NEW! YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

Win! £200 WORTH OF PRIZES MUST BE WON!

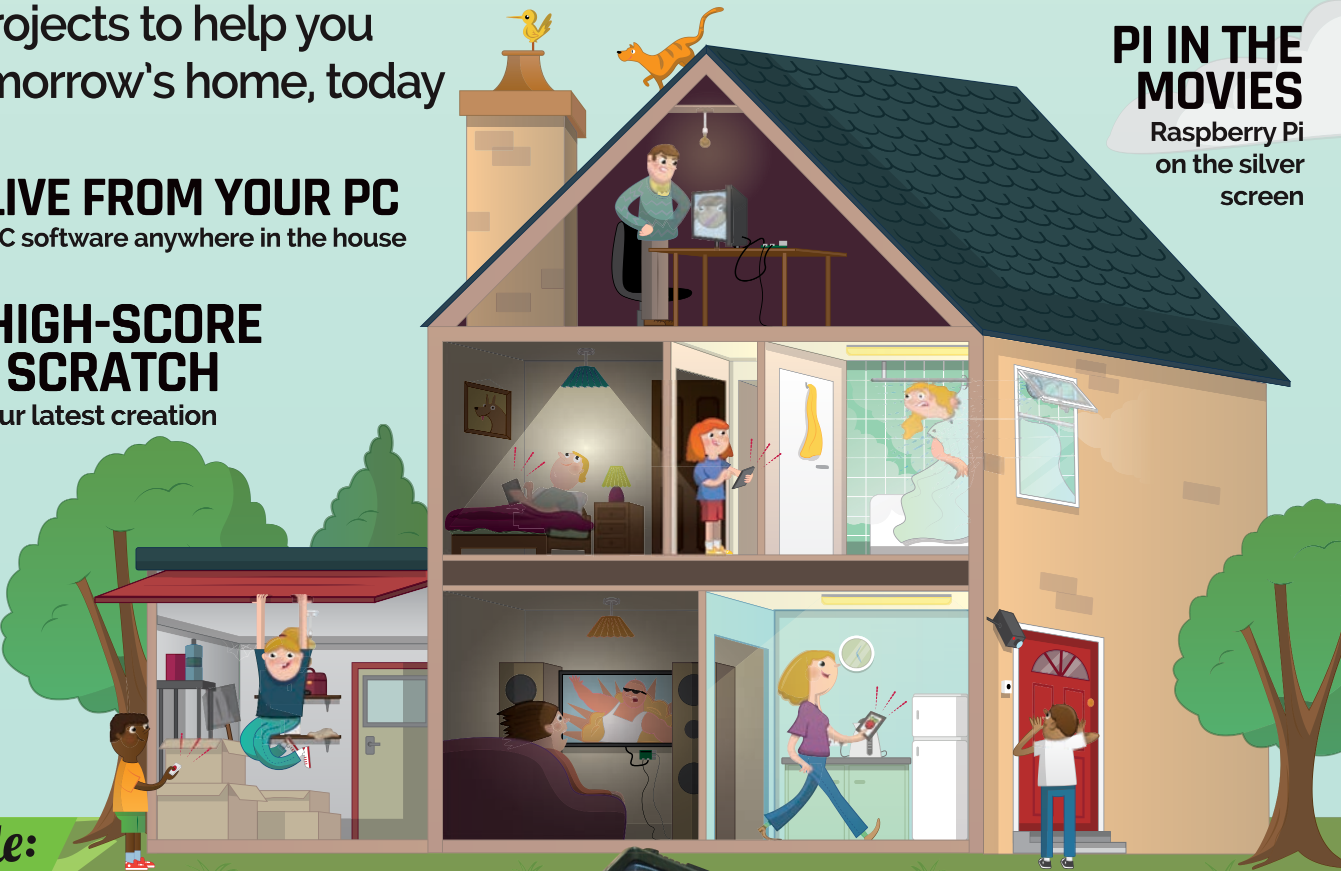
100 pages of hacking & making

Issue 37 September 2015

raspberrypi.org/magpi

RASPBERRY PI DIGITAL HOME

Eight Pi projects to help you create tomorrow's home, today



PI IN THE MOVIES

Raspberry Pi on the silver screen

STREAM LIVE FROM YOUR PC

Use the hottest PC software anywhere in the house

MAKE A HIGH-SCORE TABLE IN SCRATCH

Add polish to your latest creation

BUILD A FRIDGE MONITOR

Everyday engineering brought to life

Also inside:

- COLLISIONS IN PYGAME
- SONIC PI V2.6 REVIEWED
- HOW TO INSPIRE CHILDREN WITH STEM
- BUILD A COMPUTER VISION SEQUENCER

THE FALLOUT PIPBOY

How one man made science fiction fact, just in time for Fallout 4!

Issue 37 • Sept 2015 • £5.99

09

9 772051 998001

THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY

WELCOME TO YOUR OFFICIAL PI MAGAZINE!

The dream of a digital home has been around much longer than the technology needed to make it a reality. I can remember watching the wonderful foward-looking BBC program, Tomorrow's World, when I was growing up and wondering just what it would be like to live in the 'homes of tomorrow' that regularly featured on the show.

While technical marvels like optical storage, evergreen plastic grass, affordable home computers, electronic banking and mobile telephones have all become vital parts of our everyday lives, the digital homes of the future have still very much remained part of tomorrow's world. At least, until now.

This month we take you on a journey through eight Raspberry Pi-powered digital home projects that you can use to revolutionise the way you live today. Starting on page 18, we'll demonstrate and demystify things like automated lighting, Twitter-powered door bells, affordable home CCTV and smartphone-triggered garage doors (among other things), so you can save time and trouble by giving your home the power to better serve you. We might not be at the point where our own Robbie the Robot can warn us of the dangers of the modern kitchen, but I honestly think we're actually getting there.

Enjoy the issue!

Russell Barnes

SUBSCRIBE & SAVE 25%
SEE PAGE 26 FOR DETAILS

THIS MONTH:

FIND US ONLINE raspberrypi.org/magpi

GET IN TOUCH magpi@raspberrypi.org

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**

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

Contributors: **Sam Aaron, Tim Anderson, William Bell, Simon Brew, Mike Cook, David Crookes, Gareth Halfacree, Lucy Hattersley, Phil King, Sean McManus, Simon Monk, Martin O'Hanlon, Les Pounder, Matt Richardson, Richard Smedley, Sean M Tracey, Richard Waterworth & Robin Withers.**

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

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2051-9982.

NEW! YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Win! £200 WORTH OF PRIZES
RASPBERRY PI DIGITAL HOME
STREAM LIVE FROM YOUR PC
MAKE A HIGH-SCORE TABLE IN SCRATCH
BUILD A FRIDGE MONITOR
THE FALLOUT PIPBOY

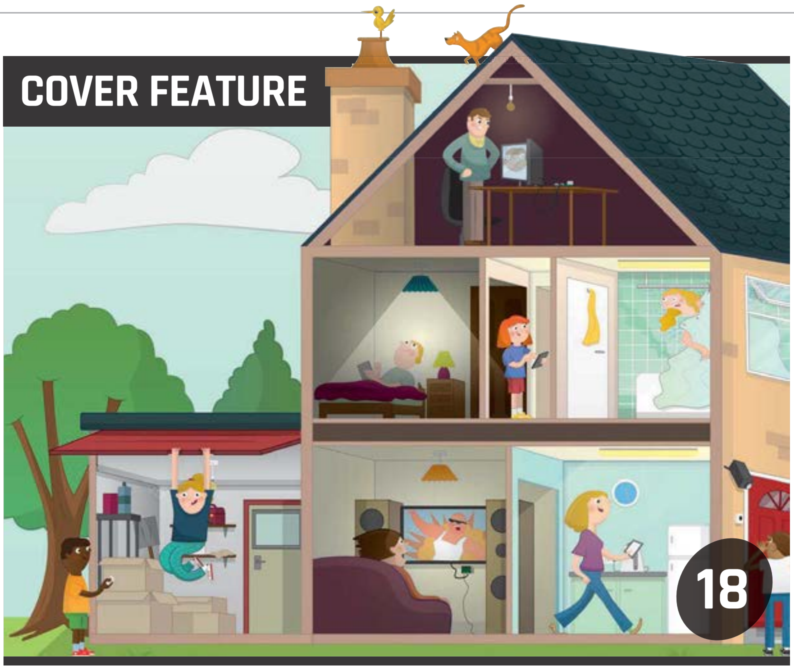
Contents

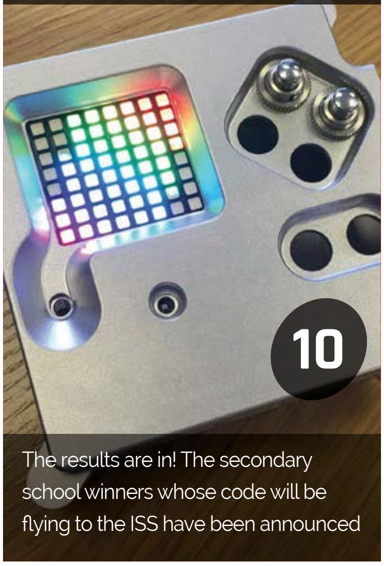
Issue 37 September 2015 raspberrypi.org/magpi

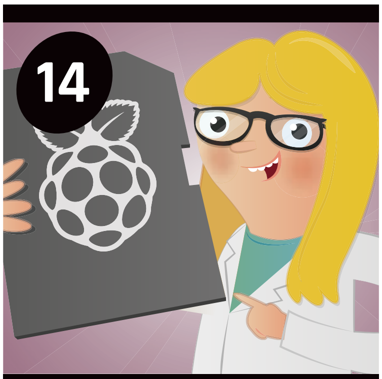
TUTORIALS

IN THE NEWS

COVER FEATURE







Contents

The MagPi OUT NOW!

You've been buying the print edition of The MagPi in droves. Here's just a small selection of our favourite pictures you've been sending us on Twitter...

NEW! YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Win! 10 RASPI 2s THANKS TO ELEMENT14.COM
100 pages of hacking & making
Issue 36 August 2015 raspberrypi.org/magpi
GET STARTED WITH RASPBERRY PI
How to unbox your credit card-sized PC and start hacking today!
RETRO CLASSICS The very best games of yesteryear you can bring back to life...
SIMULATE PHYSICS WITH PYTHON Create a whole solar system on your Raspberry Pi
3D PRINTING WITH RASPBERRY PI Control and view your printer with OctoPrint
CREATE A START SCREEN IN SCRATCH How to put the finishing touches to your latest creation
MINECRAFT SPLAT Recreate Nintendo's brilliant Splatoon on your Raspberry Pi
Also inside:
CONQUER THE COMMAND LINE
BUILD YOUR OWN MOOD LIGHT
MAKE A SIMON SAYS ELECTRONIC GAME
IQAUDIO: THE ULTIMATE AUDIO ADD-ONS
THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY
Issue 36 Aug 2015 £5.99

For the first time ever, *The MagPi* is available in print on the high street and we're really pleased to report that you've been buying it in droves! We're really pleased you've enjoyed the magazine and we look forward to many years of bringing you the very latest news, projects, tutorials and reviews dedicated to the world's favourite credit card-sized PC.

Some of you have had some difficulty finding the magazine on the high street, but don't worry – if you'd like a copy, but your local store doesn't stock it yet, you can simply visit the Raspberry Pi Swag Store online at **swag.raspberrypi.org** and have a print copy delivered anywhere in the world. If you'd like a copy of the magazine to arrive on your doorstep at the same time every month, you could subscribe (and save up to 25% in the process).

Visit **bit.ly/MagPiSubs** or see pages 26-27 for more details!

Phil Grimley hiding behind his hot-off-the-press first edition.

Allen Heard is looking particularly pleased with his latest purchase.

Our youngest reader to date, sent in by Chris Leach. It just goes to prove that you're never too young to learn how to get started with Raspberry Pi.

One of this year's Creative Technologists, Connor, with a selfie within a selfie. Thanks!

Raspberry Pi's own Eben Upton rearranging the shelves to ensure The MagPi is front and centre.

Raspberry Pi's own Liz Upton didn't have to pay for hers. There's got to be some perks to such a tough job!

The latest Picademy contingent in the largest magazine selfie we've had so far!

Image: Jarle Teigland

PI WARS II: THE RETURN

The dedicated Raspberry Pi robotics contest is back, bigger and better...

Christmas came early last December when a host of Pi-powered robots descended on Cambridge to contest the first ever Pi Wars. Well, Santa must have already read our 2015 wish-list, because the event is back! Pi Wars 2015 will be held on 5 December at the University of Cambridge Computer Laboratory, a larger venue that will enable 32 teams to compete, and accommodate over 300 spectators and a 'Show and Tell' area.

Organised by Mike Horne and Tim Richardson, the idea for Pi Wars originated from the high level of interest in robotics at the regular CamJam events run by the pair. "In particular, people were very excited about the possibilities of using robotics to teach about electronics and programming in schools and at home," says Mike. "We had both been avid fans of the BBC television series *Robot Wars*, and Tim came up with the initial thought of having some kind of robotics competition. The ideas just wouldn't stop flowing from then on!"

Mike tells us that last year's event was a great success, despite not everything going to plan and some frantic adding up of scores at the end. "All of the courses were well received and the majority of the teams were able to compete in the majority of the challenges, which is exactly what we wanted."

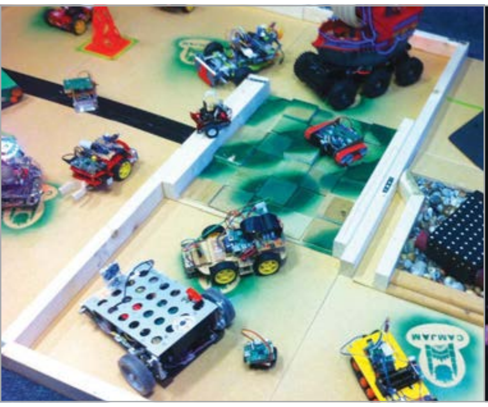
"We had all kinds of ages taking part," adds Tim,"from children as young as seven to grandfathers... We were particularly happy to welcome a lot of families, both as competing teams and as spectators."

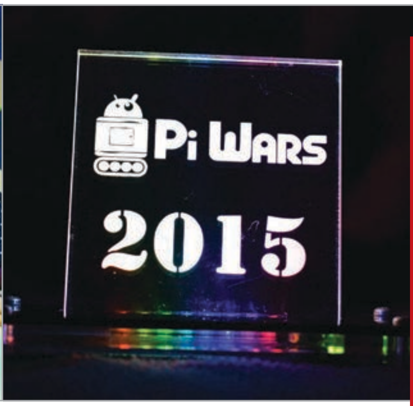
Mike and Tim were also pleasantly surprised by the range of entries, some of which were very slick while others were cobbled together from whatever was available. "We were particularly surprised by the success of some of the less professional-looking but extremely well-built entries," says Mike. "The most surprising robot was probably the one that looked like a pirate ship!"

THE CHALLENGES

Pi Wars 2015 will feature a broad range of remote-control and autonomous challenges...

- Obstacle course (RC)
- Skittles (RC)
- Pi Noon (RC)
- Straight-line speed test (auto or RC)
- Three-point turn (auto)
- Line follower course (auto)
- Proximity alert (auto)
- Code quality
- Aesthetics
- Build quality
- Blogging





Pi Noon

While the broad structure of Pi Wars 2015 will remain the same, Mike and Tim have a few changes and surprises lined up, including a mysterious 'Pi Noon' duel which replaces the Sumo challenge and requires robots to hold a wire. They're remaining tight-lipped on the details of this non-destructive challenge, but as Mike reveals, "One thing's for sure: it's going to be explosive!"

Other changes include Golf being replaced by a new Skittles challenge. The Obstacle Course is also being overhauled and will feature new obstacles that are modular and replaceable, while the Line Follower track will be improved from last year's 'paper and Sellotape' version. In addition, they're introducing a couple of new on-paper challenges, judging the build quality of the robots and how well the teams have blogged about their experiences of preparing for the competition.

Another change is in the way robots are categorised: this year they'll be split into two groups based on size rather than build cost. "The reason for the change is a practical one," explains Tim. "The size of a robot doesn't tend to change much past initial conception, whereas the cost of a robot could change enough to tip a team over into the higher category, which means we had to rejig things at short notice last year."

Finally, there will also be a 'Show and Tell' area where non-competing robots are exhibited. "At our regular Raspberry Jams, we always have a Show and Tell section and we wanted to include something similar at Pi Wars," explains Mike. There's also a practical reason: "Due to the way the main competition needs to be scheduled, there are some times where we have gaps. By having Show and Tell, it gives people something extra to look at!"

If it's anything like last year's event, Pi Wars 2015 will be a lot of fun. While the 32 competing teams have now been selected, spectator tickets (free for under-16s) are available via **piwars.org.**

> "The most surprising robot was probably the one that looked like a pirate ship!"

DESIGN THE PROGRAMME

A competition is being run to design the cover for the Pi Wars 2015 programme that will be given to every competitor and spectator. There will be a prize for the winning design. The closing date for entries is 1 October – visit **piwars.org** for more details.

I, ROBOTEER

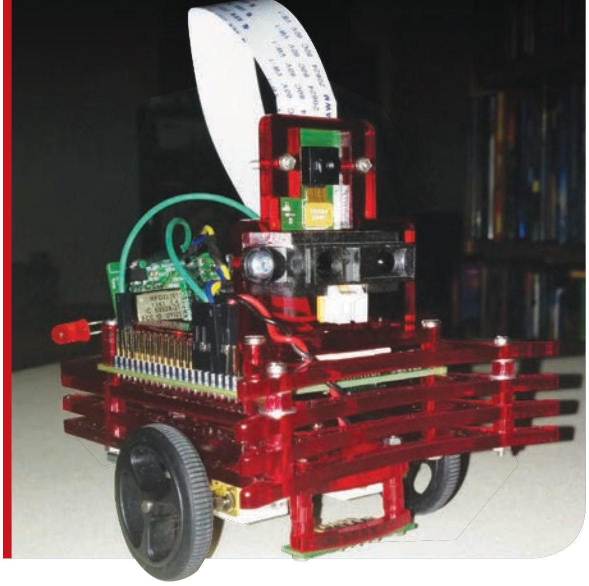
We chatted to Pi Wars robot maker Brian Corteil, who is also making this year's trophies…

What prompted you to enter the first Pi Wars?
It look like it would be a fun day, and it was an excuse to build a cool robot. PyroBot came about a third [of the way] up the field and it was the programmer to blame!

Were you surprised by some of the other robots at the event?
Yes, there were some cool robots on the day. My favourites were the ones from schools and made by young engineers. Oh, and PiBorg's DoodleBorg is amazing.

What was the atmosphere like? Was there a lot of robot rebuilding going on?
The atmosphere was great, very friendly and competitive at the same time. Yes, parts were added and removed from lots of robots. PyroBot had a change of motors and attachments for the different challenges, plus adjustments were being made all the time.

Can you tell us a bit more about the design of PyroBot and how you've improved it for 2015?
The design of PyroBot was inspired by Pimoroni's iconic case, and Cannybots' classic line-following robots. I even used an early prototype of their BlueBrain robot controller for driving the motors and controlled Pyro with their custom controller. I wanted PyroBot to be no larger than a Model B+ Raspberry Pi – in the end, I used the A+. This year, PyroBot has been upgraded with the latest version of the BlueBrain and will have caterpillar tracks, a complete overhaul of his code, plus a fancy new docking station.



ASTRO PI SECONDARY SCHOOL WINNERS ANNOUNCED

Results are in from the judges, and code from UK schoolchildren will run on the ISS, on Astro Pi: a Raspberry Pi with a Sense HAT

YOUR ASTRO PI

The Astro Pi's Sense HAT, with its 8×8 RGB LED matrix, accelerometer, gyroscope, magnetometer, pressure sensor, and combined temperature/humidity sensor, will be available to buy. Keep watching **raspberrypi.org** for updates!

When the competition started, the UK Space Agency's astronaut flight education programme manager Libby Jackson told *The MagPi*: "Every time we run competitions, children always have amazing ideas that we haven't even thought of." This competition was no exception, and the judges – from the UK Space Agency, ESERO UK, and leading UK space firms – who met at the Raspberry Pi Foundation in Cambridge were "blown away by how clever some of [the entries] were!" Judging followed the BBC's code of conduct for competitions – as used on the *Blue Peter* Mission Patch competition – and after hours of poring over entries some rather wonderful winners were announced, to accompany Major Tim Peake on his mission to the International Space Station (ISS).

Below Creative, cross-curricular, and entertaining: Thirsk pupils produced a great entry

Key Stage 3

The two winners of the KS3 (11-14 years) competition both impressed the judges. Andy Powell, of the Knowledge Transfer Network, said: "All of the judges were impressed by the quality of work and the effort that had gone into the winning KS3 projects, and they produced useful, well thought through and entertaining results."

Thirsk School's team, Space-Byrds, will use telemetry data provided by NORAD (North American Aerospace Defense Command, perhaps best known for its @noradsanta efforts), along with the real-time clock on the Astro Pi, to work out what country's territory the ISS is above and then show its flag on the LED matrix, along with a short phrase in the local language.

Thirsk's ICT head, Dan Aldred, is well known for contributions to the Pi community, including reference sheets for using the Astro Pi in the classroom and for the competition. Similarly, Martin O'Hanlon – who helped Key Stage 2 winner Hannah Belshaw from Cumnor House Girls' School implement her *Minecraft* data visualisation [see *The MagPi* #34 and **github.com/astro-pi**] – has put together a getting started tutorial for the Astro Pi board which will ensure that its educational use as a sensor board and data logger lives well beyond the end of the mission to ISS.

Each of these Astro Pi programs has been assigned an operational codename that will be used when talking about it over the space-to-ground radio. Thirsk has the 'ops' name FLAGS, while the other Key Stage 3 winner, an environmental system monitor by Kieran Wand, a year nine pupil at Cottenham Village College, will be referred to by mission control under the ops name MISSION CONTROL.

Kieran's program measures the temperature, pressure and humidity, and displays these in a cycling, split-screen display, raising alarms if these measurements move outside acceptable parameters. The judges noted that they were "especially impressed that code had been written to compensate for thermal transfer between the Pi CPU and Astro Pi sensors."

Key Stage 4

Westminster School's EnviroPi team (ops name: TREES) will use the Astro Pi NoIR camera in the cupola of the ISS to take pictures of the ground. These will then be analysed using false colour image processing to produce a Normalised Differentiated Vegetation Index (NDVI) for each image, which is a measure of plant health. One piece of code will run on the ISS to capture the images, and another will run on the ground after the mission, to post-process and analyse the images captured.

Team Terminal, from Lincoln University Technical College (ops name: REACTION GAMES), also won the prize for the best overall secondary school submission, earning them a photograph of their school taken from space by an Airbus or SSTL satellite. They made a suite of reaction games that record response times, to investigate how crew reaction time changes over the course of a long-term space flight. The judges noted that "this entry caused all work to cease during the judging for about half an hour!"

> "RADIATION took their Astro Pi and Camera Module to the Rutherford Appleton Laboratory and fired a neutron cannon at it to test it was working"

Radiation

Using the Pi for radiation is a very cool application indeed for the tiny board. Three pupils from Magdalen College School in Oxford (ops name: RADIATION) have used the Camera Module, blanked off against light entry, to produce just such a device. The blanked-off camera sensor still detects the impact of high-energy space radiation particles as tiny specks of light, the intensity of which is calculated by the OpenCV computer vision library.

Two of the schools seized fantastic opportunities to test their entries. EnviroPi's TREES had the chance to test their code by sending the board up in a light aircraft to take pictures of the ground, and RADIATION took their Astro Pi and Camera Module to the Rutherford Appleton Laboratory and fired a neutron cannon at it to test it was working. Their code can even compensate for dead pixels in the camera sensor, to allow for rough conditions during launch as well as space radiation damage. This is fortunate, as before it even faced the neutron cannon the camera lost a couple of pixels during the journey there! Note that while the Pi's image-processing pipeline does have automatic bad pixel correction, this will be disabled by the Astro Pi team, as it would interfere with the detection of cosmic rays.

To infinity, and beyond!

This December, Major Tim will be strapped to a cylinder containing several thousand tons of rocket fuel, and blast off towards the ISS with luggage containing all the winning Astro Pi boards, including the winners of the primary school competition (see *The MagPi* #34): Hannah Belshaw of Cumnor House Girl's School's *Minecraft* data visualisation, and Cranmere Code Club's very scientific astronaut detection system.

The SD cards have already been burned and code testing is well under way. Other entries are also on the SD cards, so if Major Tim has any downtime during the mission, some other code may be run manually – this means several schools will be on tenterhooks waiting to see if their experiments run too. They won't be the only ones: the whole Pi community will be watching the skies, and the internet, for news from space.

RETRO ALREADY?

Although the Astro Pi's Sense HAT is compatible with the Pi 2 (and the A+), the B+ board will be accompanying it into space. Stability and predictability in hardware and software is very important, as computer errors can have serious consequences when you are in a small metal box 250 miles from your home planet.

Because of this, in order to maintain the old computers on space shuttles, NASA has been noted in the past for collecting large stocks of vintage computer parts. The space shuttle landed after its final mission in 2011, still running its original 30-year-old computers with 8086 chips.

Older equipment was also more robust against cosmic rays and other radiation – fortunately for the Pi and the astronauts, this isn't a major problem inside the ISS.

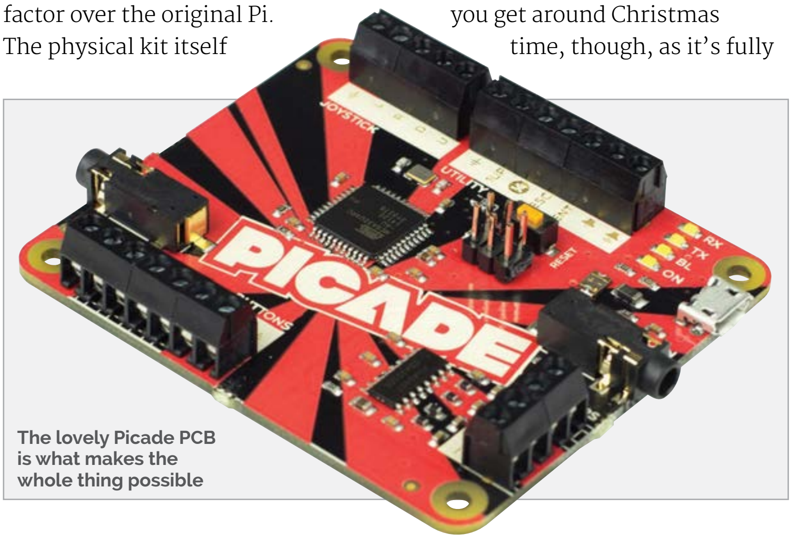
PICADE FINALLY RELEASED

Celebrate Pimoroni's third birthday with two versions of the Picade...

If you cast your mind all the way back to 2012, or Raspberry Pi Year One in comic-book origin terms, you'll remember one of the very first cases to really stand out for the Raspberry Pi was Pimoroni's Pibow: a laser-cut, acrylic kit you assembled yourself that was, and still is, very fun.

Not long after this success, Pimoroni started a Kickstarter for the Raspberry Pi-powered arcade machine kit, Picade. Made up of a special, custom PCB and a cabinet kit that made use of the team's established case skills, the Kickstarter ignited the imagination of hundreds of people. 625 of them backed the project, funding Pimoroni to actually make the arcade kit.

It's been a while since then, and though the Picade has been delayed a bit, the final product is now here, just in time for Pimoroni's third birthday. In that time, a lot of changes have happened in the Raspberry Pi scene, and the biggest change from Picade's original spec is that it now supports the Raspberry Pi 2 and B+ form factor over the original Pi. The physical kit itself looks lovely, with black powder-coated panels for that proper arcade machine look and feel, along with great art and a full six-button layout. Perfect for failing to overthrow M. Bison in *Street Fighter II* (again)!



The lovely Picade PCB is what makes the whole thing possible

> If you want to celebrate Pimoroni's birthday in style

Below The Picade Console is an excellent alternative if you want the same thing on a bigger screen

As well as the full original kit, Pimoroni also has a surprise in the shape of the Picade Console. It's a stripped-down version of the Picade which lacks the LCD screen, basically functioning as an arcade stick that you can plug into your TV to play *Sonic* and other classic games. It's a lot better than the 100-in-1 Mega Drive knockoffs you get around Christmas time, though, as it's fully moddable and will play any game compatible with the selection of emulators included.

To celebrate Pimoroni's birthday in style, go to **shop.pimoroni.com** and grab a Picade for £180, or a Picade Console for £90. You can also find the Pibow cases that made them famous three years ago. Happy birthday, Pimoroni! Here's to many more years of great Raspberry Pi accessories!

DEXTER INDUSTRIES

SAVE 15%
"MagPi15"
discount code

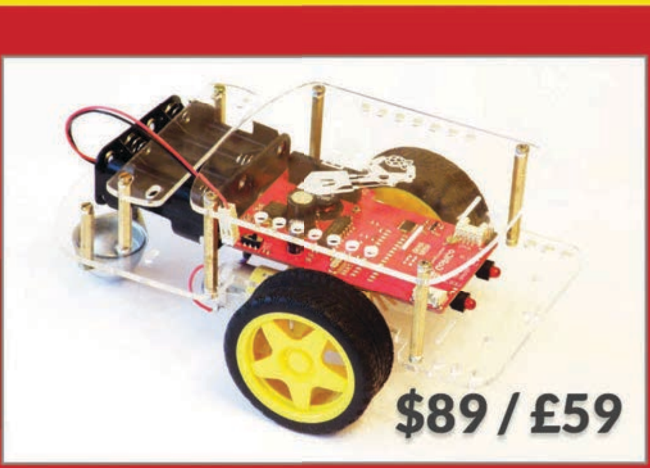
$89 / £59

BrickPi

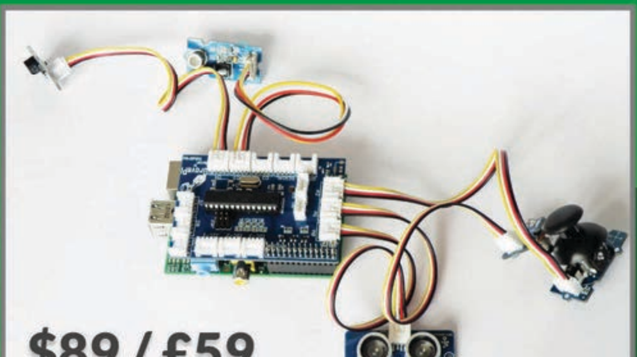
Build a LEGO robot with your Raspberry Pi!

GoPiGo

Everything you need to build a Raspberry Pi robot!



$89 / £59

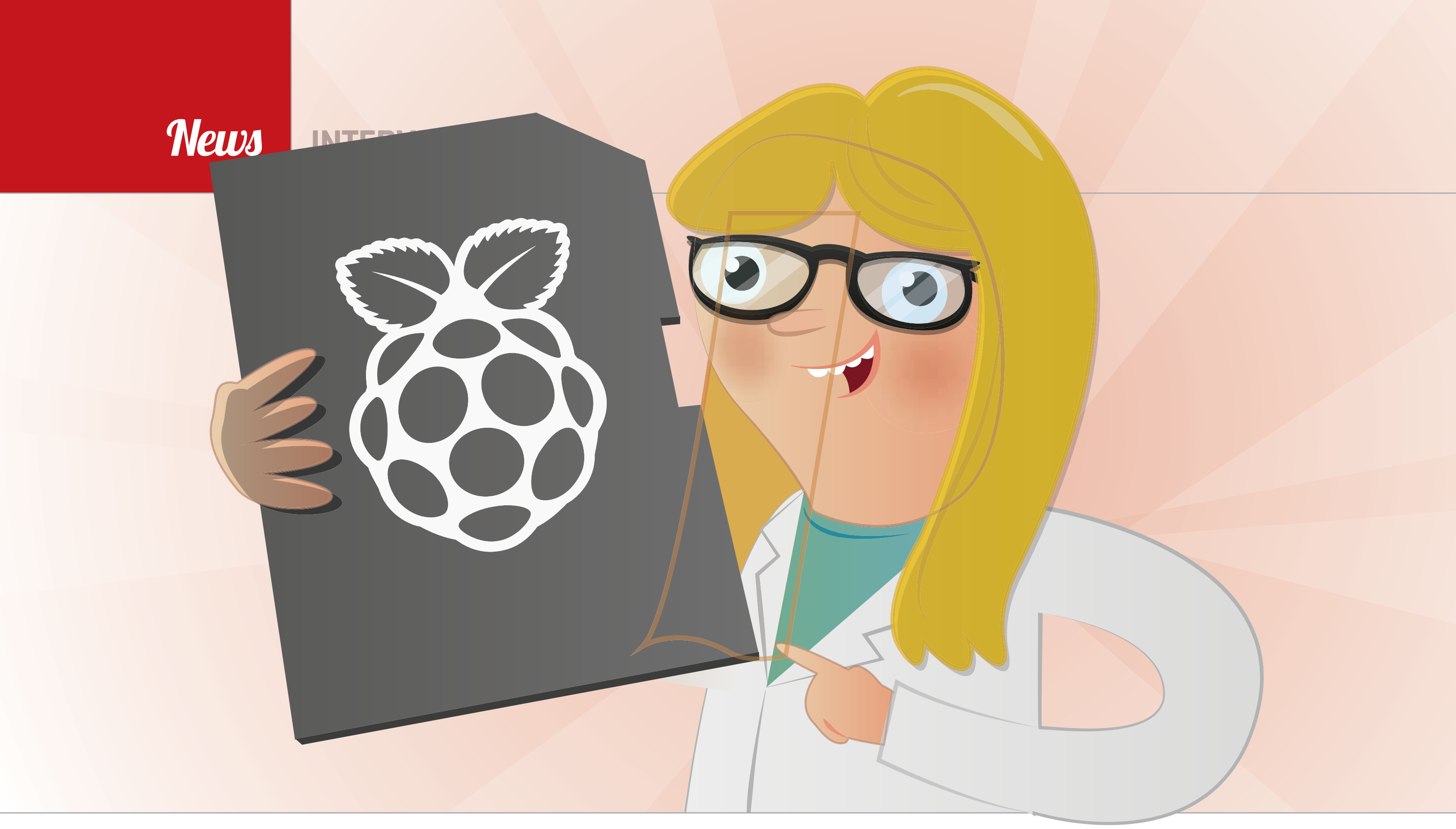


$89 / £59

GrovePi

Connect hundreds of sensors to your Raspberry Pi!

www.dexterindustries.com



CARRIE ANNE PHILBIN | PI, PYTHON & PIONEERING EDUCATION

The Raspberry Pi Foundation's education pioneer **Carrie Anne Philbin** tells us about her mission to revolutionise the teaching of computer science

"It's the best job in the world!" exclaims Carrie Anne Philbin about working for the Raspberry Pi Foundation. While she enjoys the chance to play with new technology and robots, however, she takes her role as the Foundation's education pioneer extremely seriously: "I always say I'm trying to change the world and I really believe that at Raspberry Pi, that's exactly what we're trying to do."

> I'm trying to change the world and I really believe that at Raspberry Pi, that's exactly what we're trying to do

As such, her job involves flying around the globe to talk to teachers and speak at conferences. We caught up with her at PyCon Australia, where she was giving a keynote speech.

As a newly elected member of the Python Software Foundation (PSF) board, Carrie Anne is keen to influence the direction of the popular programming language to make it easier to use in schools. "For me, it just felt like the PSF weren't really doing anything specifically for education. So one of the first things I did when I was elected to the board was send this really long email saying 'What is our mission statement with regards to education? These are the things I want to do'." This resulted in PSF founding member Marc-André Lemburg suggesting she set up a working group for education, which was announced by Carrie Anne at the recent EuroPython 2015 conference in Bilbao, Spain, and has since been approved by the board.

It seems things have come full circle, since Carrie Anne reveals that it was the Python community that was the first to welcome her when she was still working as an ICT teacher at an East London school. And, while

EDUCATION SUMMIT

EuroPython 2015 featured an Education Summit organised by Carrie Anne's team at the Raspberry Pi Foundation. While she concedes that it wasn't that well attended by teachers, mainly due to the newness of the event, she thinks the numbers will grow in subsequent years. In addition, she notes that more and more conferences are featuring education tracks, including PyCon Australia: "This one was really interesting because most of the presentations were made by teachers, which for me was really exciting because I'd never been to a developers' conference and had teachers so well represented. I keep asking people in my keynotes, if you run conferences then try and add an education track to them. So I definitely think they're going to become more popular."

she is a big fan of the community and says they can't really do enough for anybody, she admits to being really nervous before talking about some education-related Python bugbears in her EuroPython keynote speech. "Before I went, I spoke to a few Python developers, [including] Dave Jones, who created the PiCamera library… and he went 'Ooh, I wouldn't do that!' I spoke to [Sonic Pi creator] Sam Aaron too and he said 'Well, good luck!' However, afterwards the reaction has been really, really positive."

Python barriers

In that memorable EuroPython keynote, Carrie Anne listed a series of barriers to teaching Python effectively in schools. One was a confusing lack of consistency from developers in the naming of functions, although she admits this won't be easy to solve: "You're almost asking people to change the way they use the language." Another issue raised was the compatibility problems between Python 2 and 3, for which she has a simple solution: everyone should use Python 3!

