspberry Pi 2, used to gather telemetry data and send it to the team on the trackside.

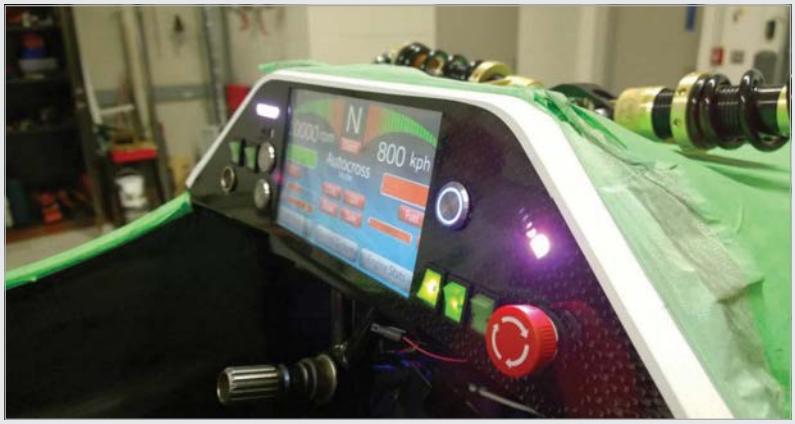
We caught up with Jonathan Moscardini, LV lead for the electrical division of McMaster University's Formula Hybrid and EcoCAR 3 teams. He tells us that the Raspberry Pi has been crucial to the car's success. "We're a pretty big team, between 70 and 90 people. But our project is entirely student designed, built, and tested. The students do everything."

McMaster has been entering the Formula Hybrid and EcoCAR 3 competitions for several years now and had pretty much perfected the physical car: "It has a very clever monobox," asserts Jonathan. "It's one big fibre tub."

Physically the car was fine, but that's not enough to win a race. "We do a lot of electronics in the car," says Jonathan, "a lot more than we need to" and that's where the Raspberry Pi steps in. "What the Pi does for us is handle all of our communications. Essentially, it's both the dashboard computer and our team radio. It also gives us a few new features along the way, simply because it's so powerful."

Good telemetric data is essential when building a race car. "We have live up-to-date information about everything that's happening in the car," says Jonathan. "We use a wireless adapter known as a Bullet [BULLET WirelessHART Adapter], which is built for a variety of outdoor uses. We have one at either end: one in the car and one in the pits."

The telemetry data enabled McMaster students to analyse the car as it raced around the track, which helped them fix a variety of



Above The Raspberry Pi 2 sits behind the dashboard and powers the display. When the car is racing, it automatically sends telemetric data to the team in the pits

engineering challenges. "We were having reliability issues," divulges Jonathan. The team used the Raspberry Pi with various sensors to analyse parts of the car as it went around the track. "When we installed a Raspberry Pi 2 it made life a lot easier."

While they initially installed a Raspberry Pi 2 to gather telemetry data, its use soon expanded to other aspects of the car. McMaster engineering students quickly realised they had enough power to build an electronic dashboard. "We also installed the PiCam [Raspberry Pi Camera Module]," says Jonathan, to get first-person video recording from the car. The Raspberry Pi was sealed behind the steering wheel, to keep it well protected. "You really need to be able to seal and waterproof the computer," Jonathan tells us. "You need to keep it away from the elements".

So what if you want to start integrating your Raspberry Pi with a track car? "Planning and testing is one of the biggest things," advises Jonathan. "Race cars are not the place to be trying things out. As far as implementing goes, planning and testing is really important because it's a lot easier to work on things when you're not in the car. So you really need to have everything installed before you start."

> "We use a wireless adapter known as a Bullet, which is built for a variety of outdoor uses..."

Above The car is much smaller than a Formula 1 car, but with a top speed of 150km/h it is much more than a standard kart

INSTALLING A RACE-CAR COMPUTER

>STEP-01
Raspberry Pi

The Raspberry Pi is connected to custom PCBs built by Advanced Circuits (**4pcb.com**), and a stock video display (picked up from eBay). The Raspberry Pi Camera Module is also added (to provide video recording of the race circuit).

>STEP-02
Sealed in

The Raspberry Pi electronics kit is located behind the steering wheel and is sealed in using Loctite 243 and Henkel (**henkel.com**) adhesives, sealants, and functional coatings. This protects it from the elements. The display is mounted on the dashboard behind the (detachable) steering wheel.

>STEP-03
Trackside telemetry

The Raspberry Pi is hooked up to a BULLET WirelessHART Adapter, which communicates with another Bullet at the trackside. Data captured by the Raspberry Pi is sent wirelessly to the team working on the car.

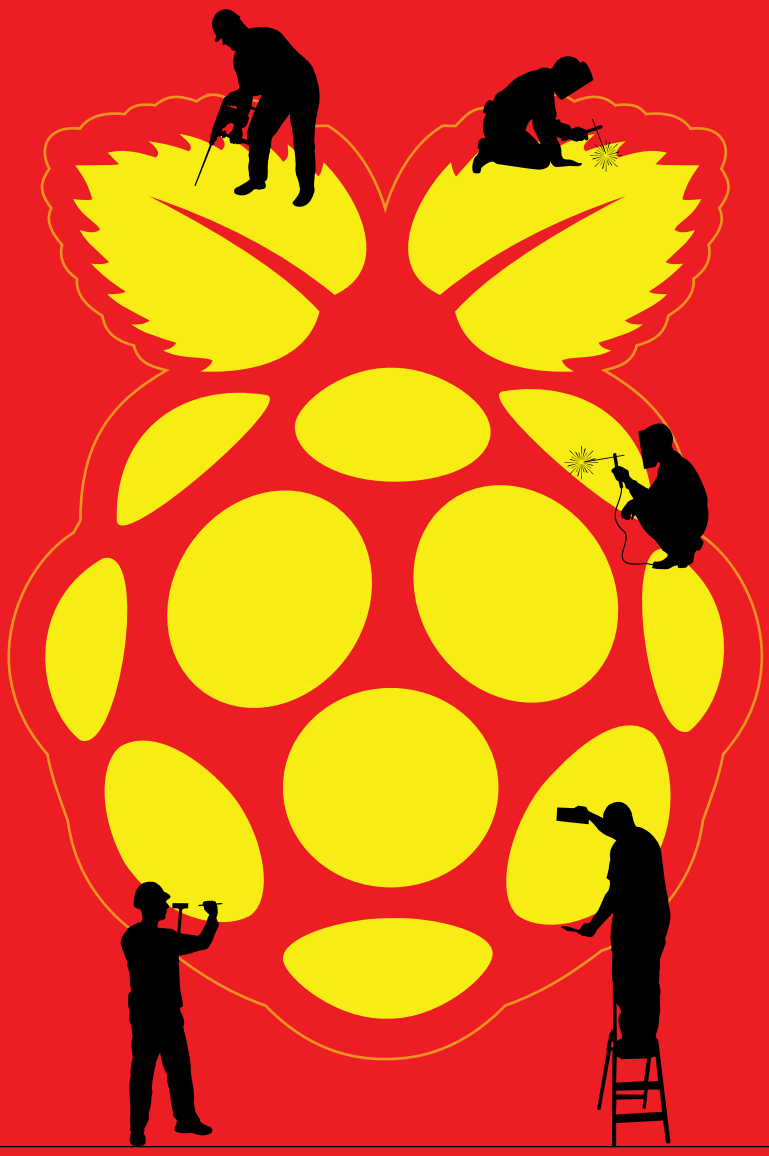
AMAZING PI PROJECTS

8 imaginative creations that *innovate* and *inspire*

While the Raspberry Pi is to all intents and purposes a simple Linux-powered PC, it's sometimes difficult to convince people on the street that they merely need to connect a standard TV or monitor and keyboard and mouse to use it like any regular computer or laptop. They struggle to comprehend that something so small and strange in appearance is anything other than alien in origin. The difficulty is usually intensified when you talk to them about all the incredible things people do with the Pi. Sending Raspberry Pis to the edge of space to take pictures of the curvature of the Earth, configuring them to adjust the temperature or control the lighting in their home… how can they hope to even turn it on, let alone take control of their home?

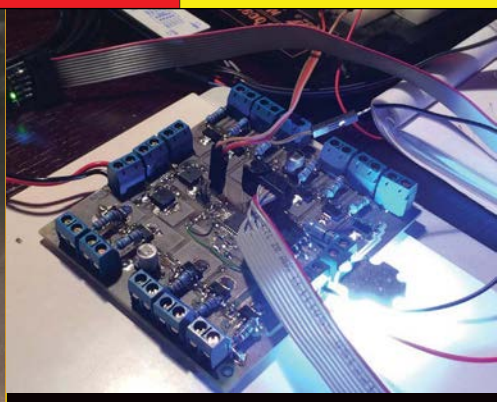
Amazingly, turning on a light is usually the first step for anyone wanting to make use of the Raspberry Pi's extensive physical computing repertoire. All it takes is a few lines of code and a basic grasp of elementary electronics (if that). Once they've cracked it, though, everything else is common sense and trial and error. One minute a tiny LED light is blinking; the next, your robot is deciding whether to turn left or right.

Once that light bulb moment occurs and the initial culture shock has subsided, the final blocker is usually indecision. What do you do with a credit-card-sized PC that can go anywhere and do anything? That's the focus of this feature – ideas, inspiration and advice from some of the community's best hackers and makers.



Project Aquarius

Two brothers recreated the Amazon rainforest in their home, including indigenous flora and fauna. Weather, sound, and lighting effects produce accurate day and night cycles...



Above It uses a custom PCB to aid control

Above It simulates light, sound, and weather

Wojciech 'Poopi' Lazarski is a technology evangelist from Poland. Poopi was once a developer in the demo scene, way back when the ZX Spectrum and Amiga were the best show in town. These days Poopi and his brother-in-law Piter are more concerned with recreating the Peruvian rainforest – specifically, a region called Rio Tahuayo – in their home with their custom-made paludarium.

"One day my brother-in-law showed me a video on YouTube and said it would be cool to have something like this," explains Poopie. "He said that making the tank wouldn't be a problem, but creating [the project] would be impossible for him. I asked him to give me time to digest it and a week later I said yes, but that I'd like to do it my way. I didn't just want to add lighting effects so I started thinking about a more sophisticated solution."

At the start, it controlled lights to create the illusion of night-time with the phases of the moon, and sunrises and sunsets with all the associated colour effects. Everything is accurately calculated based on the target location. Later, wind simulation was added using two fans, and then mist and rain effects using solenoids. Since delicate flora are present, careful temperature monitoring is also of utmost importance.

You can see Project Aquarius in action on YouTube (**youtu.be/FeS5zqL8frk**), simulating various times of day and weather conditions.

Simulating THE AMAZONIAN RAINFOREST

Poopi and Piter's paludarium is controlled by Raspberry Pi and four ATmega 168Ps. Here is its extensive feature list:

- 6× independent sections of halogen lights
- 27× independently controlled 1W LEDs for various effects
- 3× independent 3W RGB LEDs for ambient colour effects
- 3× independent 3W LEDs for thunder and moon simulation
- 3× independent 10W LEDs for paludarium lighting
- 2× independent fans for wind simulation
- 3× fog generators
- 2× independent solenoids for rain control
- Temperature monitoring

All lights are fully dimmable, and Poopi and Piter also have full control over fan speed.

One Controller to Control Them All

Retro gaming is a popular avenue for the Raspberry Pi, but few have done it with as much flair and precision as Brian Corteil…

All images courtesy of Alex Eames www.raspi.tv

Brian Corteil is a Cambridge Raspberry Jam regular and founding member of the Cambridge Makespace (**makespace.org**). Brian is well known in the Raspberry Pi community for the remarkable quality of his projects and their stunning attention to detail.

"I wanted to make something my two boys could set up and play with, without me being around," he tells us about his distinctive retro gaming project. "It's also a great honeypot to draw people in to my table when I show off at Maker Faires and Raspberry Jams [see sidebar on the next page]."

Brian's retro gaming console-cum-arcade controller is certainly that, and it should be of little surprise that it's just as neatly presented on the inside as it is on the outside.

"I used one of the laser cutters at the Cambridge Makespace to cut [and] engrave the box and back-plate. I also used Inkscape (**inkscape.org**) to design the back-plate and modify the box design from Steve Upton's box-generating script (**officeoffairetrade.com**). The RetroPie SD card image takes care of the software side of things (**bit.ly/1lYZkDg**)." digitalLumberjac's arcade joystick driver (**bit.ly/1xHKQCz**) also proved to be the best way for Brian to interface the joystick and buttons to the Raspberry Pi's GPIO pins.

Brian keeps an excellent log of his many Raspberry Pi projects, including the building of robots, on his website: **corteil.co.uk/Blog**.

Make it HAPPEN

Brian is one of the founding members of Cambridge Makespace (**makespace.org**), which gives him full access to the tools and equipment (like 3D printers and laser cutters) he uses in his projects, not to mention the expertise of fellow members.

While trailing its US equivalent, the UK maker scene is growing fast, with new locations popping up all the time. Try searching online for 'maker spaces' or visit **hackspace.org.uk**.

There are also numerous online services that will do the 3D printing or laser cutting for you and send the results by post.

Digital Zoetrope

Brian's second project is a deliciously modern take on a pre-film animation device that produces the illusion of movement with still images...

"I started the Digital Zoetrope over Christmas 2014 and I'm still refining it," Brian Corteil tells us about his remarkablae animation device. "I was inspired from the work of Eadweard Muybridge, an early pioneer of high-speed photography [who] was the first person to show that a horse takes all of its feet off the ground when it's running. Researching his work led on to Zoetrope and I thought it would be a wonderful project."

As with his retro-gaming console, Brian utilised the laser cutter at Makespace, and Inkscape for the computer-aided design work. Despite bundling a wealth of technology, his Digital Zoetrope is moved by hand. Like the original designs, you spin the device and look through the slats to see movement in the still images as they rotate.

Since his project uses 12 OLED displays with the Raspberry Pi, it's actually possible to update the frames in real-time, so you could watch an entire film if you wanted. Moreover, using the technological trickery, it's possible for two people to view entirely independent animations when looking into the Digital Zoetrope from different angles.

Organise
YOUR OWN JAM EVENT

Raspberry Jams are a great way to get together with like-minded hackers and makers. They're also the best place to find inspiration and help to build amazing Raspberry Pi projects. You can find a Raspberry Jam near you by pointing your web browser to **raspberrypi.org/jam**. Can't find an event near you? Don't worry – it's easy to set up your own! Here are a few tips from Cambridge Raspberry Jam (**camjam.me**) organisers Mike Horne and Tim Richardson:

1. **Find a partner and team:** Don't try to organise everything on your own. Find a core team of two or three people who can meet regularly and gather helpers on the day.
2. **Get a venue:** It must be accessible with good parking and transport links, and have the space you need to do what you want. It could be as small as your kitchen table or as large as a local community or sports hall.
3. **Get talking:** Everyone likes 'show and tells' and talks, so start with those. You can add things like workshops, sponsorship and vendors later.
4. **Walkthroughs:** In the run-up to every Jam, put yourself in an attendee's shoes. Imagine walking in through the front door for the first time. What do you see? What do you need to do?
5. **Questionnaires:** Just after the end of the event, send out a questionnaire to all attendees to find out what was good and what could be improved. Use it to make your next Jam even better!

Internet of Things Chessboard

16-year-old Ben James spent two months designing and building his IoT chessboard, and three months developing the software. The results speak for themselves...

Ben James is a home-educated Raspberry Pi enthusiast who plans on becoming a robotics engineer when he's older.

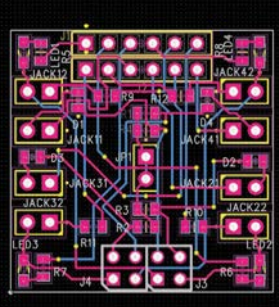
"The inspiration for this project came about one afternoon when I felt like playing a game of chess, but there was no one to play with. I would have gone online and played, but that's a 2D diagram view that feels really unintuitive, so I decided to make a real physical chessboard that could play online."

To play, your opponent requests a new game on the chessboard's own website. "When they make a move by dragging a piece on the web interface, the relevant piece and square will flash on the IoT Chessboard. Whoever's sitting in front of it will move that piece to update the board, then will make their own move. The board detects this, then sends that move back to the opponent's web UI, where the piece is moved on the screen."

According to Ben, the biggest challenge of the build was desiging the circuitry. "Each of the 64 squares on the chessboard has to have an individually addressable LED on its surface, an LED on the piece sitting on top of it, and a detection circuit to determine whether there is a piece on it."

Instead of controlling an unwieldy 192 input/outputs, the project utilises multiplexing. "Each group of four squares is handled by a PCB, then the 16 PCBs are connected horizontally and vertically in groups of four." While it makes I/O much more managable, it required over 2,000 solder joints and made troubleshooting issues a gargantuan task. "It was more efficient in the end, though, cutting the I/Os down to under 60, and it meant the software was a lot easier to write."

See it in action at **youtu.be/bWeObKths-I** and learn more at **engineercheer.wordpress.com**.

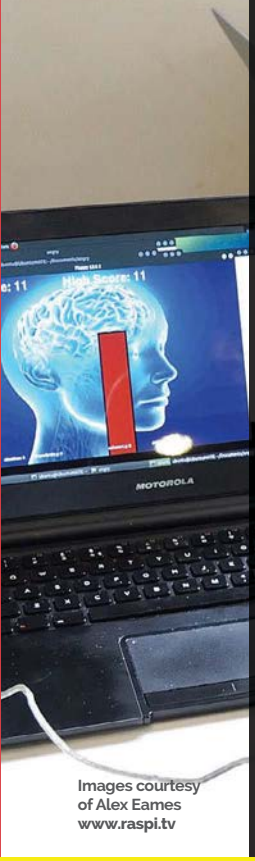


Above The circuitry is impressive

Above Ben had to solder 2,000+ joints

Flappy Brain

Control a Flappy Bird clone with your brainwaves? It's mind-blowing stuff...



Images courtesy of Alex Eames www.raspi.tv

Albert Hickey is a coordinator of the Egham Raspberry Jam. His latest project uses a MindFlex band (a Mattel game from 2009) to read your brainwaves and convert the data to allow you to play a version of *Flappy Bird* with nothing but the power of your mind.

In a recent interview with Alex Eames on RasPi.TV (**bit.ly/1FEvmNK**), Albert explained more: "The headband features a little piece of metal that you put against your temple and it reads your brainwaves." It's connected to an Arduino that converts this information into usable data that's then passed over to the Raspberry Pi and used as controller input for the game. "Just search online for 'MindFlex Arduino Hack' and you'll come up with a page that explains how do it."

Albert used Pygame to create *Flappy Brain* on the Raspberry Pi and, much like the game on which it's based, you need to move the protagonist (in this case a tiny brain) past a series of obstacles.

Unlike *Flappy Bird*, though, the brain automatically moves up the screen – to move it down (and past the obstacles), the user needs to concentrate; Albert demonstrates by doing mental arithmetic to move it down. To allow it to move up again, Albert blinks repeatedly to break his concentration. Amazing! Learn more at **winkleink.blogspot.co.uk**.

PiWars COMPETITION

If you're looking to find amazing Raspberry Pi robots, look no further than Pi Wars (**piwars.org**) – it's an annual UK-based competition organised by the makers of Cambridge Raspberry Jam (**CamJam.me**) to bring home-made Raspberry Pi robots together to compete in a number of challenges and categories.

You don't need to be a robotics expert to attend, but check out this excellent video by RaspberryPiIVBeginners from the 2014 event to get your creative juices flowing: **youtu.be/PQy0DZzQJIY**

Above One of last year's smallest bots

Pi WARS
The Raspberry Pi Robotics Challenge Competition

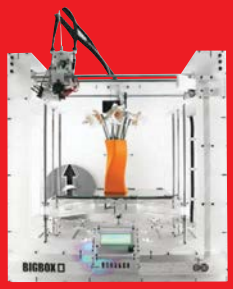
Crowdfunding SUCCESS

As we're seeing in this feature, Raspberry Pi projects come in all shapes and sizes. If you've got a really big idea and you'd like to take it to the next level, you might want to try crowdfunding it on a platform like Kickstarter. Here are two examples we've got our eyes on...

NATUREBYTES WILDLIFE CAM KIT

The Naturebytes Wildlife Cam Kit incorporates a Raspberry Pi and a Raspberry Pi Camera Module that's triggered by a PIR motion sensor, to help just about anyone take candid shots of birds and animals in their natural habitat. The team's Kickstarter campaign (**kck.st/1Ndsg99**) proved a great success. Find out more about this heat-sensitive camera trap at **naturebytes.org**.

BIGBOX-3D



BigBox-3D incorporates two of our favourite things in the world – Raspberry Pis and 3D printers – so we're naturally rather excited about the project, the subject of a successful Kickstarter campaign.

It's the brainchild of E3D, a British engineering company with extensive 3D printing experience, and LittleBox, which has previously found success on Kickstarter with the MicroSlice mini laser cutter.

The idea of the BigBox-3D is to make it easy and affordable to achieve high-quality 3D prints. The project uses OctoPi (a cloud-based print solution) and the Raspberry Pi Camera Module for a remote live view of your prints as they happen. Learn more at **bigbox-3D.com**.

FERRAN FÀBREGAS

A Spanish computer scientist who works at a consultancy for urban ecology by day, by night he is an active member of the Barcelona maker community.
lifebox.ferranfabregas.info

LIFEBOX

Quick Facts

- The project took two months to complete
- Ferran learnt how to laser-cut wood for the project
- It is apparently not inspired by Conway's Game of Life
- The code is available on GitHub: **bit.ly/1T8VKtC**
- There's a mini version in development

Creating artificial intelligence on the Pi may sound like the start of the robot uprising, but the LifeBox isn't taking any chances, and has imprisoned them in lights...

There are so many different type of light-display projects on Raspberry Pi that you could be forgiven for thinking that the LifeBox was just another neat little programmed series of LEDs. This would be a huge mistake to make because the little lights are a lot cleverer than you could imagine: they're alive. Well, sort of. At the very least, they have been programmed with behaviour. Instead of launching into a thesis on when artificial intelligence can be classed as alive, we're going to concentrate on the LifeBox itself. Here's what it says on the side of the box:

"In this box live two pixelic entities, the blue and yellow species. These two species compete to survive and reproduce, feeding with the white mana that grows under [their] feet.

"Each species has eight configurable variables that can change their behaviour. The white mana also has five parameters that determine their behaviour and also rule the future of the two species that feeds.

"Learn the basic concepts of programming and biology being the god of these entities, varying all the parameters and seeing the consequences of your actions in the LifeBox!"

Its creator, Ferran Fàbregas, explains it to us in a less poetic manner, but one that makes more technical sense.

"LifeBox, in short, is a virtual ecosystem simulator on a 32×32 RGB led panel," Ferran tells us. "It's composed of two species that compete for the resources (mana) to get energy, survive, reproduce, and grow.

"Both species and the mana (which actually acts as a different species itself) have a user-defined parameterization that allows [the user] to change their behaviour and see the consequences on the panel, acting as a god of the virtual ecosystem."

> "Learn the basic concepts of programming and biology being the god of these entities"

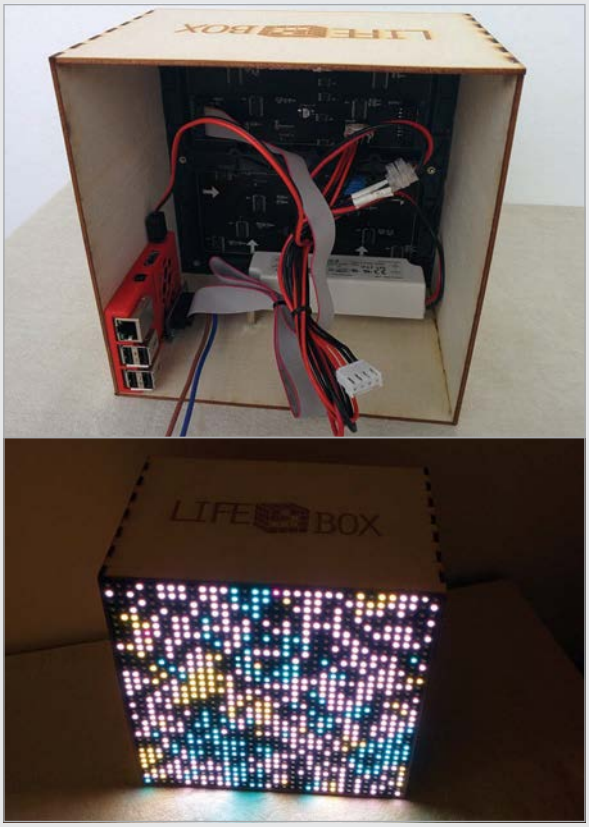
Ferran's god complex has been with him a while, as well as the interest in simulating these kind of ecosystems:

"Since I was a child, I was attracted to robotics and the possibilities of simple life simulations. I programmed some basic life simulators software before, but one day I found a shiny, beautiful, and cheap 32×32 RGB LED panel at a fair and I decided to create the LifeBox. I was captivated by the idea of seeing the evolution of a programmable ecosystem on a beautiful box on my dining room [table], like other people can enjoy with a fishbowl."

Inside the box is just a Pi, the LED panel, and a driver to connect the two. Because of this, the code is split up into two sections: one part to control the driver, and one part that controls the simulation.

"The driver is based on the great work of the original C++ driver by Henner Zeller and a reimplementation in C by Peter Onion with slight modifications," Ferran explains.

"The simulator is build in C and the main goal is to maintain it as simple as possible, so anybody can change not only the species parameterization but also the simulation algorithm itself (although this is not the objective, because it can result in an unusable LifeBox situation)."



The whole thing is configurable via specific files, and you don't need to recompile it each time either. The code is still being improved, and you can find it on GitHub (**bit.ly/1T8VKtC**), or, if you really like the project, you can look at getting one of the LifeBox kits by checking out the crowdfunding page at: **lifebox.ferranfabregas.info**.

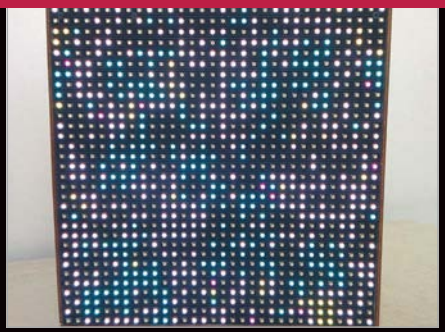
Top It looks messy inside, but it's generally quite a simple setup once you break it all down

Above The most interesting night light you've ever used: maybe it will be this decade's lava lamp?

BRING YOUR RASPBERRY PI TO LIFE

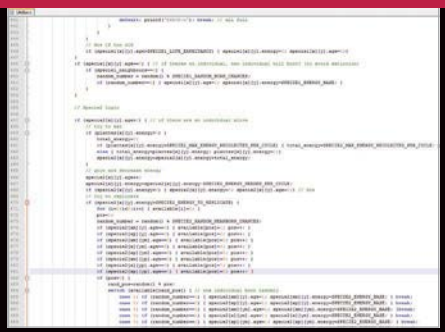
>STEP-01
Set the parameters

Delve into the config files and define how your life forms and mana work: life expectancy, energy requirements, reproductive rates, and so on.



>STEP-02
Start the experiment

Power on the LifeBox and the simulation begins. The simulation will run for as long as you supply it with power, and any resets will start it from the beginning.



>STEP-03
A new beginning

Once you've completed the experiment, it's time to start again. Go back into the parameters and reprogram the life forms and mana.

ROBERT DOERR

Robert Doerr runs Robot Workshop and has built robots that compete in *BattleBots* (the US version of *Robot Wars*).
robotworkshop.com/robotweb

Quick Facts

- PiPlateBot took two weeks to design and build
- Turtle robots were first used in the 1940s
- They were developed because few computers had monitors
- They're often controlled using a language called Logo
- They're named after *Alice in Wonderland*'s Mock Turtle

PIPLATEBOT

Drawing inspiration from the turtle robots of old, **Robert Doerr** created a Raspberry Pi robot with the components hidden inside a Bud Pi Plate case. Say hello to PiPlateBot...

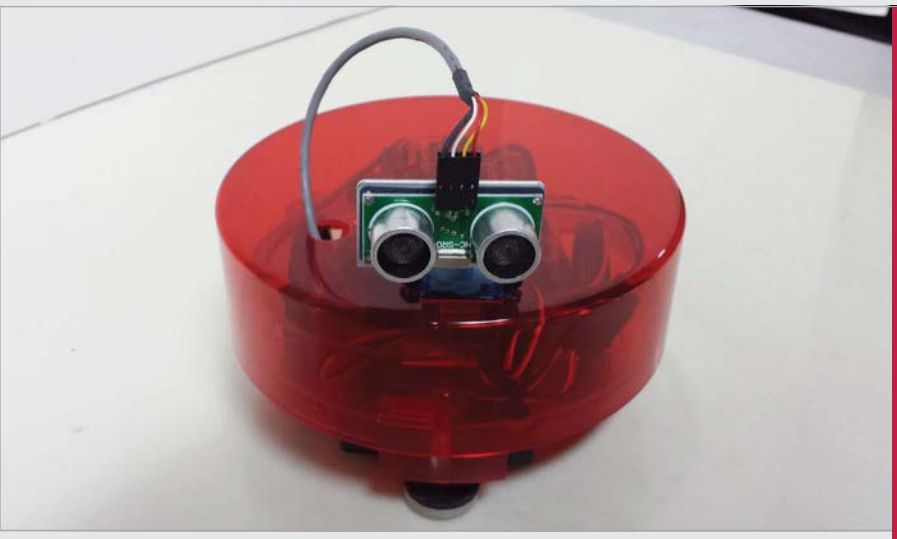
Robert Doeer is no stranger to building robots. He's the owner of Robot Workshop, an organisation dedicated to restoring classic robots, and has entered two robots into *BattleBots* (the US version of *Robot Wars*): Crash Test Dummy and Crash Test Junior.

But this is no battle droid. When Robert saw a Bud Pi Plate case (**budind.com**), he was struck by how similar it looked to the turtle-style robots used to train computer science students. It was "definitely [like] the early turtle robots like the Terrapin Turtle [ones] and the early Tasman Turtle robots," says Robert.

The Pi Plate's circular design enables easy access (you twist off the top) and there is space inside for additional components. Robert immediately decided to see if he could place a Raspberry Pi, along with all the parts required to build a moving robot, inside the case. "It is the only Raspberry Pi-based robot that I know of built using an off-the-shelf Raspberry Pi case," he claims. "I tried to use as many Raspberry Pi-type products in the construction as I could."

"Getting everything to fit was the biggest hurdle," says Robert. He cut two rectangular holes in the base of the Pi Plate enclosure, and glued servos to the bottom of the case. An EZO power bank sits on top of the Raspberry Pi and RoboPi boards, and works as a battery. Finally, a USB WiFi adaptor enables wireless communication via SSH. "Those that have seen it really like the robot and ask where I bought it," Robert says with pride. "The completed robot really looks like a finished project, so they assume it may be sold in stores."

The powerful RoboPi board (**mikronauts.com**) is an important component. "The Raspberry Pi is great at the high-level thinking, while the Parallax Propeller chip on the RoboPi board is a great I/O controller for offloading all the real-time tasks."



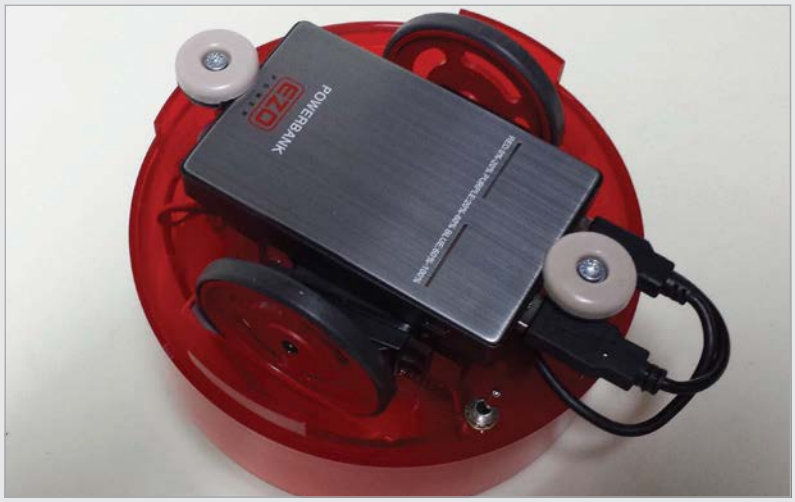
Above The finished product is a friendly-looking turtle-style robot that can sense objects in front of it

> If you have an idea for a project, just go for it! You'll never get it done just thinking about it

"I have been programming the robot in C," Robert tells us. "The RoboPi controller has libraries available for both C and Python. Eventually it would be fun to write a Logo interpreter so it could also use Logo and emulate the early turtle robots."

Finally, Robert added an HC-SR04 sonar sensor (ultrasonic transducer) so the PiPlateBot could measure objects directly in front of it. "[It] was built over the course of a couple weeks during my spare time in the evenings," he says. "It could have been built in a few evenings, but I had to wait for some parts to come in." There is still a bit of room left on the robot for additional sensors like contact bump, IR line following, or even a I²C compass.

Having created, and rescued, countless robots, Robert has good advice for budding robot builders: "If you have an idea for a project, just go for it! You'll never get it done just thinking about it. Even if the first iteration doesn't work out, you can always change it along the way and you'll learn a lot as you go."

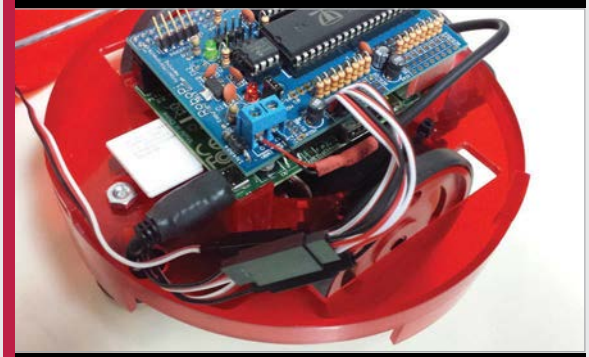


Above A standard smartphone battery pack is squeezed on top of the boards and inside the case

BUILDING THE PIPLATEBOT

>STEP-01
Cutting and mounting

Holes cut in the base allow the wheels to fit, and glue sets the servos in place. Furniture gliders attached to the front and rear stop PiPlateBot from wobbling.



>STEP-02
Assembling the components

The Raspberry Pi is fitted and a USB WiFi adaptor is connected. The RoboPi board is stacked on top of the Raspberry Pi board. A smartphone charger provides power.

>STEP-03
Power and motion

The RoboPi comes with libraries for both C and Python that are used to control the servos. An HC-SR04 sonar sensor is fitted: this enables PiPlateBot to sense its surroundings.

GRAHAM GELDING

Graham is a software developer, working on Linux to make graphical interfaces with Qt. In his spare time he has a number of hobbies, including learning to be a blacksmith.
instructables.com/id/Coffee-Table-Pi

COFFEE TABLE PI

Quick Facts

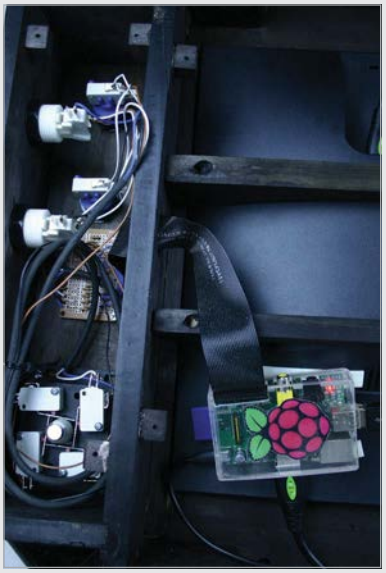
- The project took a few weeks to complete
- The Perspex over the screen can get scratched easily by kids
- *Donkey Kong* is Graham's favourite game to play on it
- It does actually get used for resting cups of coffee on
- There will be no kit, so use the Instructables guide

This fully fledged cocktail arcade cabinet, apparently masquerading as a coffee table, is one of those Pi projects everyone wants to do...

As you probably know, the Raspberry Pi, while an excellent educational tool, is big among the maker community. The kind of folks who like to create machines are always on the lookout for a tiny computer controller, one that can power their project with the smallest footprint. Along with the makers, any tiny computer released to the market also tends to attract a great deal of attention from the arcade gaming community, looking for the next thing to power their work-in-progress MAME cabinet. With this cross-section of interests for one device, it's a wonder why we haven't seen full-size arcade cabinets powered by a Raspberry Pi in every single issue of the magazine. Graham Gelding is one of the few to take on this task. Not only that, he's gone a step further and created the ultimate in classy, grown-up arcade gaming apparatus: a cocktail arcade cabinet (although it has a slight twist, as he likes to refer to it as his 'coffee table').

"It was an attempt to recreate the classic arcade cocktail cabinet," Graham tells us, "but in a way that can fit into a lounge room. It's also a way of introducing my kids to the games that I had growing up.

"I also wanted to try some woodworking and needed a project for my Raspberry Pi. The idea of

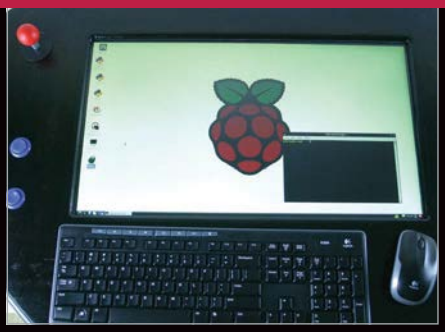


Above The insides are neatly arranged and very chunky, just like any good arcade machine should be

USING GRAHAM'S COCKTAIL CABINET

>STEP-01
Table or arcade machine

When the coffee table is off… it's still a table. To turn it into its true arcade machine form, you need to switch it on.



>STEP-02
Select your game

Like a lot of Pi emulation software, you're met with a selection of pre-loaded games that you've added yourself. Selecting a game will launch it.

>STEP-03
King of Kong

Play your game and have fun! Try not to put your coffee cup directly on top of the screen, and try not to knock it off while screaming at Mario.

mixing the two projects seemed perfect. I could make use of my experience with Linux, but also learn about woodworking."

He's not joking about the woodwork either. While you might think it was just a table that had been modified, or an existing cocktail cabinet that had been gutted, Graham made the whole table from scratch using pine from an old bookshelf, as well as installing the screen, arcade controls, and the Raspberry Pi itself that powers it. This made the most expensive parts of the project the LCD screen and some of the controls.