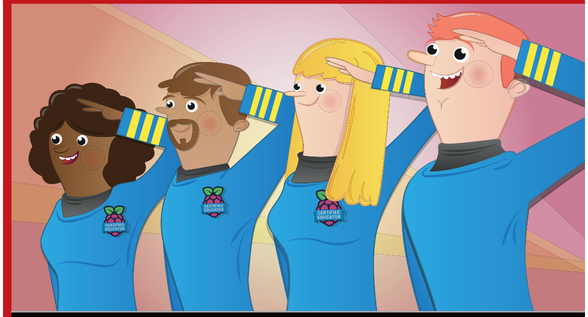
Possibly a bigger obstacle is the need for teachers to install extra libraries on a classroom of computers, which can result in many learners falling down early on. It's a particular hassle since, for the protection of pupils, many schools keep classroom computers offline or have firewalls that prevent the downloading of libraries. When Carrie Anne discussed this issue with Pygame Zero creator Daniel Pope, he came up with the idea of an 'educational bundle' of commonly used dependencies that would be downloaded on top of the standard Python library. Carrie Anne explains the concept: "Educators [will] download the bundle that includes Python 3, things like Pygame Zero, extra libraries they might need, and this new text editor."

The text editor in question is an intended replacement for IDLE, another of her Python barriers. Although there are many alternatives, she points out that IDLE is what children have first access to since it comes included with Python. So it needs to be quick and easy to get started with, particularly as it can be difficult

WHY PYTHON?

Out of the many text-based programming languages, Carrie Anne believes Python is the ideal one for teaching purposes. "[It's] probably one of the easiest languages to get started with. If you look at writing 'Hello world' in Java, you've got four or five lines, curly braces, really random words in there; whereas with Python you just write 'print' and then you've got your brackets and whatever your string is… that's one line." As well as its ease of use, she points out that Python has real-world applications: "It's not just an education language; it's also used every day by developers in places like Google, CERN, and NASA. It's a real language used by real developers around the world, with a growing community."

EXPANDING PICADEMY



Founded by Carrie Anne in early 2014, Picademy is a free professional development experience for primary and secondary teachers. Over the course of two days, 24 teachers get hands-on with computing and discover the many ways in which the Raspberry Pi can be used in the classroom. While the main Picademy programmes are run at the Raspberry Pi Foundation's Cambridge HQ, the initiative has recently been extended to Google Garages in Leeds and Birmingham in the UK; run by volunteers in these venues, there are standard two-day courses as well as 'Bytes' evening classes. There are also plans for four Picademy events in the USA next year, and Carrie Anne is constantly receiving requests to take it out to more countries. However, she says it will take some time to come up with a viable strategy for scaling it up: "We're still a small team and we need to think about how we can best reach teachers globally. The answer is probably not running a two-day course in every country around the world."

Picademy is branching out to more venues

for pupils to make the leap from visual programming languages like Scratch. While she's taken a look at Al Sweigart's IDLE Reimagined project (and hopes he'll join her working group), she feels it doesn't really go far enough. After talking to prominent members of the Python community, including Python creator Guido van Rossum, Carrie Anne reckons it'll make more sense to move away from IDLE and start again with something new.

"There are some good things about IDLE, as much as we all hate it," she laughs, "and one of the things we want to keep is how simple it is." Rather than having the text editor in a separate window, she'd like to see everything placed together in a single paned window. "That would be really helpful, and just some nice buttons that say 'Run' and 'Stop', and maybe some nice error reporting, but still be really, really lightweight."

The four Rs

Carrie Anne thinks computer science should be on a par with the other fundamentals of education: "Reading, 'riting, 'rithmetic, and Raspberry Pi… the four Rs!" Naturally, the Pi really comes into its own for physical computing, which is another important aspect of teaching computer science

Carrie Anne (far left) joins attendees at Picademy Leeds

> “We write materials, we run events, we have an annual diversity conference…”

since it gives learners immediate feedback. “They can see instantly what’s happening and say ‘oh, I can just modify this bit of code and something else happens – that’s really exciting’,” says Carrie Anne. Even so, younger children may lack the motor skills required to work directly with the Pi’s GPIO pins, which is where add-on boards prove useful: “Things like the Pi-Stop [traffic light] are fantastic, [while] Pimoroni’s Flotilla is a great idea to connect much bigger cables.”

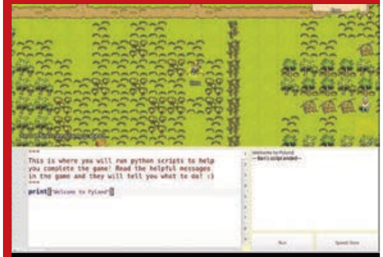
It’s not just the pupils who may need extra help, however. According to a recent *TES* survey, 60% of teachers didn’t feel confident about delivering the new computing curriculum in England. And with little government funding (about £170 per school) available for upskilling, the Pi Foundation’s free Picademy teacher training events are likely to be in even greater demand. Fortunately, the programme has recently been extended to provide courses in Leeds and Birmingham (see ‘Expanding Picademy’ box).

Carrie Anne is hugely excited about the new computing curriculum; launched in September 2014, it requires primary schoolchildren to be taught how to apply the fundamental principles and concepts of computer science, which includes learning how to program. She says it’s a good starting point, but thinks there’s still a long way to go and it isn’t necessarily being taught everywhere or taught well. “I think it’ll be interesting in five or six years’ time, [when] the children who at age five were learning this new curriculum move to secondary school, to see what kind of impact that has.”

However, she insists that it’s not all about coding. “We’re teaching computational thinking, problem solving… It being called a coding curriculum [by the media] is a real red herring. Coding is one strand out of three, and digital literacy is still part of the curriculum, [as is] understanding how the internet works.”

Carrie Anne strongly believes that improved teaching of computer science in general will benefit society in the future. “Not just the curriculum or children learning computational thinking, but if they learn about open source and free software, use it and are enabled by it, or make their own programs and make them free to use by other people, that kind of philosophy will help society greatly. Not all of them are going to grow up and be professional developers – what we want is for children to grow up and be able to solve problems in a creative way… and change society for the better.” and I think they will.”

WELCOME TO PYLAND



```
"""
This is where you will run python scripts to help
you complete the game! Read the helpful messages
in the game and they will tell you what to do! :)
"""
print("Welcome to Pyland")
```

```
Welcome to Pyland
---Bot's script ended---
```

After helping Sam Aaron to develop Sonic Pi, Carrie Anne is now involved in the creation of a new educational tool called Pyland. Developed by a team of interns led by Alex Bradbury at the Cambridge Computer Lab, this is a game that will run on the Raspberry Pi and teach children how to program in Python. It’s an attempt to solve the problem of moving children from a visual programming language to a text-based one.

“I’m really excited about this game,” says Carrie Anne. “I’ve seen demos and I think by the end of the summer we might have something we can start testing with children. [It’ll] be really exciting to have something that’s like Sonic Pi but is Python-based and maybe a bit different.”

WOMEN IN TECHNOLOGY

While Carrie Anne is impressed by the level of diversity in the Python community, including eight of the 12 members of the PSF board being women, she’s concerned about technology being a male-dominated field. One reason is a ‘leaky bucket’ effect, as women working in technology are “leaving in their droves”. She says one possible solution is having more female groups for women who code to meet up. “I’ve been to a few PyLadies meetups and actually, it’s been really good.”

The other problem is a ‘leaky pipeline’: “Girls in schools are less likely to end up in the tech industry. So I think, first of all, introducing the [computing] curriculum at the age of five will help… I think we need to keep up this idea that we [teach] girls and boys with this gender-neutral approach.”

Carrie Anne is also the chair of Computing At School’s ‘#include’ initiative to get more girls and minority groups into computing. “We write materials, we run events, we have [an annual] diversity conference. We ran something recently called ‘hack the curriculum’ where we had teachers and industry people together in a room creating a free wiki book… We’re just a group of volunteers, so I think what we’ve achieved has been staggering really.”

In addition, Carrie Anne runs the award-winning ‘Geek Gurl Diaries’ YouTube channel, which she says is still going, despite a lack of recent new content: “I am going to try desperately to make some more videos, but time is my enemy.”

DIGITAL HOME AUTOMATION

The Raspberry Pi is perfect for livening up the home. Discover some of the smartest home automation projects and make your house a whole lot smarter

Lucy Hattersley

Back in the 1950s, scientists seemed obsessed with creating the home of the future. In this mythical home, everything would be smart and things would just get done for you.

There's been a real resurgence in this idea lately due to the Internet of Things (IoT) concept, where tech firms are linking everyday items to the internet.

Thanks to the power of the Raspberry Pi, you can build the home of the future right now. You may not get to live like the Jetsons, but you'll end up with a smarter home and you'll learn a lot along the way.

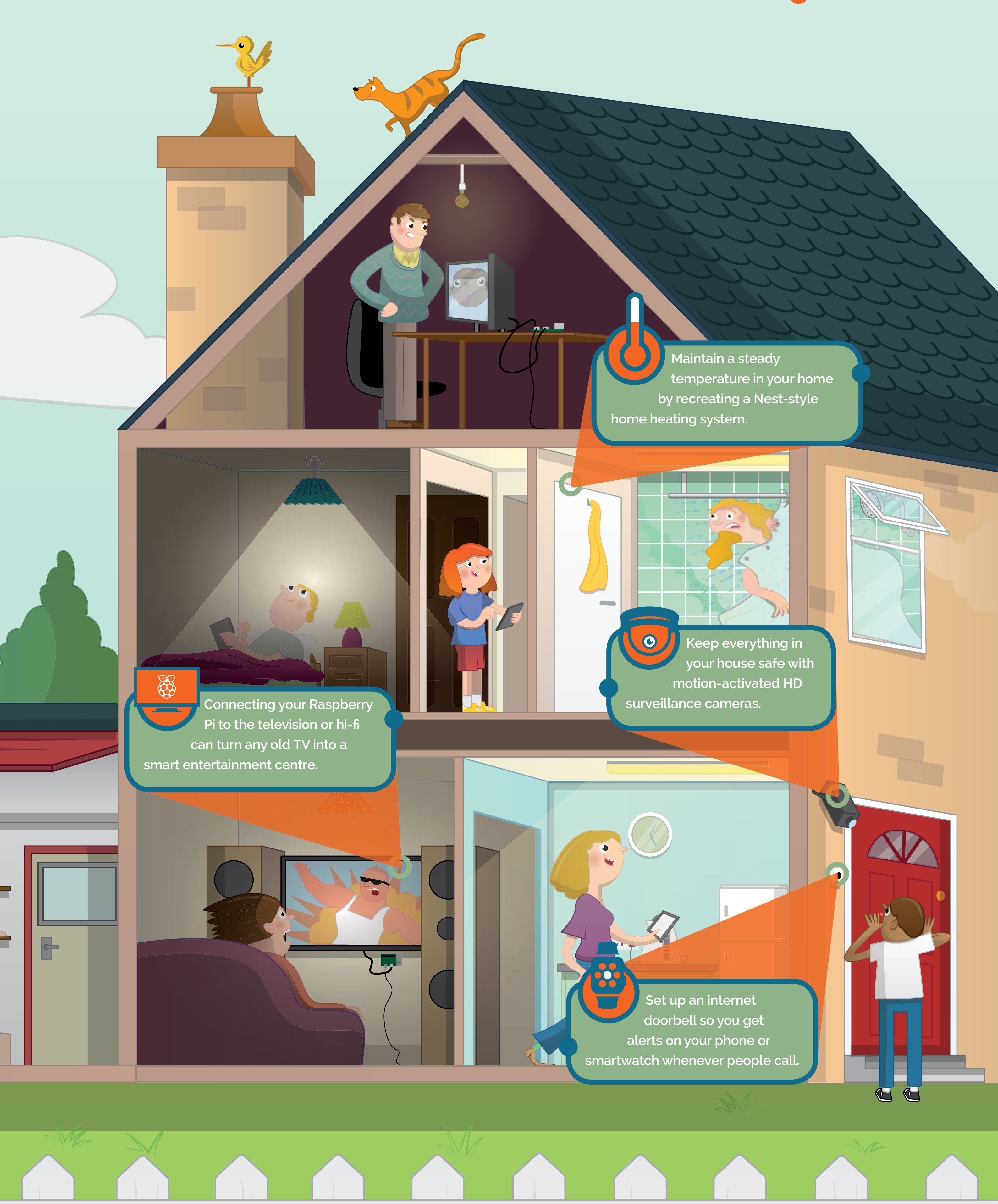
All you need are a few good tutorials. In this feature, we've found the best home automation projects around, and asked the project creators to share their secrets.

The Raspberry Pi is perfect for home projects. It's small, and it drains relatively little power. And with the Internet of Things promising to hook all our household gadgets up to smart data, there's never been a better time to learn about building a smart home.

The great news is that lots of people are way ahead, and there's a huge range of practical projects out there with good instructions to copy.

Read on to start building your dream home...

Hooking up a garage door to a Raspberry Pi enables you to open and close it from your smartphone.



Maintain a steady temperature in your home by recreating a Nest-style home heating system.

Keep everything in your house safe with motion-activated HD surveillance cameras.

Connecting your Raspberry Pi to the television or hi-fi can turn any old TV into a smart entertainment centre.

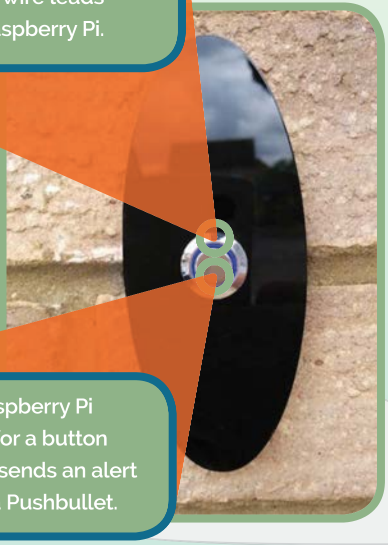
Set up an internet doorbell so you get alerts on your phone or smartwatch whenever people call.

DANIEL GARDEN

Daniel is an engineer, geek, and programmer who works at Plastic Logic in Cambridge.

muacksandglomps.com

The button is mounted in a custom laser-cut case. The wire leads through to the Raspberry Pi.



The Raspberry Pi listens for a button push, and sends an alert to your device via Pushbullet.

2/62
A minute ago
Garden Network
Doorbell: Ding Dong!

The watch vibrates and a message appears, letting you know that somebody has pushed the doorbell.

PEBBLYPI INTERNET DOORBELL

Daniel Garden never misses a visitor. His doorbell is hooked up to a Raspberry Pi, which sends messages to his smartphone and Pebble watch when pushed

You'll Need

- Rugged metal push-button adafruit.com/products/481
- Android device or iPhone
- Optional: Pebble smartwatch or Apple Watch
- 10K Pull-up resistor
- Multicore cable (e.g. telephone cable)
- PC speakers
- Optional: Adafruit Pi Plate
- Edimax Wi-Fi 150Mbps mini USB adaptor

The doorbell is a great place to start smarting-up your home. People push it and a speaker makes a noise, usally a ding-dong. It's simple and already has a button, but where's the fun in a regular doorbell?

It's easy to hook that button up to a Raspberry Pi and get it to notify you in a more creative way. And one person who's created a doorbell smarter than most is Daniel Garden. His PebblyPi Doorbell sends a message to his Pebble smartwatch whenever somebody pushes the bell.

"When I am in the garage, I can't hear the bell," says Daniel. The Pebble watch has haptic feedback (thanks to a vibration motor inside it) and its tap is hard to miss.

"A smart doorbell is one that can provide status information to other devices," Daniel tells us. "The PebblyPi can do this by initiating push notifications to my smartphone, and my watch, using a free online service called Pushover."

The parts needed to make a PebblyPi are reasonable easy to source. You need a push-button for the doorbell itself, some telephone cable, a WiFi module, and some old computer speakers.

Daniel wired his custom button up to a Raspberry Pi Model B and used an Adafruit Pi Plate to make the wiring easier. It's a pretty cheap project: the push-button cost around $5 / £3 and the WiFi module $12 / £8. The Pi Plate is $15 / £10, but this is an optional extra.

Daniel had a stylish mount for the button created using a laser cutter and hooked it up to the Raspberry Pi. Daniel explains: "The Raspberry Pi provides the button press detection. It also hosts the Python script that takes this button detection, plays an MP3, and pushes a notification to Pushover (**pushover.net**).

"My Raspberry Pi runs out-of-the-box Raspbian," continues Daniel. "I wrote the script that handles all the functionality in Python, and it autoruns on startup." You install Pushover on your smartphone. It's enabled through an API key and set up to work with the Pebble smartwatch. Pushover can also send messages to Android and iPhone devices, as well as to the Apple Watch.

Daniel wired his custom button up to a Raspberry Pi Model B and used an Adafruit Pi Plate to make the wiring easier

Once you have everything figured out, it's a quick installation and setup. Building the custom button took an hour with a laser cutter, and putting the code together took Daniel even less time. "It's easily a weekend project," he tells us.

Remember to "go open source and post your code so others can build on it," says Daniel. "There are a lot of closed smart home ecosystems, but when it comes to home automation, I want to know how it all works. Data security is a big deal and the only way to know that is through open source."

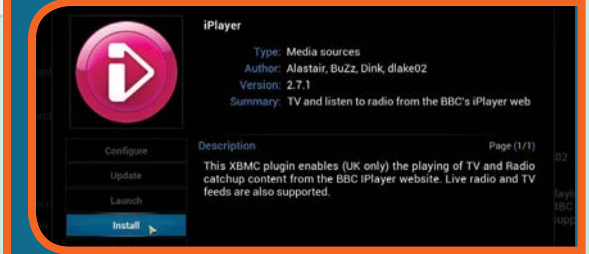
Whenever the doorbell button is pushed, the connected Raspberry Pi sends an alert directly to your iPhone or Android phone

If you have a smartwatch (such as a Pebble) connected to your phone, the doorbell alert will appear on that, too

SETTING UP A MEDIA SYSTEM

Installing software

It's easy to hook a Raspberry Pi up to your television or hi-fi and create a home media system. The main tool used to turn your Raspberry Pi into a smart television is called Kodi (**kodi.wiki/view/Raspberry_Pi**). The easiest way to set Kodi up on your Raspberry is via NOOBS. Use the Raspberry Pi's HDMI socket to connect directly to an HDTV.



iPlayer

Type: Media sources
Author: Alastair, BuZz, Dink, dlake02
Version: 2.7.1
Summary: TV and listen to radio from the BBC's iPlayer web

Description Page (1/1)

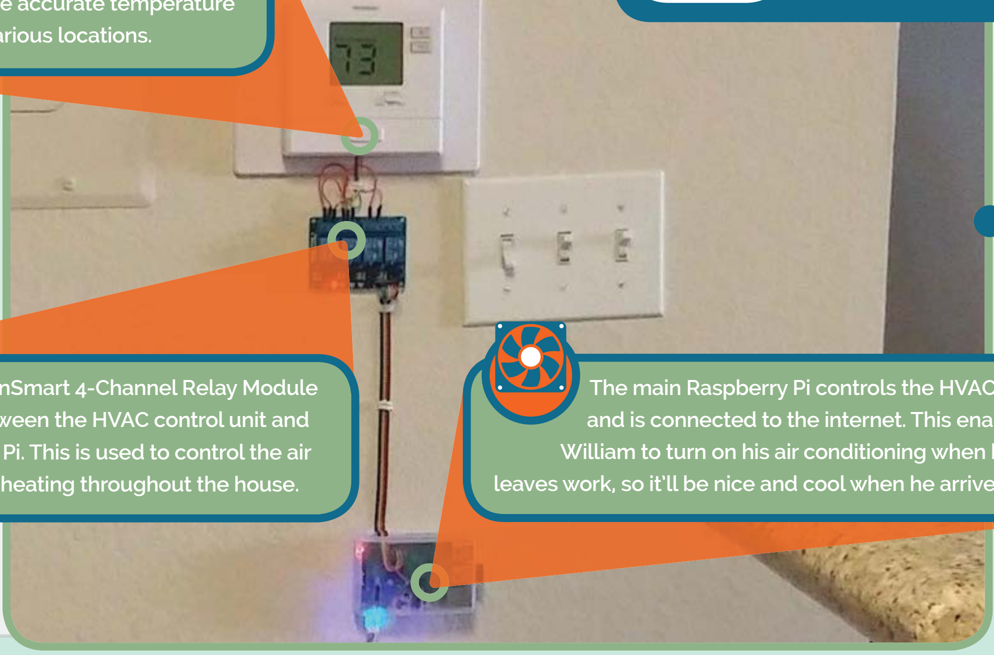
This XBMC plugin enables (UK only) the playing of TV and Radio catchup content from the BBC IPlayer website. Live radio and TV feeds are also supported.

Hook up to a hi-fi

If you're more interested in creating a music centre, use Pi MusicBox (**pimusicbox.com**).The Raspberry Pi's 3.5mm mini-jack can be used to plug directly into most music systems. In both cases, you'll probably need a USB Wi-Fi adaptor (**thepihut.com/products/usb-wifi-adapter-for-the-raspberry-pi**).

Making a remote

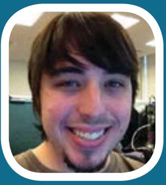
Controlling a home media system from a keyboard and mouse is largely impractical, so it's best to set up a remote control once the system is tested and working. The best way to do this is to turn a smartphone into a remote. Pi Musicbox has a built-in web interface, whereas Kodi (**kodi.tv**) has an official remote app available for both Android and iPhone.



Separate Raspberry Pi units are located throughout the house to provide accurate temperature information in various locations.

The SainSmart 4-Channel Relay Module sits between the HVAC control unit and Raspberry Pi. This is used to control the air conditioning and heating throughout the house.

The main Raspberry Pi controls the HVAC unit and is connected to the internet. This enables William to turn on his air conditioning when he leaves work, so it'll be nice and cool when he arrives home.



WILLIAM THOMAS

William Thomas graduated from Cal Poly Pomona and is now a software developer for EyeSeeThree.

willseph.com

BUILDING A NEST-STYLE SMART THERMOSTAT

William Thomas didn't buy a Nest like everybody else; he built his own home heating system with several Raspberry Pi units

William Thomas is no stranger to air conditioning, and his Raspberry Pi Thermostat is a great introduction to home temperature control.

"I've worked with Nest in the past," says William. "My last job used it for their air conditioning in the office. I had to do tech support on it, and that got me wondering if I could build my own with a Raspberry Pi. I could have bought a Nest, but I like the challenge of making things myself."

Maintaining a decent room temperature is an issue that plagues people throughout the world. "I wanted to solve the problem at my apartment of not being able to control or monitor the temperature remotely, as well as the temperature differences between the living room and the bedroom," explains William. "My bedroom heats up faster than the living room, but the original thermometer is in the living room, so there was a problem there."

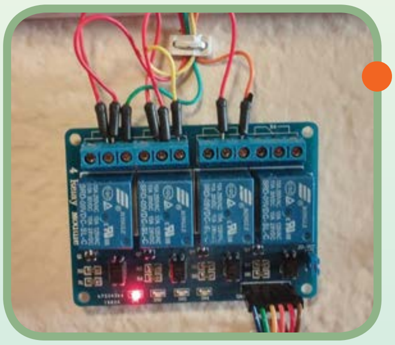
The Raspberry Pi thermostat hooks up to William's air-conditioning unit or heater. It can intelligently monitor the temperature, but, more importantly, it can also be used remotely to turn both systems on and off.

You'll Need

- 3× Raspberry Pi devices
- 2× DS18B20 temperature sensors uk.rs-online.com/web/p/temperature-humidity-sensors/5402805/
- SainSmart 4-channel relay module www.sainsmart.com/4-channel-5v-relay-module-for-pic-arm-avr-dsp-arduino-msp430-ttl-logic.html
- HVAC controller software github.com/Willseph/RaspberryPiThermostat

"Because the Raspberry Pi is connected to the thermostat," discloses William, "I'm now able to turn on the air conditioner or heater [while] away from home. During Los Angeles summers, turning the air conditioning on as I'm leaving work is really convenient."

The SainSmart relay is wired between the Raspberry Pi and the HVAC control unit



William used three Raspberry Pis in his project. Two were hooked up to thermometers to monitor temperature, while the third was connected to his HVAC (heating, ventilation, and air conditioning) control unit. The layers of software are placed into separate Raspberry Pis and they all communicate with each other.

"I have one Raspberry Pi attached to the HVAC controller on the wall," William tells us, "and I have multiple Raspberry Pis that each have a thermometer wired to their GPIO pins." William used DS18B20 digital sensors to monitor temperature.

The other Raspberry Pi is connected to the HVAC unit via a SainSmart 4-channel relay module. The multiple Raspberry Pi units increase the cost of the project, but aside from that, the parts are relatively cheap. "Each temperature sensor was about $4 [£3]," reveals William, "and the relay module was under $10 [£6]. I had some GPIO jumper wires left over from previous projects, but they're pretty inexpensive."

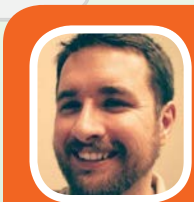
You could get away with just two Raspberry Pi units, or even one if you place the temperature sensor with the air-conditioning unit, although this would give an inaccurate reading of the temperature. "The Raspberry Pi acts as the internet bridge for the project, as well as the HVAC controller.

"Because I'm using two thermometer Pis (one in the living room and one in the bedroom), I'm able to get a much more accurate average temperature within my apartment."

All of the Pi software was written in Python and is available on GitHub. "The web interface uses basic PHP and MySQL," says William. "For the Raspberry Pis, I developed the system using a 'separation of responsibilities' approach which became five separate Python scripts that do different things. The main script for the HVAC controller Pi takes into consideration the set temperatures and the current temperatures, and interacts with the relay module accordingly. The main script for the thermometer Pis interacts with the DS18B20 sensor to record the current temperature at that location."

It's not an easy project to take on, but building a smart thermostat is very rewarding. William tells us he worked for a month on and off, but he got the software done weeks before installing the HVAC.

However, it's essential that you make sure you know exactly what you're doing when working with electrical devices. "Do not mess with mains electricity," warns William. "The HVAC panel in my apartment used a low-voltage DC current (which made the relay module somewhat overkill). However, some are built using mains electricity, meaning high-voltage AC current (in the US, anyway). Always test home circuits with a multimeter, and know where your breaker box is and how to use it. The worst thing you could ever do is start touching random wires inside your walls without knowing what you're doing. Safety is always the most important thing."



CHRIS DRISCOLL

Chris is a engineer from Sioux Falls, South Dakota. He works as a personal systems analyst at Sanford Health.

driscocity.com/idiots-guide-to-a-raspberry-pi-garage-door-opener/

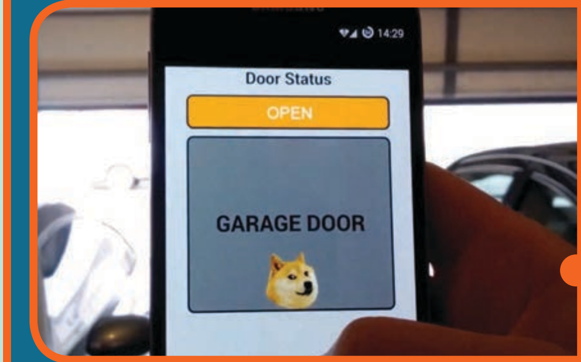
GARAGE DOOR OPENER

Chris Driscoll's Idiot's Guide to a Raspberry Pi Garage Door Opener is the perfect way to learn about door automation

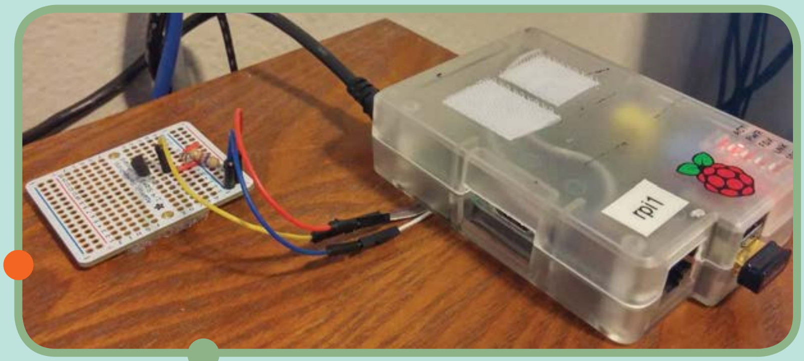
Garage doors are inherently annoying. You typically have to get out of your car, open the garage door, get back in the car, drive into the garage, get out of the car again, and then close the garage door.

It's no wonder that garage doors are one of the first things people look to automate in the home. "I had never used, or really thought of using a Raspberry Pi for anything at all, or anything related to home automation," says Chris, "but this really piqued my interest. It's just really cool."

His blog post details how he went about building the automated garage door. Like William Thomas's smart thermostat, he uses a SainSmart 4-channel relay module between the Raspberry Pi and garage door opening system. The relay module enables you to control equipment with a large current.



A web app is accessed on a smartphone to connect to the Raspberry Pi. Pushing the on-screen button opens or closes the garage door

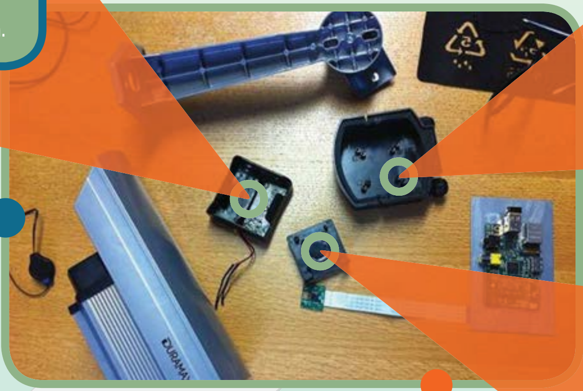


This Raspberry Pi is connected to a temperature sensor, and relays this data to the main Raspberry Pi unit connected the air-conditioning unit

CHRISTOPH BÜNGER

Christoph Bünger is the CEO of Scavix Software Ltd, his own software development and consultant company.

scavix.com/christophb/



A USB Wi-Fi adaptor is used to connect to the Raspberry Pi and enables video viewing from a remote location.

A fake CCTV case is used to house the Raspberry Pi and Pi Camera Module.

The Pi Camera Module is attached directly to the Raspberry Pi board.

HD SURVEILLANCE CAM

Create a ‘fake’ CCTV cam that’s better than the real thing. This HD Surveillance Cam keeps an eye on your things when you’re not around

You’ll Need

- Raspberry Pi
- Pi Camera Module
- Fake CCTV housing
- Motion lavrsen.dk/foswiki/bin/view/Motion/

Few people are keen on the high number of CCTV systems in use in the UK, but there’s a good case for watching over your own place with a remote camera.

Surveillance cameras can be extremely expensive to purchase, but with a Raspberry Pi and the Pi Camera Module, you can create a camera recording system that’s much smarter than your average surveillance cam.

Christoph Bünger took things a step further, and created a Raspberry Pi camera unit housed inside a fake CCTV camera case (these are sold on Amazon to deter would-be thieves). “The very special thing about this cam,” reveals Christoph, “is that it uses a camera housing that is sold very cheap on the net as fake security cams. These housings are perfect for the Raspberry Pi and cost only a few euros.

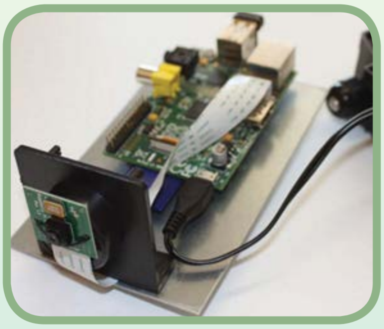
“Our Raspberry Pi surveillance cam project is a manual for everybody to build their own surveillance cam to monitor a house, office, or whatever. It’s based on the Raspberry Pi, which is very versatile and cheap. Everybody with beginner-level skills in electronics can build it.”

Even though it’s cheaper, and housed in a fake case, the Raspberry Pi Camera Module is much more capable than CCTV cameras that typically cost thousands of pounds. The Pi is connected via SSH to any computer on your local network, which can then snap photographs and record video. You can also access a live-stream of the camera remotely,

> With a Raspberry Pi and Camera Module, you can create a camera recording system that’s much smarter than your average surveillance cam

If you position the Raspberry Pi carefully inside the CCTV housing, it’s difficult to spot, and most would-be thieves won’t examine the insides of the housing

The camera is installed outdoors and looks just as convincing as a regular CCTV unit, only it's far more powerful



Christoph's smart surveillance project places a Raspberry Pi and Camera Module inside a fake CCTV camera housing

so it's ideal for keeping an eye on your property whenever you're away from home.

"The Raspberry Pi captures the stream from the Camera Module and saves it to the SD card or a shared folder on a network PC when it detects movement," explains Christoph. "So you can look through the videos later and see what happened when you were not at home."

In order to build the project, you'll need a Raspberry Pi, official Camera Module, and that fake CCTV housing.

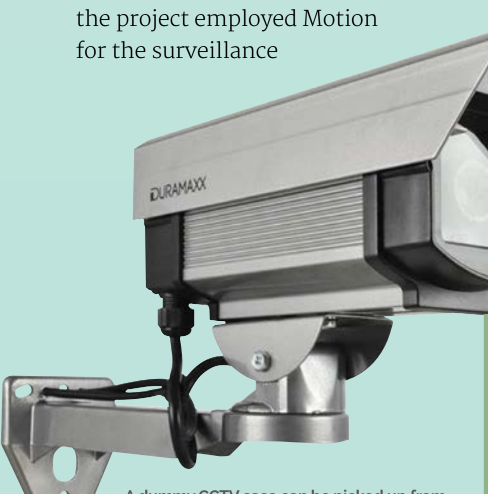
"The Camera Module is the most expensive part," says Christoph, "but even that only costs around 25 euros." You can pick up a fake CCTV camera housing from Amazon for as little as £10 / $16.

"It took us a half day figuring out how to assemble the hardware," says Christoph, "then another few hours on installing and configuring the software."

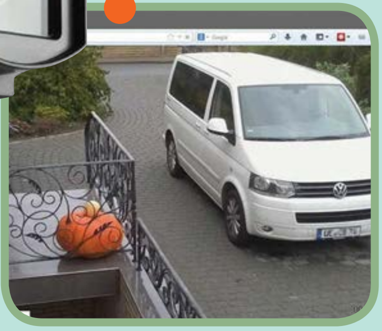
While the original version of the project employed Motion for the surveillance software, Christoph says he's now using motionEye (**bitbucket.org/ccrisan/motioneye/overview**).

After putting the project together, Christoph uploaded a tutorial to CodeProject (**bit.ly/1Es982j**) and step-by-step instructions to Instructables (**bit.ly/1Ms6j9C**). These tutorials have had over two million views so far, so there are a lot of other people out there interested in DIY home surveillance.

Whenever you're making a project, don't give up, advises Christoph. "There are a few great tutorial websites available that describe how to get to a goal with pictures and step-by-step instructions."



A dummy CCTV case can be picked up from Amazon for less than £10 / $16. It's perfect for housing a Raspberry Pi and Camera Module



The HD surveillance camera is much more adept than a regular CCTV cam. Here we are looking at a live feed from the camera inside a web browser

MORE PROJECTS

WiFi cat feeder

Create your own cat feeder to look after your feline friend whenever you're not around.

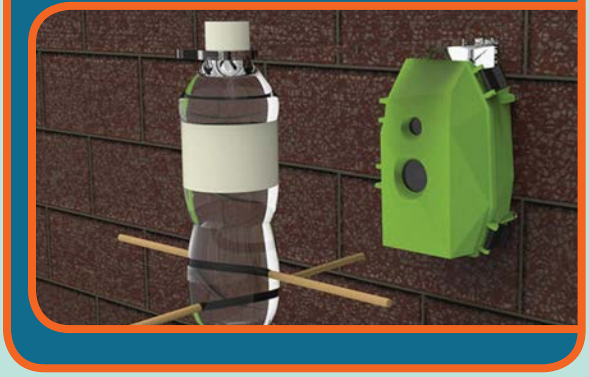
David Bryan's cunning Cat Feeder (**bit.ly/1gGSJ4u**) is a purr-fect project to get your claws into (sorry!). The Raspberry Pi controls servos that dispense food to Bryan's girlfriend's two cats.

PiPlanter

If you have houseplants, then build a PiPlanter (**esologic.com/?p=665**) to look after them. The PiPlanter monitors temperature, ambient light, humidity and soil humidity, and irrigates your plants accordingly. It even sends all the data to a MySQL database so you can check graphs on their growth.

Wildlife Cam Kit

There's no reason to automate your house and stop at the garden. If you want to take your Raspberry Pi outdoors, check out Naturebytes' Wildlife Cam Kit (**naturebytes.org**). Place a Pi inside this weatherproof housing and it uses heat sensors to capture photos of the critters visiting your garden.



SUBSCRIBE TODAY!

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

Subscription benefits

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

100 PAGES OF RASPBERRY PI

NEW! YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Win! £200 WORTH OF PRIZES
100 pages of hacking & making
Issue 37 September 2015
RASPBERRY PI DIGITAL HOME
Eight Pi projects to help you create tomorrow's home, today
STREAM LIVE FROM YOUR PC
MAKE A HIGH-SCORE TABLE IN SCRATCH
BUILD A FRIDGE MONITOR
Also inside:
THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY
GET STARTED WITH RASPBERRY PI
How to unbox your credit card-sized PC and start hacking today
RETRO CLASSICS
SIMULATE PHYSICS WITH PYTHON
3D PRINTING WITH RASPBERRY PI
CREATE A START SCREEN IN SCRATCH
MINECRAFT SPLASH
AMAZING PI PROJECTS

SAVE UP TO 25%

Pricing

Get the first six issues:

£30 (UK)

£45 (EU)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

£90 (Rest of World)

Direct Debit

UK readers can pay **£12.99** by Direct Debit every three months.

Four ways to subscribe:

- Visit **www.bit.ly/MagPiSubs**
- Call +44 (1)1202 586848
- Use the form to the right
- Search 'The MagPi' on app stores

Available on the App Store

Get it on Google play

SUBSCRIPTION FORM

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

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

Mag#37

YOUR DETAILS Mr □ Mrs □ Miss □ Ms □

First name .... Surname ....

Address ....

....

Postcode .... Email ....