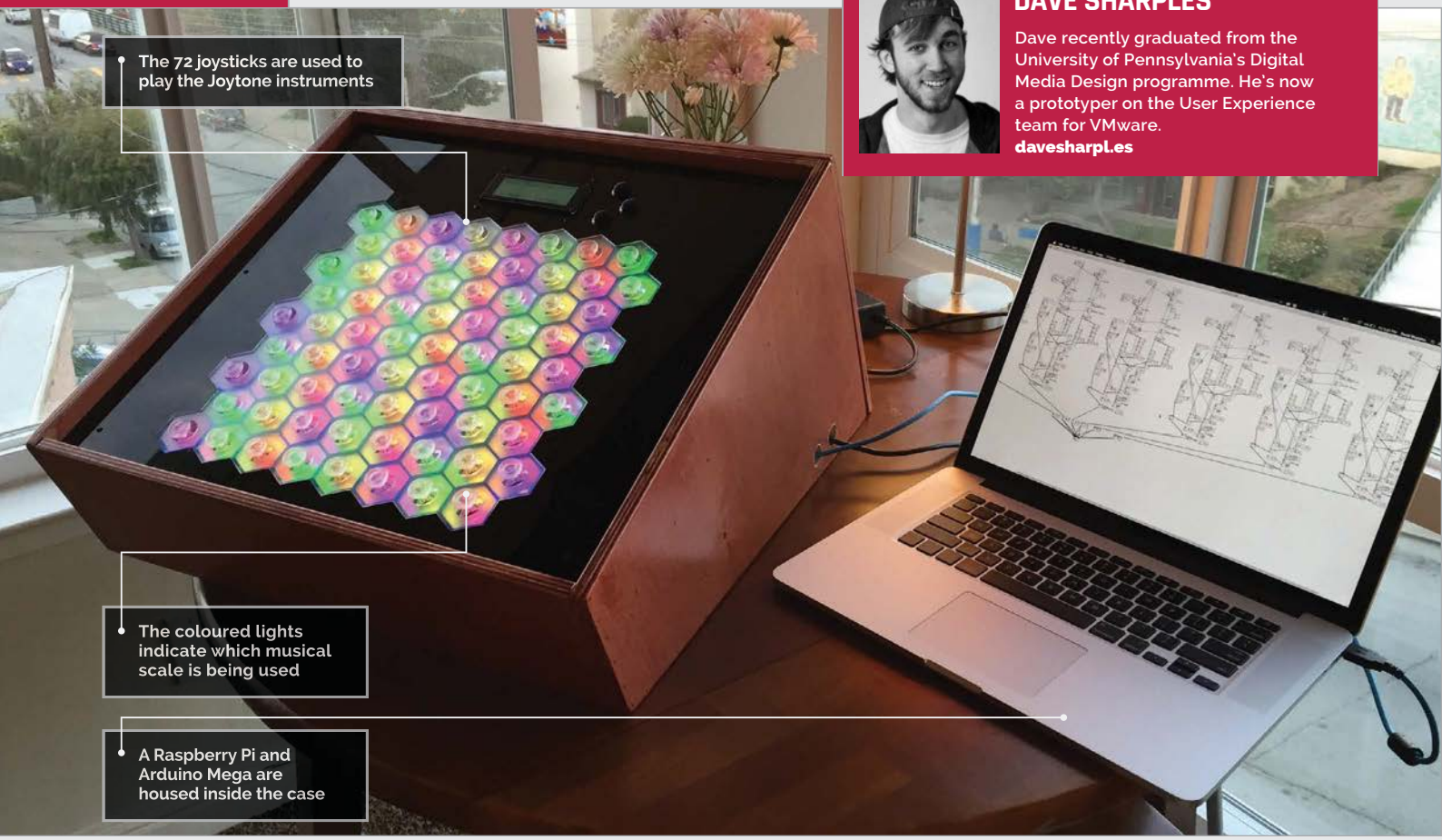
As well as those main components, Graham gutted some old PC speakers for their drivers and transformers, installed a sheet of clear Perspex over the screen to protect it, and did a lot of custom wiring and powering.

As for the future of the project, Graham is just happy with what he's made. However, he has plans for his next Raspberry Pi thing, telling us: "I would like to do something combining the Pi with the Oculus Rift virtual-reality headset."

The headset is finally coming to the consumer market, so marrying the two, if possible, would be a big leap in Raspberry Pi projects.

> "It was an attempt to recreate the classic arcade cocktail cabinet, but in a way that can fit into a lounge room"

Above The table is set up for games that use a vertical screen, so most arcade games during and before the 1980s



DAVE SHARPLES

Dave recently graduated from the University of Pennsylvania's Digital Media Design programme. He's now a prototyper on the User Experience team for VMware.
davesharpl.es

Quick Facts

- Dave used Express PCB (**expresspcb.com**) to do the board layout
- The Joytone survived six weeks in the kids' area of the Toronto Film Festival
- The parts for the Joytone cost around $1,050
- Around 16,000 people saw it at the Toronto Film Festival

JOYTONE

A unique musical instrument made from joysticks, lights, and powered by the Raspberry Pi. We talk to the Joytone's maker...

Every so often we come across a project so spectacular we have to share it. Joytone is one such creation. Designed and crafted by engineering expert Dave Sharples, the Joytone is a unique musical instrument played using an array of mini-joysticks.

"I've always wanted to be able to play a musical instrument," says Dave, "and a couple years ago I took a music theory class to see if that could help me learn piano."

Rather than learning the piano, Dave had a revelation about musical structures. "I became fascinated with the patterns associated with musical structures and realised how beautifully simple music can be," he reveals.

"Acoustic instruments are designed around the physical phenomena that produce sound," says Dave. "A violin is smaller than a cello because shorter strings make higher notes, not because it's convenient for the player."

Electronic instruments, like synthesizers and electric guitars, don't have to reproduce these limitations. There's no inherent reason for an electronic instrument to resemble the acoustic tool it is derived from.

So Dave set about creating a unique new musical instrument that made sense. "Joytone is a unique new musical instrument that features a hexagonal grid of 72 joysticks," he tells us. "The Joytone's hexagonal grid exposes musical patterns that are normally obscured by the quirks of common acoustic-style interfaces, like the white and black keys of a piano.

"Each joystick plays one note and the motion of the joystick affects the volume and character of the note," explains Dave. "The way the notes are distributed across the perfectly hexagonal grid means that all kinds of musical patterns become clear.

"Every major chord has the same finger shape, no matter what note you start on," he continues. "This is true of minor chords, scales or any other kind of musical structure, making [the Joytone] much easier to learn and play."

Above A laser cutter being used to cut the piece of acrylic.

> "There's no inherent reason for an electronic instrument to resemble the acoustic tool it is derived from"

Building the Joytone

The 'keys' of the Joytone are created using 72 joysticks. These are Xbox-style thumbsticks and were replacement parts that Dave found on eBay. They have clear plastic grips to let the LED lights shine through.

The lights (also sourced from eBay) are used to indicate which notes belong to the selected musical key. They are connected to NeoPixel (WS2812) strands from Adafruit (**adafruit.com**).

"Each joystick is really just a pair of potentiometers connected to a little plastic post," Dave explains. "One measures movement along the X-axis; the other measures movement along the Y-axis."

These are connected to an Arduino Mega via a series of custom circuit boards designed by Dave himself. "With two signals per joystick and eight joysticks per row, there are 16 analogue signals generated by each row of joysticks – a grand total of 140 for the whole instrument," he calculates.

"The Arduino only has 16 analogue inputs, so the Joytone makes use of multiplexers to handle all those analog signals. I designed custom circuit boards for the rows of joysticks, and at the end of each board is a 16-channel multiplexer.

"A multiplexer is like a big switch," says Dave. "The output wires from the multiplexers are connected to analogue inputs on the Arduino, [which] can set all the multiplexers to forward channel 0, then read all nine inputs. It then sets the multiplexers to forward channel 1, then reads all nine inputs again, and so on. Once it knows the positions of every joystick on the board, it can go through and figure out which ones are being moved.

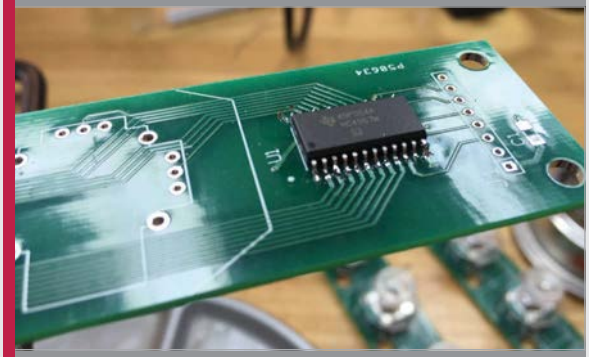
"For each active joystick," continues Dave, "the Arduino looks up the MIDI note it represents, then bundles that information up with the two values coming from the joystick sensor and sends a little MIDI message to the Raspberry Pi."

WHAT PARTS ARE USED?

>STEP-01

72 joysticks

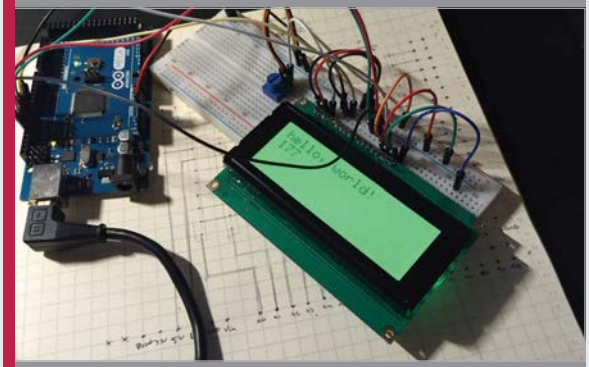
The interface of the Joytone comprises these mini-joysticks (picked up from eBay). The clear plastic grips enable light from LEDs to shine through.



>STEP-02

Multiplexer and Arduino Mega

An Arduino Mega and multiplexer are used to assess the stick positions. The Arduino only has 16 analogue inputs, so the multiplexer enables it to handle all 72 joystick signals.



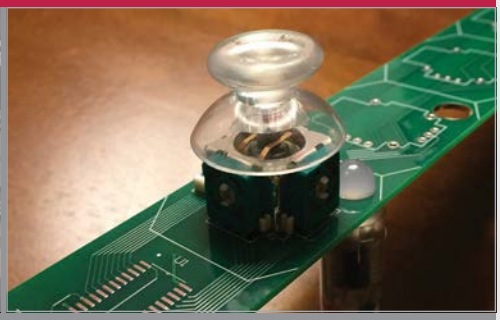
>STEP-03

Screen and Raspberry Pi

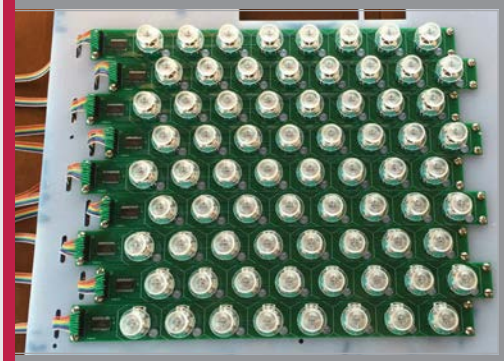
The screen is an RGB backlight positive LCD 20×40 from Adafruit (**adafruit.com**). The screen provides feedback to the user. The Arduino Mega sends MIDI information to the Raspberry Pi (which plays the audio).

PUTTING JOYTONE TOGETHER

>STEP-01
Custom circuit boards
At the heart of the Joytone are nine printed circuit boards (PCBs). These make the Joytone more reliable and easier to fix than earlier models. Dave signed up for TechShop in San Francisco to learn how to design PCBs.



>STEP-02
Fitting the joysticks
The clear joysticks and lights are fitted into the PCBs. Rather than use individual lights, the Joytone employs a strand of 25 Digital RGB LED Pixels (**adafruit.com**). These are easier to fit and more reliable than individual LEDs.



>STEP-03
Arranging the boards
Nine PCBs, each containing eight joysticks, are arranged into a square in this offset pattern. With two signals per joystick, there are 16 analogue signals generated by each row.

>STEP-04
Multiplexer and Arduino
Multiplexers are used to connect the PCBs to the Arduino Mega. They enable the Arduino to examine the position and movement of all 72 joysticks at once.

>STEP-05
Raspberry Pi
The Arduino looks up the MIDI note it represents and sends that information up to the Raspberry Pi, which then uses it to create the audio sound.

>STEP-06
Played with sticks
A sheet of acrylic is laser-cut into a honeycomb shape to hold the joysticks. A box holds all the equipment, and the device is ready to play. A single finger is used to play each stick and you can play up to six notes at once.

The software
The Joytone depends on PureData (**puredata.info**): "You can connect blocks with lines to direct the flow of data through a series of mathematical operations, to produce all kinds of strange and delightful behaviour."

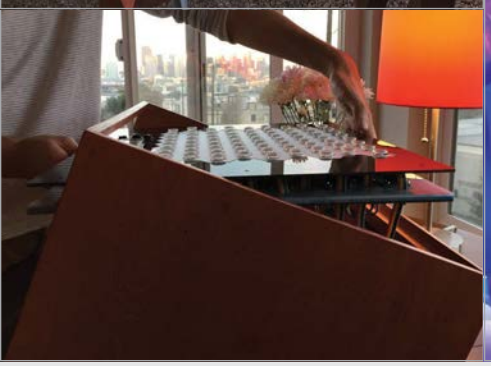
Dave built a PureData patch for the Joytone. "[It] receives MIDI messages, then unpacks them and passes the values inside into a group of blocks that produces a synthesized note," he explains.

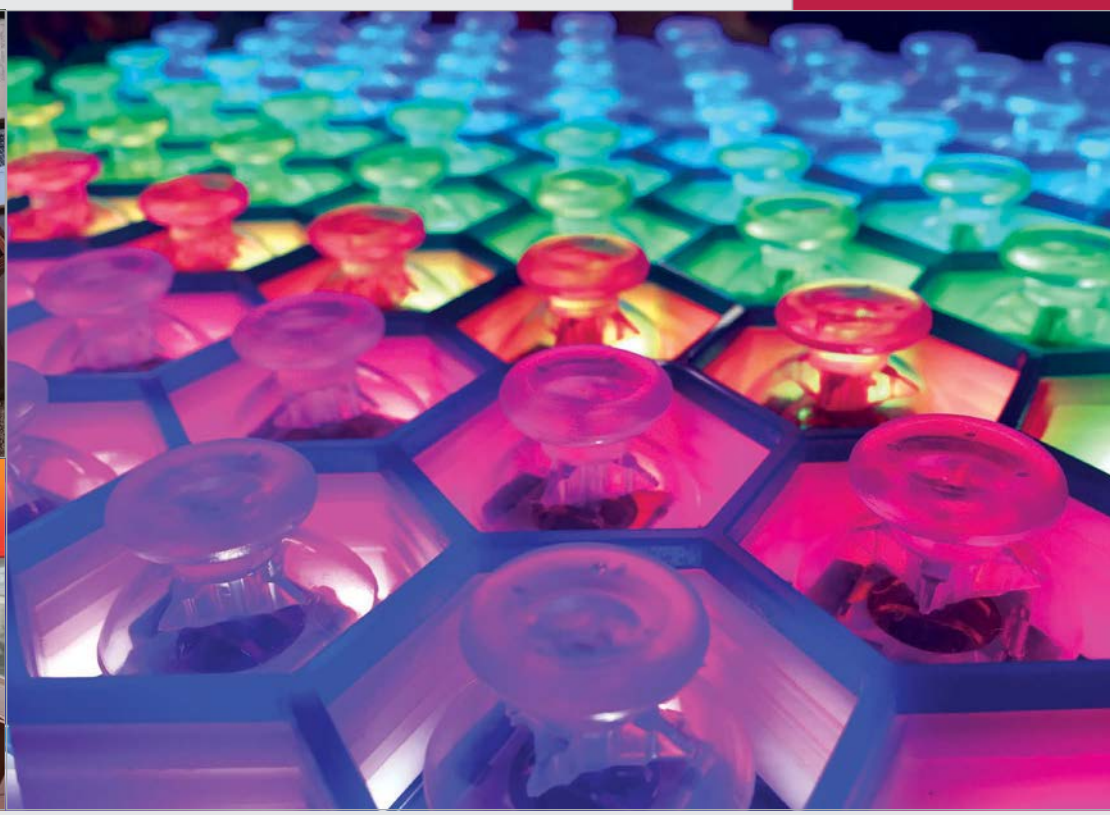
"The pitch is controlled by the joystick the user chose to push," says Dave. "One axis of the joystick controls the kind of waveform produced. It fades from a bright trumpet-like sawtooth wave to a darker triangle wave that sounds like a bell." The other axis of motion controls how flat or wide the note sounds by making a small tuning difference between the pair of oscillators that produces the note.

"All of that expressive potential is represented in the group of blocks in PureData," Dave tells us. "There are six of those groups in the patch, meaning the Joytone can play up to six notes simultaneously."

Playing the Joytone
While the idea behind the Joytone is complex, playing it is surprisingly easy. "My friend who helped me build the first one in school is a very talented musician," affirms Dave,"and the first time we got it working, he played with a couple of the joysticks, then paused and thought for a second, then immediately played a Bach fugue he was familiar with. It was an awesome moment of success after a string of very long nights."

Playing the Joytone is remarkably straightforward. The joysticks are played with a single finger. "It's easy to hold your hands as if you were typing on a keyboard," advises Dave, "and play many notes at once. The joysticks are pretty close together, so the player has access to a large musical range with pretty limited hand motion."





Integrating the Raspberry Pi

This build is the second iteration of Joytone and is the one that introduces the Raspberry Pi to the design. "The first Joytone had to function though a nightmarish rat's nest of wires inside," Dave recalls.

"In the two-brain design I used for the Joytone, the Arduino does all the analogue-to-digital conversion and the Raspberry Pi does all the audio synthesis. I like to use it on a Raspberry Pi because I can just tuck it inside the enclosure and focus completely on the instrument. I taught myself a little bit about PCB routing, and with the help of some friends they came out perfectly," he says.

Showing off the Joytone

Part of the inspiration for building a more powerful iteration of the Joytone was the Toronto International Film Festival (TIFF). Dave was asked to provide Joytone for part of the digiPlaySpace exhibition at the show.

"I had to design it to run continuously and withstand thousands of visitors over a period of six weeks," Dave says, "so I did some research at the Exploratorium in San Francisco to see how they build their exhibits.

"[The Joytone] saw about 16,000 visitors this year and I've been showing it around to my friends. Everyone seems to have a good time with it, and I'm curious to see what could happen with some serious practising. My favourite moment so far was a write-up from a six-year-old blogger who visited the TIFF exhibit.

"I'm glad that the Joytone is back from the TIFF exhibition because I can practise on it now. I only finished it a day or two before I had to ship it out! Even with the relatively short amount of practice time I've had, I can play almost any scale or chord with ease."

Despite the apparent complexity, Dave claims that the Joytone was pretty simple to build. It was "extremely tedious because there are so many joysticks."

It isn't a cheap project to make, either: "The parts for the Joytone cost about $600, most of which is for the lights and joysticks." Dave also spent an additional $450 on the custom PCBs, though he tells us he's got enough leftover parts to build another instrument.

"I have lots of plans for other musical inventions," he says. "I'm going to iterate on the joystick idea but investigate some other form factors. I'd also like to make something more compact and portable."

Making musical instruments

If you are interested in following in Dave's footsteps, it's worth knowing that you don't have to build a device as complex as the Joytone. "Anyone could wire up a joystick to an Arduino and start sending MIDI notes to their computer," says Dave, and "you could make a simple musical instrument in an hour or two."

Budding musical creators should just "go for it," he tells us. "There's a great community of people building new musical instruments, and lots of inspiring work already documented. PureData is a wonderful tool to experiment with quickly, and the Raspberry Pi is a friendly platform for musical inventions. Learning a little music theory is also really helpful."

Above Pushing a stick up controls the kind of waveform produced; left to right controls how flat or wide the note is

Top left The Joytone is a spectacular-looking modern musical instrument

Left A sturdy wooden case is used to house all of the components

RASPBERRY PI NOTEBOOK

Adafruit's star makers, the **Ruiz brothers**, are back with another stunning handheld Raspberry Pi project...

Here's a brilliant project for you or your family to test your hacking and making skills with this weekend. This beautiful retro-styled mini-notebook, built by Adafruit's Ruiz brothers (**bit.ly/1MkrxGe**), is powered by a Raspberry Pi 2 and an Adafruit 3.5" PiTFT touchscreen rechargeable battery, which sits sandwiched between the Pi 2 and the screen, is managed by a PowerBoost 1000C, a load-sharing DC/DC boost converter capable of doling out 5.2V and charging the battery while your gadget is in use. Finally, a small amp is connected to a tiny speaker for audio output.