Daytime phone .... Mobile ....

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

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

First name .... Surname ....

Address ....

Postcode .... Email ....

PAYMENT OPTIONS

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

DIRECT Debit

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

Service user number 8 3 8 7 7 3

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

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

Address ....

....

.... Postcode ....

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

Branch sort code □□□□□□ Account number □□□□□□□□

Reference □□□□□□□□□□□□□□□□□□ (Official use only)

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

Signature .... Date □□/□□/□□

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

SUBSCRIPTION PRICING WHEN PAYING BY CHEQUE OR CREDIT/DEBIT CARD

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

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

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

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

Card number □□□□□□□□□□□□□□□□

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

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

Signature .... Date □□/□□/□□

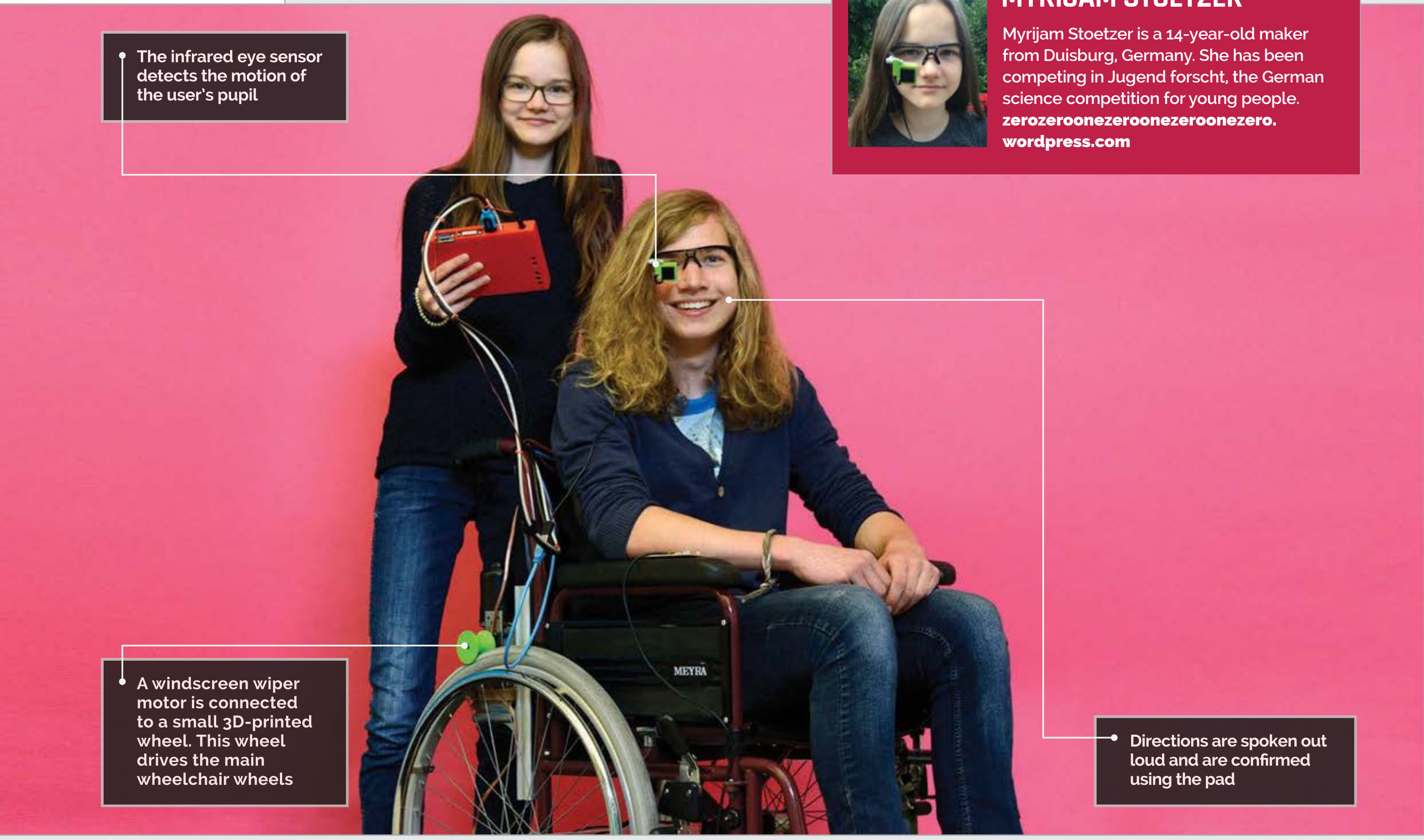
I would like my subscription to begin from issue .... (month + year)

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

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

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

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



The infrared eye sensor detects the motion of the user's pupil

A windscreen wiper motor is connected to a small 3D-printed wheel. This wheel drives the main wheelchair wheels

Directions are spoken out loud and are confirmed using the pad

MYRIJAM STOETZER

Myrijam Stoetzer is a 14-year-old maker from Duisburg, Germany. She has been competing in Jugend forscht, the German science competition for young people.
zerozeroonezeroonezeroonezero.wordpress.com

Quick Facts

- The entire project took about 10-12 months to complete
- The files have been released under the CC BY NC licence
- Originally, an Odroid was used instead of a Raspberry Pi
- A robot was used to test the software
- The system can be used with any non-powered wheelchair

EYE-CONTROLLED WHEELCHAIR

Myrijam Stoetzer and **Paul Foltin** built an incredible eye-controlled wheelchair from 3D-printed parts, motors from a windscreen wiper, and a hacked webcam. **Lucy Hattersley** finds out all about it...

Myrijam Stoetzer and Paul Foltin, from Duisberg, Germany, have built this fantastic eye-controlled wheelchair. We caught up with Myrijam to talk about it.

"We did it because eye-tracking was so interesting," says Myrijam, "and then it occurred to us that we might steer something.

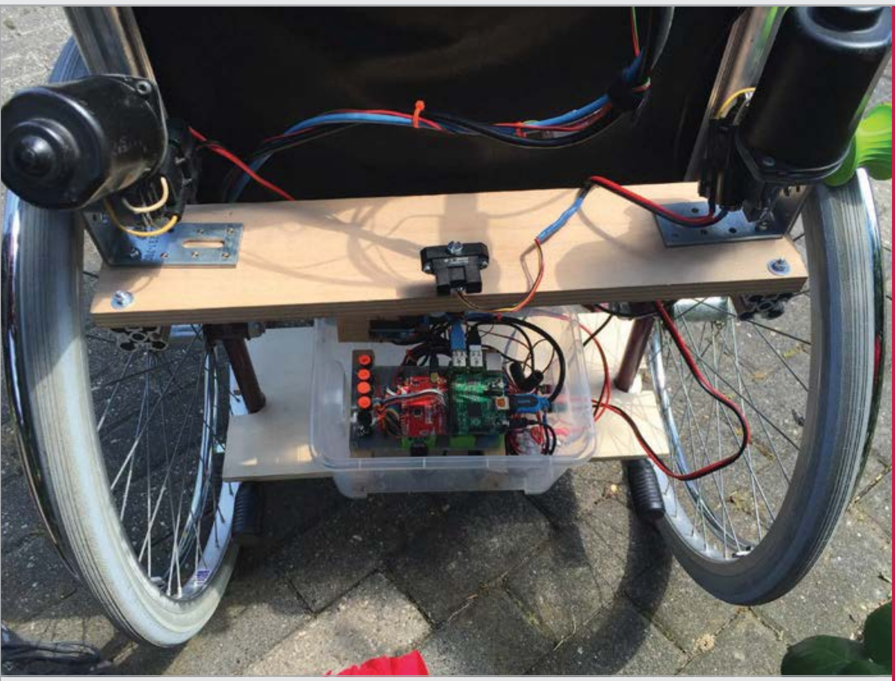
"While looking at different ways to build an eye tracker, we stumbled upon the Eyewriter project (**eyewriter.org**) from graffiti artist Tony Quan."

Tony suffers from ALS (amyotrophic lateral sclerosis) and can only move his eyes. "Tony was isolated for seven years before his friends helped him to paint and draw again using an eye tracker," Myrijam tells us. "We wanted to help people like him be able to move again, because it is so horrible to imagine you couldn't move any more."

Tracking eyes

The eye tracker is a webcam modified to use infrared light and mounted on the frame of safety glasses. "We replaced the infrared blocking filter with a piece of analogue film that blocks nearly everything but infrared light," explains Myrijam, "[then] we soldered in two infrared SMD LEDs in exchange for the white LEDs. This way, the eye will be illuminated by IR light, and the pupil reflects it back."

The camera is mounted inside a custom 3D-printed case and attached to the safety glasses frame.



Above A windscreen wiper motor is connected to a small 3D-printed wheel. This small wheel drives the main wheelchair wheels

"The camera films the eye, and the software extracts the position of the pupil and checks if it is within the range of four areas corresponding to the four movements," says Myrijam.

In the current version, the command is read out to the user via speech synthesis, and it has to be confirmed. Then the motors are switched on, depending on the direction. When the wheelchair is moving, infrared rangefinders check for obstacles in the path.

> Our biggest achievement was getting the tracker to find the coordinates of the pupil

Raspberry Pi-powered

The Raspberry Pi is used to capture images from the webcam, and OpenCV (**opencv.org**) is used for image manipulation. "The software captures images from the webcam with a frame rate of 10-12 fps," says Myrijam. The image is converted to greyscale, before being blurred and converted to black and white. "OpenCV offers a great filter that calculates the centre of gravity of a given shape," she explains. "We use this function to get the coordinates of the pupil that has been filtered out so far: this was the biggest breakthrough."

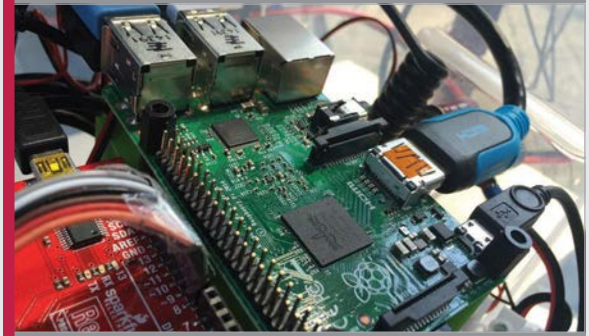
The wheels are powered by windscreen wiper motors rescued from the scrapyard. "We wanted to keep it possible so everybody can simply rebuild it on their own," Myrijam tells us. These motors are connected to the wheelchair wheels using a small 3D-printed wheel contoured to fit around the tyre.

It was a challenging project: "It was very difficult to work on the code because this is our first time of leaving Lego Mindstorms behind," says Myrijam. "Our biggest achievement was getting the tracker to find the coordinates of the pupil. After that, we knew it was possible. I learned to do 3D design, 3D printing, Linux, Python, and image processing. It was amazing to see that you can integrate computer vision into your projects and combine that with physical computing."

EYE TRACKING WHEELCHAIR

>STEP-01 Infrared camera

The eye tracker consists of a webcam modified to use infrared (IR) light. The IR blocking filter is removed, and a film is used to block everything but IR light. IR LEDs replace the standard LEDs. A 3D-printed case is used to mount the device on safety glasses.

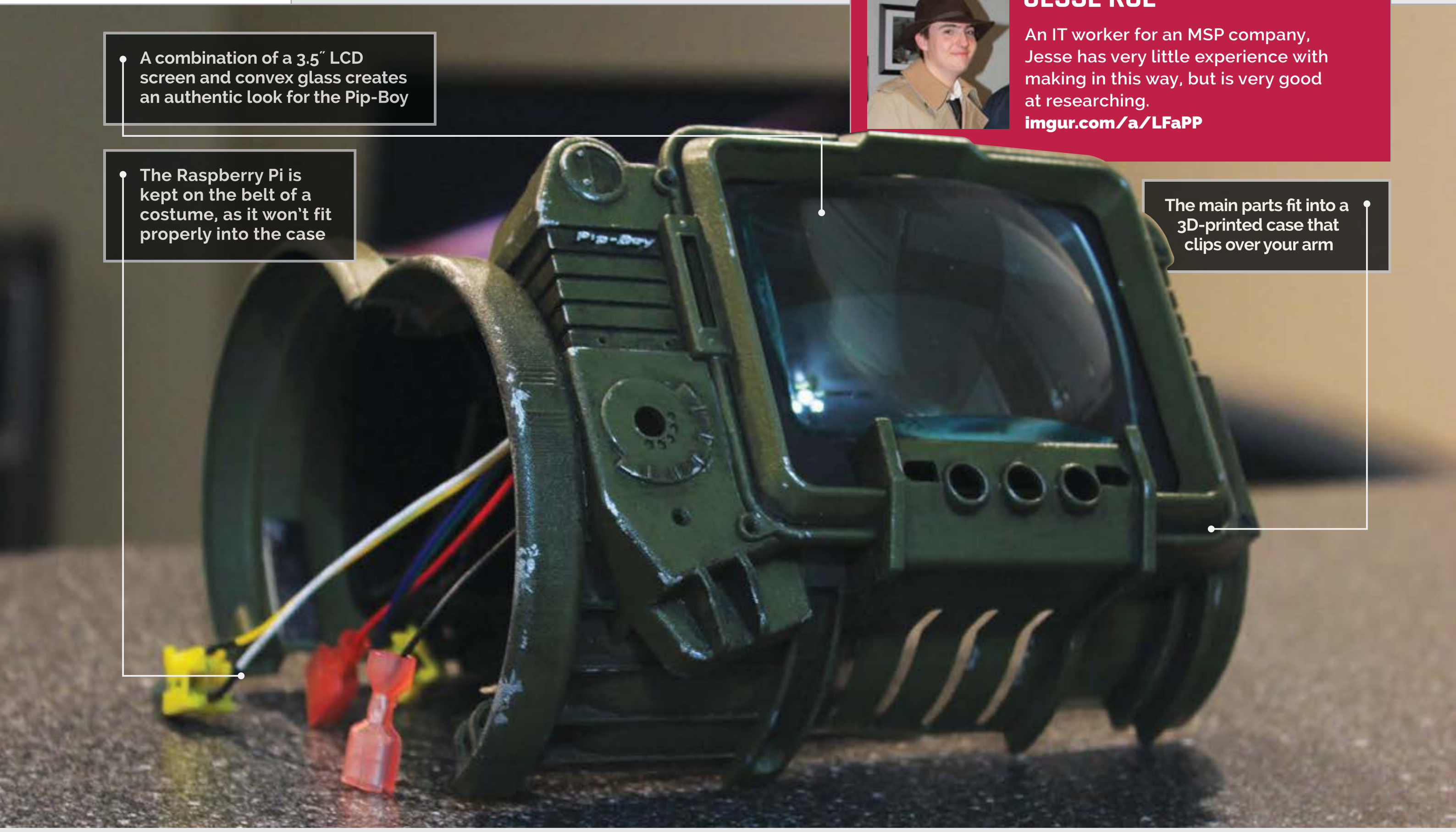


>STEP-02 Pupil detection

The Raspberry Pi 2 snaps images of the user's eye from the IR camera at a rate of 10-12fps. A series of filters are applied to each image, increasing the contrast and making the pupil more easily identifiable. OpenCV software is used to map the coordinates of the pupil.

>STEP-03 Motor control

The wheelchair is controlled by motors salvaged from a windscreen wiper unit. These are connected directly to the wheel on the wheelchair using a custom 3D-printed profile (designed to match the wheels of the chair).



A combination of a 3.5″ LCD screen and convex glass creates an authentic look for the Pip-Boy

The Raspberry Pi is kept on the belt of a costume, as it won't fit properly into the case

The main parts fit into a 3D-printed case that clips over your arm

JESSE ROE

An IT worker for an MSP company, Jesse has very little experience with making in this way, but is very good at researching.
imgur.com/a/LFaPP

Quick Facts

- Jesse completed the project between October and Christmas 2014
- There are still a few buttons to add, but it's functionally complete
- *Fallout 3* and *New Vegas* are his favourite *Fallout* games
- Jesse now wants to make a robot tank army with Pis
- The cast for the case was from Nakamura Shop, and the RasPipBoy UI by Neal Corbett

RASPBERRY PIP-BOY 3000A

Survive the radioactive wastelands or a cloudy day in town with a custom-made, wrist-worn computer

The Pip-Boy from the *Fallout* video games is quite an iconic piece of kit these days. The fictional wrist computer's functionality may be less amazing now than it was when the series debuted in the 1990s, but the retro-futuristic look has kept it in certain corners of pop culture. Now, thanks to the reignition of interest in the *Fallout* franchise, the rise of makers, and the exploding popularity of cosplaying, every so often a new, home-made version of the Pip-Boy will turn up online and go a bit viral. Even with the impending release of a special-edition version of *Fallout 4* with an official Pip-Boy, people are still making their own versions, like Jesse Roe.

"The project was an attempt to make a fully functional Pip-Boy 3000A. Not something to stick a phone in, but an actual working device," Jesse tells us, referencing the official Pip-Boy which will require a smartphone.

Having never found a perfect use for his Raspberry Pi, Jesse decided to use it for this project he was making for a friend. Using a 3D-printed case that he modified himself, the build wasn't simple.

"I worked on this probably about 70 hours total, with a lot of that being just research," Jesse explains about his build process. "There was a lot of stuff out there on making a Pip-Boy, where to get the cast from, materials, etc. The main piece to get was the Pip-Boy cast itself, which I ordered from Nakamura Shop on Shapeways.

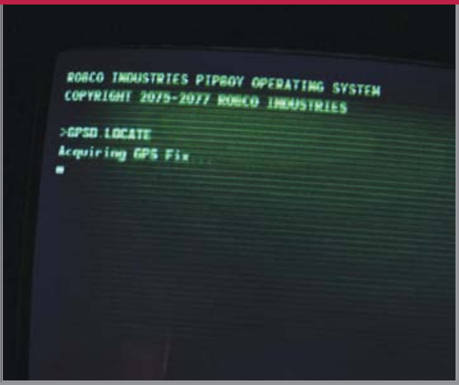
"Once it was in, we sanded it down and used model paint for the base with a darker green.

USE A CUSTOM PIP-BOY

>STEP-01

Set it up

You will need to get into a full Vault-dwelling outfit with a belt to house the Raspberry Pi, so that you can plug the Pip-Boy screen into it, and the Pi into some power.

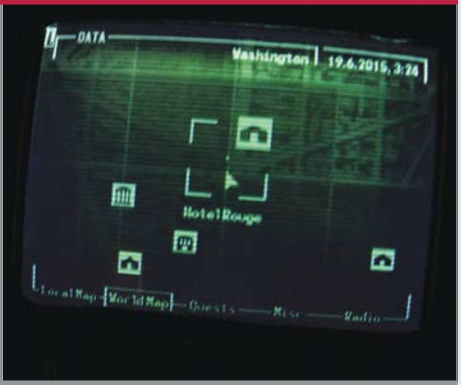


```
ROBCO INDUSTRIES PIPBOY OPERATING SYSTEM
COPYRIGHT 2075-2077 ROBCO INDUSTRIES

>GPSD.LOCATE
Acquiring GPS Fix...
```

>STEP-02

Boot it up

You can enjoy the custom bootup messages, perfect for when you're cosplaying and making your way to a convention floor to show off your fully functioning Pip-Boy.



>STEP-03

Show it off

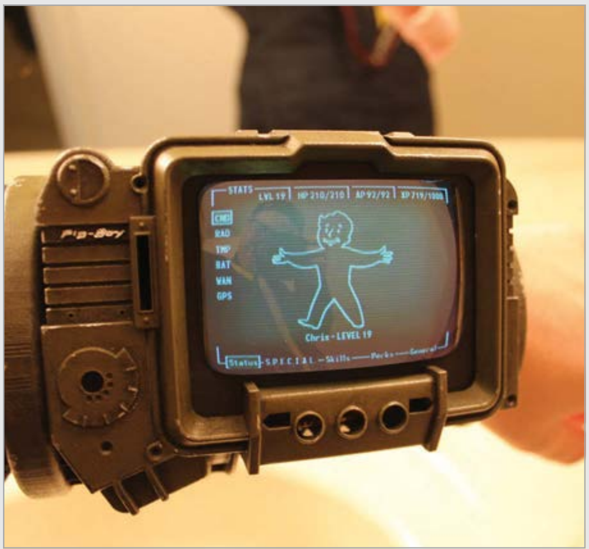
The custom interface, available from **bitbucket.org/selectnone/raspipboy**, has many functions you can show to people who are interested: GPS, VATS simulator, and a Wi-Fi strength monitor.

We detailed it up with scratches and other abrasions to look like a used model. I knew the suit as a whole needed to be separated. Too many people are so close to having a working Pip-Boy and get stuck when they say they can't fit everything in there! I took a mechanics jumpsuit and put the 101 logo on the back as well as other details... I used an old army surplus belt with lots of compartments for a part of the suit as well, and put a hole in the suit so all the wiring could go from the belt, up the back, and to the Pip-Boy.

> “I worked on this probably about 70 hours total, with a lot of that being just research”

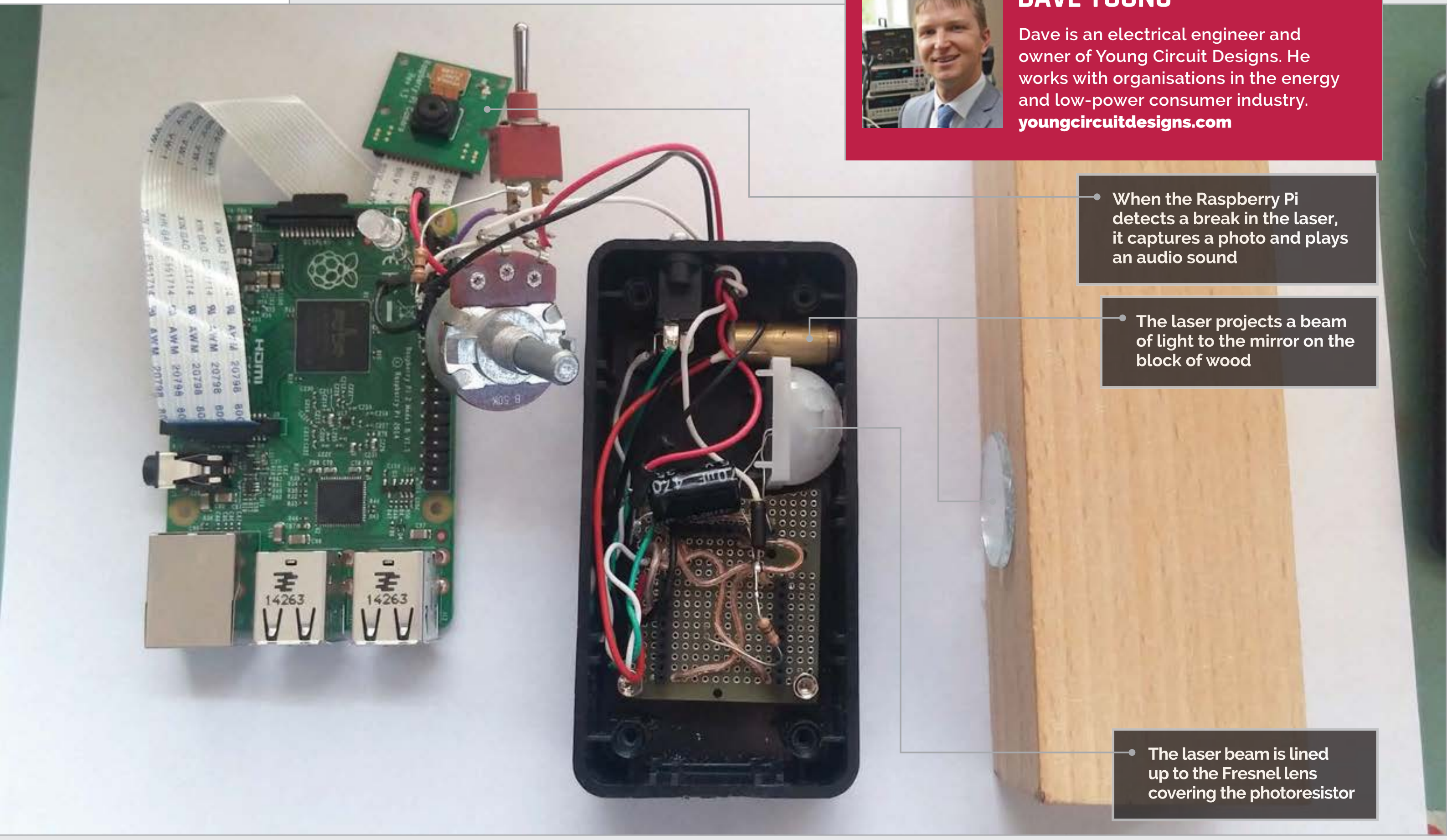
"Next was getting the Pip-Boy to boot into the slightly tweaked 'OS' on startup, which was fairly simple. I ended up editing the code to have a different startup message for the 'BIOS' and whatnot, to make it more complete, but I didn't mess with the software too much. The hardest and almost final part of the build was the screen. I ended up using a 3.5″ TFT from Adafruit. The way the top bezel comes down and how you have to mount the screen means laying it down on something that covers all, for lips behind the bezel. Then after that, I had to lay a piece of glass on top of the screen and glue it very lightly (but not too lightly) around the screen. Any mistake and you have to get a new screen because either you're going to glue it down off-centre or you'll put too much on the screen."

Since Jesse hadn't done much programming before, the modifications and setting up were a little tricky for him, but as you can see, in the end he managed to complete the project and walk around with a sweet Raspberry Pi-powered wrist computer.



Above Dials are to come, but the interface perfectly captures *Fallout 3*'s Pip-Boy

Left A handy army belt allowed Jesse to hide the electronics in the suit



When the Raspberry Pi detects a break in the laser, it captures a photo and plays an audio sound

The laser projects a beam of light to the mirror on the block of wood

The laser beam is lined up to the Fresnel lens covering the photoresistor

DAVE YOUNG

Dave is an electrical engineer and owner of Young Circuit Designs. He works with organisations in the energy and low-power consumer industry.
youngcircuitdesigns.com

LASER DOG WATCHER

Penny is a well-trained dog, thanks to her master's Raspberry Pi-powered Laser Dog Watcher. **Lucy Hattersley** investigates…

Quick Facts

- Raspberry Pi aside, the parts only cost around £15
- The hardest part is getting the mirror to line up
- It can play any audio file of your choosing
- A Fresnel lens is used to make it easier to line up the laser
- It's based on an Arduino XBee shield project

When Dave Young, the owner of Young Circuit Designs, was training his dog Penny, he hit upon an idea. Why not turn a Raspberry Pi into a dog monitor that could issue voice commands when he's not around?

"My dog Penny is a delightful girl," says Dave. "We got her from the Denver Dumb Friends League [a local dog rescue] a few years ago. My wife and I were just going in to take a look. When I saw Penny, I knew she was my dog. We walked out with her that afternoon."

Penny is a very clever dog. "She's smart as a whip," explains Dave, "and is quite good at playing people like a fiddle. She uses her very slightly lazy eye to make the saddest face in the world. Especially when food is involved. She's food crazy."

It took a long time to train Penny not to jump up on the counter and eat Dave's food. "I didn't want to use a shock collar," he says, "and I toyed with using some very high-pitched noise as a deterrent, but settled on an array of recordings of me reprimanding her."

The voice commands worked, but only when Dave was around to issue them. That's when he hit upon the idea of using his Raspberry Pi to monitor Penny and issue spoken commands when he's not around.

The Laser Dog Watcher has a laser beam similar to security systems (650nm, 6mm, 3V, 5mW Mini Dot Diode). "It sends a laser out to a mirror and measures if it makes it back to a sensor. If the sensor can't see the laser, there is a circuit that tells the Raspberry

Pi that the laser beam has been broken. The Pi then takes a photo of the area near the sensor and plays an audio file of my voice saying one of a few things that I use to reprimand her.

"I recessed a photoresistor [Excelitas Tech VT935G] into the box to block out ambient light and put the system on one side of the counter, and a mirror on the other side. I aim the laser to go from the system, bounce off the mirror, and then come back to the sensor. When the laser is blocked, the photoresistor sees less light [and] changes its output voltage, which is fed into a digital input pin of the Pi."

> "The Pi then takes a photo of the area near the sensor and plays an audio file of my voice"

Building a Laser Dog Watcher wasn't as easy as it sounds. "I'd put it at medium to medium-hard difficulty," says Dave. "The electrical circuit is easy, but creating a mechanical system that lines up nicely takes a bit of thought." Helpfully, for anyone wishing to replicate the project, Dave has placed all the design schematics on the Element14 community website (**bit.ly/1J3p1ls**).

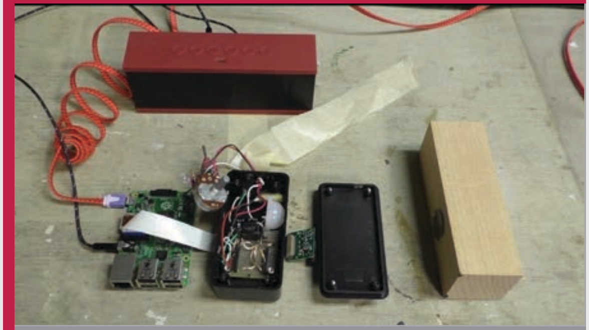
"Think hard about the mechanical system," advises Dave. "I ended up using a Fresnel lens to make it easier to aim the laser. The Fresnel lens allows for the laser to be directed at a larger target area. A photoresistor is a pretty small sensor, and accurately hitting it from a 10-foot total laser length is a real challenge. If the laser hits anywhere on the lens, it will send the light to the photoresistor sensor.

"I also added some error correcting," says Dave, "so that it doesn't continue until the laser beam connection is remade. That way, if she knocks the aim of the system off, it won't continually play my voice over and over."

The system is designed to cycle through three audio files of Dave saying 'no' in a variety of ways, along with capturing a photograph of Penny. The laser tripwire could be put to a variety of different uses, though; it would benefit anything that responds to motion.

Below Penny is mostly a good dog, but she can't stop herself from jumping on counters and eating food

MAKING THE DOG WATCHER



>STEP-01

Laser tripwire

A laser is recessed into the device and placed at one end of the table. The laser is aimed at a mirror at the other end of the table, and the laser beam is reflected into a photoresistor (back on the main device).

>STEP-02

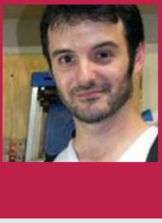
Take photo

When Penny jumps up to the table, she blocks the laser. The photoresistor detects the change in light and sends a signal to the Raspberry Pi. It then captures a photograph.

>STEP-03

Make a noise

The Raspberry Pi then plays one of three verbal commands (personally recorded by Dave). These are all variations of 'no' which encourage Penny to move down from the table.



PETER ROCA

VP of technology, and lead software developer, at a small company in Washington, DC. He has a degree in Neurobiology and has been coding for 15 years.
peterroca.com

BEDBOT

An over-engineered alarm clock built into a bedside table that wakes you up with music. It's also powered by a Raspberry Pi

Quick Facts

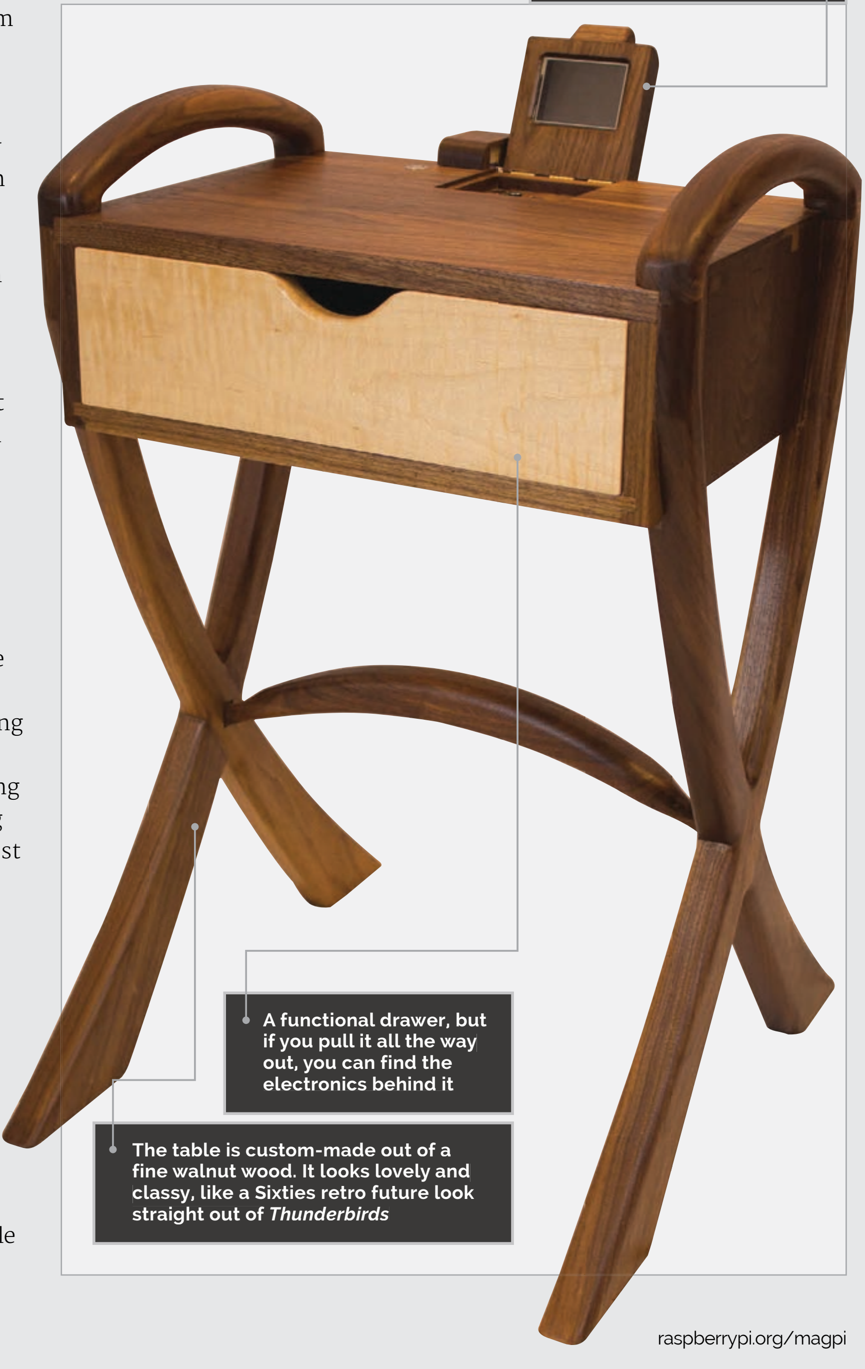
- The whole thing took Peter 400 hours over six months
- It was Peter's first Raspberry Pi project
- All the source code is available for free on GitHub
- Peter had never used Python before this project
- When we asked him about it, he said "At least it looks nice."

When you think of an alarm clock, you probably get an image of a silver, analogue clock with bells on top. Thanks, television. No one really uses that any more, starting with the invention of clock radios and the Teasmade, and right up to smartphones acting as our alarm clock and morning newspaper. What's the next step in the evolution of waking up in the morning? Well, it probably won't be the Bedbot, but it's really cool either way.

The Bedbot is, according to its creator Peter, "My Daytime Initialization Assistant – a digital alarm clock/radio with a fancy wooden enclosure. It is a crude, over-engineered, expensive piece of hardware."

The idea came to him after being a little fed up with the kind of radio programming he was getting when he woke up in the morning (apparently a little later than most people). "I realised that I could use my new RPi to create a radio device that could bypass trivial things such as NPR's morning program schedule."

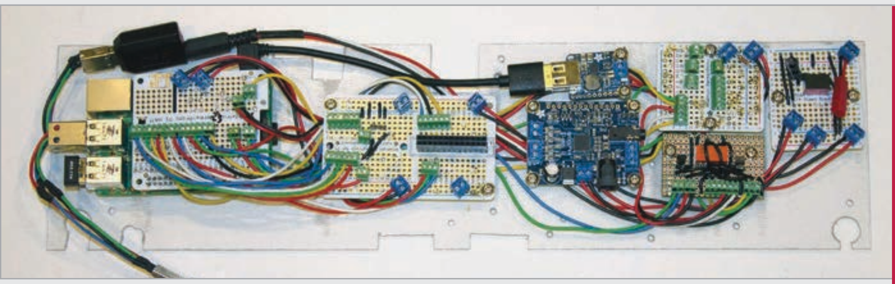
The Raspberry Pi is built into a walnut and maple side table, with a little screen that flips up when the alarm starts the radio. From the images, you can see that the relatively small table has a lot of electronics packed into it. The table was also made by Peter.



Normally sitting flush, this screen flips up during the alarm, revealing the buttons underneath to control it

A functional drawer, but if you pull it all the way out, you can find the electronics behind it

The table is custom-made out of a fine walnut wood. It looks lovely and classy, like a Sixties retro future look straight out of *Thunderbirds*



Above The circuitry is quite complicated for what you could feasibly consider to be a simple function, but it makes it high-quality

“I started woodworking about four years ago in my free time,” he tells us. “I really enjoy creating interactive systems which is why this project was so fun for me. Working in my shop is cathartic and it offsets the lethargy that working in an office inevitably encourages. I think a lot of software developers eventually pick up some kind of hobby that requires using your hands for something other than typing.”

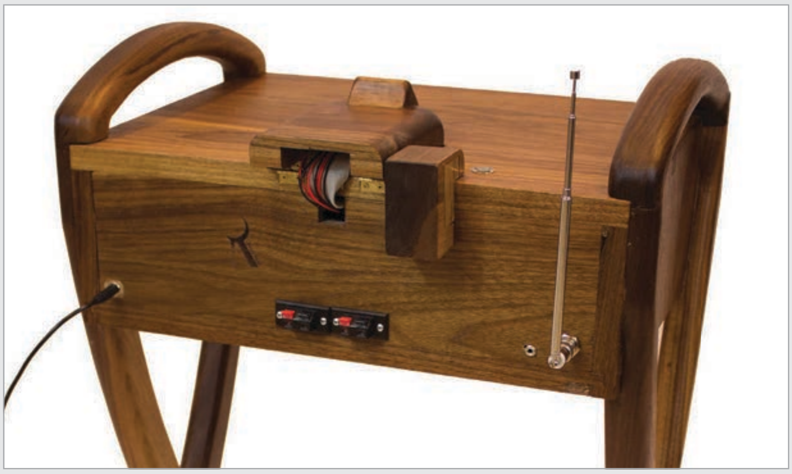
> Each major feature of Bedbot is considered a ‘module’ and has no dependencies on other modules

The electronics comprise the Pi, WiFi and Adafruit PiTFT screen, along with a small OLED, USB audio cards and custom speaker system, servos for the screen to flip up, several buttons, and lots of other custom-built bits to the get the look and effect he wanted. Next came the programming, which wasn’t too complex.

“The only complicated part is how I designed the [Python] module system. Each major feature of Bedbot is considered a ‘module’ and has no dependencies on other modules, so that modules can be added or removed without breaking anything else… if it’s configured correctly, Bedbot will find the module and load it automatically into the menu system. There are hooks to allow for modules to communicate with each other through a broadcast system. I designed it this way to allow for anyone to write their own module without having to dig into the code of the entire system.”

The proof is in the pudding, as they say, so how well does it work?

“Well, I haven’t been late to work yet!”



Above The rear of the table gives away that this is more than just a bit of furniture. Sit it up against a wall to hide these parts away

PROGRAM YOUR TABLE ALARM

>STEP-01

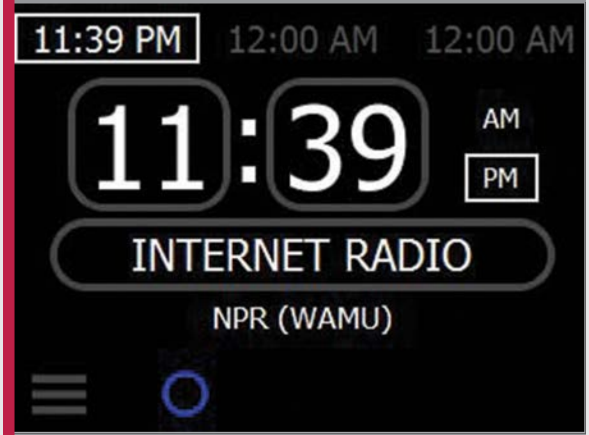
Set your alarm

As well as time, you need to tell BedBot what kind of wake up you want. Normal radio? Internet radio? Play from an MP3 player? The choice is yours.

>STEP-02

Go to sleep

As humble writers about the Raspberry Pi, we can’t really tell you how to accomplish that. Big Bird taught us how to get to sleep, though; try him.



11:39 PM 12:00 AM 12:00 AM

11:39 AM PM

INTERNET RADIO

NPR (WAMU)

>STEP-03

Wake up!

The screen lifts up when the alarm time comes around, and your selected audio starts playing. There is a snooze button for five minutes’ more kip, though.

STARRING THE RASPBERRY PI

Slowly but surely, the Raspberry Pi has been turning up in movies and TV shows. **Simon Brew** tracks its screen appearances to date...

The lure of gazillions of dollars in product placement revenue, or perhaps just swagging some free kit from the Apple Store, may lead you to think that nobody in the movies would be seen dead with anything less than a MacBook and some overpriced accessories. Heck, in the film *Percy Jackson & The Olympians: The Lightning Thief*, no longer could the mythical Medusa now be defeated be spotting her reflection in a sword and shield. An iPod, that's what they needed.

But could times be changing? Even movie and TV productions face austere times, and thus the days of every ten-year-old owning an array of top-end products from PC World may be coming to an end. Instead? The flexibility, quality, and affordability of the Raspberry Pi have led to it sneaking more and more onto screens, big and small. You don't need us to tell you just what the Raspberry Pi can and can't do. But will someone need to tell Hollywood? Let's take a look. And remember that all these Pis were billable to their respective productions...

COMPUTERS ON SCREEN

It's little secret that the depiction of technology in film and TV leaves something to be desired. Here are just a few of the common faux pas that slip through the net, which tend to irritate the most...

- When a character appears on the screen, a beeping noise of sorts has to be made.
- Switching the screen off tends to switch the computer off.
- Also, unplugging one computer stops a network hack.
- Photo-enhancing software can enhance any photo on planet Earth.
- Complicated passwords can be cracked in a matter of minutes. In the case of the, er, all-time classic *Swordfish*, this can be achieved when the hacker in question is 'multitasking'.
- They don't crash. Ever.
- Remote signals can be sent to individual components, in turn causing them to blow up.
- Every teenager in America seems to own an Apple Mac. And can afford all the accessories. And likes to angle it so the logo is pointed towards, coincidentally, wherever the camera happens to be pointing. Lucky, that.

POINT BREAK

In brief:

Remake of much-loved action movie. Not many people seem to be looking forward to it, judging by 86% of words written about it on the internet so far.

What does the Pi do?

Blows stuff up. Er, that seems to be it. But as the film isn't out yet, it may yet turn out to be a major supporting character. You never know.

The details:

Nothing cheers people up like a beloved film getting the Hollywood remake treatment.

A lie, clearly, as a quick look around internet message boards will testify. They're remaking *Point Break*? And none of the original cast will be anywhere near it? As Edmund Blackadder once mused, "I think the phrase rhymes with clucking bell".

The new *Point Break* movie doesn't arrive until Christmas 2015, so let's err on the side of optimistic and suggest that it's going to be an Oscar-winning classic. But already, in the trailer for the film, we can see that the Raspberry Pi is taking on a significant supporting role.

It's good to see that *Point Break* is tackling the current lack of quality movie villains by, well, sort of taking their place. So, from what we can make out, the idea is that you attach your Pi to some explosives, and then all you need is one of those wonderful movie products where you have to press one of two buttons to blow seven shades out of everything in the immediate vicinity. We've never hung around long enough to find out what the other button does.

We saw this using boring old technology in that other awards-attracting masterpiece, *Die Hard 4.0*, where the wheeze that time was basically to have an exploding hard drive. Those of us of a certain vintage may remember times when the humble hard drive did have less of a reliable quality to it, but attacking computers in a manner that leaves Bruce Willis diving around your place of residence? It's a bit of a stretch.

Given that just a few minutes of *Point Break* have thus far been released, we'll have to assume that the trailer has happily glossed over the bit where the antagonists excitedly take delivery of their Pi and customise it a bit. But we're sure that's going to be in the final cut of the film. It'd be a startling omission otherwise.

REVOLUTION

In brief:

A TV series set in post-apocalyptic times, where the world is on the receiving end of a permanent electrical power blackout.

What does the Pi do?

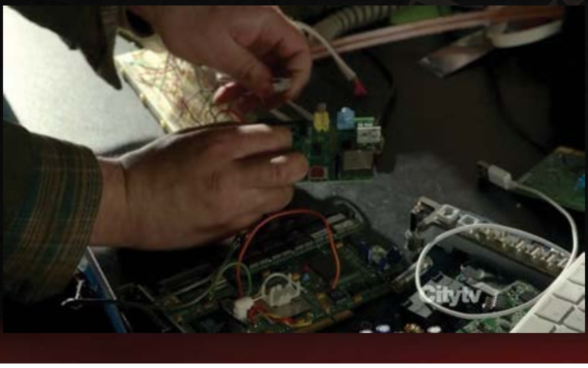
It still works! Even though pretty much nothing else does. That, friends, is value for money.

The details:

The idea behind the TV series *Revolution*, which ran for two seasons between 2012 and 2014, is that electricity's days are numbered. Thanks to one of those pesky apocalypses that science fiction likes to foist on us from time to time, all electrical devices on planet Earth are done for. A permanent electrical blackout shrouds the world.

The result of all this is an intriguing television show about how humans might cope without *Candy Crush Saga*, mean tweets, and cat GIFs.

But what's this? A working Raspberry Pi snuck in there? How could that happen? That's the beauty of such a low-power device, clearly. Even when there's no electricity, the damn thing still works!



Actually, that's not quite true. As *Revolution* progresses with its intriguing premise, it starts to cheat a bit. Small pockets of power are generated, and so, with that in mind, electronics are sought and utilised. Thus, in the episode The Longest Day, which is the 17th instalment of the first season, the character of Aaron is trying to get an old Mac up and running, but actually – as you can see from the screenshot we lovingly grabbed with our electricity-free supercomputer – something rather familiar, and more affordable, has crept into shot.

Good news, though: the device is being brought into being with the intention of saving someone's life. So already in this feature, we've gone from a Raspberry Pi being a tool of villainy to one employed by heroes. That's far more like it.

The idea is that apparently early nanotechnology is being assembled, so that it can take hold of energy from someone's capsule and knit together the wounds of a dying woman. Naturally enough, if it doesn't go to plan, then we're back where we were with *Point Break*. We don't want to give too many spoilers away here, but the Raspberry Pi at the very least survives. So that's something.

It's hard to judge just how realistic an application for the Pi this actually is, as *Revolution* is set in the year 2027, and a lot can happen in the next 12 years. Rest assured, when a $35 piece of hardware starts performing life-saving surgery, it'll make more than the cover of this magazine. Especially if it saves a Kardashian bot or something.

PERSON OF INTEREST

In brief:

A still-running American TV drama series, about a man who seems to have seen *Minority Report* and therefore invents a system that can help prevent crimes before they happen. Thus far, Tom Cruise has not appeared.

What does the Pi do?

Brings internet to that scary bloke from *Lost*. Having seen *Lost*, we would not refuse his request.

The details:

Persons Of Interest is a bit of a bumpy series, and certainly the first season of the show did it few favours. But like many American dramas, it has matured, become more interesting, and tried to placate people like ourselves by inserting agreeable technology in from time to time. We're easily bought like that.

The show has enjoyed four seasons so far, and the fifth is in production. Reports suggest that this might be the last one, so the production team are running out of chances to squeeze a further Raspberry Pi in there somewhere, as is their constitutional duty. But at least they've already managed one.

Okay, so it's not a starring role, but a fresh Hollywood agent should soon sort that out. And at least the Pi gets to sit in the centre of a shot. For now then, Harold Finch – played by Michael Emerson – has shown his technical taste by using a Pi in the season three premiere of the show. Here he is, with a Pi in an Adafruit case attached to the back of his laptop. Because sometimes, you can't trust a wireless hotspot, right? Better to invest a few quid and bring some cables out with you.

THE ANT-MAN TINY TECH CHALLENGE

While Marvel's latest superhero opus, *Ant-Man*, doesn't actually feature a Raspberry Pi itself, the marketing boffins behind it were keen to harness the idea of tiny technology for a tie-in promotion. Given that the film has banked over $300m at the time of writing, it seemed to work, too. That, and all the posters over the side of buses and stuff.

In conjunction with Dolby, Marvel Studios, Disney, and Visa Signature, the Raspberry Pi-backed Ant-Man Micro-Tech Challenge was launched. It was an American competition, aimed at girls aged 14-18. The objective? To make a DIY project using at least one readily available micro-technology component.

The winners were announced at the end of June. Allison White came up with a water-limiter for shower heads, to alert people that they were using too much water. Anna Nixon turned a teddy bear into an interactive character, complete with face detection, a GUI, and voice recognition. The aim is to add health detection features in the future.

Ashita Patel made a robotic arm that draws images created on a computer, while Elizabeth Almasy made a bubble maker in the shape of an octopus!

The winners all went to see the *Ant-Man* movie at its premiere, and got to spend time with the people who make theme park rides at Disneyland. It made us wish that we'd submitted our Candy Crush Saga-Destroying Nano Bots (TM) in time.

Full details of the challenge can be found here: **www.ant-manchallenge.com/winners.php**

Oddly, it's a practical use for the Raspberry Pi, which goes against the way computers tend to be dramatised on screen. So even though it's implicit use, and the Pi isn't arresting a terrorist or the like, we can but salute it.

Relatively realistic tech usage in Hollywood productions? At this rate, they'll be making funny Adam Sandler films again soon. Or is that where we get too far-fetched?

CSI: CYBER

In brief:

The *CSI* show that's not set in Las Vegas, Miami, New York, London, Los Angeles, or Bridlington. The one that is also laughed at a lot by people who have used computers for more than ten minutes.

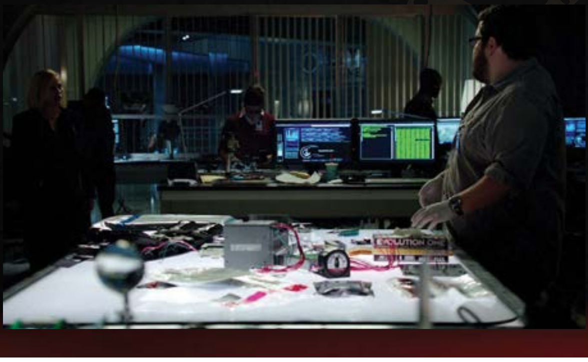
What does the Pi do?

Potentially brings down the entire world. Messes around with lift doors. Gets a price increase.

The details:

The never-ending *CSI* television franchise has of late decided to turn its documentary-esque levels of accuracy and focus (go with us on this) towards cybercrime. Naturally, therefore, they all use movie-style computers that make a little noise whenever a single digit of text appears on the screen, but let's gloss over that for the minute. In fact, if you're looking for realism, it's probably best to gloss over most of the show. But that's concessions to TV drama for you.

For there are atrocities to battle! And, wouldn't you know it, the Raspberry Pi is sat near the heart of them. In this case, it's a roller-coaster crash with assorted fatalities, and the team soon realise that a computer had been hacked to switch off the necessary safeguards. *Final Destination 3* ensues. Then, we learn that it's being used to help control subway trains, too. By the end, we wouldn't be surprised if they'd said it was being used to put the wrong flavouring in crisps.



Don't worry, though, as there's a non-stereotypical bearded, slightly overweight nerd on hand to get to the bottom of things. He helpfully points out to Patricia Arquette's boss lady that these devices "can be found online for about $50", and then demonstrates how one can be used to comedically open and close lift doors while people are still in it. "Anyone can hack in," he warns, sagely, standing next to his non-stereotypical messy desk.

So, can all this be done? Can't say we've tried, to be honest, and, just for the sake of clarity, we have no intention of doing so. Can a Raspberry Pi be used to control the operations of things like doors, and, er, roller coasters? Well, you know the answer to that.

We look forward to future episodes of *CSI: Cyber*, though, given that it's got a second season on the way. Expect the Raspberry Pi to be at the heart of further villainy, and inaccurate price projects. After all, $50? They should shop around a bit...

CHARLIE BROOKER'S ELECTION WIPE

In brief

Ahead of the UK general election, Charlie Brooker and a chum or two dissected the campaign, and fitted in a Raspberry Pi joke.

What does the Pi do?

It allows Charlie Brooker and Philomena Cunk to talk to each other, if, ahem, you take it at face value. But mainly, it lets them poke fun at election coverage.

The details:

Via his series of *Weekly Wipe* programmes on the BBC, Charlie Brooker has regularly taken a spear to the conventions of news reporting, and exposed its limitations. He does it with cutting humour, too. We like him.

> "It's as if he's making a joke about people not being able to wrap their head around technology or something"

But then Brooker cut his teeth on computer magazines, and for some time was a writer on the long-lost *PC Zone*, a British computer magazine we still miss. For his *Election Wipe* programme, broadcast ahead of, as you might expect, the 2015 UK General Election, he turned his gaze to the most "unpredictable" election that the British Isles had faced since, er, the last one. Naturally enough, his satirical dart-throwing was accurate.

So where does the Raspberry Pi fit in? As a punchline of a gag, as it happens.

It pops in at the end of a sequence where 'Philomena Cunk' is trying to make head or tail of the assorted of graphics and data that surround an election campaign. All the gubbins, basically, that you get from experts ahead of an election, who prove to be entirely incorrect, and then turn up a few years later to do it all again. As Cunk switches back to Brooker, we see that he is holding a Pi, and using it to talk into. It's as if he's making a joke about people not being able to wrap their head around technology or something.

This is the bit where it's important to recall that Brooker more than knows his way around technology. The joke doesn't work so well otherwise. So chortle along and just remember that he's one of us. He's just paid a lot more.

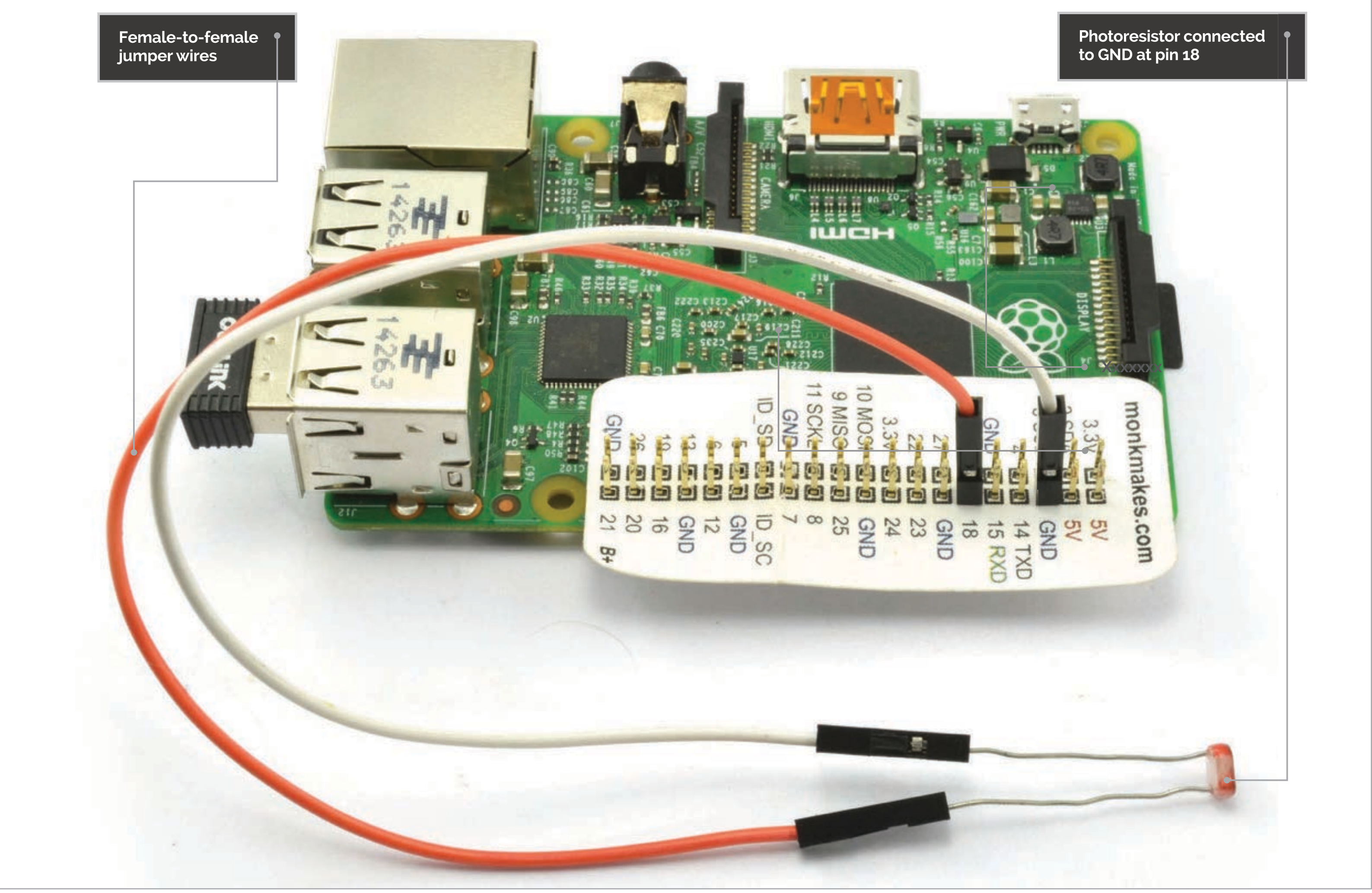
EVERYDAY ENGINEERING PART 7

SIMON MONK

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

simonmonk.org

monkmakes.com



Female-to-female jumper wires

Photoresistor connected to GND at pin 18

FRIDGE MONITOR

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

You'll Need

- Photoresistor (almost any one will work for this project)
- 2× female-to-female jumper wires
- USB Wi-Fi adaptor (unless your fridge has an Ethernet socket nearby)

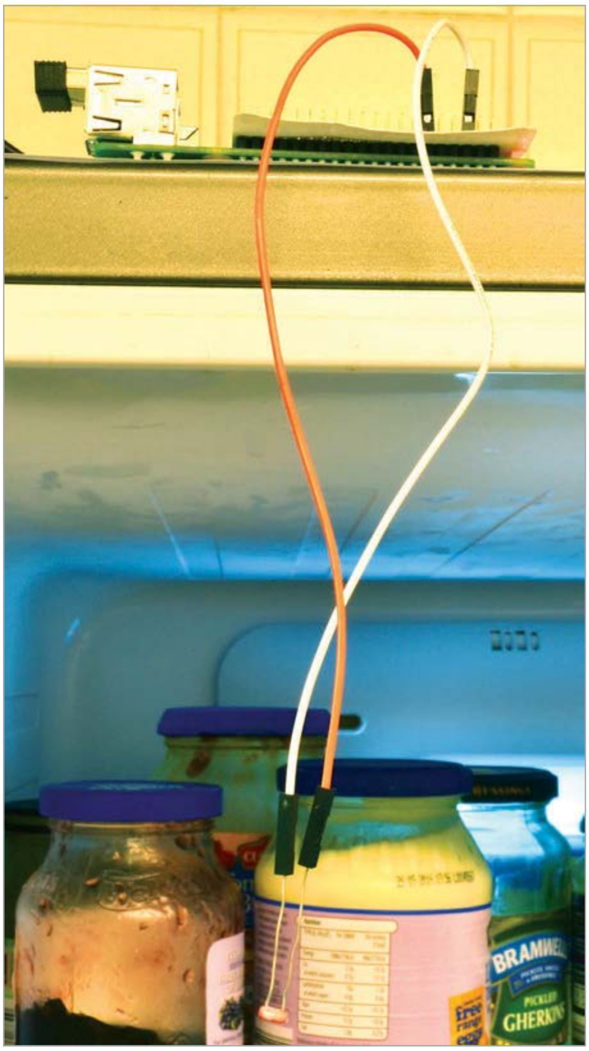
The house is infested with cheese fairies who raid the fridge at all times of day and consume considerable quantities of tasty cheese. In this project, any fridge-door-opening activity will result in a notification email – or, if you prefer, a tweet or Facebook update – telling you the time that the fridge door was opened.

This may seem a somewhat trivial example, but it really serves to show just how easy it is to hook up a sensor that will cause your Raspberry Pi to notify you of events in a variety of ways using the If This Then That framework.

As you'll see from the list of required components on the left, this project uses a photoresistor connected to the Raspberry Pi GPIO header using a pair of female-to-female jumper wires.

Detecting Darkness

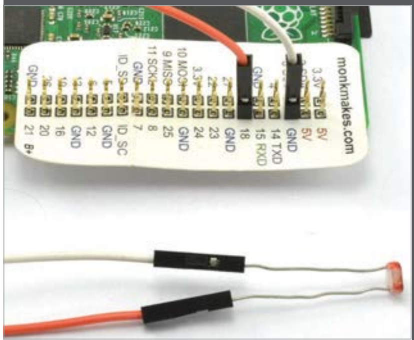
Unlike an Arduino, for example, a Raspberry Pi does not have analogue inputs that can measure a voltage. It does, however, have digital inputs; if the voltage at a digital input exceeds about 1.65V (half of 3.3V), then the input is read by the Raspberry Pi as HIGH, otherwise it is counted as LOW. So, although



Above Fridge monitor

BUILDING THE PROJECT

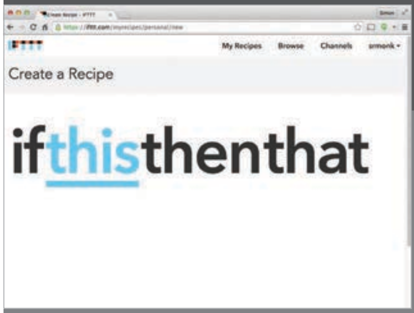
This is a really easy project to make. There is no soldering to be done and just two wires to connect up. In fact, most of the work is in setting up IFTTT.



>STEP-01
Connect the photoresistor

Connect the two leads of the photoresistor to the female-to-female jumper wires. It does not matter which way around the photoresistor is connected. The other ends of the jumper wires are connected to GND and GPIO 18 of the Raspberry Pi.

If the photoresistor leads are a bit loose in the jumper wire sockets, bend a little zigzag into them so that they stay firmly in the sockets. If you are worried about the legs of the photoresistor touching each other, you can shorten them with snips.



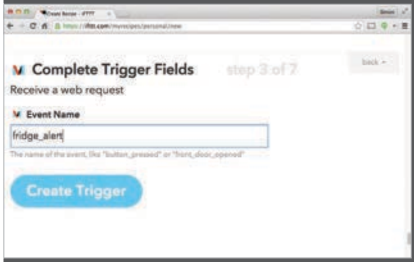
IFTTT My Recipes Browse Channels smonk

Create a Recipe

ifthisthenthat

>STEP-02
Start a new recipe in IFTTT

If you don't have one already, create an account on IFTTT (**ifttt.com**). Then click on the 'Create a Recipe' button.



Complete Trigger Fields step 3 of 7

Receive a web request

Event Name

fridge_alert

The name of the event, like "button_pressed" or "front_door_opened"

Create Trigger

>STEP-03
Create a Maker Channel trigger

Click on the IF part of the new recipe and search for the Maker Channel in the list of channel icons that are shown. Select the only trigger option available (Receive a web request) and enter the text 'fridge_alert' into the Event Name field.

you can't use a photoresistor with a Raspberry Pi to give a measurement of the light level (not without a few extra components anyway), you can use a photoresistor to tell if it's dark or light.

The digital input pin (GPIO pin 18) has its internal pull-up resistor enabled. So, while the pull-up resistor is trying to pull pin 18 HIGH, the photoresistor will try to pull it low. They are in a of tug-of-war and if the pull-up resistor is winning, the input will be HIGH; if the photoresistor is winning, it will be LOW.

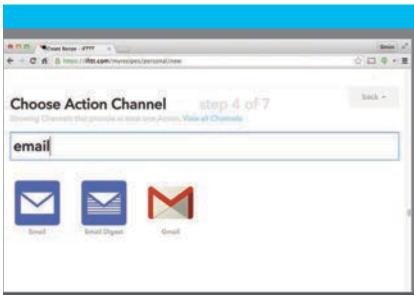
The more light falling on a photoresistor, the lower the resistance (the stronger it pulls pin 18 towards 0V). Typically, a photoresistor will have a resistance of a few hundred to a few thousand ohms when it's in the light. If you make a photoresistor really dark (say inside a fridge), then its resistance rises to several mega-ohms (millions of ohms). At this point, the built-in pull-up resistor (50 to 100 kilo-ohms) will be easily winning and the input will be HIGH.

If This Then That (IFTTT)

IFTTT is a web service that allows you to set up triggers that then cause an action. For example, you could create an IFTTT 'Recipe' that sends you an email (action) whenever someone mentions you on Twitter (trigger). As well as actions and triggers from all sorts of social media and email services, IFTTT can be set up to work with physical events, such as your fridge door opening.

The way this works is that the Raspberry Pi sends a web request to IFTTT as a trigger and then IFTTT performs whatever action you have specified for it. In this case, that is sending an email.

The various channels available to IFTTT often require their own logins, so IFTTT will, from time to time, ask you to enter a user name and password for that channel.



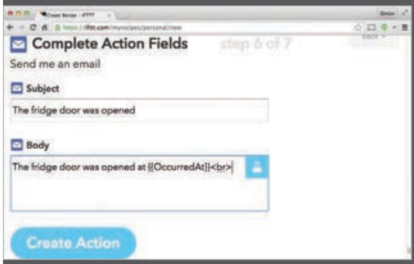
Choose Action Channel step 4 of 7

email

>STEP-04

Choose an action

Now we come to the THAT part of IFTTT. So, click on Action, choose the Email action channel and then select 'Send me an email'.



Complete Action Fields step 6 of 7

Send me an email

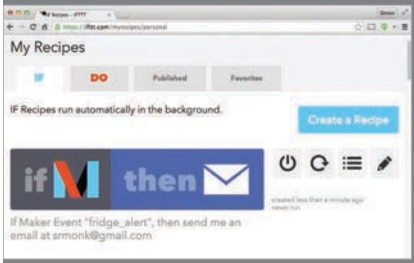
Subject: The fridge door was opened

Body: The fridge door was opened at {{OccurredAt}}

>STEP-05

Complete the action fields

You now need to fill in the fields. The 'subject' field will become the email subject, while the body field will contain the text of the email. Notice the use of the special names contained between {{ and }}. These are called 'ingredients' and allow you to customise the text. In this case, the 'OccuredAt' is a timestamp for when the trigger event happened.



My Recipes

IF Recipes run automatically in the background.

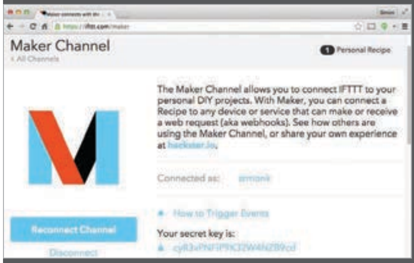
If Maker Event "fridge_alert", then send me an email at srmonk@gmail.com

created less than a minute ago
never run

>STEP-06

Complete the recipe

After you have completed the action, you will return to a summary of the recipe. Click the 'Create a Recipe' button to actually create the recipe and make it active.



Maker Channel

The Maker Channel allows you to connect IFTTT to your personal DIY projects. With Maker, you can connect a Recipe to any device or service that can make or receive a web request (aka webhooks). See how others are using the Maker Channel, or share your own experience at hackster.io.

Connected as: srmonk

Your secret key is: cjvR3xPNFIPfKC2W4NZB9cd

>STEP-07

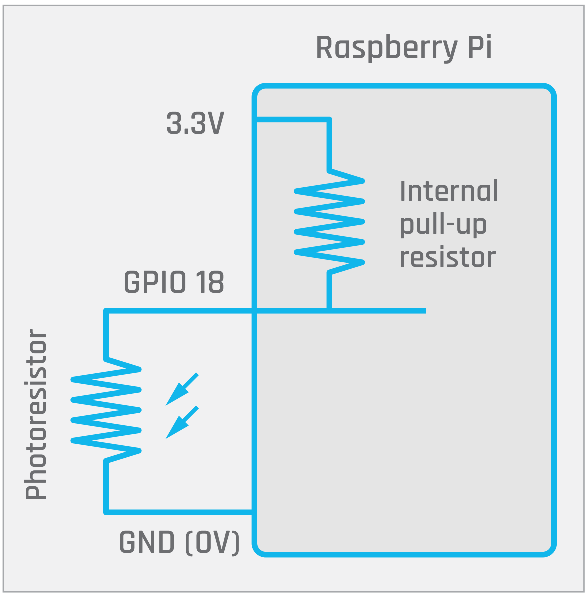
Find your secret key

Bad things could happen if just anyone were allowed to trigger emails to be sent using IFTTT. So, when the web request is sent from Python to trigger the email, it must be accompanied by a secret key. To find that key, click on the Channels tab at the top of the IFTTT webpage and then find the Maker Channel. In a little while, you are going to need to paste the secret key into your Python program.

Now that the hardware side of the project is complete, and IFTTT has been configured, we need to look at the Python program that will send the web requests to trigger IFTTT.

Download the Python program for this project using the following commands:

```
cd /home/pi
git clone https://github.com/simonmonk/
pi_magazine.git
```



Above Detecting darkness with a photoresistor

This command will actually bring down the code for all the projects in the author's *MagPi* series, so if you have already issued this command for one of the earlier articles, change directory to **pi_magazine** and run the following command to update your directory with this project (07_fridge_monitor):

```
git pull
```

How the code works

The Python code for this fridge monitor project is pretty straightforward. If you are interested in how the code works, load it up into a text editor while we go through it.

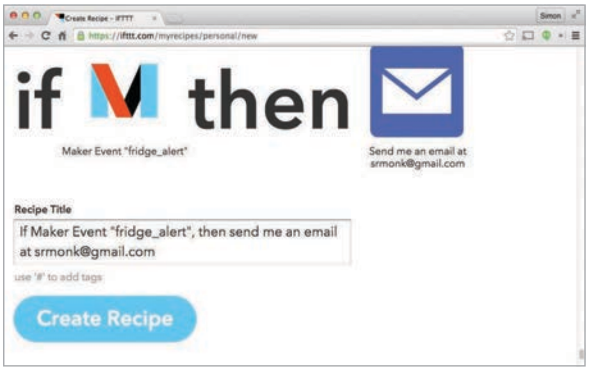
The first section of code includes all of the libraries you need. This includes the **RPi.GPIO** library that is needed to interface with the photoresistor and the **urllib2** library responsible for sending the web request to IFTTT.

The next section contains variables that are used to configure the program. **EVENT** must match the text that you specified as the Event Name when you created the trigger. You need to replace the value of the **KEY** variable with your key that you found in step 7 (see box on left).

The function **send_event** will be called by the main part of the program when IFTTT needs to be notified. The response from the web request is printed out, just to provide useful information while you are getting the project working.

The main program loop is contained in a **try: finally:** block, so that the GPIO pins will be tidied up when the program exits. The main loop inside **while True:** will keep running until you quit the program with **CTRL+C**. It checks to see if **LIGHT_PIN** is 0, indicating that the fridge door has been opened. When this happens, **send_event** is called.

To prevent a whole series of events being sent every time the fridge door is opened, two measures



if then

Maker Event "fridge_alert"

Send me an email at srmonk@gmail.com

Recipe Title

If Maker Event "fridge_alert", then send me an email at srmonk@gmail.com

use '#' to add tags

Above An IFTTT recipe is used to trigger an email

are taken. First, a **while** loop does nothing while the door is still open, to prevent repeat triggering every time around the outer loop. Second, when the door is finally shut, the program sleeps for a minute to prevent any further triggerings within that minute.

Using your fridge monitor

If you want to make your project a bit more permanent, you could tape the photoresistor to the top-inside surface of the fridge, rather than just letting it dangle like we did.

Before running the Python program, you can test out your recipe from the command line using the wget utility. Type the following command into LXTerminal. You will need to substitute your secret key.

```
curl https://maker.ifttt.com/trigger/fridge_alert/with/key/cyR3vPNF1P9K32W4NZB9cd
```

> “You might like to experiment by editing the recipe”

You’ll then see the message, ‘Congratulations! You’ve fired the fridge_alert event’.

Check your inbox and you should see an email from IFTTT. If anything went wrong, you should get an error message in place of the congratulations. You can also see the logs for your recipe on the IFTTT website by selecting the recipe and then clicking the icon that looks like a series of bullet points.

Start the fridge monitor program with the following command:

```
sudo python fridge_monitor.py
```

IFTTT is a very powerful system, so you might like to experiment by editing the recipe and changing the action so that it tweets or changes your Facebook status instead of sending an email.