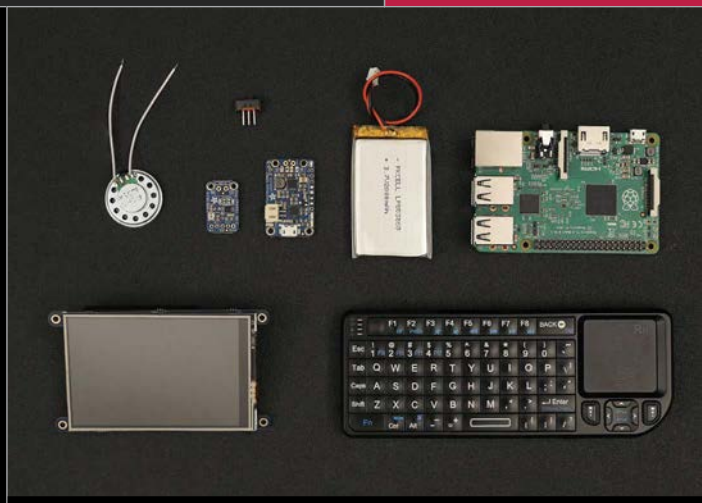
> "The project comes hot on the heels of the Ruiz brothers' excellent Pocket PiGRRL"

and, frankly, not a great deal more! The project comes hot on the heels of the Ruiz brothers' excellent Pocket PiGRRL – as covered on page 46 – a home-brew Nintendo Game Boy build we're still swooning over a few months after discovering it.

Besides the Raspberry Pi 2 and PiTFT display, for control the project features a mini-chiclet keyboard with built-in trackpad. It's a widely available wireless input device that's both affordable and easy to use. The 200mAh

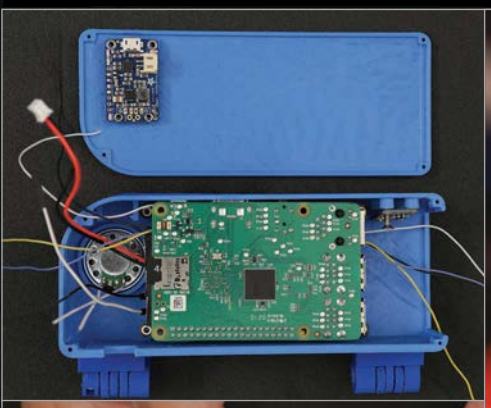
While the hardware is the really exciting bit, the 3D-printed chassis is a work of art, too. Take, for example, its totally modular hinged design. While it works really well on this Raspberry Pi mini-notebook, you could reuse it for 101 different hardware projects.

Like all of Adafruit's excellent Raspberry Pi projects, you can find a full shopping list of parts, software and 3D printing files for the Raspberry Pi Notebook on the Adafuit Learning system at **learn.adafruit.com**.



Above All the components laid out – it's not particularly complex

Above The 3D-printed case, complete with modular hinges



Above It's a thing of beauty and a cracking weekend build, assuming you have access to a 3D printer

DANIEL SPIES
A Dutch magician who likes to build his own tricks, and has taken to more advanced electronics and microcontrollers to improve his shows.
facebook.com/raspberrynin10do

NIN10DO

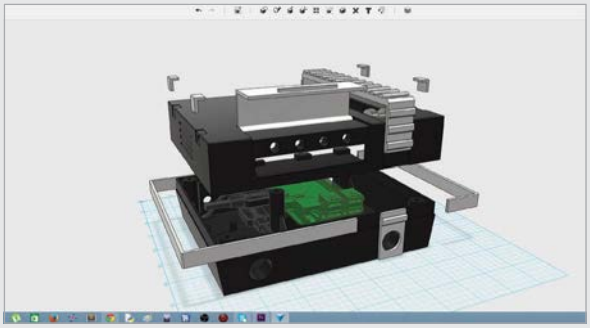
A 3D-printed Raspberry Pi-powered Nintendo Entertainment System that can also play Sega games

Quick Facts

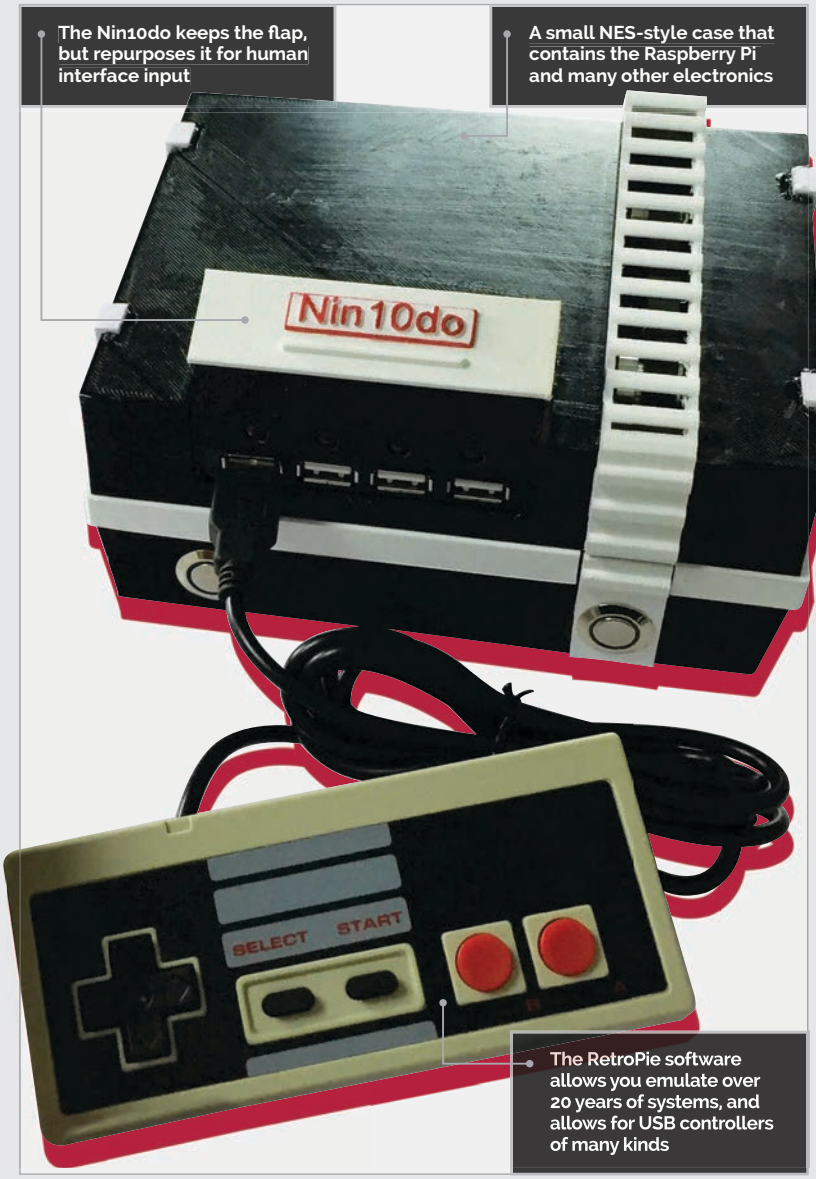
- The first version took about two months
- The case is made of XT-CO-Polyester instead of the normal ABS plastic
- Daniel now has plans and the confidence to develop his Pi box of tricks
- A future version will be made with laser-cut acrylic
- It's not a trick, it's an illusion

When you're a magician looking to make your tricks – sorry, *illusions* – better, you have several paths to go down. You could start buying or recreating other people's work; however, that's not entirely original, and you risk other people having seen it already. What truly great magicians do is to create their own illusions – original tricks that wow an audience with a wonderful show. Creating tricks and illusions for people-sized magic sounds similar to another hobby: that of being a maker. Makers love the Raspberry Pi, and this is where professional magician Daniel Spies found himself at the end of 2014.

"I wanted to start learning how to program in Python so I could integrate a Raspberry Pi in complex magic acts," Daniel tells us. "The Raspberry Pi would be great for starting special music



Below A 3D modelled and printed chassis allows for the perfect fit for the project



or sound effects, operating small smoke machines or even electromagnets. After I mastered the basics (servos, LEDs, etc.) it was time to build something as a practice project. It had to include as many different skills as possible, like CAD drawing, 3D printing, mechanical engineering, programming, and electronics. Then I saw guys on YouTube using their Raspberry Pi to emulate classic games on their TV. The idea was born."

Daniel decided to 3D-print his own custom NES case for the Raspberry Pi; it was smaller, sleeker and with a few more tricks than the 30-year-old console, including motors, lots of flashing lights, and the ability to play N64 games. There were some rules for the project to make it worthwhile as a test run for bigger things, though. "[These] rules included the Nin10do must look like a firm professional game console; it must run my Python script in the background but *not* sacrifice any speed or usability; it must have a option to be turned on and off without damaging the software or the SD card; and the cover must not damage itself if, for some reason, it is opened twice by the stepper motor."

With this in mind, Daniel went to work. He created full schematics for the electronics, making use of a series of LEDs, stepper motors, driver boards, timing belts, and lots of extenders to reach the I/O ports to make it actually usable. All of the components used were basic ones, so while there was a lot of soldering involved, he wasn't gutting any existing devices to make the project work.

> "Daniel designed and 3D-printed a case reminiscent of the NES that could house his Raspberry Pi"

With this and the coding in place (which you can find on his GitHub page – **bit.ly/1RkBirf**), Daniel designed and 3D-printed a case reminiscent of the NES that could house his Raspberry Pi and electronics. The important question, though, was whether or not he succeeded...

"The latest version runs very stable!" Daniel reveals. "I added some parts a while ago (small capacitor in the second momentary switch) and changed the USB hub to a better version. This solved the last (minor) bugs."

The full build process, printable 3D models and code are all available online from Daniel, so if you want to give it a go yourself, the tools are there. There may even be a kit coming in the future...

Above The flap is motorised, revealing the USB ports when turned on, thanks to a custom Python script

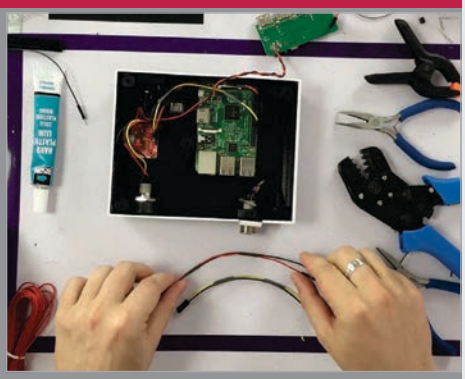
HOW TO CREATE A NIN10DO

>STEP-01
Electronics

Put together all the necessary components you need for the project. Apart from the Pi, this includes the LEDs, the other controllers, USB extensions, and the motors necessary for the mechanics section.

>STEP-02
Programming

The main emulation is done by the RetroPie software. However, you also need to create a Python script that controls the buttons, motors and LEDs, depending on the situation.



>STEP-03
Mechanics

Finally, fit everything into a custom, 3D-printed case. This step includes assembling the mechanised flap, along with installing the USB ports and physical buttons.

DENIS PAPATHANASIOU

Denis Papathanasiou is a managing director at Banrai LLC, an analytical technology firm that he co-founded in 2001.

denis.papathanasiou.org

PISCAN

One project builder turned a Raspberry Pi into a home-made Amazon Dash scanner. The maker of PiScan tells us more…

Quick Facts

- It uses the Open Product Data database to match products
- Unlike Amazon's Dash, you can order any product from Amazon with it
- It cost around $70 to build (including the cost of the Raspberry Pi)
- All the software and installation instructions are on GitHub
- A similar project called Oscar served as the initial inspiration

Wouldn't it be amazing if you could just scan a barcode and get another one of that particular product through your door the next day? Amazon clearly thought so; that's why it built a mini device called Dash that did just that. No more online shopping, just scan and go…

PiScan is an open-source version of Amazon Dash created by Denis Papathanasiou. With it, you can scan products using a Raspberry Pi and order them directly. It's great fun and incredibly practical, and it's even more powerful than Amazon's official device.

"It's the ultimate in lazy-person shopping," says Denis. "PiScan will read the barcode on any consumer product and order it for you from an online vendor."

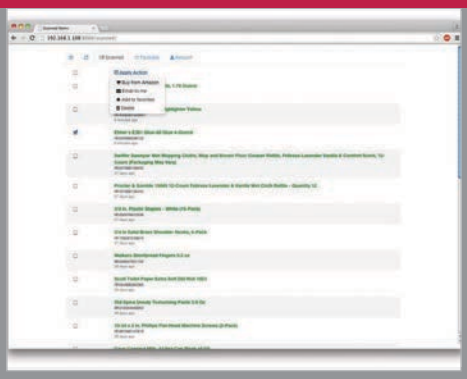
The inspiration came from another Raspberry Pi project called Oscar. That just converted "product barcodes into a grocery list," explains Denis, "but I thought it would be nice to take it one step further."

PiScan converts scanned items into a list that you can use to order products, using Amazon's API.

Right A list of recently scanned products. Placing a tick next to the product enables you to shop for that item

SETTING UP PI SCAN

>STEP-01
Laser USB scanner
The main piece of hardware is a USB laser barcode scanner. This is used to read barcodes on products and send the digital information to the Raspberry Pi (which then matches it to a database).



>STEP-02
Software installation
With the scanner connected, you install the software. Denis has put a pre-built ARM binary on GitHub (plus the source code). He's hosting an open barcode database (**saruzai.com**), or you can create one of your own.

>STEP-03
Scan and shop
Use the barcode scanner to scan products. An open-source database of products is used to match the barcodes. You can then tick products in the list and shop for them automatically on Amazon.

In terms of hardware, PiScan is pretty basic. "I used a Raspberry Pi Model B with a WiFi dongle and a USB laser barcode scanner," says Denis. The scanner is the only extra hardware requirement to a regular setup, and you can pick one up from Amazon for less than £20.

"I wrote software for the Raspberry Pi to listen for input from the barcode scanner," adds Denis. "The scanner works just like a keyboard, except its input comes in short bursts of characters."

> "PiScan converts scanned items into a list that you can use to order products using Amazon"

The input is a 10- to 13-digit number matched to the Open Product Data database (**product-open-data.com**).

"If there's a match," explains Denis, "it will put the name of a product into a list." The Raspberry Pi delivers the list to you as an email, or you can tick items to add to your Amazon shopping cart.

Building the PiScan "was simple and straightforward," says Denis. "There was nothing to solder and I didn't need to use any of the Pi's GPIO pins... Most of the work went into the software design, to make sure that the input from the barcode scanner was being read correctly.

"It's great," he tells us. "The dedicated scanner device reports barcode numbers with high fidelity.

"I've been using it to buy staple products regularly."

It's even better than Amazon's Dash scanner, claims Denis. "Dash only works with certain brand products, and it doesn't give you any say about quantity or price."

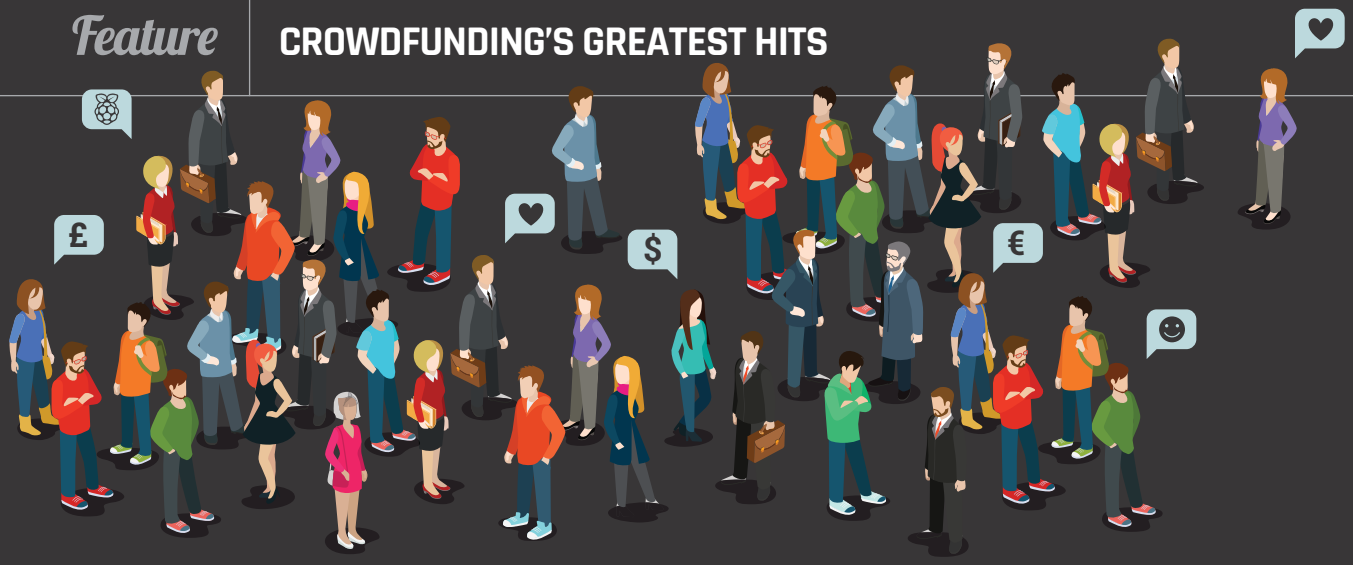
The only vendor so far is Amazon, but Denis is looking to expand to other retailers.

"Tesco in both the UK and Korea supposedly offer similar APIs," says Denis, "and there are probably other vendors out there that I'm not aware of."

If you fancy making your own PiScan personal product scanner, the software is freely available to download from GitHub (**github.com/Banrai/PiScan**), along with installation instructions.

"I've tried to make them simple to understand," says Denis, "but feedback is welcome."

Above The finished PiScan device is attached to a wall and orders products online when you scan them



CROWDFUNDING'S GREATEST HITS

The Raspberry Pi is the perfect maker tool, so it makes sense that it has played a pivotal role in some of the best crowdfunding stories ever told...

The low cost of the Raspberry Pi, coupled with its appeal to hackers and tinkerers, has launched an explosion of hardware projects – from simple plastic cases, to entire spin-off devices powered by a Pi at their heart. For many, the only thing stopping them from taking their project from the workbench to market is a lack of funds, something crowdfunding sites like Kickstarter and Indiegogo aim to solve.

Crowdfunding serves several valuable purposes for smaller businesses and individual makers: it allows them to raise funds to bring a product to market without having to go through traditional funding channels, like bank loans or venture capital; it enables them to quickly gauge the demand for a particular product and receive feedback on its design ahead of mass-production; and it can help build buzz around a product launch.

For customers – 'backers,' in crowdfunding parlance – it provides access to hardware that would otherwise not be readily available, along with a feeling of being part of the journey to market that doesn't come from simple pre-orders or registrations of interest on a corporate website. While only around two in five Kickstarter campaigns reach their funding goal, the right idea at the right time can earn a fortune, as proven by our rundown of the biggest successes from the Raspberry Pi community.

It's a heady mixture that has featured some stunning successes, along with a perhaps surprisingly small number of failures. If you've ever wondered about the secret of crowdfunding success, we've spoken to those who have been there and done that, to bring you the advice you need to know in order to succeed, while avoiding the pitfalls that have caused others before you serious upset along the way.

KANO

The top-grossing crowdfunding campaign in Raspberry Pi history by quite some margin, Kano raised nearly £1 million in November 2013, with a premise that at first seems laughably basic: a Raspberry Pi starter kit bundle.

Positioning the project as 'a computer you make yourself', Kano bundled an off-the-shelf Raspberry Pi with carefully selected accessories, including an eye-catching orange keyboard and trackpad, and a customised operating system. Its real potential came with the unveiling of Kano Blocks, a bundled graphical programming language inspired by MIT's Scratch and Google's Blockly, along with built-in lessons to walk the user through everything from assembling the kit to programming a game.

Despite its record-breaking funding run, Kano – as is common with crowdfunding projects – ran into some trouble during production. The first Kano units were due to ship in June and July 2014, but problems with the HDMI cables, keyboard battery, flashing the SD cards, the power supplies, and even the plastic case, meant that the first kits didn't ship until September and it was October before the majority of backers received their rewards.