Having a keyboard, mouse, and monitor attached to your fridge-mounted Pi is fine during testing, but really, it would be better to have the fridge monitor program start automatically when the Raspberry Pi first starts up. To do this, run the following command to make the program executable.

```
sudo chmod +x fridge_monitor.py
```

Then, edit the file **/etc/rc.local** with the command:

```
sudo nano /etc/rc.local
```

Add the following line after the first block of comment lines that begin with #.

```
sudo /home/pi/pi_magazine/07_fridge_monitor/fridge_monitor.py &
```

Restart your Raspberry Pi and this time the fridge monitor program should start up automatically.

fridge_monitor.py

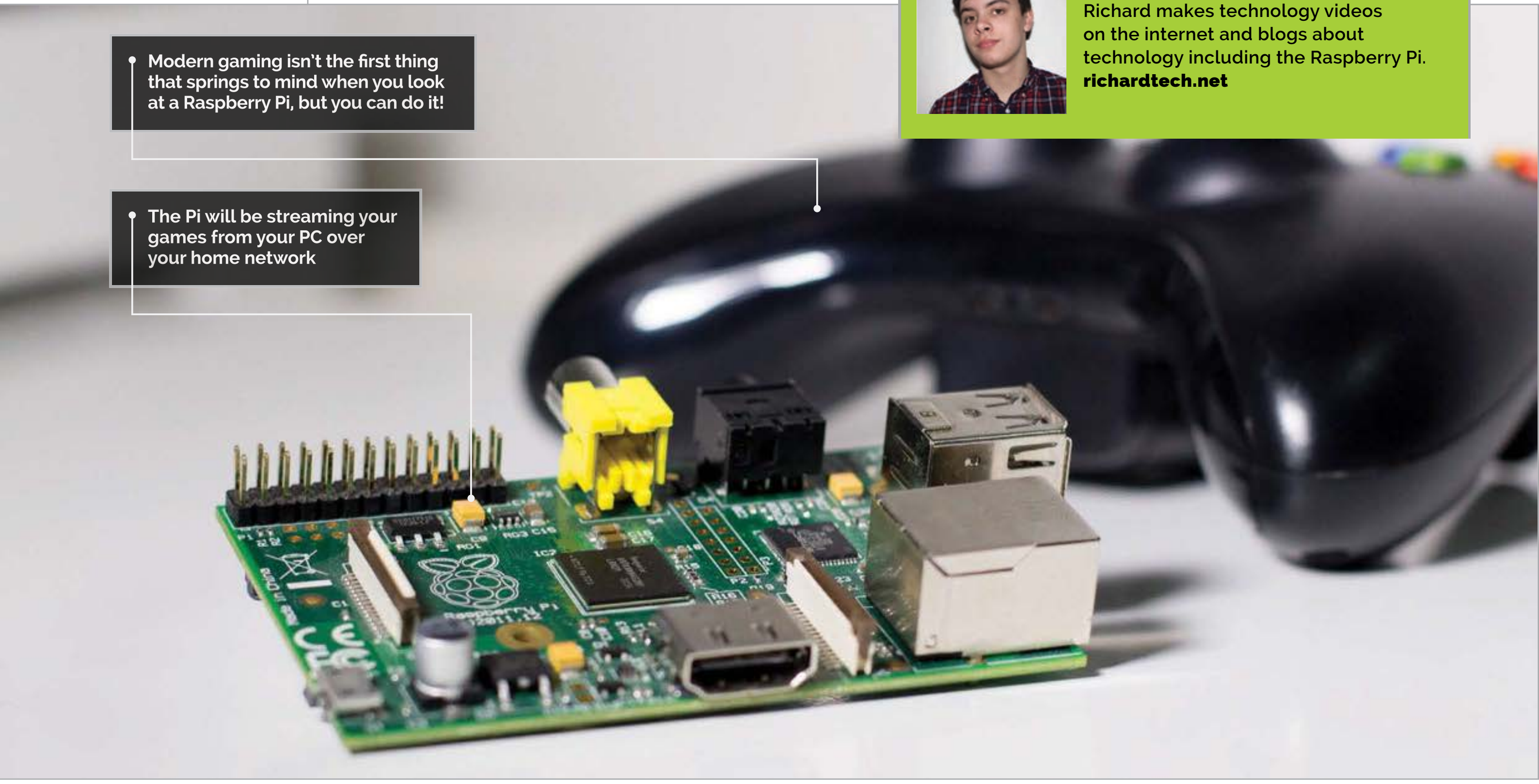
Language
>PYTHON

DOWNLOAD:
Github.com/simonmonk/pi_magazine/tree/master/07_fridge_monitor

```
import time
import RPi.GPIO as GPIO
import urllib2

LIGHT_PIN = 18   # photoresistor pin
EVENT = ‘fridge_alert’
BASE_URL = ‘https://maker.ifttt.com/trigger/’
KEY = ‘cyR3vPNF1P9K32W4NZB9cd’

# Configure the GPIO pin
GPIO.setmode(GPIO.BCM)
GPIO.setup(LIGHT_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

def send_event():
    response = urllib2.urlopen(
BASE_URL + EVENT + ‘/with/key/’ + KEY)
    print(response.read())

try:
    while True:
        if GPIO.input(LIGHT_PIN) == 0:
            # Its light (door open)
            send_event()
            # Do nothing until the doior is closed again
            while GPIO.input(LIGHT_PIN) == 0:
                time.sleep(0.1)
            # Do nothing for a further minute anyway
            print(“Wait a minute”)
            time.sleep(60)
            print(“Monitoring again”)

finally:
    print(‘Cleaning up GPIO’)
    GPIO.cleanup()
```

NEXT MONTH

In the next project in this series, we’ll create a garden waterer that will keep your plants alive while you go on holiday.



Modern gaming isn't the first thing that springs to mind when you look at a Raspberry Pi, but you can do it!

The Pi will be streaming your games from your PC over your home network

RICHARD WATERWORTH

Richard makes technology videos on the internet and blogs about technology including the Raspberry Pi.
richardtech.net

STREAM STEAM GAMES TO YOUR RASPBERRY PI

The Raspberry Pi can do a lot of things, but maybe you didn't know that you can use it to stream your favourite games from your PC

You'll Need

- Nvidia GTX 650 graphics card or higher
- GeForce Experience **geforce.com/geforce-experience**
- Raspbian
- Internet connection

Sometimes it's just not convenient to play the latest PC games on your clunky, noisy PC. It's either tucked in the corner of the bedroom or under the stairs, right where it's bound to disturb other people from the moment you press Start to play.

Never fear, though, because it's possible to play your favourite triple-A PC games from the comfort of any TV in the house using nothing more than a suitable graphics card, a Raspberry Pi and this guide…

>STEP-01

Install dependencies

Before we begin, you will need to make sure that you have all the dependencies installed; typically, these will already be installed with Raspbian. The dependencies are essential for Moonlight, the application we are using on the Pi to stream, to work. To make sure you have all of the dependencies installed, use the following command:

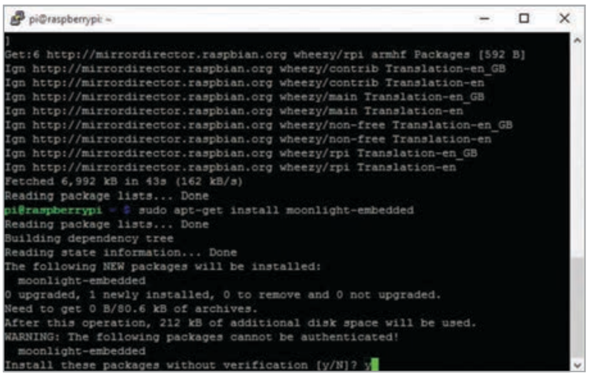
```
 sudo apt-get install libopus0 libasound2
libudev0 libavahi-client3 libcurl3 libevdev2
```

Once this has been done, we can install Moonlight, after we have configured the sound.

>STEP-02

Configuring sound

To make sure that we have sound coming out of the HDMI port, we need to edit the boot configuration file and add a line of text. To edit the boot configuration file, you will need to enter the following command:



```
]
Get:6 http://mirrordirector.raspbian.org wheezy/rpi armhf Packages [592 B]
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 6,992 kB in 43s (162 kB/s)
Reading package lists... Done
pi@raspberrypi ~ $ sudo apt-get install moonlight-embedded
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  moonlight-embedded
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/80.6 kB of archives.
After this operation, 212 kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  moonlight-embedded
Install these packages without verification [y/N]? y
```

Above Setting up game-streaming is as simple as typing in a selection of console commands

```
sudo nano /boot/config.txt
```

This will then bring up the Pi's nano text editor. You will want to go to the end of the file and enter this line:

```
hdmi_drive=2
```

Once this is done, the sound should be ready to go!

>STEP-03

Downloading and installing Moonlight

Now we're getting somewhere! But just before we can download Moonlight, we need to add its source to the Raspberry Pi's sources list. To edit the sources list, type in this command:

```
sudo nano /etc/apt/sources.list
```

This should open up the sources list in the nano text editor. Now all we need to do is add the following line to our sources list to add Moonlight's source:

```
deb http://archive.itimmer.nl/raspbian/
moonlight wheezy main
```

Once this has been done, we can save and quit the file, and then check for updates on the Pi:

```
sudo apt-get update
```

Once this command has been run, we can finally install Moonlight!

```
sudo apt-get install moonlight-embedded
```

So, now we've finally managed to get Moonlight installed, let's move on to how we can use it.

>STEP-04

Setting up Moonlight

First of all, you will need to get the IP address of your Windows PC on the network. Go to the Command Prompt (in Accessories) on the PC and enter:

```
ipconfig
```

This will obtain the IP address of your PC. If there is a big list, your IP address should begin with '192.168'.

Now we can pair Moonlight with the computer by using the following command on the Pi:

```
moonlight pair YOUR-IP
```

...replacing **YOUR-IP** with the IP address of your Windows PC. Once you've entered this command, a PIN code will appear in the terminal, which you will need to enter into the GeForce Experience box that pops up on your Windows PC.

> "There are many options that can be added to the moonlight command"

>STEP-05

Streaming with Moonlight

Streaming with Moonlight is very simple; all you need to do is type into the terminal:

```
moonlight stream YOUR-IP
```

Steam Big Picture mode will automatically pop up. By default, the stream will be 720p at 60fps, but we've had better luck on our network at 30fps. There are many options that can be added to the command. To stream at 30fps, for instance, you would enter:

```
moonlight stream -30fps YOUR-IP
```

You can find all the options for the **moonlight** command at **bit.ly/moonlight-options**. These include 1080p streaming, changing bitrates, resolutions and much more!

USING ETHERNET VS WI-FI

Using Ethernet will allow you to get a better frame rate and resolution compared to using Wi-Fi.

USE A CONTROLLER

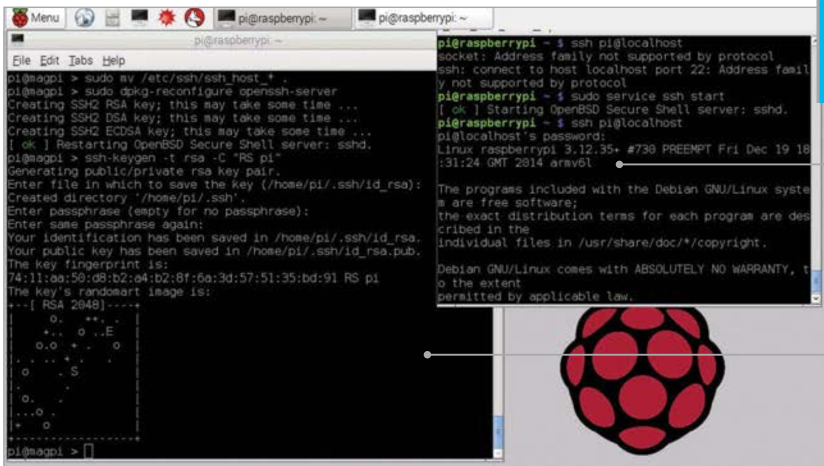
Moonlight does support controllers, so you can game as if the Pi were a console. Sadly, we didn't have much luck with our controller, but yours may be more of a success.

Below Play some of the hottest games on your Pi, by streaming them from your PC's Steam games library

RICHARD SMEDLEY

Having often found words better than pointing at things, Richard stuck with the command line when all around had fled.

twitter.com/RichardSmedley



```
pi@magpi > sudo mv /etc/ssh/ssh_host_* .
pi@magpi > sudo dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...
Creating SSH2 DSA key; this may take some time ...
Creating SSH2 ECDSA key; this may take some time ...
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
pi@magpi > ssh-keygen -t rsa -C "RS pi"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pi/.ssh/id_rsa):
Created directory '/home/pi/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pi/.ssh/id_rsa.
Your public key has been saved in /home/pi/.ssh/id_rsa.pub.
The key fingerprint is:
74:11:aa:50:d8:b2:a4:b2:8f:6a:3d:57:51:35:bd:91 RS pi
The key's randomart image is:
+--[ RSA 2048]----+
|     o.   ++. .  |
|    +..  o ..E   |
|   o.o  + .   o  |
|. . .. + .    .  |
| o     . S       |
|.       .        |
| o.    .         |
|...o .           |
|+    o           |
+-----------------+
pi@magpi > 
```

```
pi@raspberrypi ~ $ ssh pi@localhost
socket: Address family not supported by protocol
ssh: connect to host localhost port 22: Address famil
y not supported by protocol
pi@raspberrypi ~ $ sudo service ssh start
[ ok ] Starting OpenBSD Secure Shell server: sshd.
pi@raspberrypi ~ $ ssh pi@localhost
pi@localhost's password:
Linux raspberrypi 3.12.35+ #730 PREEMPT Fri Dec 19 18
:31:24 GMT 2014 armv6l

The programs included with the Debian GNU/Linux syste
m are free software;
the exact distribution terms for each program are des
cribed in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, t
o the extent
permitted by applicable law.
```

You can test on the Pi if SSH is running, and start the service from the command line – as you can any service (look in **/etc/init.d/** and **/etc/init/** if you're curious about other services)

The Raspbian install image shares its keys with everyone else who has a copy. Generate your own, and personal keys for the user account, for secure remote access

COMMAND LINE PI PART 7: REMOTE PI

Richard Smedley presents your cut-out-and-keep guide to using the command line on the Raspberry Pi. In part 7, he covers accessing the Pi from remote PCs and devices, with Secure Shell

You'll Need

- Raspbian **raspberrypi.org/downloads/** – though most of the tutorial series will work with the command line running the Linux default Bash shell on any GNU/Linux PC.

INTERRUPTED SERVICE

While you can restart most services with **sudo service ssh restart**, replacing **restart** with **reload** will allow configuration changes to be registered with less disruption – important for some projects.

It's great that the Raspberry Pi is so portable, but sometimes you may want to use it without taking it with you. Here, the Pi's default operating system is a real strength, as Unix-like OSs have been used this way for over 40 years.

Over the decades, as the internet has given the opportunity for malicious users to connect to your computers, old standards like telnet and rlogin have been replaced by Secure Shell (SSH), based on public-key cryptography. The good news is that once set up, secure connections are just as easy, and are open to scripted, automatic connection for your projects.

With Raspbian, the SSH server is enabled by default. If not, run **sudo raspi-config**, then enable SSH (found in the advanced settings). Check the IP address assigned to the Pi with **ifconfig** ('inet addr' for the etho interface). Now you can try connecting from another computer on your network.

Connecting with SSH

From a Mac or GNU/Linux computer, just use **ssh** from a terminal to connect to your Pi. Assuming a default setup, and ifconfig revealing an IP address of 192.168.0.2, connect with:

```
ssh pi@192.168.0.2
```

...and enter your password when asked. From a Windows PC, you'll need to install an SSH client: we recommend Putty (**bit.ly/1dTlX8g**), which also works with scp, telnet, and rlogin, and can connect to a serial port. Android users can try the ConnectBot client.

You should now be at the command-line interface of your Pi. If you got any sort of error, check from the Pi that SSH is really up and running by entering **ssh@localhost** on the Pi itself – if that works, SSH is up and running on the Pi, so take a closer look at network settings at both ends.

Hello, World

Now we can access the Pi on the local network, it's time to share with the world. Step one, before even thinking about going further, change the **PermitRootLogin yes** entry in **/etc/ssh/sshd_config** to read:

```
PermitRootLogin no
```

...using **sudo nano**. After making any changes to the SSH server's configuration, you must restart the service for them to take effect, or at least reload the configuration file:

```
sudo service ssh reload
```

Note there's also a file in /etc/ssh/ called ssh_config, which is for the SSH client; the **d** in **sshd_config** is short for ‘daemon’, the Unix term for a service which runs constantly in the background.

Other useful changes include port 22 to any unlikely number – but check it works afterwards. You’ll need to begin **ssh -p 12123** or whatever to tell your client you’re not using the default port 22.

To reach your Pi from anywhere on the internet, you need an IP address, which will connect you to your board even though it’s behind an ADSL router (of course, if your Pi is in a data centre, with its own public IP address, you don’t need any workaround).

There are numerous services such as DuckDNS.org providing free-of-charge dynamic DNS (DDNS), to point a constant IP address to the changing one allocated to you by your ISP – but since the largest of them, DynDNS, ended its free service, that does provide a reminder that you cannot expect a free service to be around for ever.

There are several steps to configuring a DDNS setup, no matter which service and software client you choose. Some are detailed in the RaspberryPi.org forums, and there’s a good guide to *ddclient* at **samhobbs.co.uk**.

Otherwise, if your broadband router can handle both port forwarding and dynamic DNS, you can set it up to point to port 22 (or a chosen alternate port) on the Pi. You may even find your ISP offers static IP addresses.

Bye bye FTP

FTP, dating back to 1971, was not designed for security – data, and even passwords, are transmitted unencrypted. *scp*, the Secure Copy Program, which runs over SSH, is included in the Pi’s SSH packages. The syntax of the command is fairly familiar, as it mimics the command-line *cp* program, adding in the path for the remote side of the transaction’s network address.

```
scp pi@192.168.0.2:/home/pi/bin/lein .
```

Here we’re transferring a file from the Pi, across a local network, to the current location (the dot shortcut). Note that you can use wildcards for groups of similarly named files, and can recursively copy directories and their contents with the **-r** switch after **scp**.

A secure key

If you’re trying this on something other than Raspbian, you may not have the SSH server installed. It’s often found in a package called **openssh-server**. With Raspbian, you have a pair of keys (public and private) in **/etc/ssh/** – unfortunately, they’ll be the same as those held by everyone else with a copy of the Raspbian image that you downloaded. Follow these steps to create new keys.

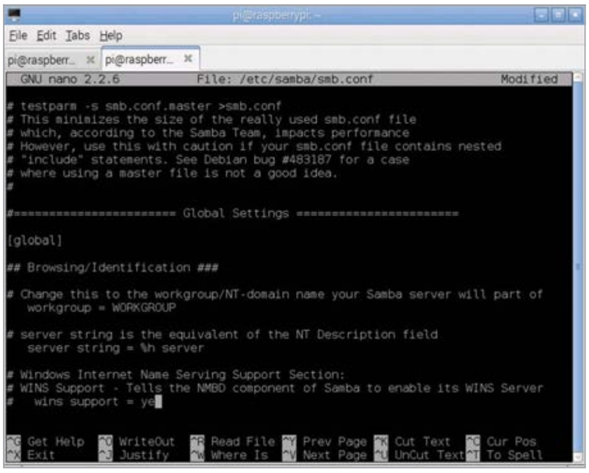
First, remove the existing keys:

```
sudo rm /etc/ssh/ssh_host_*
```

...or move them somewhere out of the way. Regenerate the system-wide keys with:

```
sudo dpkg-reconfigure openssh-server
```

But for keys personal to you as a user, type **ssh-keygen -t rsa -C “comment”** - where “comment” is anything you want to identify the key with: name, email, or machine and project, for example. If you press **ENTER** at the passphrase step, you’ll get a key with no passphrase, which makes life easier when making scripted (automated) connections, but removes an extra layer of security. You can create keys from any computer with the SSH package, and move the public key to the Pi – but we’ll work on the assumption that the Pi is the only handy *nix computer, generating the keys there.



```
  GNU nano 2.2.6            File: /etc/samba/smb.conf                  Modified

# testparm -s smb.conf.master >smb.conf
# This minimizes the size of the really used smb.conf file
# which, according to the Samba Team, impacts performance
# However, use this with caution if your smb.conf file contains nested
# "include" statements. See Debian bug #483187 for a case
# where using a master file is not a good idea.
#

#======================= Global Settings =======================

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
   server string = %h server

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
#   wins support = ye

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Left There’s a lot of configuration in Samba, but simply adding your WORKGROUP name to the default settings should get you up and running

If you accepted the defaults, your personal keys will now be in **~/.ssh** with the correct permissions. By default, sshd looks in **~/.ssh/authorized_keys** for public keys, so copy the contents of **id_rsa.pub** to there – the following will work even if you already have an **authorized_keys** with contents (make sure you use both **>>** symbols with no gap between them):

```
cd ~/.ssh && cat id_rsa.pub
 >> authorized_keys
```

Copy the private key to **~/.ssh** on your laptop, or wherever you will access the Pi from, using scp, removing it from the Pi if it’s to act as the server. Once you confirm SSH works without passwords, you can edit **/etc/ssh/sshd_config** to include **PasswordAuthentication no** – or if sticking with passwords, replace **raspberry** with something stronger.

Shared drive

You may be using a service like Dropbox to share files between machines – no need to do this on a local network, as putting Samba on the Pi lets even Windows PCs see it in Network Neighbourhood, and access it as a shared drive:

```
sudo apt-get install samba samba-common-bin
```

Now edit **/etc/samba/smb.conf** with a WORKGROUP value (for Windows XP and earlier – try **workgroup = WORKGROUP**) and/or HOME (For Windows 7 and above). Ensure Samba knows **pi** is a network user:

```
sudo smbpasswd -a pi
```

Then restart with:

```
sudo service samba restart
```

The Pi should now show up in Windows Explorer under Network. You can fine-tune **smb.conf** for what’s shared (including printers), and permissions.

LOST KEYS?

The private key half of your key pair should be kept secure – but safe, too. Keep a backup of the private key on a memory card in a safe place.

SAMBA STEPS

Samba is *extremely* well documented, with separate man pages for everything from **smb.conf** to **smbpasswd**, and excellent online books at **samba.org** – look for smb.conf examples.

SEAN MCMANUS

Sean McManus is a Code Club volunteer and wrote the book *Scratch Programming in Easy Steps*. He co-wrote *Raspberry Pi For Dummies*.

sean.co.uk
twitter.com/musicandwords

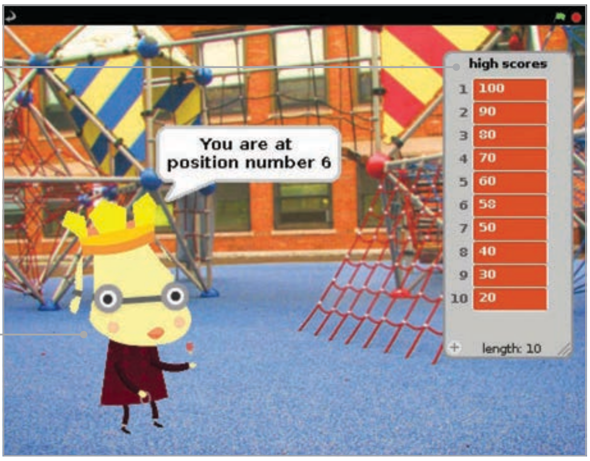
ADD A HIGH SCORE TABLE TO YOUR SCRATCH GAME

Keep players coming back for more by keeping a record of the best scores, and telling them how they measure up

You'll Need

- One of your Scratch games. Pick a favourite!

This project features scripts that enable you to create a high score table, and then add new scores to it if they're high enough. There isn't an easy way to display and hide a list from within your program, so the scripts also tell players how they ranked and what the next highest score is, so they know how close they came to beating it. This code will work with most simple games, but you might need to make some changes if your game invites players to play again, or has scripts that continue when the game has ended.



high scores

1	100
2	90
3	80
4	70
5	60
6	58
7	50
8	40
9	30
10	20

length: 10

Tick the box in the Blocks Palette to see the list and edit its values. No cheating, now!

The sprite checks the player's score and tells them how they did

>STEP-01

Make your game

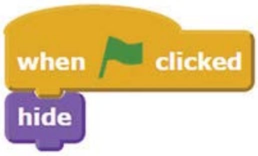
You'll need a game to add this script to – either one of your own, or one that you've programmed from a book or magazine. Try playing the game a few times to work out the likely scores. Some games award a few points, some hundreds, and some thousands. The starting numbers in your high score table should present a challenge to players, but not be completely unachievable. Take care with your own games: if you've spent days playing them in development, they'll be much easier for you than anyone else.

>STEP-02

Add your high score sprite

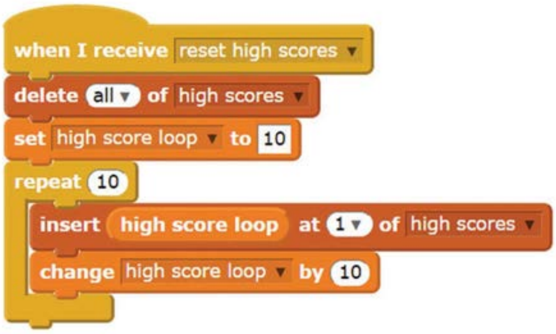
The scripts for the high score can all go on the same sprite. This sprite will tell players if they got a high score. It could be the main character of your game, the sprite used on the title screen (see *The MagPi* #36), or it could be a new sprite. We've added the sprite `royalperson` for our high score table. You'll find it in the 'people' folder, even though it looks like a dog. It'll be in the way during the game, so add **Listing 1** to hide it when the green flag is clicked.

LISTING 1



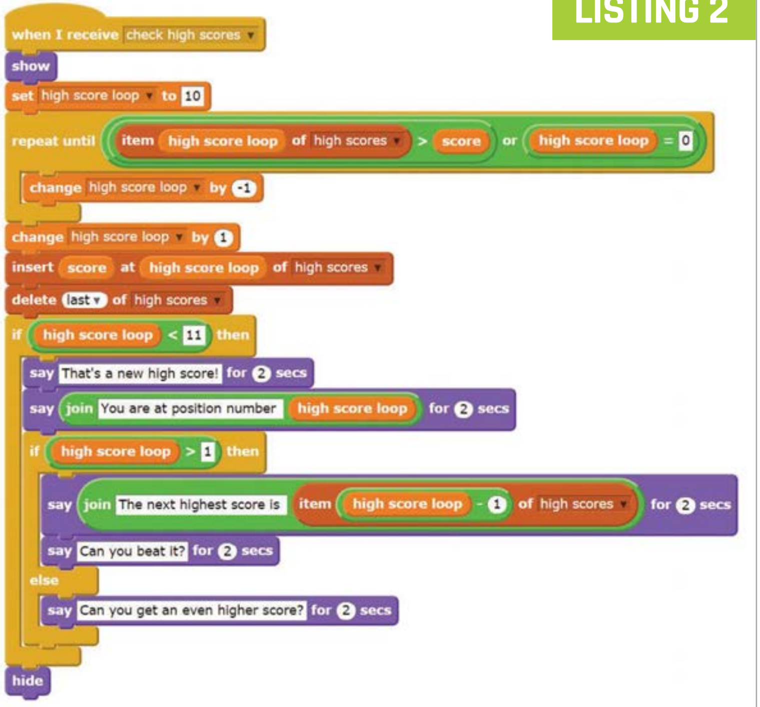
```
when [flag] clicked
hide
```

LISTING 3



```
when I receive [reset high scores ▾]
delete (all ▾) of [high scores ▾]
set [high score loop ▾] to (10)
repeat (10)
    insert (high score loop) at (1 ▾) of [high scores ▾]
    change [high score loop ▾] by (10)
end
```

LISTING 2



```
when I receive [check high scores ▾]
show
set [high score loop ▾] to (10)
repeat until <<(item (high score loop) of [high scores ▾]) > (score)> or <(high score loop) = (0)>>
    change [high score loop ▾] by (-1)
end
change [high score loop ▾] by (1)
insert (score) at (high score loop) of [high scores ▾]
delete (last ▾) of [high scores ▾]
if <(high score loop) < (11)> then
    say [That's a new high score!] for (2) secs
    say (join [You are at position number ] (high score loop)) for (2) secs
    if <(high score loop) > (1)> then
        say (join [The next highest score is ] (item ((high score loop) - (1)) of [high scores ▾])) for (2) secs
        say [Can you beat it?] for (2) secs
    else
        say [Can you get an even higher score?] for (2) secs
    end
end
hide
```

> “This code will work with most simple games, but you might need to make some changes if your game invites players to play again”

>STEP-03

Set up your list

Your high score table will be stored in a list. Click the Variables button above the Blocks Palette, click the button to make a list and call it ‘high scores’. In the Blocks Palette, you can click the tickbox beside the list name to show or hide the list on the Stage. This is a handy way to view the whole list, and you can edit the values in it by clicking them and typing on them. The list gets in the way of your game, so we recommend unticking the box.

>STEP-04

Set your starting scores

You can type some starting scores into the list on the Stage, but it’s better to use a script to generate your high scores. **Listing 2** does this. It runs if it receives the broadcast **reset high scores**, but you can also click the script once to reset your scores. To change the lowest score, change the value in the **set high score loop** block. To change how much scores go up by, edit the value in the **change high score loop** block. Note: the pointed Operator blocks are shown as rounded in our code because of limitations in the Scratchblocks software we use for laying out code for the magazine.

>STEP-05

Add your high score code

Listing 3 checks the score and adds it to the high score table in the correct position if high enough. It also tells the player how well they did. Add it to your high score sprite. Take care with building the script that goes in the hole of the **repeat until** block. You’ll need to drag in blocks in a similar order to this: **or**, **>**, **item 1 of high scores**, **high score loop**, **=**, **high score loop**. When the next highest score is announced, add blocks in the order: **say Hello! for 2 secs**, **join hello world**, **item 1 of high scores**, **-**, **high score loop**.

>STEP-05

Patch it in to your game

To finish, connect your high score script to your game. If the game doesn’t already use the variable **score**, click on Variables and make that variable for all sprites. You want the high score script to run when your game ends, so you need to add some code at that point in your game. Add a block to set **score** to your game’s score variable if you’re not already using the variable **score** in the game. Finally, add a block to broadcast **check high scores**. To keep your high scores, simply save your game. When you save a Scratch program, the list values, including your high score table in this case, are saved too.

MIKE'S PI BAKERY

MIKE COOK

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

COMPUTER VISION SEQUENCER

Build up a musical piece by putting coloured pegs into holes with this exciting computer vision project...

You'll Need

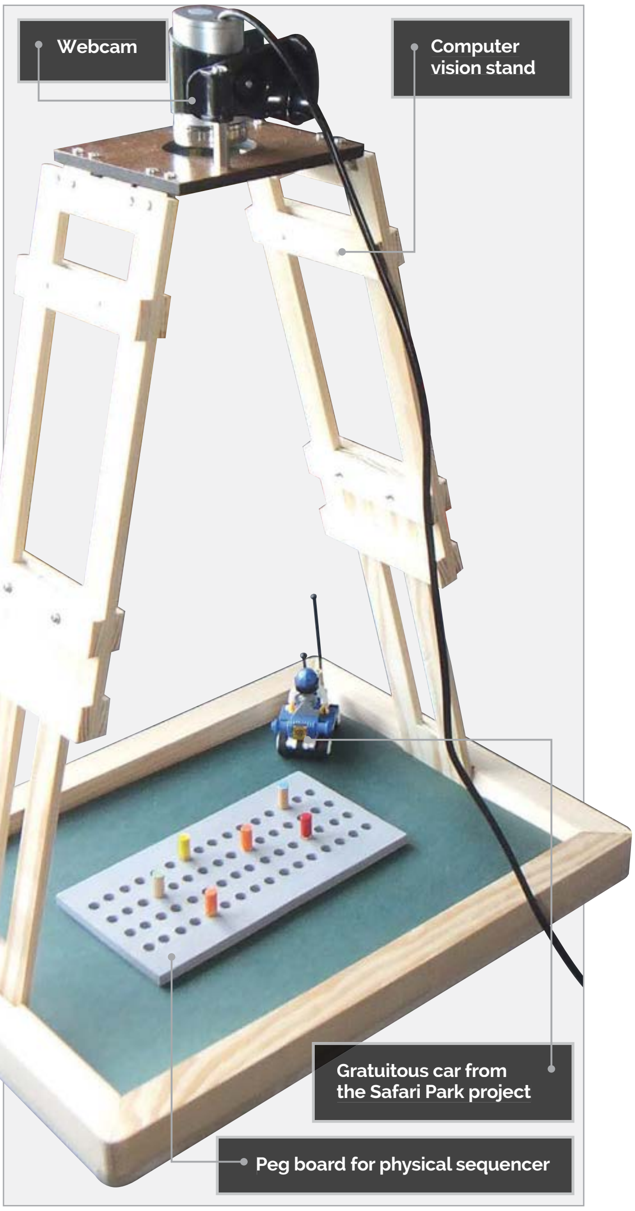
- Webcam
- Raspberry Pi 2 Model B
- Multiple colours of acrylic paint and primer
- Adjustable neutral density filter (optional)
- 20mm M3 threaded spacer
- 20 x 20mm pine strip
- 17 x 6mm pine strip
- 6mm MDF
- 6mm plywood
- 6mm dowel
- 4x miniature hinges
- 1mm foam strip
- Assorted nuts and bolts

One of the new territories the Raspberry Pi opens up for exploration is video processing/computer vision, and we have a number of ideas for projects we can do using these techniques. This month we want to see how to get a basic setup going and make a seemingly simple project with it. However, as you will learn, nothing is quite as easy as it seems when it comes to computer vision.

The basics

We will be using the OpenCV computer vision library for these projects. If you don't already have it installed on your Raspberry Pi, then we would advise you to look up *The MagPi* #32, where Willem Koopman wrote an excellent article (page 48) on how to install it and do a bit of facial recognition. It can take a few hours to install, depending on your internet connection speed and what Pi model you have. We would recommend using a Pi 2 for all computer vision work, as you need all the processing speed you can get.

Next, you need to get hold of a compatible webcam. You could use the Raspberry Pi's own Camera Module, but this is not recommended because of its high resolution. A high resolution means lots of pixels, which means lots of pixels to process, and that takes time. It is normal to work on images as small as possible with computer vision work, to minimise the amount of processing you have to do. Many webcams have a curved clip arrangement for hooking on top of your monitor; while this looks unpromising from an adaptation point of view, it in fact proves to be very easy to mount on a piece of your own hardware. Simply unscrew the joint, remove the bottom part of the clip, and mount the camera on a single pillar or spacer.



Webcam

Computer vision stand

Gratuitous car from the Safari Park project

Peg board for physical sequencer

The webcam stand

One thing that takes a lot of processing time is the location of objects. You can circumvent this by making a stand and tray for the webcam. We made a bipod – that is, a tripod but with only two legs. Normally this would not be stable, but by bracing it in slots either side of the frame and having a base board fill the frame, something robust enough for our experiments could be made. We made the webcam stand from a combination of strip pine, plywood, and MDF. The construction is in two parts: a stand and an A4-size base. We made the stand capable of being adjusted to two heights. The tallest height was designed so that the whole of the base could be seen by the camera, and the lowest so the camera field could be concentrated on a small area of the base. This added a bit of complication to the design, but was well worth it. For this project, we used the stand at its lower position. The legs are held in position by M3 bolts; no nuts were needed (see the step-by-step guide on page 55 for full details of construction).

While the webcam we got worked fine using other software, when it came to using it with OpenCV the images were very much overexposed and it was way too sensitive to light, resulting in the colours being all washed out. Despite trying many software remedies to control the exposure, our webcam stubbornly refused to succumb to any type of meaningful control. In the end, we reverted to buying an adjustable neutral density filter (from eBay at just under £7) to control the exposure of the image. You might be luckier with your camera.

The software - principal

There are a great many tutorials online detailing how to use OpenCV with Python, and some of them actually work! The majority, however, do not; this is due to changes in the way commands work and the way they are specified in the latest version of the library. All the tutorials seem to be one or two software revisions behind the current one. This can lead to a bit of frustration from a beginner's point of view and make it difficult to learn the language. The other frustration is that the window OpenCV generates to show the image does not behave like normal windows; the close X box will not shut it down. It will appear to close, but then it will open up another window in a random place on the desktop. It seems there is no simple solution to this; there might be a complex, labyrinthine Linux solution, but this distracts from the job in hand.

One way to think of a webcam is that it is a vast array of colour sensors; if you know where to look, you can easily pick out a colour. That was the thinking behind making a physical sequencer: simply have a handful of coloured pegs and stick them into holes in a board. Then use the pegs to generate a sound sequence, with the colour determining the sound and the position determining when that sound is played. That sounds so simple, but to get a computer to do it is not so easy. The first problem is where to look; you might think this is simply evenly spaced across the area of the board, but that is not taking perspective into account. A peg only looks like it is a round shape on a grid when the camera is directly above it. If the camera is off to one side, then it will look decidedly off the grid. The solution is to have a record of where every point you want to sample actually is in the image, then your program knows what pixels to look at. Gathering these points is a tedious and repetitive operation, which is fortunately just what a computer is good at. So the first program we have is one to gather all the sample points and put them into a file. Then the sequencer software proper can simply read this file and use it to control the notes and the timing.

Handling colours

The next problem is identifying the colour of a pixel. Sure, you can just read the red, green, and blue components of any pixel on the screen, but it is not as simple as that. This is because the same object can return different RGB values depending on the lighting, the angle between the object and the camera, and even where in the camera's field of view it is. Also, different shades of a colour produce vastly different numbers. Therefore, we need a more robust way of defining a colour than just three values. What we need is that each colour is defined not by a specific number but by a range of values. It is a bit complex defining a range over three values, so it would be simpler if the three colour values were distilled down into a single value representing the colour. This is what is done with the HSV colour space model, which defines a colour as three parameters: H (hue), S (saturation), and V (value or brightness).

The H value is just what we want to categorise the pegs, but what if there is no peg? This is where the 'chroma' parameter comes in. This is the deviation a colour has from the grey line – a line in colour space which joins black to white. Colour space simply means expressing the parameters of a colour as if it were a geometrical space, hence the RGB representation of a colour is often known as a colour cube, and any colour can be thought of as a point in that cube. So we only need to consider categorising the colour of a pixel once it is known to be coloured and not grey. The H values can be simply divided up into bins and any unknown colour assigned a colour number, depending on what bin it falls into. Note that these bins are not evenly spaced, as there are a large number of H values that the eye sees as blue compared to the small number it sees as red; this applies to the other colours as well.

So now we have the theory, we can start to look at how this can be put into practice.

The calibration code

First, look at the **calSetter.py** code (page 56). This gathers sample points in the video frame and puts them into a file. As set up, it gathers eight rows of eight equally spaced points by simply clicking on the start and end points. This allows you to compensate for the perspective distortion over the whole peg board by selecting the far left side followed by the left centre peg, then the right centre peg followed by the far right peg. This is repeated for each of the four rows and then you have the opportunity of writing the resulting points out to a file. The configuration for this is set up using the **ppr** and **nor** variables.

The **main** function prints out some instructions and then enters a **while** loop, which displays the current frame from the webcam and overlays circles at the points already defined by calling the **overlay** function. The current mouse pointer is used to also draw a circle; this is tracked by the mouse callback function **track_mouse**. When a mouse click happens, the location of that click is stored and, on every other click, points are defined between the last two clicks with the **calculatePoints** function. When all the points have been defined, you can press the **S** key to save them and type in the file name you want to use to the Python console. This will then close down the program, as will pressing the **ESC** key at any time. Note that you can use this program to define any set of points over the video field; use your imagination. Don't expect a rapid response from the mouse, since there is a lot going on here; just move slowly and the circle will follow eventually.

> "The program runs under the Pygame system to handle the sounds"

The sequencer code

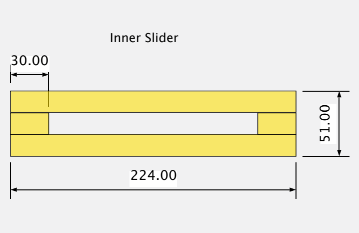
The **pickle** package is used to format the data so that it can be read in directly by another program and simply restore the **points** tuple list. This is picked up by the **videoSequencer.py** program (page 56), which is the physical sequencer proper. The program runs under the Pygame system to handle the sounds, which are stored in a **sounds** directory at the same level as the program. There are six different colours recognised, each one triggering a different sound. At each of the 16 steps, up to four different sounds can be produced; this is known as four-note polyphony. The program has two sliders attached to the window: one to switch the sequencer on and off, and the other to set the speed in BPM (beats per minute). These sliders are not as responsive as you might hope, so give it a second or so when moving them. An image is taken then analysed for the colours, and then the sounds are outputted in turn until the end of the 16-step sequence. Due to the time it takes to output the sounds, the OpenCV video buffer has filled up with four frames, so these are read off quickly so that the last image frame you read is the latest frame. This means you have the whole sequence time to change any pegs for the next run-through. The program can be quit in two ways: pressing the **ESC** key quits the program, whereas pressing the **D** key saves a CVS (comma-separated value) file, containing the colour value for each position, before quitting the program. This file can be opened with any text editor or by a spreadsheet program; it will allow you to fine-tune either the picture or the bins that define the colours in the **getColour** function. You can move the boundaries around or even add more colours if you like. The colours are taken from the average of a 3×3 pixel square centred around the target pixel, to minimise single pixel errors. The chroma threshold – that is, the point where a pixel value is considered to be a valid colour and not grey – is set by the **cromaThresh** variable. While playing the sequence, the four positions defining the sound are joined together with a thick green line. This gives a visual indication of where in the sequence you are and how close to the end it is, so you can avoid getting your hand in the next frame. However, you might want to do that intentionally, as it seems to result in a small solo that relieves the repetition. Unfortunately, mouse movement makes the operating system steal some time from the program, so the green sample lines might stop being updated if you waggle the mouse around. When the sequencer is turned off by the slider, the video is updated as fast as possible; this allows you to adjust the peg board's position to match up with the sample points that are shown outlined by a dotted square.

Customisation

Like all Bakery products, it's now ready to spread your own jam across it. You can change the sound samples or the sequence length/polyphony. We used colour swatches from the DIY store to add variety to the last row, by slipping it over the peg board for the next sequence. You can replace the peg board with a coloured drawing or photo from a magazine and see what that sounds like. You can experiment with different sample points for a picture and change the code so you can type in what sample file you want to use. You can include many sample point files and switch between them at the end of each sweep through the sequence. How about changing the sound samples to notes from an instrument, extending the number of rows, and making a virtual music box? We're sure you can think of many ways to take this project further. Have fun!

BUILDING A CAMERA STAND AND PEG BOARD

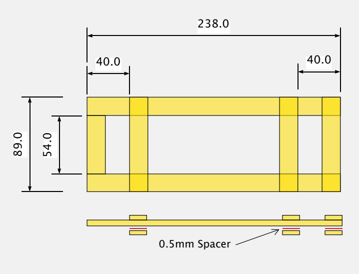
>STEP-01
The inner slider legs



Make the inner slider legs – you need two of these. At first, we used white woodworking PVA, but in the end this proved not to be strong enough, and it also took 24 hours to set. A much better joint was obtained by using 'Gorilla Glue' (a foaming adhesive), with both sides of the joint being slightly moistened before applying a thin layer of glue and clamping them together. The glue will foam out of the edges and can be removed with a sharp knife or sandpaper after it has set, which should only take about two hours.

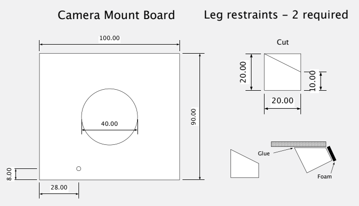
Cut and arrange the four strips of 17mm wide by 6mm thick stripboard and glue and clamp each edge. We used clamps to squeeze them together, as well as a clamp over the top to make sure this produced a flat and not warped small rectangle. To stop the work sticking to the bench, we used a sheet of cling film from the kitchen to stop the glue sticking in the wrong place. The clamping is important: the instructions on the glue say that the joint's strength depends on the tightness of the clamping.

>STEP-02
The outer support



Make the outer support – again, you need two of these. Cut the two 238mm strips of 17mm wide by 6mm thick stripboard, and also the single 54mm piece. Clamp and glue the edges together to make a long U shape. Use the inner slide wrapped in a piece of cling film as a spacer on the open end to make sure it is all square. Again, clamp to squeeze the joints together as well as clamping along the top. When the glue has set, remove the clamps and handle carefully so that you don't crack the joints. Next, clamp and glue the three 89mm strips to the top of this U shape; this time, only downward pressure clamps should be necessary. Finally, when this is set, flip it over and glue the other three 89mm strips to form the outer frame. You can insert a small square of spacer material under these strips to give a little play, to allow the inner slider to move freely but not sloppily. We didn't do this on the prototype - instead we sanded the inner slider to reduce its thickness slightly - but next time we will use a piece of 0.5mm styrene sheet to make life easier.

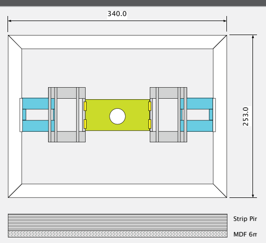
>STEP-03
The camera plate



Make the camera plate. Ours was made from a piece of 100mm by 89mm plywood, mainly because we had a piece already, but there is no reason why you can't make it from MDF. A 40mm diameter hole was cut in the middle using a saw drill. A 3mm hole was placed in the point in the diagram to allow the camera mounting pillar to be attached. The camera plate is attached to each of the sliding legs by two small hinges and M2.5 nuts and bolts. Two blocks were cut and glued onto the underside of the camera plate to act as stops for the legs.

A small piece of black foam was added to the end of the block to give a bit of springiness to the legs. If you are fitting a neutral density filter, then you might have to file a slight curve in the blocks to make room to fit the filter. The filter should be attached to the camera plate with a very small amount of hot-melt glue. If you do use a filter, you need to get a small piece of cardboard with a 10mm hole to fit on the top of the mount between the camera and the board, to suppress reflections from the top of the filter into the lens.

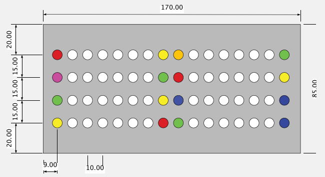
>STEP-04
The base



Once the stand has been made, it's time to construct the base. This is designed to be an A4 tray that you can mount the webcam over. The base is made from four pieces of 21mm square strip pine – two at 340mm and two at 253mm. The ends of each piece were cut off at 45 degrees. A recess was cut in the middle of the two shorter sides, to allow the legs of the stand to sit flush with the frame. The frame was glued on the diagonal corners and was held together with a 'square clamp' until the glue set.

Then a 340mm by 253mm piece of 6mm MDF was cut to make the bottom. It was drilled and countersunk so that eight screws could screw it into the frame – we used two screws for each side of the frame. When the frame was screwed to the base, the corners were rounded off using a disc sander. Screwing the base to the frame in this way allows you to remove the base to make fine adjustments to the recesses. Finally, a 296mm by 210mm piece of 6mm MDF was cut to make a tray that held the legs apart.

>STEP-05
The peg board



You can use the tray to make the peg board, or you can copy us and make the board from a 170mm by 85mm piece of MDF. Drill the holes in the peg board at 6.5mm diameter and paint it grey. We made a bit of grey paint by adding a touch of black paint to some white paint, but don't mix too much. Remember, the MDF will need a coat of MDF primer first, and don't forget to paint inside the holes.

Cut the pegs from 6mm dowel at a length of 20mm; this is a good compromise length between handleability and not producing much distortion on the camera. Paint the pegs with acrylic paint – a good source of low-cost paints can be found in the multi-tube artist sets of paints available from many retailers. Again, we found that adding a primer to the pegs first meant the colours were bright and uniform.

calSetter.py

```
# calSetter - defines a grid  of sample points
import cv2
import os, pickle

ppr = 8 #  points per row
nor = 8 #  number of rows
maxclick = nor * 2
ix,iy = -1,-1
target = [(-10,-10),(-10,-10)]
pointer = 0
showRow = False
points = [ (-10,-10) for p in range (0,ppr*nor) ]

# mouse callback function
def track_mouse(event,x,y,flags,param):
   global ix,iy,pointer, showRow
   if event == cv2.EVENT_LBUTTONDOWN:
       # save mouse position
       if pointer == maxclick :
          pointer = 0
          showRow = False
       target[pointer & 1] = (x,y)
       if (pointer & 1) == 1:
          calculatePoints(pointer / 2)
          showRow = True
       pointer +=1
       if pointer == maxclick:
          print"locate start point on row 0 again or 's' to
save points"
       else :
          if (pointer & 1)== 0:
             print "locate start point on row",pointer/2
          else :
             print "locate end point on row",pointer/2
   elif event == cv2.EVENT_MOUSEMOVE:
       ix,iy = x,y

cv2.namedWindow("Sample point define", cv2.CV_WINDOW_AUTOSIZE)
camera_index = 0
capture = cv2.VideoCapture(camera_index)
if not capture.isOpened():
    capature.open()
cv2.setMouseCallback("Sample point define",track_mouse)

def main():
   global frame
   print"Define",nor,"rows of",ppr,"points per row"
   print"press Esc to quit"
   print "locate start point on row 0"
   while(True):
     ret, frame = capture.read()
     cv2.circle(frame,(ix,iy),10,(0,0,255),1)
     cv2.line(frame,(0,240),(640,240),(0,0,225),1)
     cv2.line(frame,(320,0),(320,480),(0,0,225),1)
     overlay()
     cv2.imshow("Sample point define", frame)
     k = cv2.waitKey(1) & 0xFF
     if k == 27:
         finish()
     if (k == ord('s')) & (pointer == maxclick) :
        print "save points - click on Python console"
        fName = raw_input("type file name ")
        print"saving points to file",fName
        f = open(fName+'.txt','w')
        pickle.dump(points,f)
        f.close()
        finish()

def finish():
   print"all done"
   capture.release()
   cv2.destroyAllWindows()
   cv2.waitKey(1)
   os._exit(1)

def calculatePoints(row):
   global points
   startx = target[0][0]
   starty = target[0][1]
   endx = target[1][0]
   endy = target[1][1]
   xinc = float((endx - startx)/ (float(ppr) - 1.0) )
   yinc = float((endy - starty)/ (float(ppr) - 1.0) )
   for point in range(0,ppr):
      points[point +(ppr*row)] = (startx + int(xinc * point),starty +
int(yinc * point) )

def overlay():
   global pointer
   if pointer & 1 == 1 :
      cv2.circle(frame,target[0],10,(0,255,0),1)
   if showRow :
      # draw the sample row positions defined so far
      for i in range (0,(pointer/2)*ppr):
         cv2.circle(frame,points[i],10,(0,255,255),1)

if __name__ == '__main__':
    main()
```

videoSequencer.py

```
#!/usr/bin/env python
# Video sequencer with track bars
# and multi-pixel sampling
import time, pygame, pickle
import os, sys, math, copy
import cv2

pygame.init()  # initialise Pygame
pygame.mixer.quit()
pygame.mixer.init(frequency=22050, size=-16, channels=4, buffer=512)

pygame.event.set_allowed(None)
samples =["clap","closedhat","cowbell","crash","hitom","lotom"]
colours = ["Red","Orange","Yellow","Green","Blue","Magenta"]
maxCol = 6
seqSound = [ pygame.mixer.Sound("sounds/"+samples[sound]+".wav")
                  for sound in range(0,maxCol)]

cv2.namedWindow("Video Sequencer", cv2.CV_WINDOW_AUTOSIZE)
camera_index = 0
capture = cv2.VideoCapture(camera_index)
if not capture.isOpened():
    capture.open(camera_index)
hList = [ 0.0 for h in range (0,64)]
cList = [ 0.0 for c in range (0,64)]
nextNote = time.time()
cromaThresh = 0.23 # threshold for a colour
def track_mouse(event,x,y,flags,param):
    pass
cv2.setMouseCallback("Video Sequencer",track_mouse)

def main():
 global capture,samplePoint
 f = open('grid16x4.txt','r') # change to use another points file
 samplePoint = pickle.load(f)
 f.close()
 print"type Esc to quit - d to save colours to a CSV file and quit"
 for c in range(0,maxCol):
    print colours[c],"for",samples[c]
 cv2.createTrackbar('BPM',"Video Sequencer",120,200,nothing)
 switch = '0 : Stop \n1 : Run'
 cv2.createTrackbar(switch,"Video Sequencer",0,1,nothing)
 startTime = time.time() + 3
 while startTime > time.time(): # let web cam's AGC settle
    ret, frame = capture.read()
    cv2.imshow("Video Sequencer", frame)
    c = cv2.waitKey(1)

 while True:
    ret = True
    for i in range (0,5) : # read off 5 frames
      ret, frame = capture.read()
    points(frame)
    cv2.imshow("Video Sequencer", frame)
    getKey()
    if cv2.getTrackbarPos(switch,"Video Sequencer") == 1 :
       soundOut(frame)

def getKey():
        k = cv2.waitKey(1)& 0xFF
        if k == 27:
            terminate(False)
        if k == ord('d'):
            terminate(True)

def soundOut(frame):
    global nextNote
    bpm = 60.0 / float(cv2.getTrackbarPos('BPM',"Video Sequencer"))
    if bpm > 2.0:
      bpm = 2.0
    for i in range(0,16):
        temp = cv2.copyMakeBorder(frame,0,0,0,0,1)
        cv2.line(temp,samplePoint[i],samplePoint[i+16],(0,255,0),3)
        cv2.line(temp,samplePoint[i+16],samplePoint[i+32],(0,255,0),3)
        cv2.line(temp,samplePoint[i+32],samplePoint[i+48],(0,255,0),3)
        while nextNote > time.time() :
            pass
        nextNote = time.time() + bpm
        cv2.imshow("Video Sequencer", temp)
        getKey()
        for j in range(0,4):
          index = i + (j*16)
          if cList[index] > cromaThresh :
             seqSound[getColour(hList[index])].play()

def nothing(x): # dummy callback function for track bars
    pass

def output(): # CSV format
  print"Saving colours to file - colours.csv"
  with open('colours.csv','w') as f:
     f.write("Hole, Hue, Chroma, Colour \n")
     for c in range(0,16):
       f.write("\n")
       for r in range(0,4):
          i=(r*16) + c
          entry = str(i)+ ", "+"%.2f" % hList[i]
          f.write(entry+", ")
          entry = "%.2f" % cList[i]
          f.write(entry+", ")
          if cList[i] > cromaThresh :
             f.write(describeColour(hList[i])+"\n")
          else :
             f.write("neutral \n")
  f.close()
```

```
def getColour(h):
    colour = -1
    if h < 1 or h > 340:
        colour = 0
    if h>1 and h< 30:
        colour = 1
    elif h>30 and h< 90 :
        colour = 2
    elif h > 90 and h < 190 :
        colour = 3
    elif h> 190 and h< 300 :
        colour = 4
    elif h> 300 and h < 340 :
        colour = 5
    return colour

def describeColour(h):
    colourNumber = getColour(h)
    if colourNumber == -1:
       colour = str(h)+" is unknown"
    else:
       colour = colours[colourNumber]
    return colour

def points(frame): # outline sample area and get the colours
    for point in range(0,64):
      surround(samplePoint[point][0],samplePoint[point][1],
(0,0,0),frame,point)

def surround(x, y, col, frame, place):
    getCol(x,y, frame, place)
    frame[y, x-2] = col
    frame[y+2,x-2] = col
    frame[y-2,x-2] = col
    frame[y+2,x,] = col
    frame[y-2,x] = col
    frame[y,x+2] = col
    frame[y+2,x+2] = col
    frame[y-2,x+2] = col

def getCol(x,y, frame,place):
    global hList,cList
    bt = rt = gt = 0
    m = 255.0 * 9.0
    for ox in range(-1,2):
       for oy in range(-1,2):
          blue, green, red = frame[y+oy,x+ox]
          bt += blue
          gt += green
          rt += red
    r = float(rt) / m # normalise colours
    g = float(gt) / m
    b = float(bt) / m
    alp = 0.5*(2*r - g - b)
    bet = 0.866*(g - b)
    hList[place] = math.degrees(math.atan2(bet,alp))
    if hList[place] <0 :
        hList[place] = 360 + hList[place]
    cList[place] = math.sqrt(alp * alp + bet * bet)

def terminate(debug): # close down the program
    if debug :
       output() # colours to a csv file
    print ("Closing down please wait")
    pygame.quit() # close Pygame
    capture.release()
    cv2.destroyAllWindows()
    cv2.waitKey(1)
    os._exit(1)

if __name__ == '__main__':
    main()
```

Language

>PYTHON

DOWNLOAD:
Github.com/Grumpy-Mike/Mikes-Pi-Bakery/tree/master

PROJECT VIDEOS

Check out Mike's Bakery videos at: **vimeo.com/channels/mikespibakery**

PART 7

MAKE GAMES WITH PYTHON

SEAN M TRACEY

Sean is a technologist living in the South West of England. He spends most of his time making silly things with technology.
sean.mtracey.org

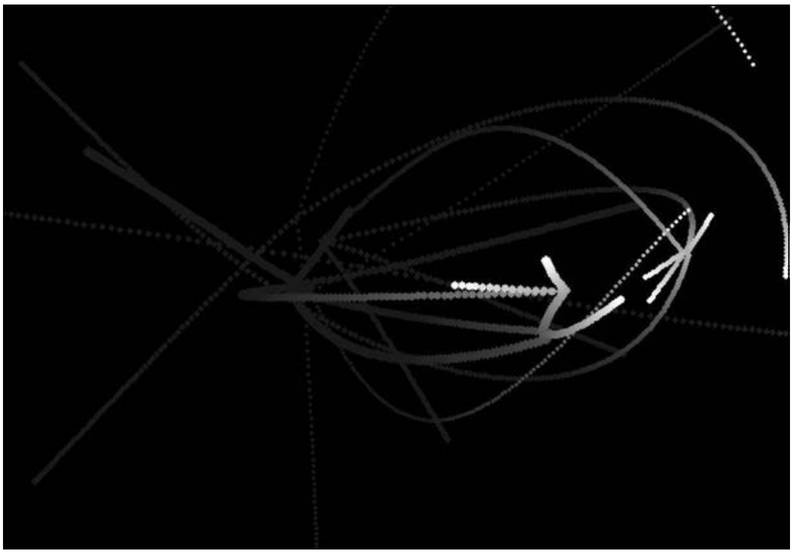
PHYSICS & COLLISIONS

In the seventh part of this series, we create circles that can bounce off one another if they collide

What happens when an unstoppable force meets an immovable object?

In our last tutorial, we built a whole solar system! Well, we simulated a good chunk of it at least. Using vectors and a dash of maths, we created a gravitational attraction between objects with mass to simulate their movement in space. Small objects would orbit larger ones, large objects would move very little when attracted to smaller objects and vice versa, and all was well in the simulated world. That said, one thing might have seemed a little odd: when two objects collide in the real world, they bounce off one another (or implode), but in our simulation they just slipped by one another as if they were ghosts. That just won't do! This time, we're going to write code that lets our objects collide and bounce off each other.

Below Simulate object collisions



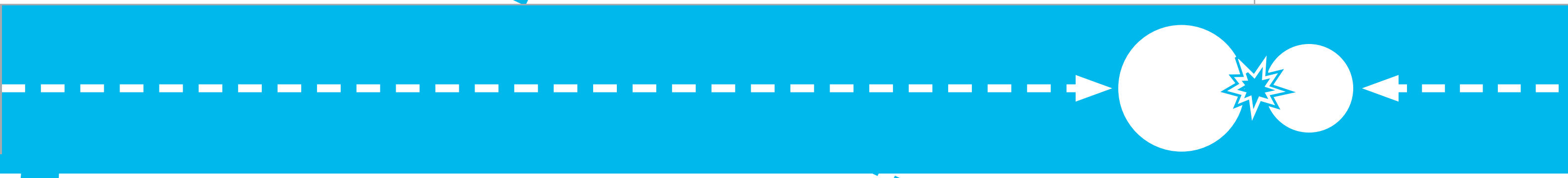
So, what are we making?

Unlike last time, we aren't going to be using planets and the solar system to prettify the effect – we're going to use basic circles for our program. Using circles makes it easier for us to use maths to calculate collisions, and we can change the properties of the circles to reflect the qualities they represent: for example, more mass or a bigger radius. That said, although we aren't using images of the solar system in this program, we can still think of the particles we'll be colliding in terms of a solar system.

The smallest of our collidable objects will be like 'meteors': they move really fast, but require less energy to do so. A medium-size object would behave much as a planet might; they move at a moderate speed and have more kinetic energy behind them. If they bump into a smaller object, they will adjust course, but not by much, whereas the smaller body will fly off!

We're going to use the code from the last tutorial as a springboard for this one. All of our objects will have a mass and will attract every other object gravitationally using the same **`calculateMovement()`** method as before.

Let's take a quick walk through our code now. Just like our previous bits of code, the top of **collisions.py** (between lines 1-24) imports the modules we'll need for our code and declares the variables that we'll be using throughout the tutorial. Obviously, these variables are very similar to the variables we used for our solar system simulator, but there's one little difference – instead of storing all of our planets in a list called **`celestialBodies`**, this time we're storing all of our objects in the **`collidables`** list.



Lines 26-84 will also seem familiar. **drawCollidables** is the same function as our **drawPlanets()** function, our **drawCurrentBody()** hasn't changed at all, and the same can be said for our **calculateMovement()** function which handles the gravity of all of our objects. Hello, old friend.

The **handleCollisions()** function on lines 86-135 is where we'll spend our time this tutorial. Here, we check for colliding objects and adjust their trajectories accordingly.

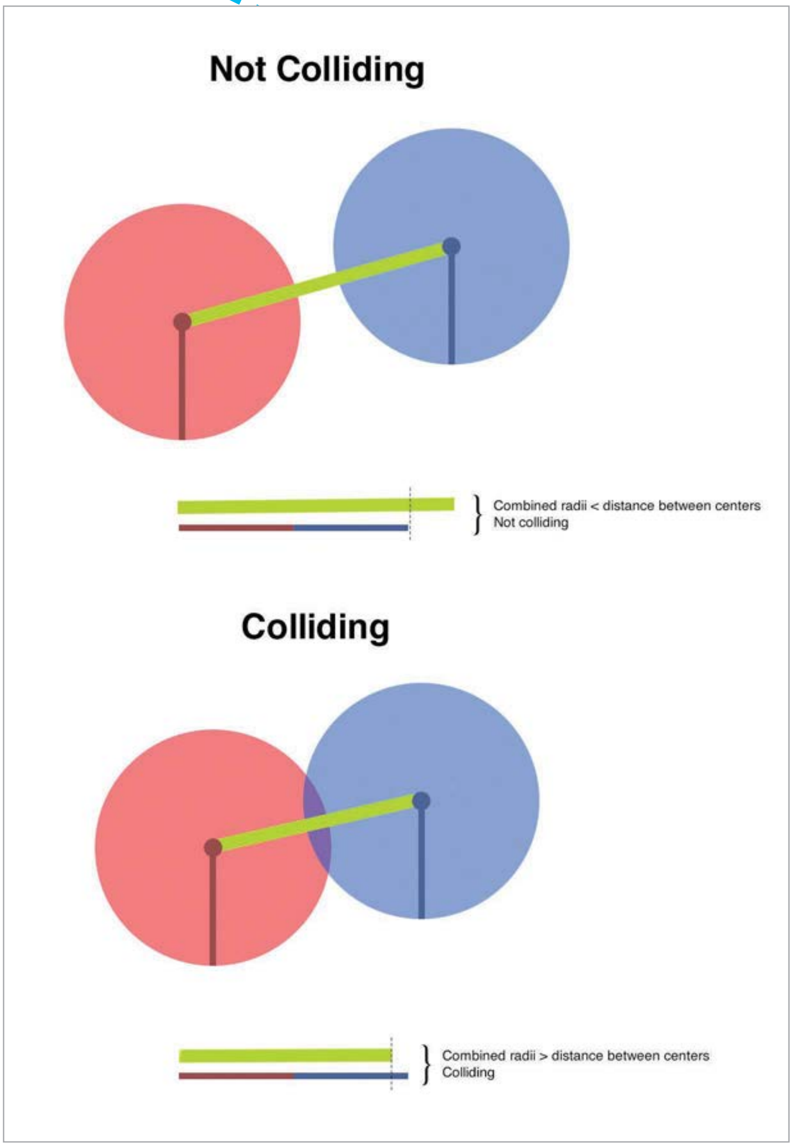
Lines 137-202 contain the logic for our keyboard and mouse interactions, as well as our main loop. Just as before, clicking in our window will create a new particle which will only affect the movement of other particles once the mouse has been released. If the mouse was moving when it was released, the particle will inherit the velocity of the mouse pointer.

What do we need to know to simulate a collision?

We need to know a couple of things before we can simulate a collision. First, we need to know which two objects, if any, are colliding. Once we know which two objects are colliding, we need to figure out how fast they're going, the angle of incidence (which we'll look at in a little while), and the mass of each of the objects.

So, how do we know which two objects are colliding? That's pretty simple... when you use circles. Circles are regular shapes: each point along the circumference is the same distance from the centre as every other point; this measurement from the edge to the centre is the radius. By measuring the distance between the centres of two objects, we can check whether or not the outline of the objects are intersecting (crossing into) the space of each other. If the distance between the centres of two circles is less than the radius of each circle added to the other, they must be colliding! Neat, huh?

On lines 88-96, we create two **for** loops that let us work through every possible collidable object in our simulation. Inside these loops, we measure the distance between the centres of every object in our **collidables** list. We do this on line 102 with our **distance** variable, using Python's maths module. If the distance between the two centres of our objects is more than the combined length of the radius of each circle, our objects are not colliding, and we then continue on measuring the distance to other circles. However, if the distance is less than the sum of the radii (plural for radius, in case you were wondering), then our objects are colliding and we can start figuring out what to do with them.


Not Colliding
Combined radii < distance between centers
Not colliding
Colliding
Combined radii > distance between centers
Colliding

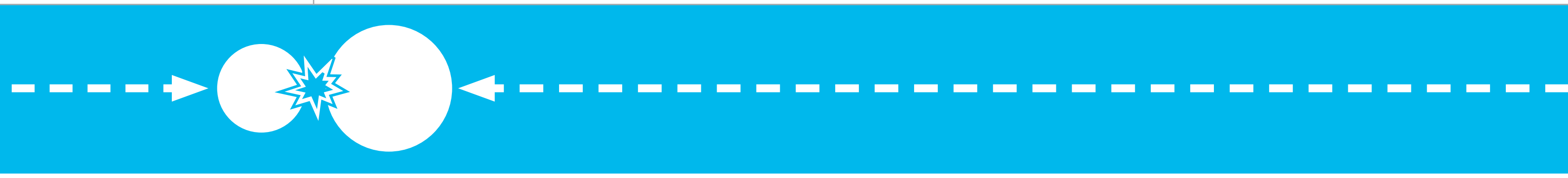
Above A diagram of all of the aspects needed to bounce one circle off another

⚠ **WARNING!**

PYTHAGORAS THEOREM AHEAD

The angle of incidence

How do we create a convincing bounce effect? In the past, whenever we've wanted to restrict or adjust movement, we've been doing so to squares or rectangles – shapes with flat sides. Going too much to the right? Okay, we'll send it back to the left. Going too far down? Okay, we'll send it back up. Now, we're dealing with circles. Instead of having eight directions we can go in, we now have 360. If our circles hit each other square on (ha!), then all



> Of the two objects colliding, the faster object will transfer energy into the slower object

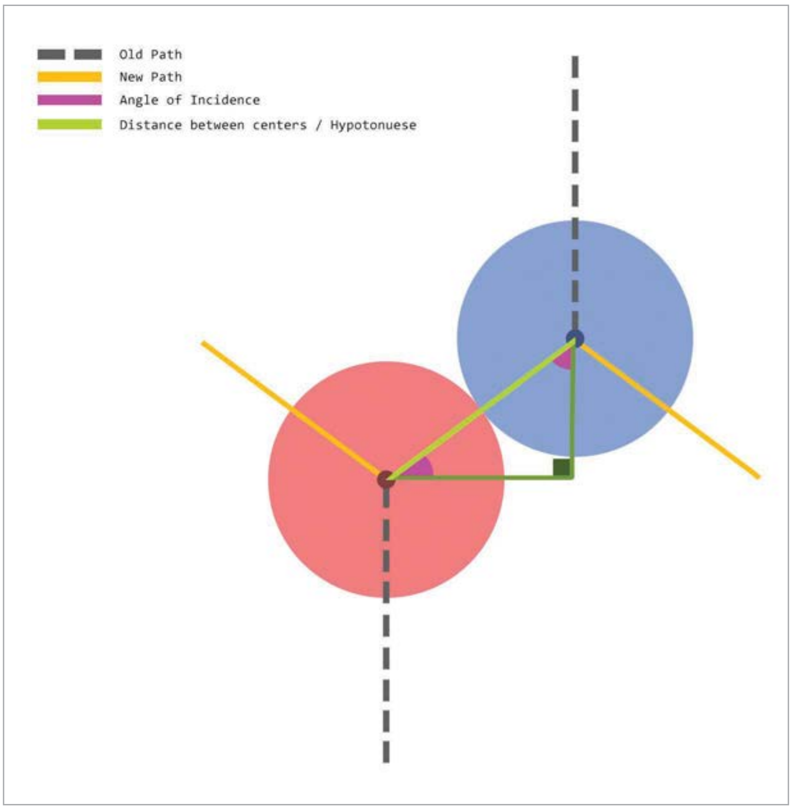
THANKS

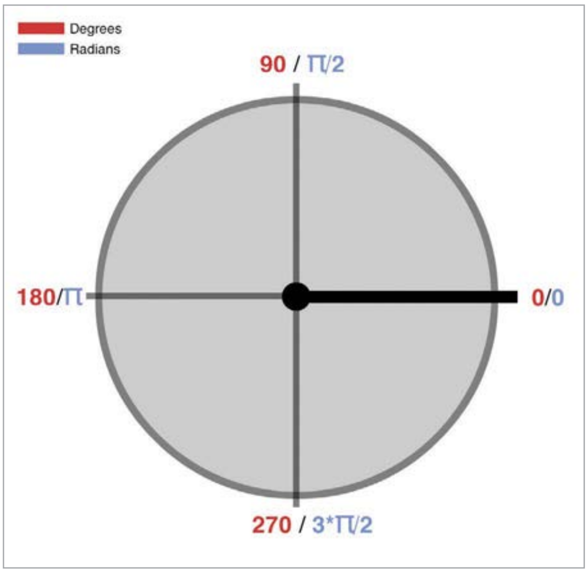
A hat tip for this tutorial goes out to Steve and Jeff Fulton. These guys put a huge amount of effort into dissecting old Flash-based physics code into its core parts and putting it back together for their book *HTML5 Canvas*, which made this tutorial possible.

Below A right-angled triangle is used to figure out the angle of incidence

we have to do is send the circles back along the path they came, but these are circles *with gravity*; hitting another circle square on along the X or Y axis is not going to happen very often. If a collision happens a little to the left or right of the centre of the X or Y axis, we need to send our objects on two new paths, but how do we know which direction to send each object? Well, that's what the angle of incidence is for. What is the angle of incidence? It's the angle at which an object is travelling as it collides with another object. If we know the angle at which two things collide, we can figure out along which angle we can send them on their merry way – that is, the angle of reflection, which is the reverse of the angle of incidence.

It's not as complicated as it sounds. Imagine a ball hitting a vertical wall at an angle 45, so its vector is (1, 1), travelling to the right and down in equal measure. After the ball hits the wall, the rate at which the ball falls to the ground is unchanged, but the direction it's travelling is reversed along its X axis; our ball is still travelling at 45 degrees, but now it's travelling away from the wall, at -45 degrees or with a vector of (-1, 1). We've coded this in an earlier tutorial, when we first started bouncing objects around a window (gosh, that was a while ago now!), but we weren't using the terms we are here. Now we know *why* we made objects move the way we did.





Above A diagram displaying angles on a circle and their equivalent values in radians

On line 107, we calculate the angle of incidence between the centre of the two circles colliding with **`math.atan2`**, which basically works out the hypotenuse of an imaginary right-angled triangle drawn using the two centre points of the circles. If you were to print out the value of the direction variable, you might expect it to read somewhere between 0 and 360 because an angle is measured in degrees, right? Nope! Instead you'll get a value between 1 and 2π (pi * 2): our angle has been measured in radians. This may seem counter-intuitive, but to a computer (and mathematicians) it makes perfect sense. If you want to see the degree value, you can simply do 'radians * (180/pi)', but we're going to stick with radians because it keeps our code tidy.

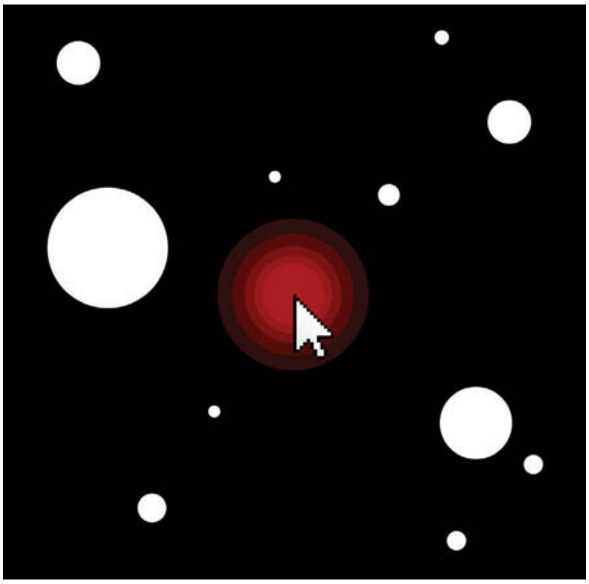
Bounce!

Now we've got the angle of incidence, we can calculate which way to send our colliding objects, but first we need to obtain a couple of other values to make everything work. Next, we need to work out the speed at which our circles are moving. 'Hang on, we have vectors, why do we need a separate speed value?' you ask. It's very astute of you to realise that we use vectors to affect the speed and direction of our objects, but that's part of the problem: our vector is a measure of speed *and* direction. As it is, we can't use the vectors to find out how many pixels our objects travel per frame; we need to separate the speed from the direction so we can perform some maths specific to each value. Fortunately, we can use maths to figure out the speed of our objects – which we do on lines 110-111, one variable for each object in the collision – and the direction each object is moving in radians, on lines 114-115.

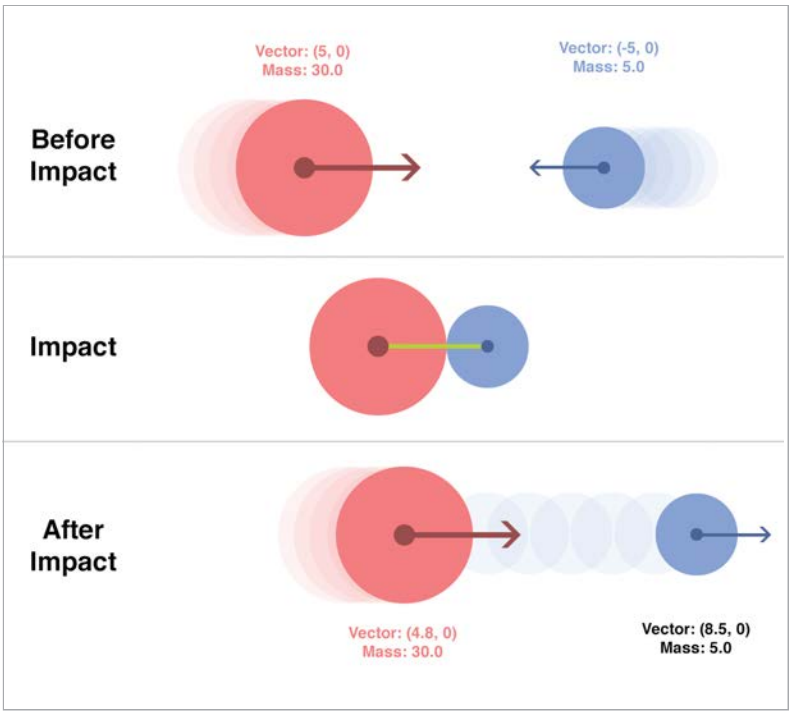
Now we have the speed and direction of each circle, we can adjust them separately to create the bouncing effect. First, we want to reverse the direction that the objects are travelling in; on lines 118-122, we create a couple of variables to calculate new velocities. Here we've recombined the speed and direction variables of each object to create new speeds for the X and Y values of our circles. When used together, guess what: we've got a new vector! But these ones will point our objects in the opposite direction of the angle of incidence – the angle of reflection. We've got the basics of our bounce.

Motion

Energy cannot be destroyed, only converted into other forms. Motion is a form of energy and when two objects collide, an energy transfer happens between them. Of the two objects colliding, the faster object will transfer energy into the slower object, speeding the slower object up and slowing itself down. The two objects will move off in different directions and at different speeds than they were travelling before the collision, but the net energy of motion - the total amount of energy moving the objects - will remain exactly the same; it's just in different quantities in different objects now.



Above Use the mouse to create a new moving object



Above An illustration of the effect of mass on vectors in a collision

On lines 125 and 126, we take into effect the mass of each of the objects colliding, as well as the speed. The result is that bigger objects will take more energy to change direction than smaller ones. With this in place, we won't have large objects being sent off at high velocities by much smaller, faster-moving objects – just like in the real world. Remember, this is all about *simulating* physics – if we wanted to accurately calculate interactions between multiple objects as if they existed in the real world, we'd need a computer much more powerful than our Raspberry Pi.

Now we have the new vectors for our colliding objects, all we have to do is apply them to our objects. We're only going to apply the X values we've calculated to each object. If we applied both the adjusted X and Y values to each object, they would bounce and follow the path they came along. That would be like throwing a ball and having it bounce straight back into your hand – it's not natural! By only applying the X value to each of our colliding objects, we create a convincing, bouncing, deflecting effect, and we do that on lines 129 and 130.

And that's it. Rinse and repeat for every possible collidable object in our simulator.

QUICK TIP

We haven't used the planet graphics or much of the user interaction code that we wrote for the solar system, but with a little bit of work, you should be able to drop the `handleCollisions()` function into last issue's code and make your planets bounce. Consider it a challenge, if you like.

Collisions.py