Since shipping Kano, the team behind it has continued to build on its success with the creation of Kano Challenges, educational contests designed to further encourage children to get involved with programming and computing.

HDMIPi

The HDMIPi campaign raised £261,250 – far in excess of its £55,000 goal – from 2,523 backers to produce a low-cost high-resolution display. HDMIPi is notable for another reason than its sky-high funding, though: the project was a joint venture between electronics professional Dave Mellor of Cyntech and Alex Eames of review and tutorial site RasPi.TV, who acted as the public face of the campaign.

"I'd been doing **RasPi.TV** full-time for 18 months, with no income from it, before HDMIPi," Alex explains. "It was Dave Mellor of Cyntech who approached me about the project in the first place, after talking to the guys at the Milton Keynes Jam.

"It was obvious to all of us that a small, inexpensive, portable screen was needed for the Pi – we all wanted one – but there wasn't anything out there under £100. We thought it ought to be possible to make that happen, and we were right, but it wasn't easy."

While the crowdfunding campaign was a great success, the project hit difficulties during fulfilment, which led to some backers receiving their displays almost a year late – a common theme in hardware campaigns. "The actual driver board design came partly from ideas [that] backers suggested during the campaign," Alex recalls. "We liked them because they made the product unique, but they were probably the biggest cause of the delays; in hindsight, we could have probably delivered months earlier if we hadn't listened."

SLICE

The creation of FiveNinjas – a group comprising members of Sheffield-based Pi accessories maker Pimoroni, the Raspberry Pi Foundation itself, and music producer and technology trainer Mo Volans – Slice raised an impressive £227,480 to build a media playback set-top box based around the then new Raspberry Pi Compute Module.

Rather than taking the retail Raspberry Pi hardware as the basis for the project, as many other campaigns have done, the Compute Module – an industrial Raspberry Pi variant based around the SODIMM form factor, designed to act as a plug-in computer-on-module for a custom-designed carrier board – allowed FiveNinjas greater control over the final design and layout of the project.

The result is a sleek aluminium box featuring a smart LED strip under user control, a wireless remote control, and an internal hard drive for storage. The Slice focuses more on playback of local content rather than streaming, although WiFi connectivity was unlocked as a stretch goal when the campaign hit £100,000. It includes a slick user interface and excellent compatibility with various file formats, along with features such as SATA storage connectivity, missing from the standard Raspberry Pi hardware.

Like the majority of crowdfunded projects, Slice has been hit by numerous delays. The rewards were scheduled to be with backers by November 2014, but it took until the end of February 2015 for the first Slice packages to begin shipping.

FLOTILLA

Flotilla, in fourth place with £146,680 raised, is the second Kickstarter campaign to come directly from the Sheffield warehouse of Pimoroni, and takes aim at the educational market with a range of smart input and output modules which can be easily programmed from a Pi.

Pimoroni made a name for itself in crowdfunding circles by launching the first project to go live on Kickstarter UK: the Picade. "We actively wanted to be first. We'd heard that Kickstarter was coming in the UK, so we made sure that our campaign was ready and [that] we were there ready to press the button," explains Jon Williamson. "We just stood there at ten to midnight, pressing F5 until the live button became live, and we were the first to press it," adds a laughing Paul Beech.

Despite not having the benefit of being first, Flotilla easily surpassed the £74,134 raised by Picade. The company has a few ideas for avoiding the delays that plagued its original project, too. "We spent a year working on Flotilla before we put it on Kickstarter," Jon reveals. "With Flotilla, we've already been through six or seven iterations of the design of the system; the software has been in development for six months. Really, it's much closer to being finished this time around, and I guess that's what we've learned: the more stuff that you leave unfinished, the more things that can go wrong to trying to finish it."

PI-TOP LAPTOP

The only top-ten grossing Raspberry Pi crowdfunding campaign to use Indiegogo rather than the more well-known Kickstarter platform, London-based Jesse Lozano and Ryan Dunwoody's Pi-Top project looked to encourage people to learn computing by building their own Raspberry Pi-powered laptop. With £112,130 raised, more than double its original goal, the campaign certainly caught the interest of the community.

While much of the publicity surrounding the campaign focused on the hardware side – including the team's decision to 3D-print the chassis' master, then injection-mould for mass-production – Pi-Top promised more: like Kano, the project positions itself as the ideal platform for teaching beginners about computing, but with a focus on hardware rather than software.

The Pi-Top campaign closed in December 2014, and the team behind it worked hard to keep backers informed, as well giving them all a free upgrade to the Raspberry Pi 2 from the planned first-generation model. While the team had hoped to ship the devices to backers in May 2015, there were delays and the first batch was finally shipped in October.

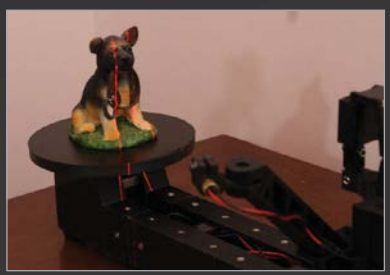
As with most crowdfunding projects, the Pi-Top is going on general sale once the backers who funded its journey from prototype to production have received their rewards. Indeed, visitors to the team's website (**pi-top.com**) can now pre-order a Pi-Top kit, with or without a Pi included.

Top Ten

RASPBERRY PI CROWD-FUNDING CAMPAIGNS

- Kano
  £987,976
  (1,522% funded)
- HDMIPi
  £261,250
  (475% funded)
- Slice
  £227,480
  (253% funded)
- Flotilla
  £146,6W80
  (448% funded)
- Atlas 3D
  £141,940
  (7,333% funded)
- Pi-Top Laptop
  £112,130
  (217% funded)
- BrickPi
  £82,286
  (6,752% funded)
- Ziphius
  £80,873
  (102% funded)
- Rapiro Robot
  £75,099
  (375% funded)
- Picade
  £74,134
  (226% funded)

ATLAS 3D



Atlas 3D, from Kentucky-based Murobo LLC, is an example of the Raspberry Pi providing the power, rather than inspiration, for a project. Raising £141,940 – a record-smashing 7,333% of its modest £1,944 goal – the Atlas 3D is a 3D scanner built around a 3D-printed chassis and the use of lasers to measure the shape of the target object.

"The Raspberry Pi was an easy choice since it has an excellent 5-megapixel camera add-on, is able to drive lasers and motors, and has enough memory and CPU power to perform a 3D scan," project creator Uriah Liggett explained about his creation during its crowdfunding campaign in early 2015. "All of the software runs on board the Raspberry Pi, so there are no required drivers or software packages to install."

Uriah's project promises to provide benefit not only to its backers, but to the community at large, too. The driving software, FreeLLS, is available under the GNU General Public License as an open-source project, while the electronic design files are open hardware.

While Uriah took a risk by promising to release his creation under an open licence, since potential backers could have been put off by the idea of paying for something that others could build themselves from free designs, his impressive funding run proves that crowdfunding and open source are not mutually exclusive.

With the first batches having been shipped to Kickstarter backers, the Atlas 3D is now available to order from the Murobo store (**store.murobo.com**), from $209 plus shipping.

CROWDFUNDING FAILURES

Not every crowdfunding project ends in success, of course. Overall, 61 percent of Kickstarter campaigns fail to make their funding goal; of these, more than half never even reach one-fifth of the way. These, however, are the lucky ones, failing gracefully and without anyone ending up out-of-pocket. More notable are those projects that reach their goal, collect backers' pledges, and then fail to deliver on their promises.

> "Projects that reach their goal, collect pledges, then fail to deliver"

Azorean's Ziphius caught the community's attention back in 2013 with the promise of an aquatic drone powered by a Raspberry Pi and controlled from a smartphone app. The campaign ended with nearly £81,000 to produce the device, with a self-imposed deadline of March 2014.

Not going as planned

By March, only the control boards were completed; in May, the company was still working on prototypes of the drone's chassis and control systems. A revised delivery plan suggested an October release date but in January 2015, Azorean admitted that it would be August before devices shipped – leaving backers clamouring for refunds.

"The delays were caused mainly by the difficulty moving from the prototype that we had at the time and the need to adapt it to the new tools of mass-production," Azorean's Cristina Gouveia explained. "We spent more, much more, time than we were expecting."

Those issues may now be resolved, but there's a more serious problem on the horizon: a lack of funds. "That delay has caused us some financial struggles, so we need to get additional investment," Cristina admitted. As to whether Azorean will be able to ship its promised rewards to backers if investment fails to materialise: "If we don't [receive investment], I don't think we can."

Above The Azorean Ziphius was a runaway crowdfunding success in 2013, but is still yet to deliver

Alfredo January 27

I am not happy at all with your last estimations (August 2015). I want a REFUND, as I am sure this product won't be released.

Frank January 30

I would like a refund. This has gone far longer than you estimated.

Justin January 22

I thought we were supposed to get an update in early January? Another deadline missed?

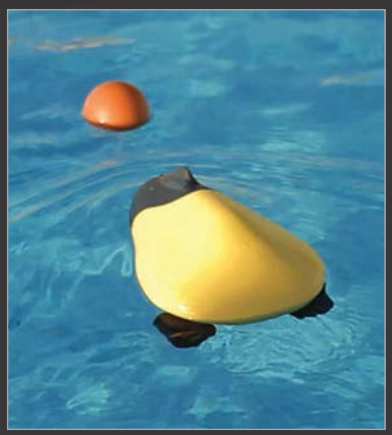
Still taking money

At the time of writing, Azorean was still accepting pre-orders for the Ziphius through its own website, priced at $269 plus shipping. There's a disclaimer about possible delays to the shipping date, although it still says 'Estimated delivery: August 2015'. We understand that the company is now planning to launch the Ziphius kit in spring 2016, followed by the ready-made version in the summer.

Azorean's case may be an extreme one, but delays to the schedules originally proposed by crowdfunding newcomers are all too common. "The skills required for prototyping and production at moderate scale are quite different," Saar Drimer, of electronics design consultancy Boldport, explains. "On top of that, the ease by which prototyping is done today can be very misleading. 3D-printing a nice enclosure is pretty easy, whilst sourcing injection-moulded enclosures in volume from China is a completely different matter. Another example is sourcing components: if you're not aware of the many pitfalls of availability and obsolescence of components, you're going to get burned."

Pimoroni's Jon Williamson can certainly attest to the latter, having run into exactly that pitfall during the Picade campaign's delayed fulfilment. "The screens were a nightmare. We picked a screen that was basically end-of-life. It was a great screen, but they were like hen's teeth. We'd get odd batches of them; we might get 20 in Hong Kong one week, and we'd just grab them and we'd have 20 screens, great, but we needed 500. It was not a pleasant experience."

Communicating with a contractor overseas, often the only way a campaign can produce hardware at an affordable level, has its own challenges, too. "The biggest problem we had was communication with China," reveals Alex Eames about the HDMIPi campaign. "Getting things done the way we wanted, meeting our specifications, and being of usable quality was really difficult."



CROWDFUNDING GLOSSARY

Backer Someone who contributes money to a crowdfunding campaign, known as a 'funder' on some sites.

Campaign An attempt to raise funds through one or more crowdfunding sites, usually limited to a 30-day period.

Creator The individual or company behind a crowdfunding campaign.

Early Bird Reward A strictly limited number of rewards offered at a discount, designed to help a campaign build momentum in its early stages.

Fees Crowdfunding sites take a cut of all money raised by a campaign, usually around 10 per cent, in exchange for providing a platform and handling payments from backers.

Flexible Funding A feature of Indiegogo which allows campaigns to receive backers' funds even if the campaign does not reach its goal.

Goal The amount of money required by a crowdfunding campaign to be successful. If this is not raised, the funds are usually returned to backers.

Limited Rewards Campaigns can set limits of the number of rewards on offer, either to boost hype through artificial scarcity, or to meet an upper limit on manufacturing runs.

Pledge The money contributed to a crowdfunding campaign by a backer.

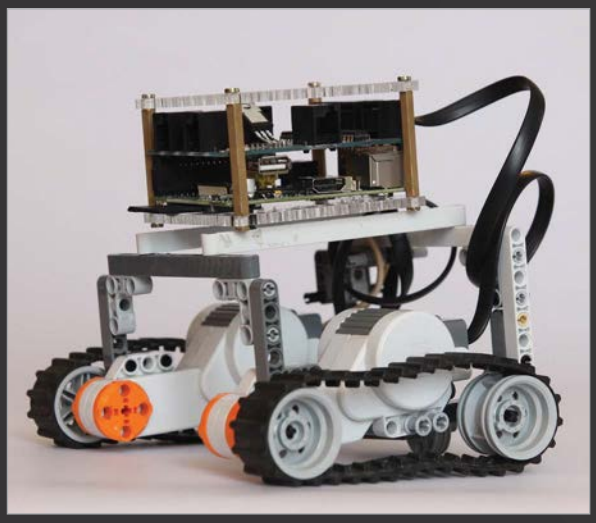
Reward The promised return for a backer's pledge, ranging from an email or T-shirt at the lower levels, to multiple units of the item and even dinner engagements with the creators at the upper end. Known as a 'perk' on some sites.

Story The pitch for the campaign, which typically includes a video presentation, a write-up of the project, information about the creator, a planned delivery schedule, and a section on potential risks.

Stretch Goal A campaign that raises more than its original goal may use the extra money to improve the product, offering a wider choice of colours, free extras, or upgraded features to backers.

Updates A good crowdfunding campaign will post regular progress reports to its page, both throughout the campaign and leading up to reward fulfilment.

FINDING CROWDFUNDING SUCCESS



Above The Dexter Laboratories' BrickPi raised over 67 times its original target

For those who want to follow in the footsteps of campaigns like Kano, HDMIPi, and Slice, it's all too easy to be blinded by the sums of money on offer and leap in without proper preparation. Convincing people to part with their cash is only the start of the process.

"You [need] to have your plan together for manufacturing, and then have a second one in the waiting if things go really, really well, before you launch your campaign," advises Dexter Industries' John Cole, whose LEGO-based BrickPi robotics kit raised £82,286 – over 67 times its original target. "That should not be an afterthought – that should actually be baked into the campaign. It ties into really concrete things, like what you price your product at and what you promise for a delivery date.

"It also factors into the softer side of developing a product, which is who are you going to be selling it to, what are their tastes, wants and needs? You should understand that clearly before you start the campaign and communicate that with your manufacturer, because they'll play a big role in whether you have a successful campaign or not."

PLEDGING FOR THE SKY

Crowdfunding campaigns typically offer a variety of reward levels, from the pound donation that gets a backer heartfelt thanks, to rewards valued at thousands of pounds. Here are some of the biggest reward levels seen...

Kano Academy

£6,490 to receive workshops for 100 people, bespoke tutorials, guides, and everything an establishment needs to become a Kano Academy, plus a T-shirt. Amazingly, one generous backer snapped the offer up.

BrickPi Namesake

£6,489 to rename the BrickPi's case to anything your heart desires, so long as it's suitable to repeat in polite company. Unsurprisingly, nobody backed the project at this level.

Rapiro Custom Design

£5,000 to have a Rapiro robot customised to your precise requirements. With the base kit priced at just £229, nobody took creator Shota Ishiwatari up on his offer.

Ziphius Prototype

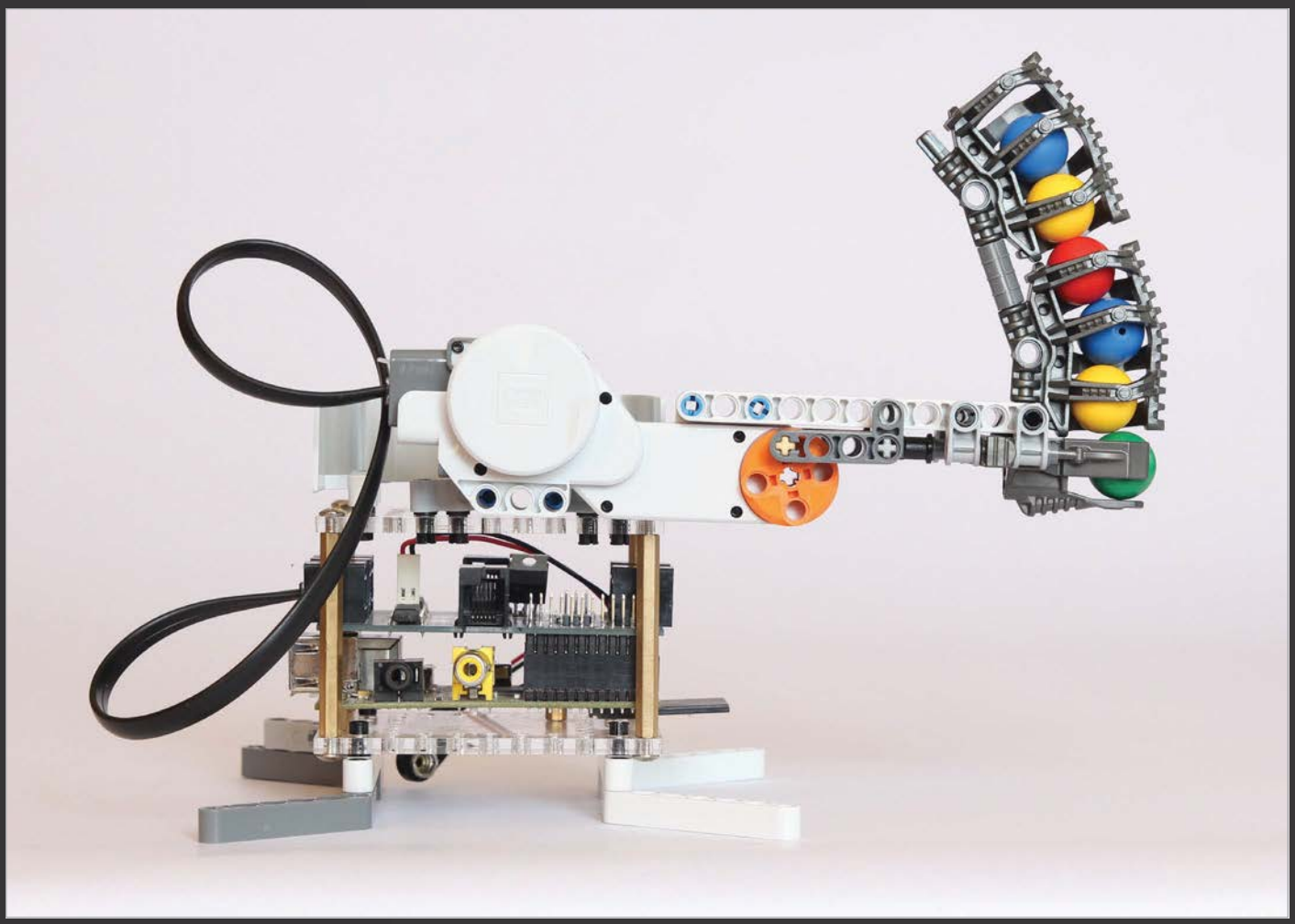
One backer gave Azorean £4,858 to receive a pre-existing Ziphius drone prototype, as used when the company entered the Engadget Expand Insert-Coin Competition.

Slice Full Custom Shop

£2,999 to work with the FiveNinjas design team to construct an entirely custom Slice media player. As with Rapiro, the cost over the £179 standard version was considered too high for anyone to pledge.

Pi-Top One of a Kind

£1,622 to discuss the creation of a custom Raspberry Pi laptop, complete with the promise of automatic hinge and integrated light show. Despite planning for five potential backers, the Pi-Top project was left with no takers for this top-tier reward level.



"There have been quite a lot of Raspberry Pi-based crowdfunded projects – even a couple of screens – that have not succeeded because the community has never heard of the people behind them," adds Alex Eames, who was chosen to front the HDMIPi campaign specifically for his recognisability in the Raspberry Pi community. "The takeaway message from this is make sizeable deposits into the community before you try to make a withdrawal. If you look at who's done well in high-value crowdfunding projects, there aren't many who've managed it without significant input into the community over a sustained period of time before they had their 'overnight success'. You won't get money out of the community unless the community thinks you deserve it and can be trusted with it."

> "If you've got a hardware idea, keep it simple and true to its core"

Minimise complications

The lessons Pimoroni's Jon and Paul are taking into the Flotilla fulfilment process are simple. "Minimise the number of complications. If you're thinking 'should I add this feature?', then I think you should err on the side of caution," Jon advises. "If you start adding features part-way through the design process, you'll risk overshooting by a month, minimum, maybe more."

"If you've got a hardware idea," agrees Paul, "keep it simple, keep it true to what the core of it is, make sure you've got an audience, and make sure you've had lots of good feedback and criticism of it so it's polished."

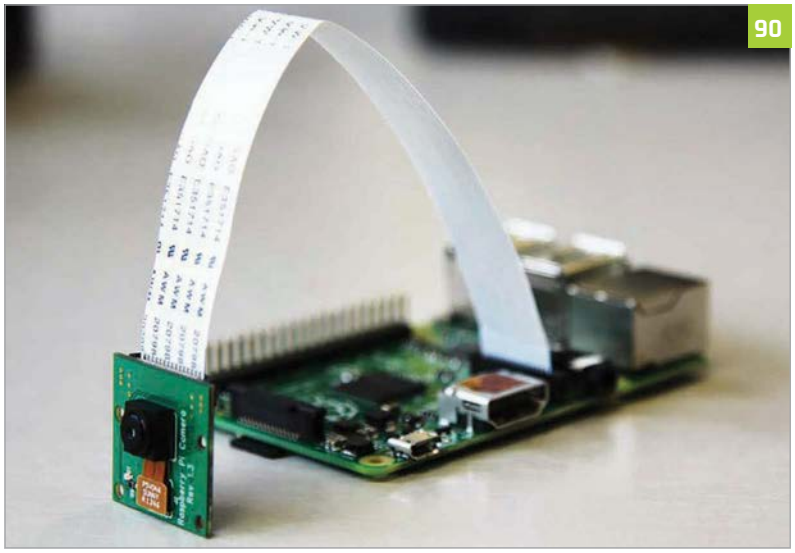
Experience is key, adds Saar Drimer, and all too often lacking in those turning to Kickstarter to get their project off the ground. "Multiply the time you think it would take by three, then by your level of confidence," he advises, "where one is 'very confident', and three is 'not confident at all.'"

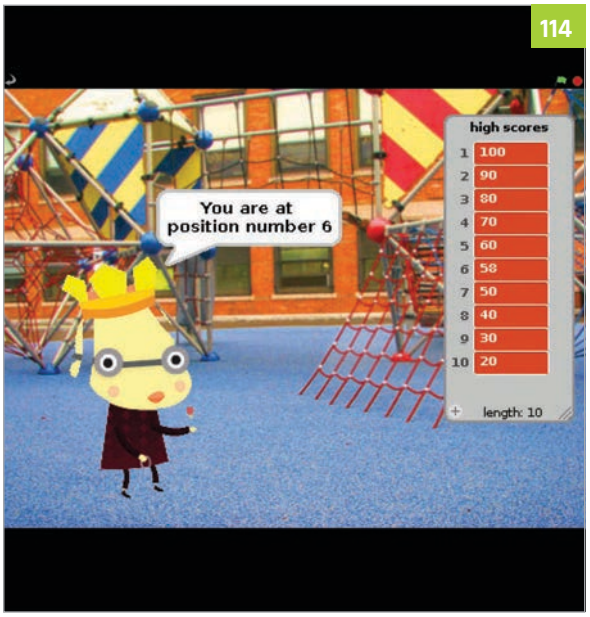
For those on the other side of the table, who are looking to back projects, Saar has a few words of warning. "Your payment is a gamble, and that's what Kickstarter was meant for. If something really appeals to you, support it – and treat the experience of actually getting it at all as a pleasant surprise."

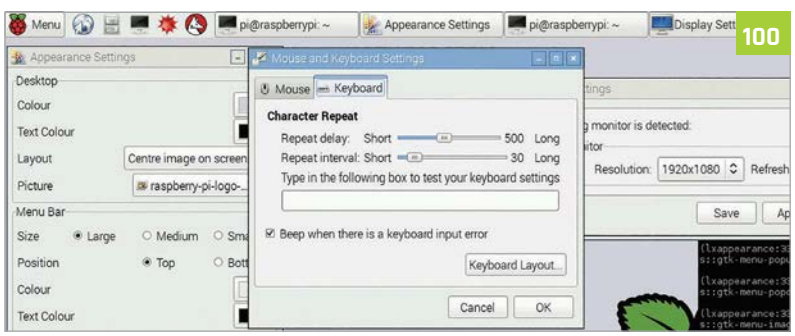
Gareth Halfacree

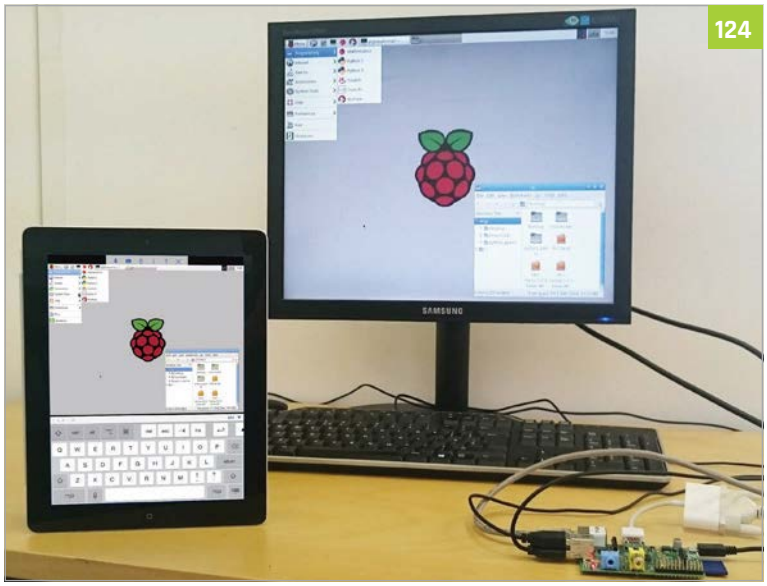
TUTORIALS

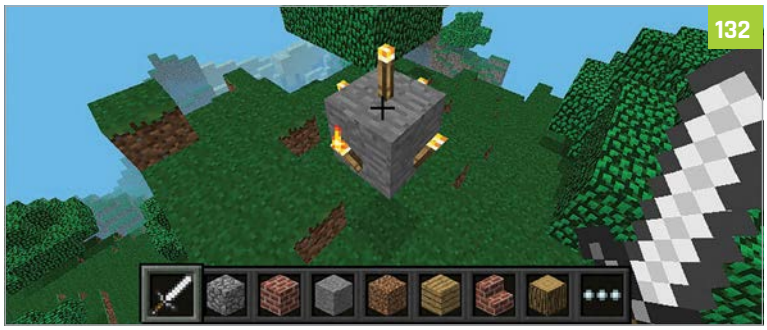
As well as giving you inspirational ideas for projects, our detailed step-by-step guides show you exactly how to make them and learn more about the Raspberry Pi…

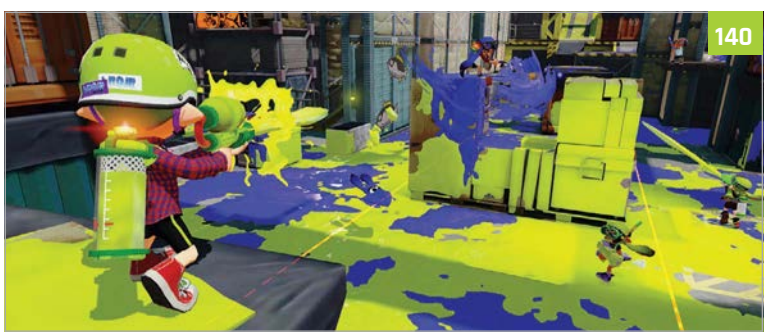












Tutorials

ALEX EAMES

Alex runs **RasPi.TV** and **RasP.iO**, and finds himself between a blog and a hardware business. You can find him as **@RasPiTV** on Twitter.
RasPi.TV

WATCH iPLAYER ON RASPBERRY PI

You'll Need

- A decent-sized SD card (minimum 8GB)
- A broadband connection
- A few lines of code: **github.com/raspitv/get_iplayer**
- A TV, or a monitor and speakers

Download BBC TV programmes as high-definition MP4 files (without any DRM) for offline viewing on your Pi or other devices

Get_iplayer is a fabulous, open source utility program which allows you to see what's currently available from the BBC's iPlayer website and download any TV programmes you want. You can choose resolutions between 512×288 and 1280×720, so you can pick one that suits your viewing device and storage capacity. There's no DRM on these files (although the BBC's terms state that you are only allowed to keep them for 30 days), and because they are BBC programmes, there are no adverts either, making it perfect to use for films.

>STEP-01

Update your package lists

We're going to install some software. The first thing you should do when you install software in Raspbian is – at the command prompt – type **`sudo apt-get update`** (and press **ENTER**), which updates your package lists. Then it's a good idea to **`sudo apt-get upgrade`** (pressing **ENTER** again) once all your software packages have been updated to their latest versions. It can take upwards of 30 minutes if you haven't done this recently. Then you need to add

Get_iplayer enables you to download films and TV shows from the BBC iPlayer service

As the BBC doesn't use DRM, you can watch shows on your Pi or any other device

Left Here's some example output from the `get_iplayer --cat film` command. Note the programme IDs (PIDs) on the left

Jon Davies's PPA (Personal Package Archive) to your **sources.list**. Copy and paste the five lines of GitHub code from **github.com/raspitv/get_iplayer/blob/master/code.txt** into a terminal window on your Raspberry Pi and press **ENTER**.

>STEP-02

Install the keyring and software

Once you've done this, you need to repeat the `sudo apt-get update` command in the terminal. You'll likely get an error message about keyrings, so now you need to install Jon's keyring as well, with the following command:

```
sudo apt-get --allow-unauthenticated -y install jonhedgerows-keyring
```

Then press **ENTER**.

Next, repeat the `sudo apt-get update` command one last time. Now we're ready to go ahead and install the **get_iplayer** program itself (notice the installation name in the command we type is hyphenated, not underscored):

```
sudo apt-get install get-iplayer
```

>STEP-03

Using get_iplayer

Before we start, it's always good to know where to find help, should you need it. To do this, type: `get_iplayer --usage` in the terminal, which should give us a list of the basic options. If you want more options, you can use `get_iplayer --help`, or even more using the command `get_iplayer --longhelp`.

There are a lot of options, so it can be a bit overwhelming, but most of them are not needed for simple searching and downloading of content. Before we can download a programme, we need to collect the index of all the available content. This is done by using the `get_iplayer` command all by itself in the terminal.

>STEP-04

Narrowing down the search

At any given time, there are a couple of thousand items available for download. That's a bit overwhelming, so we need a way to cut it down a bit. You can use categories with the `get_iplayer` command – for example, `get_iplayer --cat film`. You can choose any category from the main list: Arts, CBBC, CBeebies, Comedy, Documentaries, Drama and Soaps, Entertainment, Films, Food, History, Lifestyle, Music, News, Science & Nature, or Sport. You can also use a keyword; if matched, it'll return possible downloads.

>STEP-05

Downloading content

Looking at the list of films available, each item starts with the programme's ID number (PID). Let's choose *Wallace and Gromit – The Wrong Trousers*. This has a PID of 2210. So, to download this film at the best available resolution (1280×720), you would type:

```
get_iplayer --get 2210 --modes best
```

After about 10–15 minutes, the file is downloaded and processed into an MP4 file, which we can view, store or delete at will.

>STEP-06

Watching content

As part of the default Raspbian installation, you have a GPU-accelerated media player called omxplayer. Because it uses the GPU, it's capable of playing HD video, even on a Pi Model A. To watch the film we just downloaded, we would type the following into the terminal:

```
omxplayer [filenamehere].mp4
```

If you're dealing with long filenames, once you've typed **omxplayer** and the first few letters of the filename, you can press the **TAB** key and it will auto-complete the filename for you (then press **ENTER**). You can see the full list of omxplayer controls at **elinux.org/omxplayer**.

DO I HAVE ENOUGH SPACE?

Use the command `df -h` to see if you have enough space left on your SD card.

WANT SOUND THROUGH THE AUDIO JACK?

Use `omxplayer -o local` to send sound through the Pi's audio jack rather than the (default) HDMI port.

DISCLAIMER

The BBC's T&Cs state that all iPlayer content is for UK playback only. In addition, any downloads must not be kept beyond 30 days, and must not be distributed in other forms. Neither we nor the Raspberry Pi Foundation condone any breach of these rules. For further details, visit **bbc.co.uk/terms**

RICHARD SAVILLE

Richard runs a popular tutorial and projects blog about an average guy learning the Pi and sharing his less-than-average experiences with the community.
AverageManVsRaspberryPi.com

SHOOT IN SLOW-MOTION WITH THE CAMERA MODULE

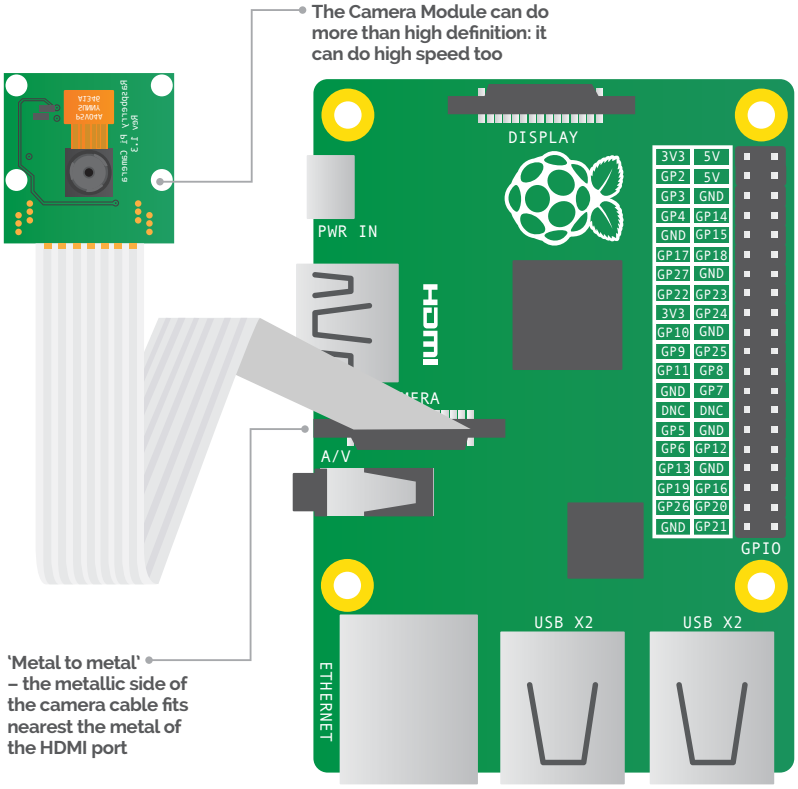
You'll Need

- Raspberry Pi Camera Module
- Internet connection
- Something fun to film

In this tutorial, the Average Man shows us how to shoot slow-motion videos with the Camera Module and convert them to play on almost any device

Slow-motion video has been used in the film industry for years – think of all those great action movie scenes with people jumping from explosions, or 'Bullet Time' made famous by the Wachowski Brothers in *The Matrix* trilogy.

It's actually really easy to make your own slow-motion videos with your Pi using the Camera Module. We'll get you set up and guide you through a short code listing that will let you record short 30-second videos that will automatically convert to MP4, so you can play it back on just about any device...



>STEP-01

Connect the Camera Module

The first thing you need to do is connect the Camera Module to your Pi. Make sure your Pi is turned off first. Be careful – the Camera Module is very sensitive to static, so ground yourself by touching something like a radiator before you start.

The Camera Module ribbon cable connects to the socket on your Pi nearest the HDMI port. Use the phrase 'metal to metal' to remember which way round to push it in – the metallic side of the camera cable should face the metal HDMI port. Gently pull up to release the clip and slip the ribbon cable in, then just push the clip back down firmly and check it's secure.

>STEP-02

Configure the Camera Module

If using Raspbian Wheezy, type `sudo raspi-config` in a terminal to enter the configuration menu. Using the arrow keys, scroll down the list that appears and select 'Enable Camera' using the right arrow key. In the next menu, select 'Enable' with the right arrow key to turn on the Camera Module, then hit **RETURN**. If using Raspbian Jessie, select Preferences>Raspberry Pi Configuration from the menu, then enable the camera in the Interfaces tab. Choose to reboot when prompted; or type `sudo reboot` in a terminal to restart the Pi.

>STEP-03

Install a video converter

The Pi records video into raw H.264 files which don't work on most of our devices. We can get the Pi to convert them to a playable format straight after we've recorded them in our script on the right. To do this, we can install a package called gpac. At the command prompt, type the following, then follow the on-screen instructions:

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

RECORDING TIPS

Be prepared

Make you have good lighting and a steady mount for your Camera Module. Also, charge your portable power options if you're shooting outside. Finally, don't forget to use an SD card with enough storage space.

What to shoot

How about shooting a remote-controlled car skidding round a corner? Perhaps a ball being thrown or other sports? What about an animation drawn on a notepad, slowed right down?

>STEP-04

Test the camera

Let's make sure everything's working as it should by testing the camera with a couple of terminal commands. With a screen connected, open a terminal window and type **`raspistill -o test.jpg`**. The picture should appear on the screen for a short time and an image should be saved to your Home directory.

If it doesn't work, check you typed the command correctly, or turn off your Pi and reconnect the camera ribbon cable before trying again.

>STEP-05

Create a Python script

We'll be using Python to create our slow-motion video script. Open your favourite text editor (the Leaf text editor in Raspbian is perfect) and copy the code opposite, being careful not to misspell anything along the way. You don't need to copy the comments (lines starting with **#**) – Python just ignores them. The script uses the OS Python library to carry out terminal commands like you've typed them in directly.

Save your file as **`slowmotion.py`** in your Home directory (**`/home/pi`**).

>STEP-06

Run the script

To run the script, simply open a terminal window, type **`cd`** and hit **RETURN** to ensure you're in the Home folder, then type **`sudo python slowmotion.py`**.

You will see the status of the script printed in your terminal window as it carries out its commands, and the Camera Module's LED will light up while it's recording.

The script will end when the video has been converted. You can watch the video on your Pi straight away by using omxplayer, which is included in Raspbian. Simply type **`omxplayer vid.mp4`**.

You could also copy your video onto any other device, like your tablet or smartphone.