```python
import pygame, sys, random, math
import pygame.locals as GAME_GLOBALS
import pygame.event as GAME_EVENTS
import pygame.time as GAME_TIME

windowWidth = 1024
windowHeight = 768

pygame.init()
surface = pygame.display.set_mode((windowWidth, windowHeight))

pygame.display.set_caption('Collisions')

previousMousePosition = [0,0]
mousePosition = None
mouseDown = False

collidables = []
currentObject = None
expanding = True

drawAttractions = False

gravity = 1.0

def drawCollidables():

    for anObject in collidables:
        anObject["position"][0] += anObject["velocity"][0]
        anObject["position"][1] += anObject["velocity"][1]

        pygame.draw.circle(surface, (255,255,255), (int(anObject["position"][0]), int(anObject["position"][1])), int(anObject["radius"]), 0)

def drawCurrentObject():

    global expanding, currentObject

    currentObject["position"][0] = mousePosition[0]
    currentObject["position"][1] = mousePosition[1]

    if expanding is True and currentObject["radius"] < 30:
        currentObject["radius"] += 0.2

        if currentObject["radius"] >= 30:
            expanding = False
            currentObject["radius"] = 9.9

    elif expanding is False and currentObject["radius"] > 1:
        currentObject["radius"] -= 0.2

        if currentObject["radius"] <= 1:
            expanding = True
            currentObject["radius"] = 1.1

    currentObject["mass"] = currentObject["radius"]

    pygame.draw.circle(surface,(255,0,0), (int(currentObject["position"][0]), int(currentObject["position"][1])), int(currentObject["radius"]), 0)


def calculateMovement():

    for anObject in collidables:

        for theOtherObject in collidables:

            if anObject is not theOtherObject:

                direction = (theOtherObject["position"][0] - anObject["position"][0], theOtherObject["position"][1] -anObject["position"][1])
                magnitude = math.hypot(theOtherObject["position"][0] - anObject["position"][0], theOtherObject["position"][1] - anObject["position"][1])
                nDirection = (direction[0] / magnitude, direction[1] / magnitude)

                if magnitude < 5:
                    magnitude = 5
                elif magnitude > 15:
                    magnitude = 15

                strength = ((gravity * anObject["mass"] * theOtherObject["mass"]) / (magnitude * magnitude)) / theOtherObject["mass"]

                appliedForce = (nDirection[0] * strength, nDirection[1] * strength)

                theOtherObject["velocity"][0] -= appliedForce[0]
                theOtherObject["velocity"][1] -= appliedForce[1]

                if drawAttractions is True:
                    pygame.draw.line(surface, (255,255,255), (anObject["position"][0],anObject["position"[1]), (theOtherObject["position"][0],theOtherObject["position"][1]), 1)

def handleCollisions():

    h = 0

    while h < len(collidables):

        i = 0

        anObject = collidables[h]

        while i < len(collidables):

            otherObject = collidables[i]

            if anObject != otherObject:

                distance = math.hypot(otherObject["position"][0] - anObject["position"][0], otherObject["position"][1] - anObject["position"][1])

                if distance < otherObject["radius"] + anObject["radius"]:

```

Language

>PYTHON

```
                          # First we get the angle of the
                          #collision between two objects
                          collisionAngle = math.atan2(anObject["position"][1] - otherObject["position"][1], anObject["position"][0] - otherObject["position"][0])

                          # Then we need to calculate the
                          #speed of each object
                          anObjectSpeed = math.sqrt(anObject["velocity"][0] * anObject["velocity"][0] + anObject["velocity"][1] * anObject["velocity"][1])
                          theOtherObjectSpeed = math.sqrt(otherObject["velocity"][0] * otherObject["velocity"][0] + otherObject["velocity"][1] * otherObject["velocity"][1])

                          # Now, we work out the direction
                          # of the objects in radians
                          anObjectDirection = math.atan2(anObject["velocity"][1], anObject["velocity"][0])
                          theOtherObjectDirection = math.atan2(otherObject["velocity"][1], otherObject["velocity"][0])

                          # Now calculate new X/Y values
                          # of each object for collision
                          anObjectsNewVelocityX = anObjectSpeed * math.cos(anObjectDirection - collisionAngle)
                          anObjectsNewVelocityY = anObjectSpeed * math.sin(anObjectDirection - collisionAngle)

                          otherObjectsNewVelocityX = theOtherObjectSpeed * math.cos(theOtherObjectDirection - collisionAngle)
                          otherObjectsNewVelocityY = theOtherObjectSpeed * math.sin(theOtherObjectDirection - collisionAngle)

                          # We adjust the velocity based
                          # on the mass of the objects
                          anObjectsFinalVelocityX = ((anObject["mass"] - otherObject["mass"]) * anObjectsNewVelocityX + (otherObject["mass"] + otherObject["mass"]) * otherObjectsNewVelocityX)/(anObject["mass"] + otherObject["mass"])
                          otherObjectsFinalVelocityX = ((anObject["mass"] + anObject["mass"]) * anObjectsNewVelocityX + (otherObject["mass"] - anObject["mass"]) * otherObjectsNewVelocityX)/(anObject["mass"] + otherObject["mass"])

                          # Now we set those values
                          anObject["velocity"][0] = anObjectsFinalVelocityX
                          otherObject["velocity"][0] = otherObjectsFinalVelocityX


                i += 1

           h += 1

def handleMouseDown():
     global currentObject
     currentObject = {
           "radius" : 3,
           "mass" : 3,
           "velocity" : [0,0],
           "position" : [0,0]
     }

def quitGame():
  pygame.quit()
  sys.exit()

# 'main' loop
while True:

  surface.fill((0,0,0))
  mousePosition = pygame.mouse.get_pos()

  # Handle user and system events
  for event in GAME_EVENTS.get():

    if event.type == pygame.KEYDOWN:

      if event.key == pygame.K_ESCAPE:
        quitGame()

    if event.type == pygame.KEYUP:

     if event.key == pygame.K_r:
           collidables = []
     if event.key == pygame.K_a:
           if drawAttractions is True:
                drawAttractions = False
           elif drawAttractions is False:
                drawAttractions = True

    if event.type == pygame.MOUSEBUTTONDOWN:
      mouseDown = True
      handleMouseDown()

    if event.type == pygame.MOUSEBUTTONUP:
     mouseDown = False

    if event.type == GAME_GLOBALS.QUIT:
      quitGame()

  calculateMovement()
  handleCollisions()
  drawCollidables()

  if currentObject is not None:
     drawCurrentObject()

     # If our user has released the mouse, add the new
     # anObject to the collidables list and let gravity
     # do its thing
     if mouseDown is False:
           currentObject["velocity"][0] = (mousePosition[0] - previousMousePosition[0]) / 4
           currentObject["velocity"][1] = (mousePosition[1] - previousMousePosition[1]) / 4
           collidables.append(currentObject)
           currentObject = None

  # Store the previous mouse coordinates to create a
  # vector when we release a new anObject
  previousMousePosition = mousePosition

  pygame.display.update()
```

WILLIAM BELL

Will is a previous contributing editor on *The MagPi* before it became the official Raspberry Pi magazine. Will enjoys solving complicated problems with elegant, bespoke solutions. For him, the bigger the challenge, the bigger the attraction.

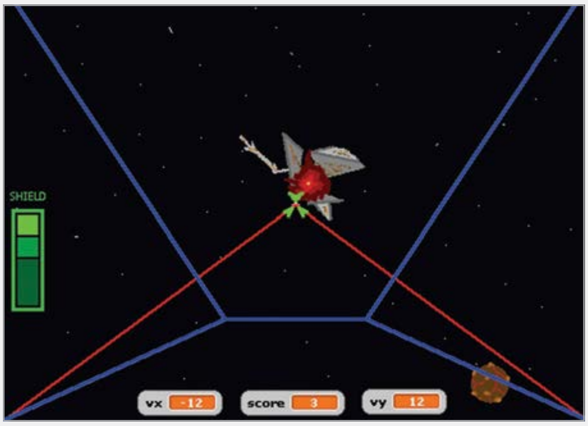
whbell.net

BUILD A SPACE SHOOTER IN SCRATCH

William H Bell walks us through how to create an impressive 3D space shooter using nothing more than Scratch and some clever coding techniques...

Scratch is a great programming language for testing out a range of concepts. Scratch programs typically involve controlling one or more sprites on the screen. Computer games where the characters are controlled from a distant view are third-person games. Games can be more exciting when the human player looks through the eyes of the central character in the game, however. This is normally referred to as a first-person game.

In this article, some of the principles of constructing a first-person game are introduced. The player is the pilot of a spaceship that is drifting through a debris field. The main engine has gone offline, causing the spaceship to drift through the debris at a constant speed. However, the spaceship still has working thrusters on the top, bottom, left and right of the craft. The main laser system is also operational. The heroic pilot has to shoot through or dodge the debris. A point is awarded each time a piece of debris is destroyed with the ship's lasers. If the debris crashes into the spacecraft, then the shield of the spacecraft will be damaged. After the shield has been completely broken, the spacecraft will explode.



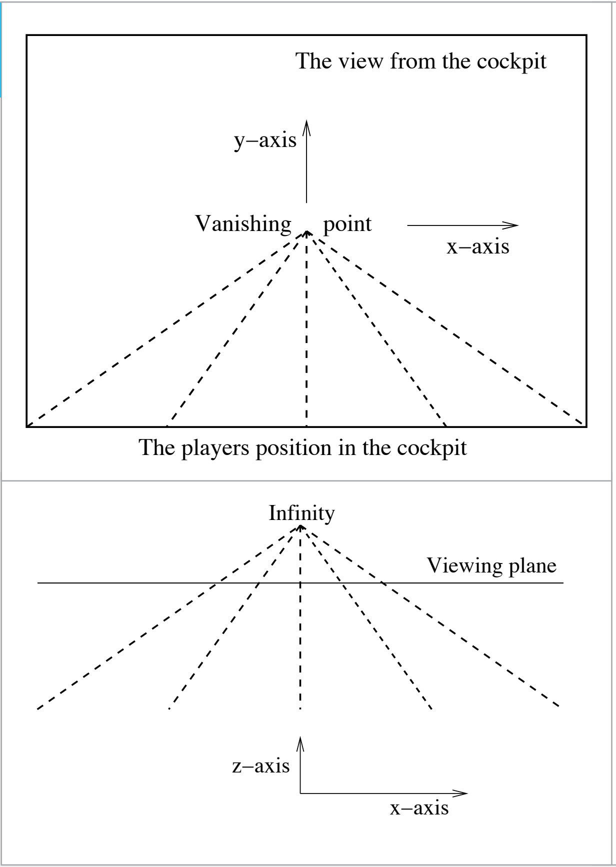
Right Hit the space bar to fire the ship's lasers to destroy debris; if it crashes into you, your shield (green bar) will deplete

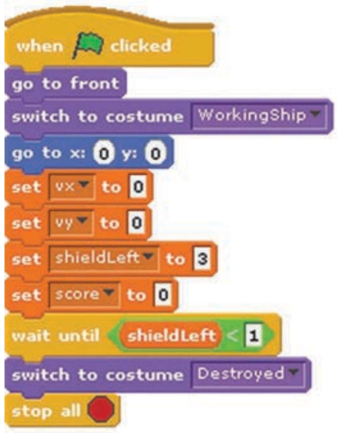
Perspective

In real life, objects that are far away appear to be smaller. One example of this is a set of railway tracks. Looking down railway tracks and into the distance, the tracks appear to become closer together. This can be applied to a computer game, where objects need to be shown as being in the distance. When an object becomes closer to a player, the object should become larger on the screen.

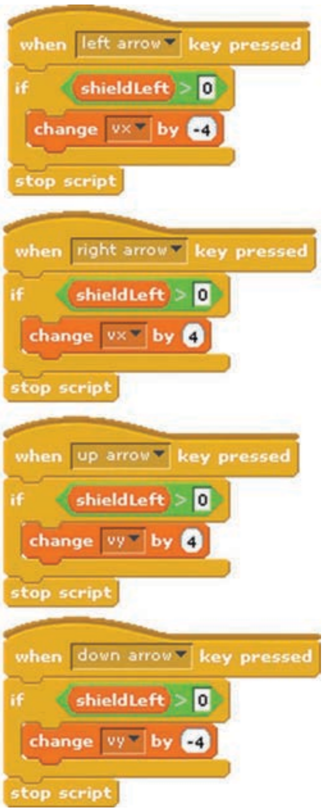
In this game, a one-point perspective is used. This means that distant objects appear to come from the centre of the screen. Rather than draw a lot of very small images at the vanishing point, it is more sensible to assume a viewing plane. The viewing plane corresponds to the distance at which objects become visible. The two diagrams at the top-left of page 65 show the vanishing point as it appears on the screen, and the position of the viewing plane. In the illustration of the viewing plane, the z-axis points from the centre of the screen straight towards the player and is perpendicular to the x-y plane.

If the spacecraft has no velocity along the x-y plane and an object appears at the viewing plane with a


The view from the cockpit
y–axis
Vanishing point
x–axis
The players position in the cockpit
Infinity
Viewing plane
z–axis
x–axis



```
when [flag] clicked
go to front
switch to costume WorkingShip
go to x: 0 y: 0
set vx to 0
set vy to 0
set shieldLeft to 3
set score to 0
wait until shieldLeft < 1
switch to costume Destroyed
stop all
```



```
when left arrow key pressed
if shieldLeft > 0
    change vx by -4
stop script

when right arrow key pressed
if shieldLeft > 0
    change vx by 4
stop script

when up arrow key pressed
if shieldLeft > 0
    change vy by 4
stop script

when down arrow key pressed
if shieldLeft > 0
    change vy by -4
stop script
```

These code blocks set the four main variables and control the spacecraft's movement (via cursor keys)

> In the game, the spacecraft is not able to turn. Since the stars in the distance are very far away, they would not appear to move relative to the spacecraft

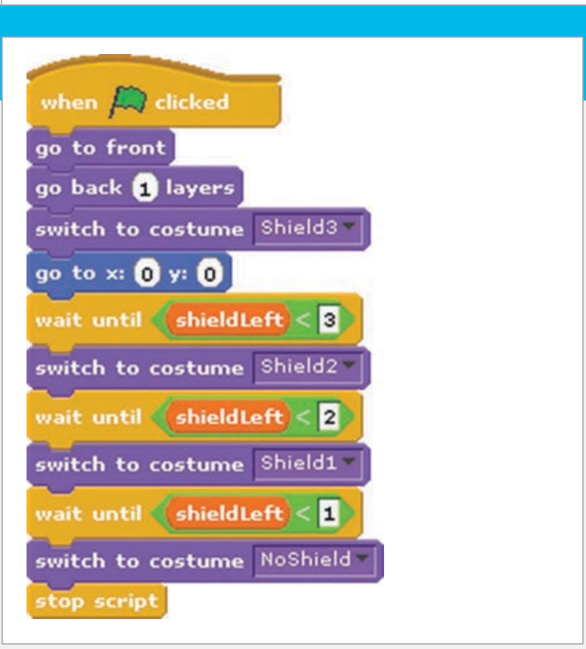
position that is not in the centre of the screen, then the object appears to have a velocity that is proportional to its distance from the centre of the screen. This is not a real velocity, but is the effect of the perspective used to display the z-axis. This effect can be observed when driving along a straight road: a vehicle that is on the other side of the road, but far in the distance, appears to move to the other side of the road as it approaches.

Spaceship and star field

In the game, the spacecraft is not able to turn. Since the stars in the distance are very far away, they would not appear to move relative to the spacecraft. Therefore, a static star field was drawn on the stage background.

The spaceship cockpit and heads-up display should stay in the foreground. This was achieved by creating a sprite that is as big as the screen. When the game starts, the **SpaceShip** sprite is set to be above other sprites. Therefore, the cockpit edges are displayed as being in the foreground.

The horizontal and vertical velocity components of the spacecraft are stored in the **vx** and **vy** variables. These variables were created as global variables, since the velocity components affect the motion of other sprites on the screen. The **shieldLeft** variable contains the number of shield points left, and the **score** contains the player's score. The **shieldLeft** variable was created as a global variable, since the other sprites that may hit the cockpit need to be able to change its value; **score** was also created as a global variable, since other sprites need to be able to increment it. The game continues until there are no shield points left. When the game starts, all four global variables are reset to zero and the spaceship is shown to be working as normal. If there are no shield points left, then the ship is shown to be destroyed by changing the costume of the **SpaceShip** sprite. The thrusters on the right, left, bottom and top of the spacecraft are controlled by the cursor keys. Since the spacecraft is in space, there is no friction to slow down its movement. Therefore, firing the thrusters in one direction will build up the velocity in that direction. To make it easier for the player to see the current status of the game, the values of the **vx**, **vy** and **score** variables were selected to be displayed at the bottom of the screen.



```
when [flag] clicked
go to front
go back (1) layers
switch to costume [Shield3]
go to x: (0) y: (0)
wait until <(shieldLeft) < (3)>
switch to costume [Shield2]
wait until <(shieldLeft) < (2)>
switch to costume [Shield1]
wait until <(shieldLeft) < (1)>
switch to costume [NoShield]
stop script
```

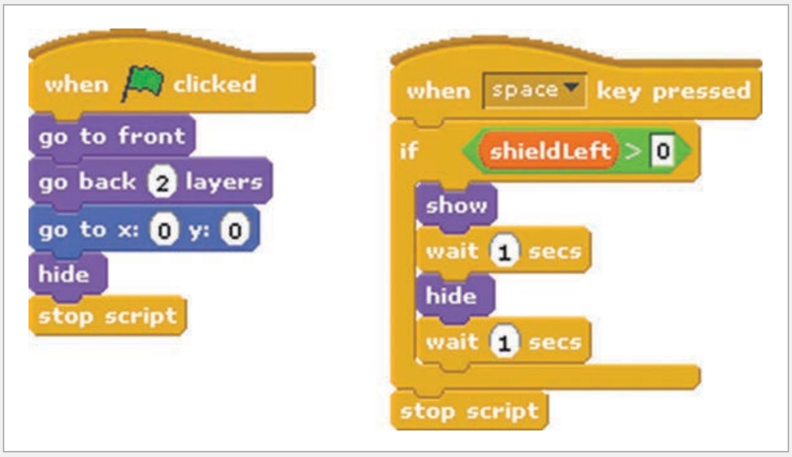
Right Each costume is used for one of four shield states

Shield heads-up display

The number of shield points remaining is shown on the left-hand side of the screen. This image is a sprite called **Shield**, which has several costumes that correspond to the different shield states. The different costumes were a copy of the first costume, each with one more green box removed.

When the green flag is pressed, the **Shield** sprite is set to be just below the main cockpit but above the other sprites. This means that the shield display stays in the foreground. The script for the **Shield** sprite waits until the number of shield points decreases and then switches to the appropriate costume.

Below These blocks implement the firing of the spacecraft's laser



```
when [flag] clicked
go to front
go back (2) layers
go to x: (0) y: (0)
hide
stop script
```

```
when [space] key pressed
if <(shieldLeft) > (0)>
    show
    wait (1) secs
    hide
    wait (1) secs
stop script
```

Lasers

The lasers were drawn as another sprite. The size of the **Laser** sprite was carefully matched to the **SpaceShip** sprite by copying the **SpaceShip** costume to check where the lasers would appear on the screen.

When the green flag is pressed, the **Laser** sprite is set to appear just below the **SpaceShip** sprite. Therefore, it is in the foreground but not as close as the cockpit. The lasers are fired by pressing the space bar. To make the game a little bit harder, the lasers fire for a second and then recharge for a second. This means that the player should not hold down the space bar, but only fire the lasers when needed. Similar to the **SpaceShip** sprite script, the **Laser** sprite only recognises the space bar when the number of shield points is greater than zero.

Space debris

Two types of space debris were created: **LavaBall** and **Scrap**. The script for the **LavaBall** sprite was copied and modified slightly for the **Scrap** sprite to prevent both sprites appearing at exactly the same time. The two sprites were also given two costumes, to show them as being **normal** or **exploded**.

> The script for the LavaBall sprite was copied and modified slightly for the Scrap sprite to prevent both sprites appearing at exactly the same time

When the green flag is pressed, the **LavaBall** is placed below the cockpit, shield display, and lasers. Then it is hidden from view. The main loop continues while the game is being played. When the **SpaceShip** sprite switches to the **destroyed** costume, it finishes the game by stopping all scripts. This includes the main loops of the space debris sprites.

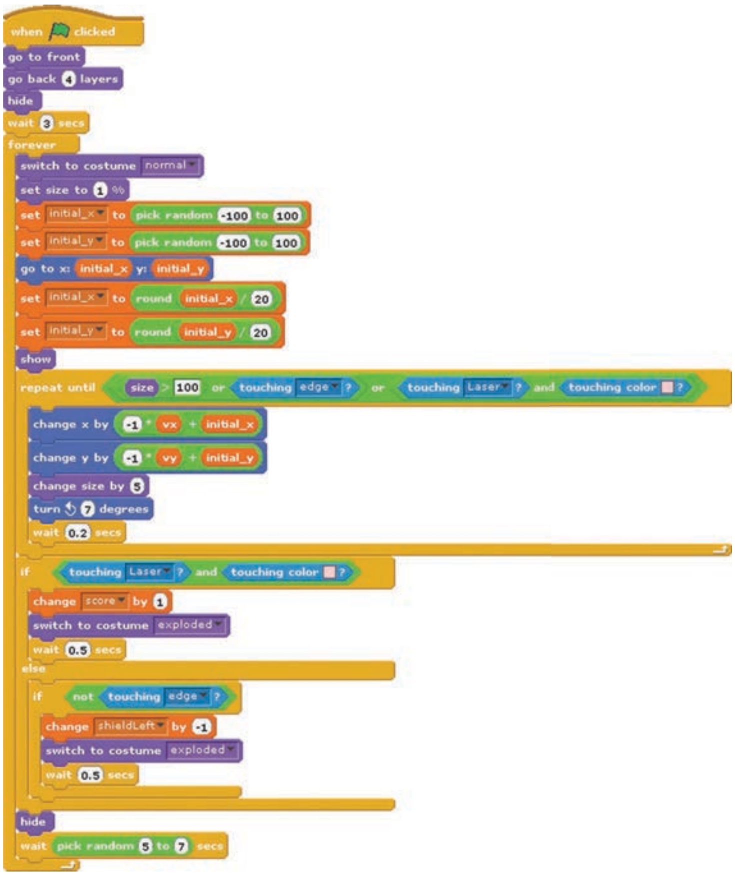
To show that it is in the distance, the **LavaBall** appears at the viewing plane at 1% of its normal size. To make the game more interesting, its starting position is chosen at random in the x-y plane. Due to the one-point perspective used, objects that are closer to the edge of the screen will quickly disappear from this location. Therefore, objects were chosen to appear within a 100 by 100 box around the centre of the screen. The initial position of the sprite, along the x- and y-axes, is stored in the **initial_x** and **initial_y** variables. Since these variables are only needed for this sprite, they were created as local variables for this sprite only. The initial position components are rescaled to produce an apparent velocity offset associated with the perspective. They are rounded to integers, since the sprite moves in numbers of pixels. The sprite is then shown on the screen. Next, the script enters another loop that continues until the sprite is full-size, has touched the edge of the screen, or has been hit by the laser beams. The point where the two laser beams meet was given a pink colour, so that this colour could be used to test if the laser beams had hit the **LavaBall**. The relative velocity of the debris along the z-axis can be raised by increasing the **change size by 5** (5%) command or by reducing the size of the **wait** within the motion loop.

In this game, the space debris is spinning but is otherwise stationary with respect to the rest of the universe. The spacecraft is drifting through the debris field at a constant speed, and starts the game at rest in the x-y plane. When the spacecraft thrusters are fired, the spacecraft moves along the x-y plane with respect to the universe. However, the game is played from the pilot's point of view, rather than from the point of view of the universe or the space debris. Therefore, when the player's spacecraft is moving to the left, the **LavaBall** is shown as moving to the right. If the spacecraft moves downwards, then the **LavaBall** moves upwards. This can be demonstrated by looking at a cup on a desk: if the person looking at the cup moves to the left, then the cup moves to the right with respect to the person's line of sight. The motion of the sprite is therefore the sum of the relative velocity and the apparent velocity, due to the object being created at a point on the viewing plane that is not in the centre of the screen.

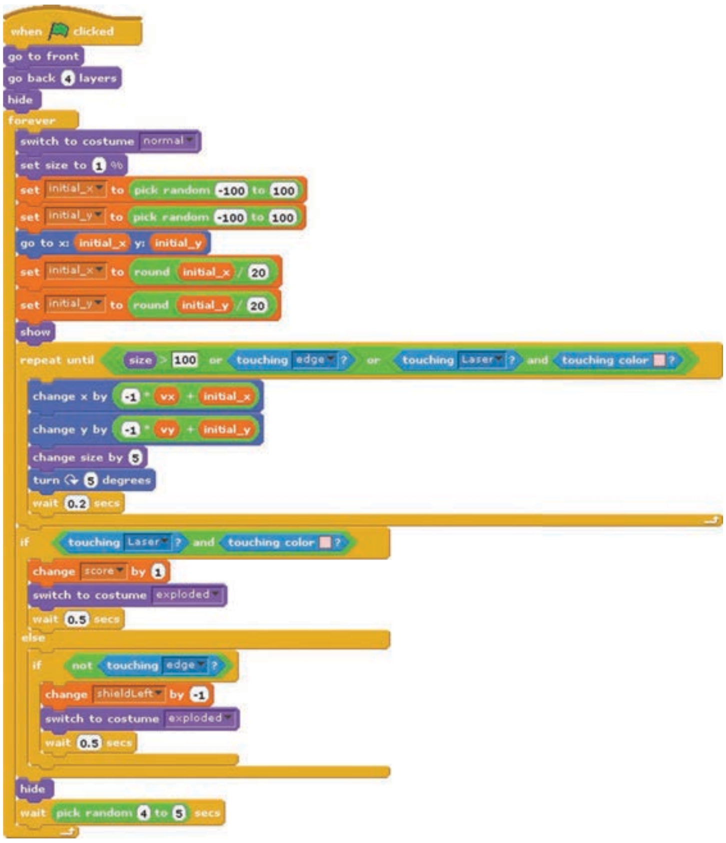
If the **LavaBall** has been hit by the laser beams, then the score is incremented and the costume is switched to the **exploded** version. The program waits for half a second for the player to view the **exploded** sprite. If the **LavaBall** has not been hit by the lasers and it has not touched the edge of the screen, then it has hit the spacecraft. If the **LavaBall** has hit the spacecraft, then the number of shield points is reduced by one and the **LavaBall** costume is switched to the **exploded** version. If the **LavaBall** has missed the spacecraft, then it disappears behind the spacecraft harmlessly. After these logic conditions, the **LavaBall** sprite is hidden and reappears somewhere else on the screen.

Possible extensions

Other features could be added to the game. The spacecraft could collect shield tokens or be able to use a wider laser beam to destroy more than one object at once. Alternatively, the principles demonstrated within this program could be used to create a first-person car racing game.



```
when 🏴 clicked
go to front
go back 4 layers
hide
wait 3 secs
forever
    switch to costume normal
    set size to 1 %
    set initial_x to pick random -100 to 100
    set initial_y to pick random -100 to 100
    go to x: initial_x y: initial_y
    set initial_x to round initial_x / 20
    set initial_y to round initial_y / 20
    show
    repeat until size > 100 or touching edge ? or touching Laser ? and touching color ■ ?
        change x by -1 * vx + initial_x
        change y by -1 * vy + initial_y
        change size by 5
        turn ↺ 7 degrees
        wait 0.2 secs
    if touching Laser ? and touching color ■ ?
        change score by 1
        switch to costume exploded
        wait 0.5 secs
    else
        if not touching edge ?
            change shieldLeft by -1
            switch to costume exploded
            wait 0.5 secs
    hide
    wait pick random 5 to 7 secs
```



```
when 🏴 clicked
go to front
go back 4 layers
hide
forever
    switch to costume normal
    set size to 1 %
    set initial_x to pick random -100 to 100
    set initial_y to pick random -100 to 100
    go to x: initial_x y: initial_y
    set initial_x to round initial_x / 20
    set initial_y to round initial_y / 20
    show
    repeat until size > 100 or touching edge ? or touching Laser ? and touching color ■ ?
        change x by -1 * vx + initial_x
        change y by -1 * vy + initial_y
        change size by 5
        turn ↻ 5 degrees
        wait 0.2 secs
    if touching Laser ? and touching color ■ ?
        change score by 1
        switch to costume exploded
        wait 0.5 secs
    else
        if not touching edge ?
            change shieldLeft by -1
            switch to costume exploded
            wait 0.5 secs
    hide
    wait pick random 4 to 5 secs
```

Sonic Pi PART 2

SAM AARON

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

CODED BEATS

Sonic Pi creator **Sam Aaron** continues his essential new series with one of the most influential drum breaks in music history...

You'll Need

- Raspberry Pi running Raspbian
- Sonic Pi v2.6+
- Speakers or headphones with a 3.5mm jack
- Update Sonic Pi: `sudo apt-get update && sudo apt-get install sonic-pi`

One of the most exciting and revolutionary technical developments in modern music was the invention of computer-based samplers in the late 1970s. These electronic boxes of tricks allowed you to record any sound into them and then manipulate and play back those sounds in many interesting ways. For example, you could take an old record, find a drum solo (or break), record it into your sampler, and then play it back on repeat at half-speed to provide the foundation for your latest beats. This is how early hip-hop music was born, and today it's almost impossible to find electronic music that doesn't incorporate samples of some kind. Using samples is a really great way of easily introducing new and interesting elements into your live-coded performances.

So where can you get a sampler? Well, you already have one: it's your Raspberry Pi! The built-in live-coding app Sonic Pi has an extremely powerful sampler built into its core. Let's play with it!

The Amen Break

One classic and immediately recognisable drum break sample is called the Amen Break. It was first performed in 1969 in the song *Amen Brother* by The Winstons, as part of a drum break. However, it was when it was discovered and sampled by early hip-hop musicians in the 1980s that it started being heavily used in a wide variety of other musical styles such as drum and bass, breakbeat, hardcore techno, and breakcore.

I'm sure you're excited to hear that it's also built right into Sonic Pi. Clear up a buffer and throw in the following code:

```
sample :loop_amen
```

Hit **Run** and boom! You're listening to one of the most influential drum breaks in the history of dance music. However, this sample wasn't famous for being played as a one-shot: it was built for being looped.

Beat stretching

Let's loop the Amen Break by using our old friend the **`live_loop`**, introduced in last month's tutorial:

```
live_loop :amen_break do
  sample :loop_amen
  sleep 2
end
```

OK, so it is looping, but there's an annoying pause every time round. That is because we asked it to sleep for **2** beats; however, with the default BPM of 60, the **`:loop_amen`** sample only lasts for **`1.753`** beats. We therefore have a silence of **`2 - 1.753 = 0.247`** beats. Even though it's short, it's still noticeable.

To fix this issue, we can use the **`beat_stretch:`** opt to ask Sonic Pi to stretch (or shrink) the sample to match the specified number of beats.

> **Sonic Pi's `sample` and `synth` functions give you a lot of control via optional parameters such as `amp:`, `cutoff:`, and `release:`. However, the term 'optional parameter' is a real mouthful,so we just call them opts to keep things nice and simple.**

```
live_loop :amen_break do
  sample :loop_amen, beat_stretch: 2
  sleep 2
end
```

Now we're dancing! Although, perhaps we want speed it up or slow it down to suit the mood.

Playing with time

OK, so what if we want to change styles to old-school hip-hop or breakcore? One simple way of doing this is to play with time or, in other words, to mess with the tempo. This is super-easy in Sonic Pi: just throw a **`use_bpm`** into your live loop...

Language
>RUBY

```
live_loop :amen_break do
  use_bpm 30
  sample :loop_amen, beat_stretch: 2
  sleep 2
end
```

Whilst you're rapping over those slow beats, notice that we're still sleeping for 2 and our BPM is 30, yet everything is in time. The **beat_stretch** opt works with the current BPM to make sure everything just works.

Now, here's the fun part. Whilst the loop is still live, change the **30** in the **use_bpm 30** line to **50**. Whoa, everything just got faster yet **kept in time**! Try going faster: up to 80…to 120…now go crazy and punch in 200!

Filtering

Now we can live-loop samples, let's look at some of the most fun opts provided by the **sample** synth. First up is **cutoff:**, which controls the cutoff filter of the sampler. This is disabled by default, but you can easily turn it on:

```
live_loop :amen_break do
  use_bpm 50
  sample :loop_amen, beat_stretch: 2,
cutoff: 70
  sleep 2
end
```

Go ahead and change the **cutoff:** opt. For example, increase it to 100, hit **Run**, and wait for the loop to cycle round to hear the change in the sound. Notice that low values like 50 sound mellow and bassy, and high values like 100 and 120 are more full-sounding and raspy. This is because the **cutoff:** opt will chop out the high-frequency parts of the sound, just like a lawnmower chops off the top of the grass. The **cutoff:** opt is like the length setting, determining how much grass is left over.

Slicing

Another great tool to play with is the slicer FX. This will chop (slice) the sound up. Wrap the **sample** line with the FX code like this:

```
live_loop :amen_break do
  use_bpm 50
  with_fx :slicer, phase: 0.25, wave: 0,
mix: 1 do
    sample :loop_amen, beat_stretch: 2,
cutoff: 100
  end
  sleep 2
end
```

Notice how the sound bounces up and down a little more. (You can hear the original sound without the FX by changing the **mix:** opt to **0**). Now, try playing around with the **phase:** opt. This is the rate (in beats) of the slicing effect. A smaller value like **0.125** will slicer faster and larger values like **0.5** will slice more slowly. Notice that successively halving or doubling the **phase:** opt value tends to always sound good. Finally, change the **wave:** opt to one of 0, 1, or 2 and hear how it changes the sound. These are the various wave shapes. **0** is a saw wave, (hard in, fade out), **1** is a square wave (hard in, hard out), and **2** is a triangle wave (fade in, fade out).

Left The Akai MPC 2000, a classic early sampler

Bringing it all together

Finally, let's revisit the early Bristol drum and bass scene. Don't worry too much about what all this code means; just type it in, hit Run, then start live-coding it by changing opt numbers and see where you can take it. Please do share what you create! See you next time…