The Camera Module has a fixed-focus lens and a wide recording angle

Slowmotion.py

Code Language
>PYTHON

```
import os
import time

print("Starting program")
time.sleep(2)

##### Record the slow motion video #####
# '-w' sets the width  # '-h' sets the height
# '-fps' sets the frames per second (90 maximum - for slow motion)
# 't' sets the time in milliseconds (30000 = 30 seconds)
# '-o' sets the output filename

print("Recording started - 30 seconds")
os.system("raspivid -w 640 -h 480 -fps 90 -t 30000 -o vid.h264")

print("Recording complete. Please wait...")
time.sleep(2)

##### Convert the raw recorded video file to playable mp4 #####
# '-add' is the name of the raw video we want to convert
# The second filename is the output mp4 file
# (we use the same name followed by '.mp4')

print("Converting video. Please wait...")
os.system("rm -f vid.mp4")
os.system("MP4Box -add vid.h264 vid.mp4")

print("Video conversion complete")
time.sleep(2)

print("Closing program")
time.sleep(2)
```

WILLEM KOOPMAN

Peripatetic sysadmin, providing grumpy solutions to grumpy problems; also maker of **whatcaniseefromtheshard.com**. Stumbled upon the field of computer vision while working in the magical world of visual effects.
secretbatcave.co.uk

FACE DETECTION WITH OPENCV

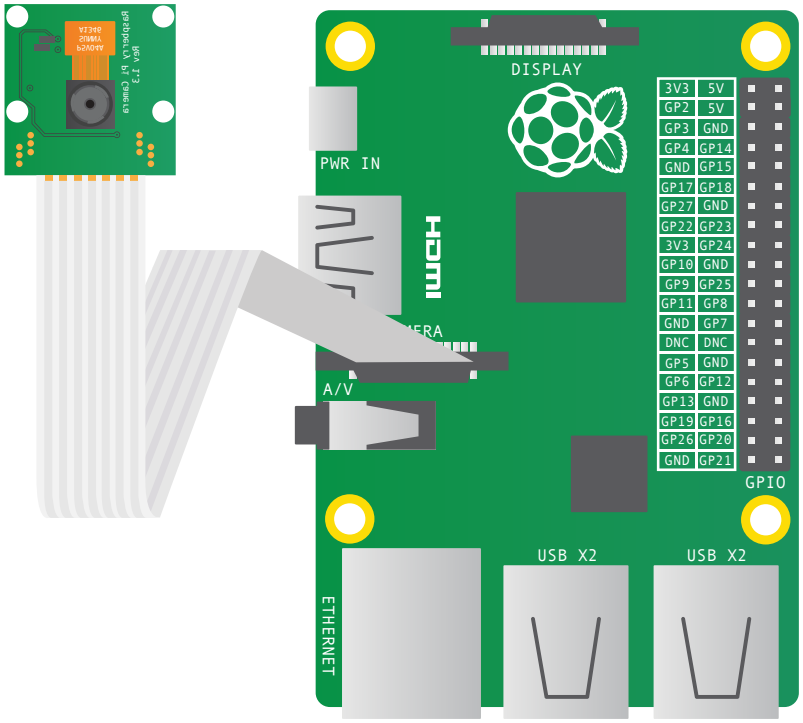
You'll Need

- An internet connection
- Camera Module (or webcam)
- OpenCV

You've got a Raspberry Pi Camera Module? You've taken a few images? Let's do something really clever and use them to detect faces...

You've set up your motion-triggered webcam, but that pesky dog keeps triggering it. How do you figure out if that alert is someone poking around or Fido searching for socks again? In this tutorial, we'll show you how to get your Raspberry Pi to separate the dogs from the faces, using the computer vision library, OpenCV.

We're going to make a simple Python script that will work its way through a directory of pictures, copying the ones that have faces in them. Not only that, it'll also draw a box around each face.



>STEP-01

Install OpenCV

By default, OpenCV isn't shipped with Raspbian. Never fear – everything is a simple **apt-get** away. First, we need to install OpenCV. In a terminal, type: **sudo apt-get update**, press **ENTER**, then: **sudo-apt-get install python-opencv libopencv-dev** and follow the instructions. We'll see if it's installed correctly by running the Python interactive interpreter and loading the OpenCV module. Type: **python** (and **ENTER**), then **import cv** (and **ENTER** again). If everything is installed correctly, you should see an empty prompt. If you see something along the lines of **ImportError**, go back and see if the **apt-get** worked correctly.

>STEP-02

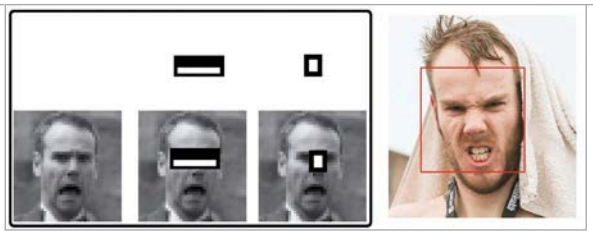
Understanding Haar-like features

We're going to use an algorithm called a 'Haar cascade'. Because computers have no understanding of what a face looks like, we have to give it a rule book. In this case, a Haar cascade describes the 'brightness signature'. A face contains two eyes surrounded by skin. The area surrounding the eye is a different intensity to the eye itself. A Haar cascade describes these patterns to provide us with a way to detect faces (and other objects).

>STEP-03

Begin the code!

Enough chit-chat! Let's write some code. We need to import the various libraries we are going to use and set some sensible defaults for the Haar detector. These defaults provide a trade-off between speed and accuracy. First, we set **minSize** to limit the smallest detectable face to a 20-pixel square. **imageScale** scales the image before we feed it into the detector; smaller



Above How the face detector works under the hood: high-contrast boxes are mapped to parts of the face

images mean faster detect times, but less accuracy. **minNeighbors** tells the detector that a match must be made up of a minimum number. Finally, **haarFlags** are special flags telling the detector what bits to ignore.

>STEP-04

Prepare the image

The first function we're going to create is **detectFace()**. Because the Haar detector only works with greyscale images, we make a greyscale copy; **grey** is resized and copied into the **small_img** container. Lastly, we equalise the histogram (using **EqualizeHist**). This evens out the contrast, making the Haar detector more effective. We pass the variables, **small_img** and **cascade**, along with the rest of the defaults we defined at the very start, into the function **cv.HaarDetectObjects**. It then spits out a list of objects, with attached coordinates, and dumps it into the variable **faces**.

>STEP-05

Mark the faces

We then 'iterate' through all the objects and extract the 'bounding box' (the area where the object detector thinks there is a face.) Now, this is where we do something vaguely confusing. Remember in the previous step, we made a few copies of the original image? Well, we didn't throw away the original. We can mark where we think the faces are on the original, so that we can save full-sized images in full colour!

We've scaled up the coordinates so that we can accurately tell **cv.Rectangle** where the top-left and bottom-right corners of the box should be.

>STEP-06

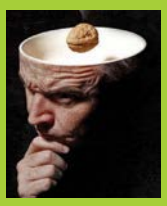
Final touches

readDirectory() goes through the directory supplied as a command-line argument and extracts files ending with '.jpg'. It then opens the image and passes it to **detectFace()**. If it finds some faces, it'll save out the marked images into a new file using **cv.SaveImage()**. To use your new program, you first need to find a Haar cascade XML file and put the path into **cv.Load()**. They can be found in **/usr/share/opencv/haarcascades/**. Running the program is as simple as storing some JPG files in the folder and typing **python facedetect.py**. With any luck, you'll see something like the following:

```
samples/ has:
  Analysing 292942_10151131251926133.jpg:
        Detected  2  object(s)
        Time = 1268.761ms
```

Facedetect.py

```
import os, sys, time
import cv2.cv as cv

minSize = (20, 20)
imageScale = 1
haarScale = 2
minNeighbors = 3
haarFlags = cv.CV_HAAR_DO_CANNY_PRUNING

def detectFace(img, cascade):
  # allocate temporary images
  gray = cv.CreateImage((img.width,img.height), 8, 1)
  small_img = cv.CreateImage((cv.Round(img.width /
imageScale),cv.Round (img.height / imageScale)), 8, 1)
  # convert color input image to grayscale
  cv.CvtColor(img, gray, cv.CV_BGR2GRAY)
  # scale input image for faster processing
  cv.Resize(gray, small_img, cv.CV_INTER_LINEAR)
  cv.EqualizeHist(small_img, small_img)
  faces = cv.HaarDetectObjects(small_img, cascade,
cv.CreateMemStorage(0),haarScale, minNeighbors, haarFlags, minSize)

  if faces:
    print "\tDetected ", len(faces), " object(s)"
    for ((x, y, w, h), n) in faces:
      #the input to cv.HaarDetectObjects was resized, scale the
      #bounding box of each face and convert it to two CvPoints
      pt1 = (int(x * imageScale), int(y * imageScale))
      pt2 = (int((x + w) * imageScale), int((y + h) *
imageScale))
      cv.Rectangle(img, pt1, pt2, cv.RGB(255, 0, 0), 3, 8, 0)
    return img

  else:
    return False

# scan all directories and subdirectories for jpg images
def readDirectory(fileLocation, cascade):
  for root, dirs, files in os.walk(fileLocation):
    print root, "has:"

    for name in files:
      if name.find(".jpg") >=1 :
        #sequentially loop, load and detect.
        print "Analysing " + name +":"
        #measure how long it takes
        t = cv.GetTickCount()
        #load in the image
        image = cv.LoadImage(os.path.join(root,name), 1)
        match = detectFace(image, cascade)

        if match:
          #save a new image with a box round each face
          cv.SaveImage( fileLocation + "/face_" + name, match)
        t = cv.GetTickCount() -t
        print "\tTime = %gms" %(t/(cv.GetTickFrequency()*1000.0))

if __name__ == '__main__':
  cdir = "/usr/share/opencv/haarcascades/"
  cascade = cv.Load(cdir + "haarcascade_frontalface_default.xml")

  if len(sys.argv) != 2:
    print 'please provide a directory to read'
    sys.exit(1)
  readDirectory(sys.argv[1], cascade)
```



DAVID HUNT

David has been making projects for the Raspberry Pi since the early days. These include a Camera Controller, TimeLapse Rail, Focus Stacker, and even a Bark-Activated Doggy Door Opener. Oh, and let's not forget the PiPhone!

DavidHunt.ie

WATER DROPLET PHOTOGRAPHY

Have you ever wanted to capture those split-second photographs of water droplets colliding? Now you can with a Raspberry Pi-controlled solenoid and camera trigger!

You'll Need

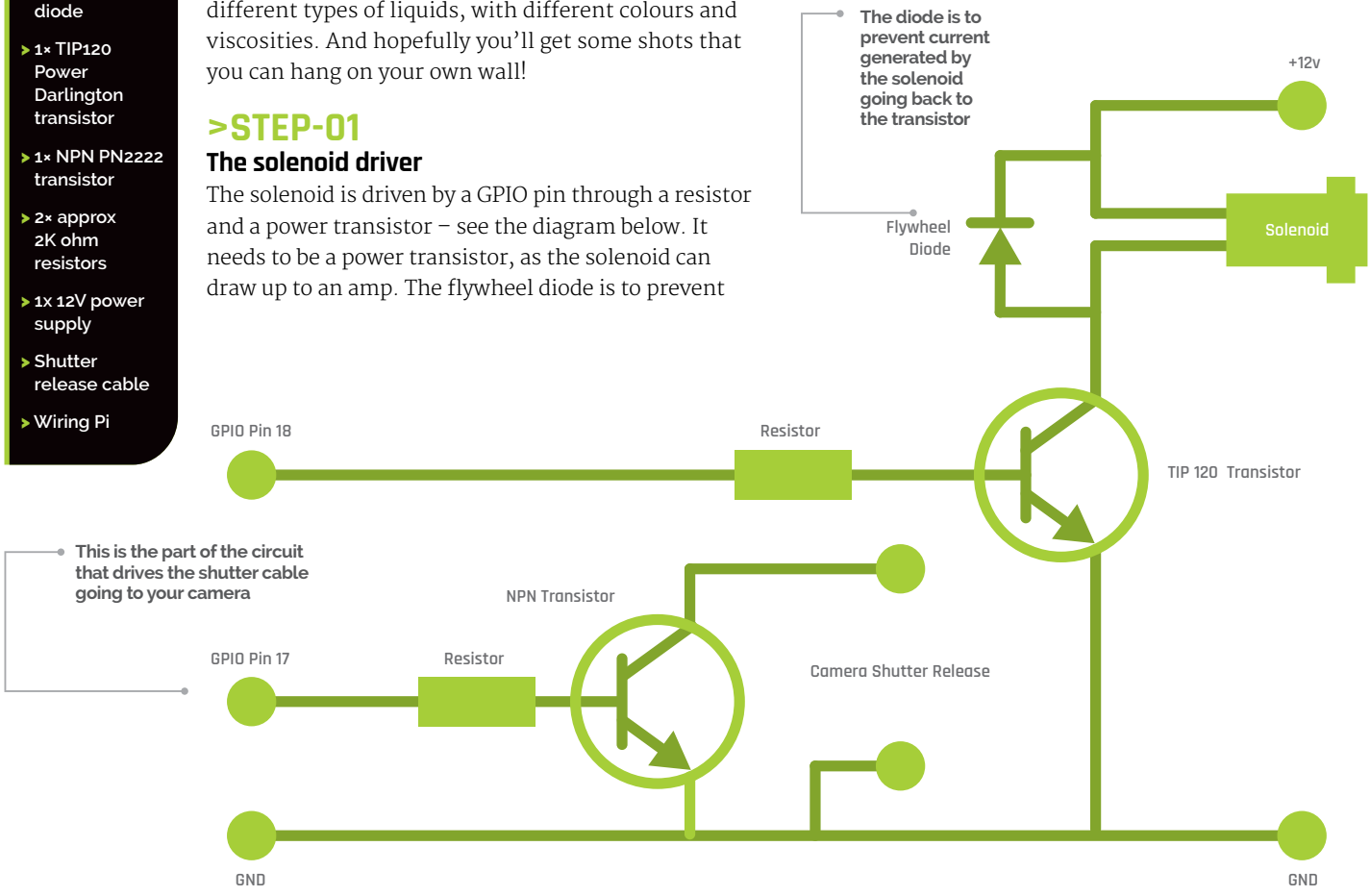
- Solenoid valve
- 1× IN4001 diode
- 1× TIP120 Power Darlington transistor
- 1× NPN PN2222 transistor
- 2× approx 2K ohm resistors
- 1x 12V power supply
- Shutter release cable
- Wiring Pi

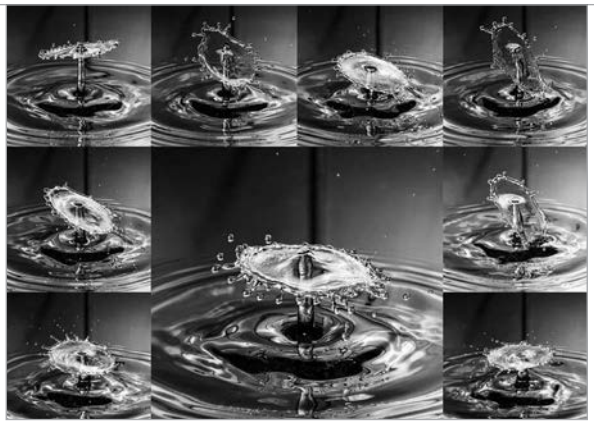
This tutorial shows you how to build a project that will allow you to capture those beautiful, carefully timed photographs where water droplets are colliding. From assembling the hardware with a solenoid, to writing the code to drive it, you'll be doing your own droplet collision photography in no time. After that, you can have all sorts of fun using different types of liquids, with different colours and viscosities. And hopefully you'll get some shots that you can hang on your own wall!

>STEP-01

The solenoid driver

The solenoid is driven by a GPIO pin through a resistor and a power transistor – see the diagram below. It needs to be a power transistor, as the solenoid can draw up to an amp. The flywheel diode is to prevent any current generated by the solenoid from going back into the NPN transistor. Once the GPIO pin goes high, the current can flow from 12V to GND, enabling the solenoid to open the valve and allowing the liquid to pass through. We only open the valve briefly, just enough to allow a drop through at a time.





Above An example of the type of image that can be achieved

>STEP-02

The camera shutter driver

The camera shutter is triggered by a low-power NPN transistor. DSLRs usually have a shutter release input which is shorted to ground, causing the camera to take a picture. In this project we're using a signal transistor to cause that (usually 3.3V) input to short, so we can get the camera to take a picture from the Python script on our Pi. You'll need to get the correct shutter release for your camera, but they can be sourced on eBay for under £5.

>STEP-03

Setting up the solenoid

This is the messy part! A drinks bottle with a small opening is ideal for attaching to the input of the solenoid. This type is often used for sports drinks, and can usually be pushed onto the solenoid input without any leaks. You can cut the bottle in half for easy top-ups. Apply 12V to the solenoid and you should get a stream of liquid through the valve; remove power and the valve should close. Attach it to the circuit you built in step 1.

>STEP-04

Trigger the camera

Now connect up your camera circuit and test it with the Python code. You will need to adjust the timings to get the camera to trigger at the right moment. But initially, you should hear two clicks of the solenoid and one click of the camera. You can adjust the timing in two ways: by changing the Python code, or altering the distance between the solenoid and the liquid container. In the code provided, the timings were good for a 50cm fall.

>STEP-05

Get the lighting right

You'll need to use a flash to freeze the movement of the liquid. Otherwise you'll get blurred images, even if your camera is on a tripod. An off-camera flashgun triggered by a sync cable is a really good idea, as it allows you to move the flash into all kinds of interesting positions. Oh, and keep the flash power low for shorter flash durations, giving you sharper images. And you can always use two or three flash units at lower power for shorter flashes still.

Drop.py

```
# Import the relevant Modules
import wiringpi2 #Learn more about this library at wiringpi.com
from time import sleep

# Set up the GPIO Pins
gpio = wiringpi.GPIO(wiringpi.GPIO.WPI_MODE_GPIO)
shutterpin = 17
solenoidpin = 18
gpio.pinMode(shutterpin,gpio.OUTPUT)
gpio.pinMode(solenoidpin,gpio.OUTPUT)
wiringpi.pinMode(shutterpin,1)
wiringpi.pinMode(solenoidpin,1)

# Release a drop of liquid
gpio.digitalWrite(solenoidpin,gpio.HIGH)
sleep(0.06)
gpio.digitalWrite(solenoidpin,gpio.LOW)

sleep(0.1)

# Release a second drop
gpio.digitalWrite(solenoidpin,gpio.HIGH)
sleep(0.05)
gpio.digitalWrite(solenoidpin,gpio.LOW)

# Wait for the droplet to hit the liquid container
sleep(0.12)

# Trigger the camera (which is set to manual mode)
gpio.digitalWrite(shutterpin,gpio.HIGH)
sleep(0.1)
gpio.digitalWrite(shutterpin,gpio.LOW)
```

>STEP-06

Adjust the camera settings

You should be shooting on manual setting, with a shutter speed as high as your camera will allow for flash. For Canons this is about 1/160th of a second, and maybe 1/250th of a second for Nikons. Use ISO 100-400 and then adjust your aperture till you get a decent exposure. You can then tweak the flash power down to get shorter flash durations, which will tend to freeze the motion of the liquid more. Open up the aperture more if you need to, but be aware that your depth-of-field will be reduced.

LIAM FRASER

Liam is the creator of the RaspberryPiTutorials YouTube channel. He is currently studying Computer Science at the University of York and has a special interest in embedded systems.
liamfraser.co.uk

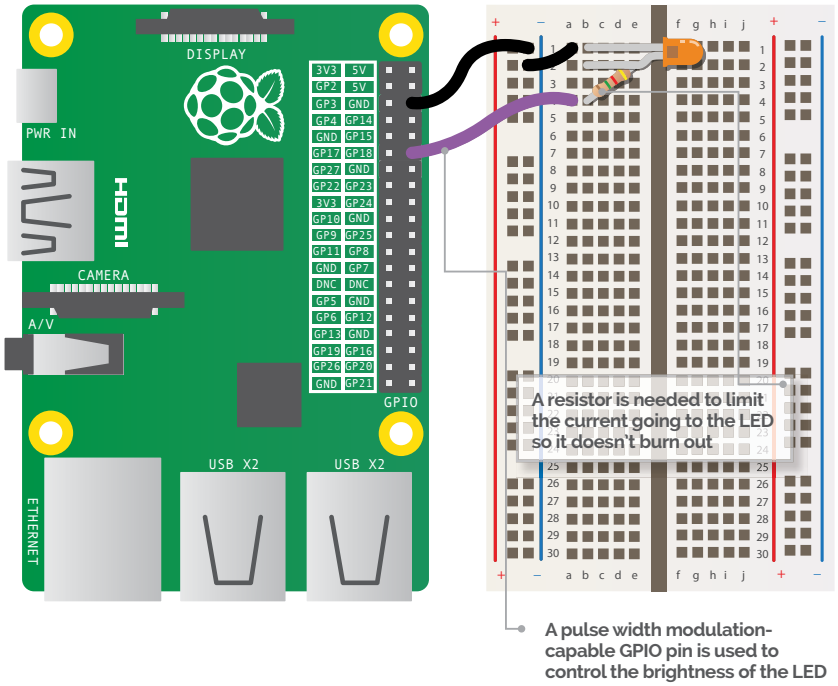
MAKE A PWM CANDLE LANTERN

You'll Need

- A coloured LED
- Breadboard
- Female-to-male jumper cables
- Resistor (100 ohm to 330 ohm)

Set a romantic mood with your Raspberry Pi by simulating a flickering candle effect using pulse-width modulation…

This tutorial is intended as a gentle – not to mention romantic – introduction to GPIO (general-purpose input and output) pins on your Raspberry Pi, and how to control them in Python. We'll be creating our romantic candle-like mood lighting using a random number generator to make an LED flicker at different intervals. In addition, its brightness will be varied using a technique called PWM (pulse-width modulation), which effectively controls what percentage of the time the LED is turned on. We will also take a look at the output of the pins on an oscilloscope, so that we can see how the code translates to the electrical signals that make things tick.



>STEP-01

Pick a resistor for your LED

A resistor will limit the current that flows through the LED. Different colour LEDs have different current limits, so you'll need to check the specifications where possible. 100 ohm or 220 ohm will definitely work, though your LEDs might end up being dimmer than usual. The equation for working out resistance is as follows:

R = (3.3V – LED VOLTAGE) / LED CURRENT

Our yellow LED needs a voltage of 1.8V – 2.2V and has a typical current of 20mA, so: **R = (3.3V – 2.0V) / 0.02** (which is 65 ohms). A resistor with a value between 65 and 130 ohms is ideal here, but a lower value will make your LED brighter.

>STEP-02

Setup the breadboard

Unplug your Pi and follow the breadboard illustration setup. Make sure you use the same GPIO pin we have, as only a couple are capable of pulse-width modulation (on the B+). We're using GPIO number 18 for PWM, which is described as **PCM_CLK/PWM0**.

The circuit path, as shown in the illustration, is **GPIO 18 > resistor > LED positive**. Finally, the LED negative leg goes into ground. The positive leg of an LED is usually longer. The negative side will have a flat edge rather than a circular one.

>STEP-03

Get coding!

Once you've wired up the project, power up your Pi and begin coding using an editor of your choice (or opening a terminal and typing **`nano candle.py`** will do).

Once we've imported the libraries we need, the **`setup`** function organises our program and starts PWM

Below This oscilloscope trace shows how LED brightness is controlled

for us. The **flicker** function sets a random brightness by calling the **set_brightness** function, then sleeps for a random time. This function is then wrapped up in an infinite **while** loop within the **loop** function, which handles the all-important cleanup of the GPIO library when **CTRL+C** is pressed by the user.

>STEP-04

Test your creation

Exit your editor and run the code by typing **sudo python2 candle.py** into a terminal (unless using Raspbian Jessie, you need root privileges to access the GPIO pins). Now you've tested it, you can exit with **CTRL+C** and we'll make it run at boot. This way, the Pi can run headless and not need any user interaction.

At the terminal, type: **sudo nano /etc/rc.local**, then add the following line: **python2 /home/pi/candle.py &** (but make sure you put this in the line before **exit 0**). Don't forget to save the changes.

The **&** means the script will go to the background and let the boot process continue. Notice how sudo isn't required because **rc.local** is executed as root. Reboot the Pi with **sudo reboot** to verify that it works.

>STEP-05

Packaging it up

Now that the script is started when the Pi boots, you could package it up into a nice container using a portable phone charger as a power supply. There are plenty of candle holders that can be fashioned out of paper if you search the internet. Paper is ideal, especially with lots of holes in, since the LED probably isn't throwing out much light.

>STEP-06

Presentation, presentation, presentation

The candlelight project is surprisingly effective, but presentation is key in matters of the heart, so you may want to spruce up your project before you use it on a loved one. Pretty lanterns are available very cheaply from most department stores; just make sure you select one that obscures the view of the interior. If the lantern isn't big enough to fit the Pi and breadboard, solder the resistor to the LED and hide the Pi behind it.

Below A candle lantern that obscures the view of the inside is perfect for disguising the LED and hiding your Pi

Candle.py

Language >PYTHON

```
import RPi.GPIO as GPIO
import time
import random

# Set the PWM output we are using for the LED
LED = 18

def setup():
        global pwm

        # GPIO uses broadcom numbering (GPIO numbers)
        GPIO.setmode(GPIO.BCM)
        # Set the LED pin as an output
        GPIO.setup(LED, GPIO.OUT)

        # Start PWM on the LED pin at 200Hz with a
        # 100% duty cycle. At lower frequencies the LED
        # would flicker even when we wanted it on solidly
        pwm = GPIO.PWM(LED, 200)

        # Start at a brightness of 100%
        pwm.start(100)

def set_brightness(new_brightness):
        # Sets brightness of the LED by changing duty cycle
        pwm.ChangeDutyCycle(new_brightness)

def flicker():
    # We want a random brightness between 0% and 100%.
    # Then then we'll hold it for a random time
    # between 0.01 and 0.1 seconds to get a nice flicker
    # effect. Play with these values to make the effect
    # suit your liking
    set_brightness(random.randrange(0, 100))
    time.sleep(random.randrange(1, 10) * 0.01)

# The wrapper around the flicker function makes sure the
# GPIO hardware is cleaned up when the user presses CTRL-C

def loop():
    try:
        while True:
            flicker()
    except KeyboardInterrupt:
        pass
    finally:
        GPIO.cleanup()

# setup the hardware
setup()

# start the flickering
loop()
```

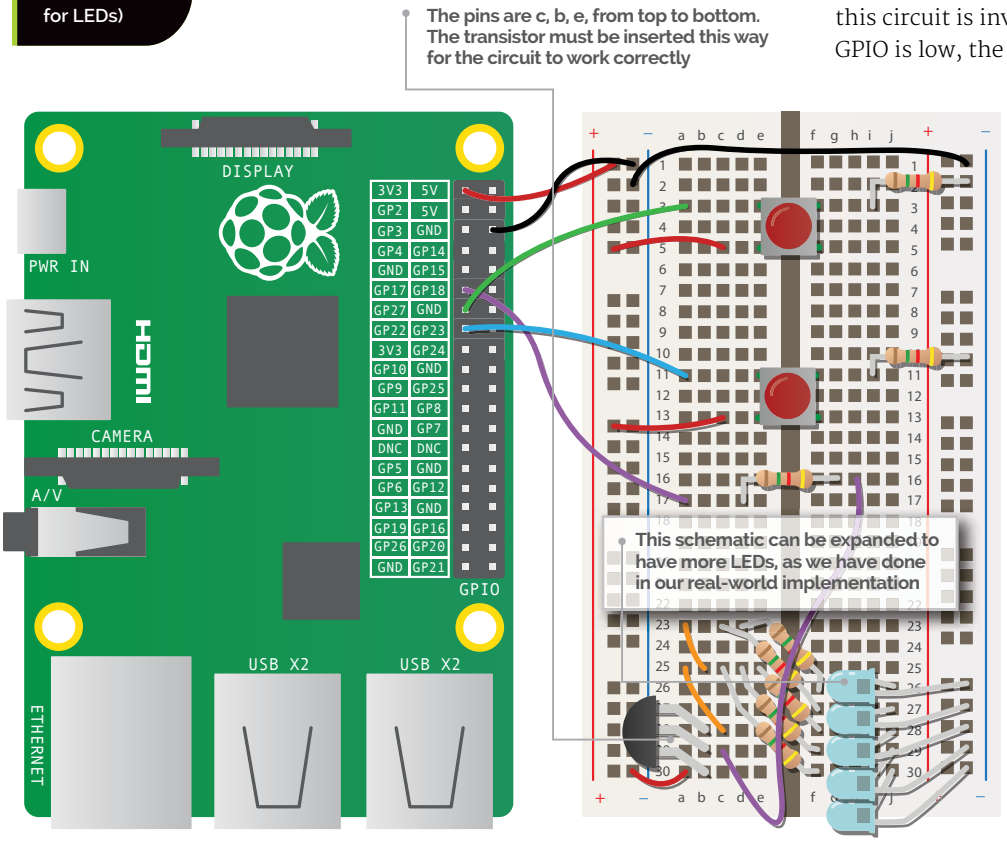
BUILD A STROBE LIGHT WITH A TRANSISTOR

You'll Need

- White LEDs (anywhere from 1 to 20)
- Breadboard
- A PNP transistor (2N2907)
- 2× push buttons
- Jumper cables
- Resistors (2× 1K ohm & 22 ohm for LEDs)

Using a transistor, it's easy to control lots of LEDs. In this tutorial, we're creating a strobe to help get the party started with our Raspberry Pi

This tutorial follows on rather neatly from the previous one (pages 96–97), where we emulated a candle using pulse-width modulation. In this guide we'll step things up a bit by throwing push buttons, a transistor, and lots of LEDs into the mix in an effort to create a strobe light that you can manually speed up and slow down. A strobe is a great accessory to any party and, with enough blue tack or duct tape, you should be able to mount this on a wall. For best effect, place it up high in a corner of a room, right where the two walls meet the ceiling.



>STEP-01

Using transistors

An LED typically uses 20mA of current. Since GPIO pins can't provide enough current to light lots of LEDs simultaneously, we're using a transistor to help us.

The 2N2907 transistor can pass 600mA, which is plenty for our needs. A PNP transistor has three pins: an emitter, a base, and a collector. Here, the emitter is connected to 3.3V, and the base is connected to a GPIO pin via a 1K ohm resistor. This pin is used to switch the transistor on and off. When the pin is 0V, current can flow; when it is 3.3V, no current can flow. Hence, this circuit is inverting: this means that when the GPIO is low, the transistor is on, and the LEDs light.

Since transistors have an amplification factor based on the current that goes through the base, we want to limit it with a resistor. Finally, the collector pin of the transistor is connected to the LEDs and eventually ends up at ground.

>STEP-02

The circuit path

Each LED path goes from the transistor, through a 22 ohm resistor to limit the current through the LED, and then into ground. Referring to the LED resistance equation from the previous tutorial, we only need a small resistor because white LEDs have a forward voltage of about 3V and a current of 20mA: **R = (POWER_RAIL – LED VOLTAGE) / LED CURRENT.**

So: **R = (3.3V – 3V) / 0.020** (which is 15 ohms, rounded up to the nearest practical value here).

WARNING!

This project uses strobe lighting. Do not use it if you or anyone are known to suffer from photosensitivity, never look directly at the LEDs, and never run it for prolonged periods or any faster than four flashes per second.

>STEP-03

Working with push buttons

Electricity always takes the path of least resistance. In the circuit diagram, you can see that a GPIO pin is connected to a leg of each push button. The leg is also connected to ground (0V) via a 1K resistor. This is termed a pull-down resistor because when the button isn't pressed, the signal is pulled down to ground. The other leg is connected to 3.3V. When the button is pressed, the two sides are connected again. There is less resistance to 3.3V and the GPIO input goes high, which we can detect in our code listing on the right.

>STEP-04

Wire up the circuit

Let's get the circuit built, but don't forget to turn your Pi off first. It's important to remember that each LED is connected to the transistor in parallel. This means that they are all effectively connected to the output of the transistor (and not chained in any way). This being the case, you can scale up the design on the breadboard as much as you want, within the specification of the transistor and power supply. In the case of our setup, that would be 20 LEDs or less.

>STEP-05

Event detection

The code for this tutorial is similar to the Candle code in the previous one, with one key difference: we're using events. The functions to speed up or slow down our strobe light will only be called when a rising edge (a transition from 0V to 3.3V) is detected on the button pins. This is a convenient way to handle button presses without having to keep checking them in your main loop.

>STEP-06

Test it out

Create and save the code on the right, then test it out using **`sudo python2 strobe.py`**. Pressing the button on the right will increase the speed of the strobe. Pressing the button on the left will slow the strobe down. Note that because the delay change is small, you'll have to press the buttons quite a few times. As always, you can start the script when you turn on your Pi, by adding it to **rc.local**:

```
sudo nano /etc/rc.local
```

Add the following line (before **`exit 0`**) and then save the changes:

```
python2 /home/pi/strobe.py &
```

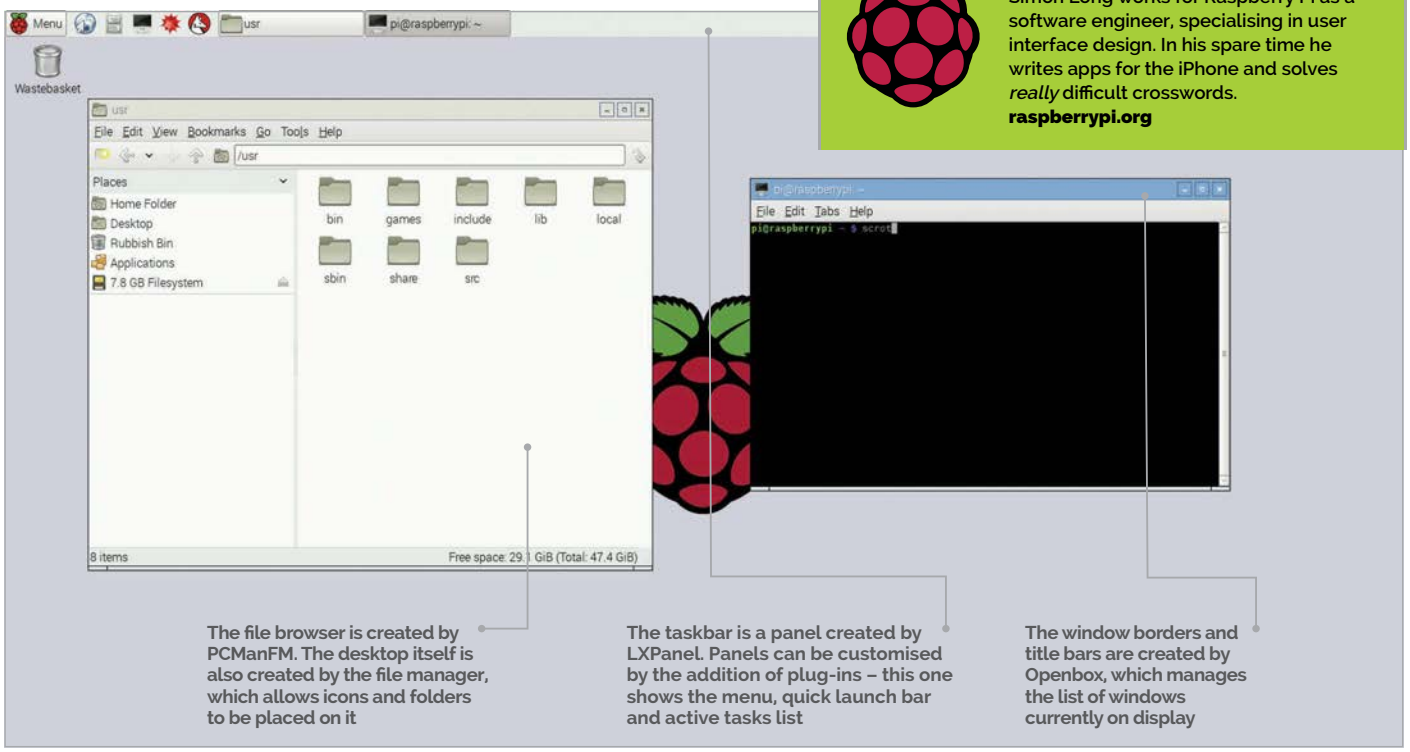
Strobe.py

Language: >PYTHON

```
# Import the GPIO and time library
import RPi.GPIO as GPIO
import time

# First we initialise some constants and variables
TRANSISTOR = 17
BTN_SPEED_UP = 27
BTN_SLOW_DOWN = 22
DELAY_CHANGE = 0.005

# Never use a strobe light any faster than 4 flashes per sec
DELAY_MIN = 0.125 # 1/8 = '4 on 4 off' flashes
delay = 0.2
def setup():
        # Next we initialise setup of the GPIO pins
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(TRANSISTOR, GPIO.OUT)
        GPIO.setup(BTN_SPEED_UP, GPIO.IN)
        GPIO.setup(BTN_SLOW_DOWN, GPIO.IN)

        # This will call a function when the speed up or slow down
        # buttons are pressed
        GPIO.add_event_detect(BTN_SPEED_UP, GPIO.RISING)
        GPIO.add_event_callback(BTN_SPEED_UP, speed_up)
        GPIO.add_event_detect(BTN_SLOW_DOWN, GPIO.RISING)
        GPIO.add_event_callback(BTN_SLOW_DOWN, slow_down)

def speed_up(channel):
        global delay
        # Take away the delay change value from the delay time.
        # Make sure the delay doesn't go less than the minimum
        # safe rate for use of stroboscopic lighting.
        delay = delay - DELAY_CHANGE
        if delay < DELAY_MIN:
                delay = DELAY_MIN

def slow_down(channel):
        global delay
        # Add the delay change value to the current delay
        delay = delay + DELAY_CHANGE

def loop():
        # The try statement makes sure we clean up properly
        # on a keyboard interrupt (Ctrl+C)
        try:
                # loop until the user presses Ctrl+C
                while True:
                        # Turn the strobe on, then wait for the
                        delay time
                        GPIO.output(TRANSISTOR, False)
                        time.sleep(delay)
                        # Turn the strobe off, then wait for the
                        delay time
                        GPIO.output(TRANSISTOR, True)
                        time.sleep(delay)
        except KeyboardInterrupt:
                pass
        finally:
                GPIO.cleanup()

# Now we setup the hardware, and start the main loop of the program

setup()

loop()
```

SIMON LONG

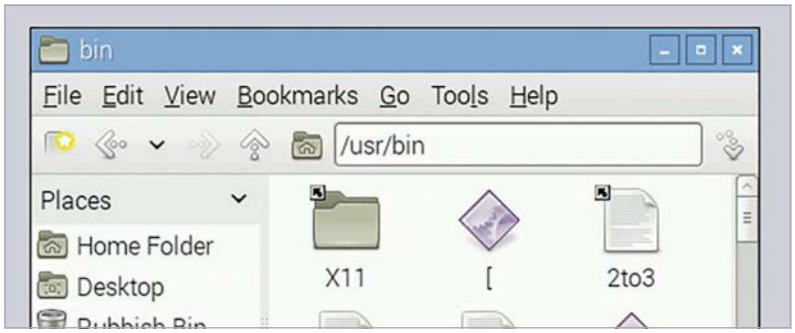
Simon Long works for Raspberry Pi as a software engineer, specialising in user interface design. In his spare time he writes apps for the iPhone and solves *really* difficult crosswords.
raspberrypi.org



HACKING THE RASPBIAN DESKTOP PART 1: WHAT IS LXDE?

In the first part in a series of four, **Simon Long** talks us through Raspbian's desktop environment, LXDE...

If you've used the desktop environment on Raspbian, rather than the command line, you've seen LXDE. The Lightweight X Desktop Environment is the software which creates the desktop graphic user interface (GUI) that appears when you type **`startx`** from the Raspbian command line – the desktop, the windows, the taskbar; all are parts of LXDE.



Below Openbox manages the appearance of window title bars, including the buttons to minimise, maximise, and close a window

LXDE is a user interface which sits on top of a system called X, a client-server windowing system. When an application starts, it requests a window in which to work from the X server program; the server also takes care of things like detecting mouse and keyboard input, and putting windows on the screen.

But X itself provides only the barest elements of a GUI – there are numerous environments that can sit on top of it and make the result look nicer, and LXDE is one of these. It's a very good fit for the Raspberry Pi because, as the name suggests, it is lightweight in terms of processor and memory usage, and so works well on a lower-powered device like the Raspberry Pi.

LXDE itself consists of several different pieces of software, all with specific tasks. The more important of these are described in the 'LXDE key components' box on the next page.