```
use_bpm 90
use_debug false
live_loop :amen_break do
  p = [0.125, 0.25, 0.5].choose
  with_fx :slicer, phase: p, wave: 0, mix: rrand(0.7, 1),
reps: 4 do
    r = [1, 1, 1, -1].choose
    sample :loop_amen, beat_stretch: 2, rate: r , amp: 2
    sleep 2
  end
end

live_loop :bass_drum do
  sample :bd_haus, cutoff: 70, amp: 1.5
  sleep 0.5
end

live_loop :landing do
  bass_line = (knit :e1, 3, [:c1, :c2].choose, 1)
  with_fx :slicer, phase: [0.25, 0.5].choose,invert_wave: 1,
wave: 0 do
    s = synth :square, note: bass_line.tick, sustain: 4,
cutoff: 60
    control s, cutoff_slide: 4, cutoff: 120
  end
  sleep 4
end
```

CLOUD RESEARCH IN MINIATURE

Think "cloud" and you'll picture massive data centres, but a Pi 2 cluster is just the thing to model cloud computing for affordable computer science research

Moore's law, describing exponential growth in computing power, has held good for 50 years now, but while processors continue to become more powerful, speeds have not increased. Rather, the number of processor cores crammed into each chip has grown, and programmers are still catching up with writing programs that run well across multiple cores, CPUs, and individual machines.

Concurrent programming, coming together with virtualisation in the data centre, and applications distributed through cloud computing, has opened up vast new areas of research for computer scientists. Can there really be a serious role for the tiny Raspberry Pi in this field?

Below The Pi cluster's Lego case was assembled by biscuit-fuelled volunteers!

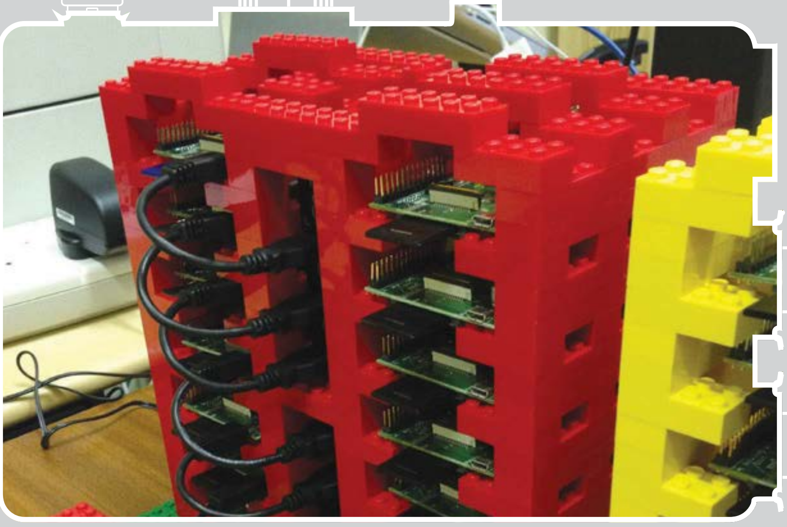
Scale model

Dr Posco Tso, a lecturer in the School of Computing and Mathematical Sciences at Liverpool John Moores University (LJMU), is researching cloud computing and big data processing, but ran up against the limitations of modelling clusters in software. Data centres are expensive beasts: Microsoft's $500m data centre in Dublin, and Google's $600m Iowa centre are at the cheaper end of the scale, with Apple's latest data centre costing the company $1bn. The affordable answer is to build a scale model, and the Pi is certainly affordable, as well as being small enough to even look like a scale model of a computer!

In 2013, while working at Glasgow University before moving to LJMU, Posco started with a 56-node cluster of Raspberry Pi Model B+ boards. Theses single-core boards were most restricted by low RAM, and weren't ideal for big data research. "Results were disappointing with B+," Posco tells us, with output 1,000 times less, at 0.1% of an Intel i7 chip.

The first cluster did give some useful results in comparing throughput between different workloads, and the B+ running 80 (idle) web servers in parallel showed the low overhead of Docker containers (see 'Docker on Pi' box on page 73). The Pi was "exciting, but still a toy."

When the Pi 2 appeared earlier this year, Posco, now in Liverpool, rigged up a test array of 14 for a simple Apache Spark setup, in standalone cluster mode (see 'Apache Spark' box), with promising results: "it renewed my hope to carry on research on it," he says. Posco had a "positive feeling" about results with Apache Spark, which were encouraging enough for him to commission the current 70-board cluster.



Below Could Raspberry Pi clusters prove a useful tool for big data work?

APACHE SPARK

Join dozens of computers together and you will need cluster management software to share your tasks out among the individual nodes. Of all the choices available – for there are many – Apache Spark is ideally suited to the Pi: Spark caches datasets in-memory, something that has seen it measured at up to one hundred times faster than the competition for some tasks.

For the Pi 2, with 1GB of RAM being just enough for Spark, the in-memory primitives free it from the relative slowness of the disk interface, and other architectural limitations. Data is shared across nodes in Resilient Distributed Datasets (RDDs), and transformed with map, filter, reduce, or join operations – programmed in Python, Scala or Java, through the Spark API (application program interface) – producing new RDDs.

This was made up of a mix of Pi 2 boards and B+ Pis, until enough Pi 2s were sourced. One bug being puzzled over is that, of the four cores, "one is always idle."

Research is also continuing with a Pi 2 cluster at Glasgow University, and between the two institutions, there exists the world's first distributed Raspberry Pi 2 cloud. At both universities, the Pi cluster serves the twin purposes of teaching and research, as students "create their own platform for virtualisation."

One cloud, many (research) angles

With his interest in cloud data-centre resource management, Posco's Pi researches have naturally targeted KVM, Xen, and LXC, then OpenStack, the free software cluster framework. Early work with Docker is continuing at the Glasgow end of the project, but in Liverpool they're now modelling data-centre clusters with one Pi per node, directly modelling a Pi 2 against a conventional data-centre virtual machine (VM).

This only removes a small overhead, but the real gain is simplifying modelling, as any bottlenecks in, for example, Spark processing can be explored for Pi-specific problems, or examined to see if they are just a function of the software's optimisation for x86 hardware, without also having to take account of bottlenecks in virtualisation: during early benchmarking a 30% loss measured in iperf (a network testing tool) at 70MB/s was put down to the virtualisation software. The virtualisation overhead with 1kB data size was a 68% loss.

Benchmarking with httperf with different sets of workloads showed the Pi 2 performing up to 100 times better than a B+. In a cloud, or data centre full of small VMs, these are the kind of throughputs which can be common, as database queries, and webpages and images, are served. Posco is also looking to "move from *cloud* to *fog*: real-time microdata mining." Current research in network function optimisation from the team may help here, particularly in the future growth of big data on ARM, and general data-centre use, as concerns over power consumption make ARM chips potentially far more attractive than power-hungry Intel processors.

Looking at software-defined networking, Posco's team is working towards "the big goal," modelling for a future "five million Pis" or other small ARM boards in data centres. "You could call us ambitious," says Posco, with admirable understatement. Yet the challenges of cooling current data-centre processors, and generating and paying for enough electricity for future data centres, has many predicting huge growth for ARM-powered servers.

As we go to press, many in the research community are gearing up for ParCo2015 in Edinburgh, where parallel computing researchers will be presenting results from FPGAs, GPUs, and other lower-power hardware. Smaller processors are no longer confined to the embedded space, and more resources should soon see optimisation of data-centre software for non-Intel platforms.

> "Between the two institutions, there exists the world's first distributed Raspberry Pi 2 cloud"

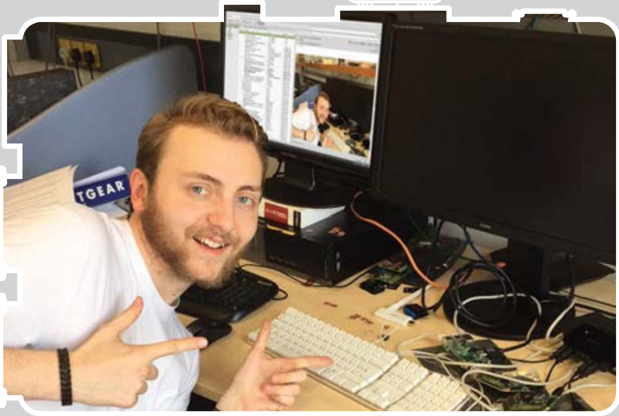
VIRTUAL PARA-HYPER-VISOR-ISATION?

Full virtualisation emulates an entire hardware platform, enabling, for example, a virtual Raspberry Pi ARM v7 system to run on your 64-bit laptop. The ***hypervisor*** – the software (or firmware, or even hardware) that creates and runs the virtual machines – emulates every aspect of CPU instructions, disk access, even starting BIOS in 16-bit legacy mode on PCs.

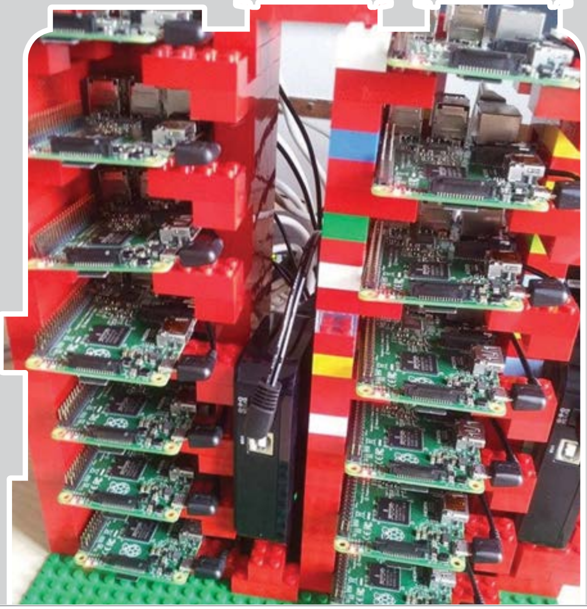
The guest operating system (OS) will run slightly slower, as everything must be translated by the hypervisor. Modified guest OSs can make calls directly to the hypervisor to speed things up, rather than running simulated machine I/O instructions; this is known as ***paravirtualisation*** (PV).

Hardware-assisted virtualisation, also known as ***accelerated virtualisation***, refers to a set of hardware extensions in the CPU to make it simpler to virtualise the processor. ***Xen*** hypervisor, running these instructions, calls this a ***hardware virtual machine*** (HVM), and supports combinations of HVM with PV drivers.

Lastly, ***operating-system-level virtualisation*** allows direct system calls to the OS kernel from multiple, isolated VM-like ***containers***, all sharing the same underlying kernel. ***OpenVZ*** is a popular example of this, often found in web hosting services to provide VMs, while ***Docker*** is a layer of abstraction around containers to make them easy to deploy. Linux also has its own ***KVM*** (Kernel-based Virtual Machine), which provides virtualisation infrastructure.



Right You need a cheerful disposition, like intern Jim Walker, to take software problems in your stride



Below A Lego case is less robust, but, for the smaller cluster, it allows easy swapping, and looks good when showing off the hardware

PayPal is an early adopter of (64-bit) ARM in the data centre, looking to save more than 80% in running costs with low energy consumption chips from Applied Micro. Qualcomm has announced it is entering the data-centre market later this year, too, with a new 64-bit ARM chip designed for servers.

Lego case

Back at the scale model end of computing, the original Pi cluster attracted some attention because of its Lego case, which was assembled cheaply by biscuit-fuelled volunteers (biscuit budget: £5!). "When we lifted it, it fell apart," Posco tells us, "because of the weight of the cables." A 3D-printed case has been developed, optimised for "minimal amount of material," ventilation, and the boards being "easy to slide out for maintenance."

There is a lot of cabling, and a substantial amount of current flowing at five volts, required to make a 70-board cluster. In Glasgow's cluster of 14 Pi 2s, they've now moved to Lego again, to ease hot-swapping of boards, and access to the HDMI port, "making network troubleshooting just that little bit quicker," according to intern Jim Walker.

No Intel

New users of Raspbian on the Pi, and the similarly Debian-based Ubuntu on laptops and PCs, are often surprised by how seamless installation of lots of new software is. However, a lot of work and testing – by Debian developers, Ubuntu and Raspbian developers, and downstream users who run the pre-releases – produces these (mostly flawless) results.

If you're making your way down a software path which is less trodden, be prepared to do more than a little legwork: putting error messages into search machines, editing configuration files, recompiling software. For 32-bit ARM v7, you're a long way from the mainstream for cloud research, and getting hypervisors and paravirtualisation software (see 'Virtual Para-hyper-visor-isation?' box) up and running isn't always the one-step install of the Intel x86 and AMD64 architectures.

Posco's team's blogs are full of posts detailing workarounds – left to help the next person stumped by an error message, and throwing themselves on the mercy of the search engine gods – on topics including problems on the team's standard PCs as well. An example is getting httperf to work on Ubuntu, for benchmarking the cluster under various HTTP workloads (to measure the throughput of the cluster, and see what size of data the tiny Pi works best with). A standard installation throws errors about FD_SETSIZE; fixing involves changing limits, then building a fresh version from source code.

Logical Volume Manager (LVM) enables adding more disks and collecting together partitions of several disks as one virtual partition. While the Pi typically has an SD card, and perhaps a USB drive for data, working with big data can involve large collections of disks, and the need to hot-swap if there's a problem. LVM makes it straightforward; what it doesn't like is a virtual machine installed on a disk partition, as this involves logical partitions within logical partitions. So, to run Xen hypervisor successfully, a logical volume must be created for each of the VM's partitions; now they can be formatted and mounted successfully.

Summer in Glasgow

Glasgow intern Jim Walker is spending the summer preparing the 14-node Pi 2 setup for Docker cluster managed with SaltStack, recently announcing "creating the first (a scoop! The first!) ARM Docker Hadoop image and Dockerfile," and writing with good-natured humour on those challenges of the road less travelled in software. After a Docker update to a version that didn't support ARM, Walker found the previous Arch Linux version of Docker unarchived, and had to build his own, only to find a fixed Docker was released the next day.

Reflecting on the necessity for updates, he writes: "It doesn't exist. Updates are a swindle and a lie. If a system is working, don't change it. Don't, for example, command all your nodes to update their most vital piece of software when there is absolutely no need for it. I really recommend that if you think that your machines do need some shiny new software, please find a good reason for updating it, and then make sure you test it. Please note that neither shiny or new are good reasons for updating software."

We can't argue with that, but do reflect that the motto for many in the maker and hacker community is 'fix it till it breaks'! Walker has also started updating the Twitter account for the Scottish end of the project – you can follow developments at **@glasgowpicloud** – and the Glasgow Pi 2 cloud may be on a tour of events this autumn.

Future cluster

Everywhere, the team is coming up against the problems of Intel bias in design decisions. The OpenStack software is billed as cross-platform, but is inefficient on ARM due to x86 optimisations. With default 64MB data chunks in the Spark tests, the Pi showed itself CPU-bound. "One way for Pi to perform better is decrease the data chunk size," says Posco, but benchmarking must initially be done with out-of-the-box settings, for proper academic comparison.

The Pi itself has limitations, it must be admitted: "The next-generation Pi needs better networking," Posco reckons. The benchmark from 100kB data size – 20% CPU, 80% idle – tells us the Pi is "limited by the network."

With plans to build a 200-board cluster next, entirely made of Pi 2s, Posco should be getting benchmarks indicating how suitable the Raspberry Pi is for some serious big data work, for a power consumption and purchase cost comparable to just a handful of rack-mounted Intel-based servers.

Since the LJMU / Glasgow cluster started, several other Pi clusters projects have appeared, many focused on cloud computing research, and even GCHQ, the UK government's Signals Intelligence, building a rather neat one (with Power-over-Ethernet to reduce cabling), to "inspire schoolchildren and students to take up science, technology, engineering and maths (STEM) subjects."

With ARM in the data centre set for massive growth in the next few years, cluster tools and infrastructure for the Pi will get better, and we look forward to hearing more from Posco and his team on the Pi's suitability for big data work.

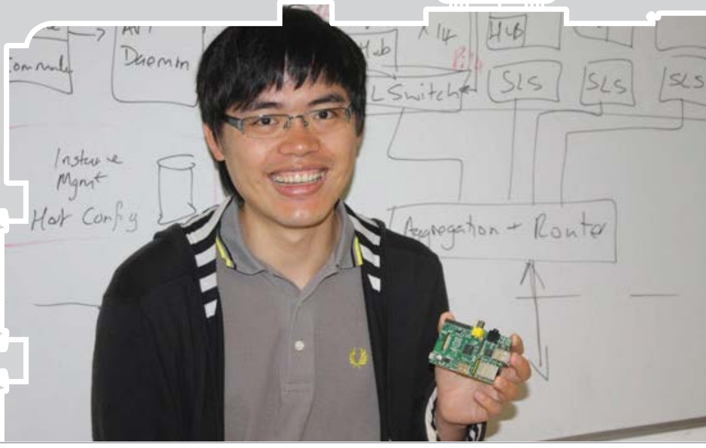
> "Jim Walker is spending the summer preparing the 14-node Pi 2 setup for Docker"

DOCKER ON PI

Docker images are available for 64-bit PCs, and 32-bit is unsupported – although many users have built it for themselves – but it is available for 32-bit ARM, in particular the Raspberry Pi, and the Pi 2's four cores make containerisation a tempting project for any Pi user.

Docker is loved by developers for freeing them from dependency problems, by expressing a series of commands to build each container (see examples at **docker.com**). However, the Docker system starts with a core image: this must be an ARM v7 image to run on Raspberry Pi so, effectively, you have to develop on a Pi. Fortunately, the quad-core Pi 2 is speedy enough to make this manageable, if not exactly swift.

Running Raspbian, you won't find Docker in your repository, but change your SD card for one with Arch Linux or Ubuntu and you will; **hypriot.com** also has pre-built images with Docker. The advantage of using Arch Linux is that it is by default a fairly minimal installation, so all the Pi's precious system resources can be put into running Docker containers and their contents.



Below Dr Posco Tso holding one little board with many applications

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved...

NEED A PROBLEM SOLVED?

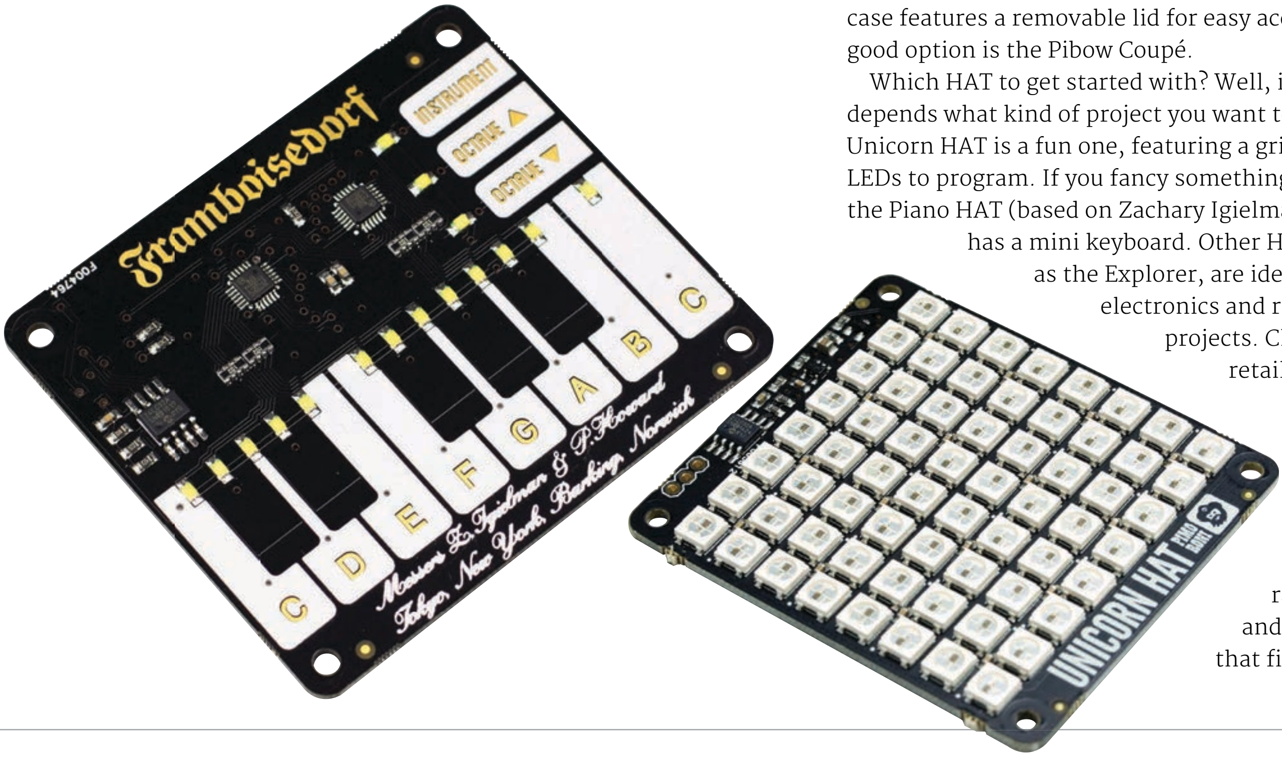
Email **magpi@raspberrypi.org** or find us on **raspberrypi.org/forums** to feature in a future issue.

A HAT FOR ANY OCCASION

I understand that you can now get HAT add-on boards for the Raspberry Pi that just fit on top of it, rather than being attached by a standard ribbon cable. Please could you tell me whether these will work with any Raspberry Pi model? And do you need to remove the Pi from its case in order to plug in the HAT? Also, do you have any suggestions as to which HATs a newcomer like me might get started with?
David Wood

The HAT (Hardware Attached on Top) specification was introduced by the Raspberry Pi Foundation last year, aiming to make add-on boards much easier to use: each plug-and-play HAT supports a special autoconfiguration system that allows automatic GPIO and driver setup. Most HATs are compatible with the Pi Model A+, B+ and 2, although you should double-check with the supplier first. While not all cases will enable you to connect a HAT, the official Raspberry Pi case features a removable lid for easy access; another good option is the Pibow Coupé.

Which HAT to get started with? Well, it really depends what kind of project you want to do. The Unicorn HAT is a fun one, featuring a grid of coloured LEDs to program. If you fancy something musical, the Piano HAT (based on Zachary Igielman's PiPiano) has a mini keyboard. Other HATs, such as the Explorer, are ideal for electronics and robotics projects. Check out retail sites such as Pimoroni, The Pi Hut and Adafruit to peruse a wide range of HATs and pick one that fits!



FROM THE RASPBERRY PI FAQ

RASPBERRYPI.ORG/HELP

How do I change the password for my Raspberry Pi?
It's actually very simple. In a terminal, type `passwd`. You'll be prompted for the current password (the default in Raspbian is 'raspberry'). After that, enter the new password, retype it and you'll get a message saying that it's been updated.

Can I use Adobe Flash Player on the Raspberry Pi?
Since Adobe has opted not to support any devices with the kind of ARM CPU used by the Pi, the official answer is no. However, there is a workaround involving the Chromium browser and a Pepper Flash Player plugin. See **bit.ly/1NaJOmo** for instructions. It might not work perfectly on every website, but we were able to play some Flash games and animations.

I put Raspbian on a 16GB microSD card using my Windows PC. Why does it show up as having just a few megabytes of space left?
Don't worry: this is due to the way Windows ignores the partitions on the SD card occupied by Linux. However, note that as with any installation of Raspbian, you'll need to expand the file space used so that the root partition takes up the entire space available on the SD card – not just the size of the original image. To do this, go to the Raspi-config menu (by entering `sudo raspi-config` in a terminal) and choose the 'expand_rootfs' option.

What kind of cable do I need to use the A/V output of my Raspberry Pi 2? I want to connect it to an old TV with no HDMI port.
The 3.5mm A/V jack on the Pi 2 and B+ is used to output composite video and stereo audio. However, you need the right sort of cable for it to work – many of them look alike, but aren't wired in the same way. A Microsoft Zune cable will work perfectly. Alternatively, a standard camcorder cable should be okay, so long as you switch the yellow (video) and red (right audio) RCA plugs. Or use an iPod cable and switch the white and red (left and right audio) plugs. To force your Pi to use this output rather than HDMI, hold SHIFT+3 at bootup for PAL video (or SHIFT+4 for NTSC).

WIFI WOES

I'm having trouble getting my Raspberry Pi to connect to my wireless router using a USB WiFi dongle. Any suggestions?
Mark Jones

You didn't say which dongle you're using, but try the following steps...

1. Enter `lsusb` in a terminal and you should see your WiFi dongle in the output. If not, try using a powered USB hub or plug the dongle into another computer to check it's working.

2. Run `lsmod` in the terminal to check the dongle's module is loaded. If so, the drivers should be loaded correctly.

3. Next, run `ifconfig -a` to check the dongle is being picked up. You should see a 'wlano' in the output. If not, you may need to install a firmware package for your dongle, using apt-get.

4. If you still can't get online, you'll need to check the WiFi configuration in **/etc/network/interfaces** (Raspbian). See **bit.ly/1KtYhqQ** for more details.

THE MAGPI APP

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

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

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

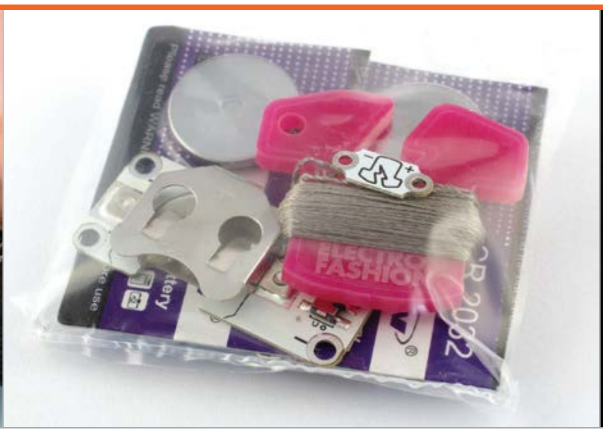
How can I search the digital magazine for keywords?
Finding direct references is really easy with *The MagPi* app – all you have to do is tap the screen to get the app's GUI to show, and then press the small magnifying glass icon in the top-right corner of the screen. Now, just type in your search term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

Available on the App Store

Get it on Google play

cpc.farnell.com

£15 / $35



KITRONIK ELECTRO FASHION DELUXE E-TEXTILES PACK

Promising an entry into the world of wearable electronics for under £15, does Kitronik's kit deliver? **Gareth Halfacree** sees...

For years, wearable electronics were either bulky and fragile or bulky and expensive. Sure, creating something like a digital watch was straightforward, but the minute you wanted your circuits to be flexible, things got complicated. Today, there are several options for those looking to make flexible circuits, with conductive thread being the most accessible to the hobbyist.

Designed as an introduction to what Kitronik describes as 'e-textiles', the Electro Fashion Deluxe pack brings together a selection of components from the firm's wearable electronics range. Inside the blister pack is a trio of battery holders – one full-size with built-in power switch, one full-size without switch, and one micro-size for smaller projects – with matching batteries. Also contained in the pack are pairs of sewable LEDs in three different colours, pairs of full-size LEDs with built-in flashing circuit (also in three different colours), a slide switch, a momentary push switch, and six metres of conductive thread.

The thread, the real gem of the kit, is impressive stuff. Built from a traditional thread coated in a thin layer of silver, it's both flexible and conductive. Roughly 18 denier, it's somewhat bulky – thicker than standard sewing thread, but not as thick as that used in upholstery and heavy-duty applications – and has a somewhat annoying tendency to come untwisted at the cut end.

While the kit comes with most of the components you'll need to begin your first project, there's something missing: a needle. You'll need a fairly large-eyed needle in order to making threading possible, and we'd recommend folding the thread in half or using a needle-threader: unlike traditional cotton thread, the silver-plated conductive thread doesn't stick together to form an easily threaded point, even when wetted.

Compared to rival kits, the lack of needle aside, Kitronik's kit is well thought out and comes with a reasonable selection of parts. The sewable LEDs are a particular delight, formed of surface-mount LEDs in white, blue, and 'sakura', a lilac-like colour, attached to tiny circuit boards with wide 'eyelets' suitable for passing the thread through. Sadly, however, they're not labelled beyond markings for positive and negative terminals and the Electro Fashion logo on the

Related

ADAFRUIT BEGINNER L.E.D. SEWING KIT

Coming complete with needles and press-stud 'switches,' the Adafruit kit is another great way to get started with conductive thread projects, but, for the money, the Kitronik kit is likely a better buy.

£12.50 / $15

adafruit.com

Maker Says

> A great way to get a little more involved with e-textiles

Kitronik

> The lack of needle aside, Kitronik's kit is well thought out

rear, meaning you'll have to juggle some thread or wire, a battery, and the LEDs to find out which colour is which.

The tiny surface-mount components continue with a pair of switches, one of which has a small slide and the other a momentary-connection push-button. Like the LEDs, these have been designed with fabric mounting in mind and are extremely compact – definitely something to keep out of reach of smaller children, owing to the potential hazard if ingested.

The larger LEDs seem out of place at first, but a glance at the bundled documentation – a full-colour booklet walking the reader through the basics of sewing a circuit – reveals a hidden feature: integrated into their bodies is a flasher unit which cycles the blue, green, or white LED on and off every couple of seconds. While they lack the dedicated sewing eyelets of the surface-mount LEDs in the pack, it's the work of moments to bend the legs into the right shape for fabric mounting.

With a needle threaded and a piece of non-conductive fabric as a base – we chose a glove for our experimentation, equipping it with one of the white LEDs on the tip of a finger – you can start sewing. For beginners to the craft, a thimble or two is a must if you want to keep your fingertips free from punctures; for those who have a some sewing experience, adding the components is little harder than sewing a button to a shirt.

The thread is flexible, but it is not elastic; sew it too tight into the bend of a glove's finger, for instance, and you can expect to lose some freedom of movement or hear the disheartening 'ping' of snapped thread or a lost component. For most uses, though, it should work just fine, and with lengths of up to 45 metres available, the supply of conductive thread can be quickly and affordably replenished when the bundled six-metre allowance runs out.

Last word

While there aren't enough components to build anything truly exciting, this starter kit offers a great taste of what's possible. Be prepared for some frustrations, though, and make sure you have a large-eyed needle to get started.

★★★★☆

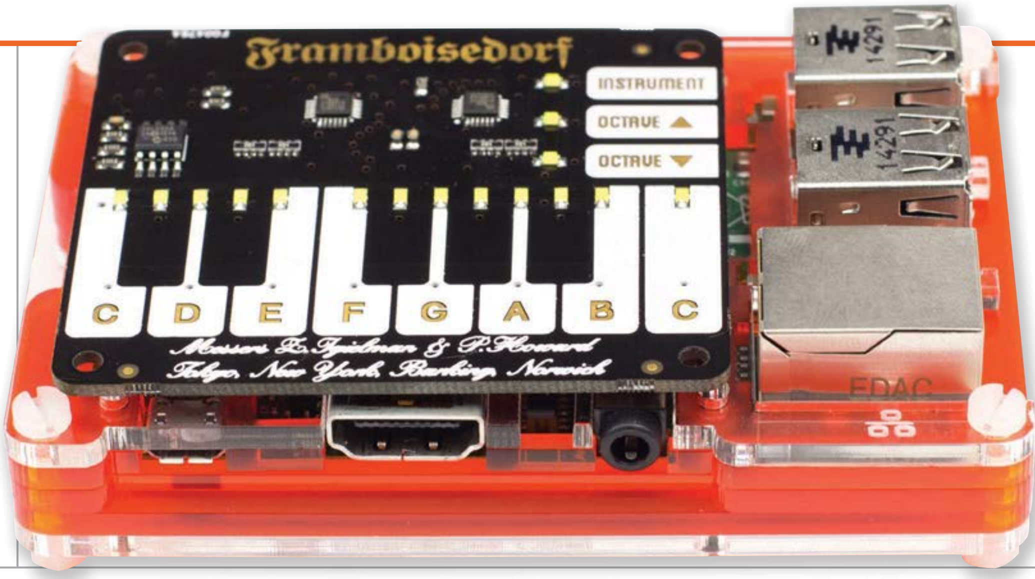
pimoroni.com

£15 / $24

Maker Says

“A Pi-based piano, transformed with Pimoroni’s signature polish”

Pimoroni


Framboisedorf
INSTRUMENT
OCTAVE ▲
OCTAVE ▼
C
D
E
F
G
A
B
C
Messers Z. Igielman & P. Howard
Tokyo, New York, Barking, Norwich

PIANO HAT

Les Pounder looks at a HAT-based piano keyboard which uses capacitive touch keys for input

In 2014, a 14-year-old called Zachary Igielman launched a successful crowdfunding campaign for his Pi Piano. Originally created using a MCP23017 GPIO expander and a series of momentary switches, the Pi Piano used a piezo buzzer to simulate the tone of a note. One of Zachary’s advisors was Phil Howard from Pimoroni, who worked with Zachary to create this new and improved version.

Crowdfunded project

The latest incarnation is now known as Piano HAT and uses a capacitive touch-based interface to simulate a portion of a traditional piano keyboard. The board measures 64mm by 56mm and fits neatly on top of the models A+, B+, and Pi 2. Being a HAT board, it also attaches to all 40 of the GPIO pins. As well as the keyboard, we have the ability to raise and lower the octaves and change instruments via three extra inputs on the board.

Hardware is just one part of this project, and the software which supports Piano HAT is all based on Python, which enables existing projects created in this language to easily integrate Piano HAT. Phil Howard has written a series of examples which use the Pygame library to demonstrate how to use Piano HAT as a typical piano via a series of audio samples, which can be swapped for any types of sounds should you wish to build your own input. Piano HAT also comes with a tutorial which demonstrates using the built-in LEDs to teach playing a simple tune, in this case ‘Twinkle, twinkle, little star’. So, using this library, we can code our own tutorial for others to learn with.

Powerful MIDI synthesiser

Software is not just limited to simple samples. In fact, Phil has supplied a number of methods to connect the Piano HAT library to a MIDI service running on your Pi. This then turns the Piano HAT into a powerful synthesiser input for software such as Yoshimi and SunVox, although a Raspberry Pi 2 is required for these applications.

The Piano HAT looks like a rather simple board, but thanks to a clear Python library, well-built hardware, and a powerful MIDI control option, we have a board for musical adventures.

Related

MAKEY MAKEY

This versatile board has been used to build a piano from stairs, a camera trigger using water, and a keyboard from pasta.

£40 / $50

makeymakey.com

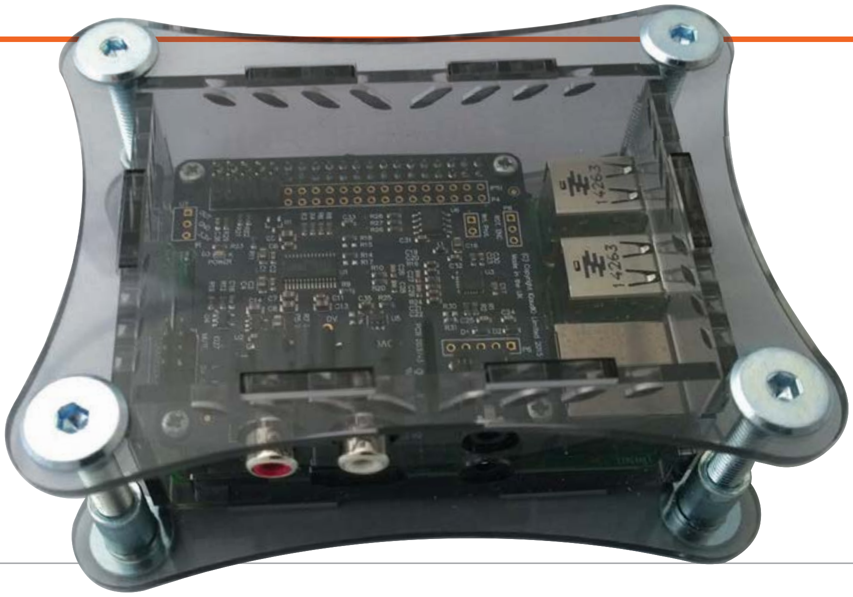
Last word

Such a simple board, but with limitless applications thanks to Python and MIDI. This is a great platform for musical experimentation and it can easily be used with *Minecraft* and other projects.

★★★★☆

Iqaudio.com

£30 / $50



Maker Says

> Outstanding sonic performance with variable phono line out and dedicated headphone amplifier
>
> IQaudIO

IQAUDIO PI-DAC+

Any Raspberry Pi becomes an audiophile music streamer with this high-resolution DAC and headphone amplifier

Even a standalone Pi makes a very decent music streamer, but add a high-quality DAC and it becomes a true audiophile device. The IQaudIO Pi-DAC plus is a HAT-compliant board which uses the I^{2}S interface for optimal transmission of digital audio between the Pi and the DAC. Once fitted, you get analogue output from either two phono sockets, for connection to an amplifier, or a 3.5mm headphone socket, along with additional pinouts for adding optional features.

No soldering required

The board comes without any instructions, but you can find these on the IQaudIO website. No soldering is required, and it works with Raspberry Pi A+, B+, and 2. You need to screw the supplied spacers to the Pi to support the board, then you simply connect it to the Pi's GPIO board, screw it down gently, and that's that.

IQaudIO can also supply a case, an acrylic affair that has suitable cut-outs for the Pi-DAC+ ports.

Once assembled, you can download a number of audio-centric distributions that have Pi-DAC+ drivers built in – including Volumio, RuneAudi, and PiCorePlayer – or follow the directions to add support to an existing installation. There is also a preconfigured Raspbian image on the IQaudIO site. We used Volumio 1.55, which worked perfectly. The music source can be anything from files on a NAS, to a streaming service such as Spotify, or a directly attached USB hard drive.

Excellent sound

The DAC sounds excellent, powered by a Texas Instruments PCM5122 DAC at resolutions up to 24-bit/192kHz. From a sonic point of view, a Pi equipped with a DAC like this can hold its own with far more exotic and expensive company. We were also impressed with the headphone output: tested with a pair of Sennheiser HD600 headphones, the sound was superb, with the clarity and spaciousness that you get from the best audio.

Expansion options at extra cost include attaching a rotary volume control or an IR sensor for remote volume control. The Pi-AMP+ is 2×20W stereo amplifier which attaches to the top of the board. You can also use the supplied optional right-angled header, designed to be soldered underneath the Pi-DAC+ to access the Pi's GPIO signals.

Related

AUDIOPHONICS I-SABRE DAC ES9023

The DAC from Audiophonics is a similar price to the Pi-DAC+, but it lacks a headphone output. It uses the Sabre ES9023 DAC instead of the Texas Instruments/ Burr Brown DAC in the Pi-DAC+, and includes a pass-through GPIO connector so you can stack additional boards.

£30 / $50

audiophonics.fr/en

Last word

Great sound and the inclusion of a headphone amplifier make the Pi-DAC+ a recommended accessory for music-loving Pi enthusiasts.

★★★★★

lightberry.eu

From £52 / $81

Maker Says

“Easy to install, simple to use, a joy to watch

Lightberry

LIGHTBERRY

Russell Barnes tests a technology designed to add a new dimension to your home cinema experience

Lightberry is inspired by a TV technology pioneered by Philips called Ambilight. While the tech naturally comes pre-fitted on Ambilight TVs, it's perfectly possible to modify any TV to add its special kind of ambience-enhancing trickery. What does it do? The premise is simple: you affix a ring of colour-switching LED lights to the back of your LCD TV and – assuming your TV is against a wall and the ambient lighting is low – the colours on the LEDs are designed to perfectly sync with the action on screen in colour and intensity, so that light appears to bleed out of the screen and dance against the wall in the background.

As with Philips' creation, Lightberry – which harnesses the power of the Raspberry Pi for computation – is all about adding a little more immersion into your viewing experience. While the idea is simple, the technology behind it certainly isn't, though for most people the results really do speak for themselves.

Sold either as a full kit starting from just over £50 / $80, or as constituent parts available separately, Lightberry is designed to take the pain out of the technical aspects, giving you as close to a plug-and-play experience as you could hope for. Obviously, sampling millions of pixels of onscreen data and converting that to the right LED, colour and brightness at exactly the right moment isn't the simplest project to tackle, but Lightberry manages to make it easy. It's also designed to work perfectly with the best software solutions for this technology, too, including Hyperion and Boblight.

Once you've picked the right length LED strip to fit your particular size of TV and ordered all the kit you need, it all arrives boxed and ready to assemble. While we found no instructions in our box, **lightberry.eu/getting-started** offers brief but functional step-by-step instructions to get you up and running. Probably the fiddliest part of the process is arranging and affixing the LEDs themselves. Elsewhere it simply requires you to connect the appropriate converter box, download software and install it onto an SD card – a process any Raspberry Pi user should already be intimately aware of.

Related

PHILIPS AMBILIGHT

The real alternative to Lightberry is an actual Ambilight TV. The cheapest we could find (at the time of writing) was £350. Ouch.

From £350 / $550

bit.ly/1JiGm9Y

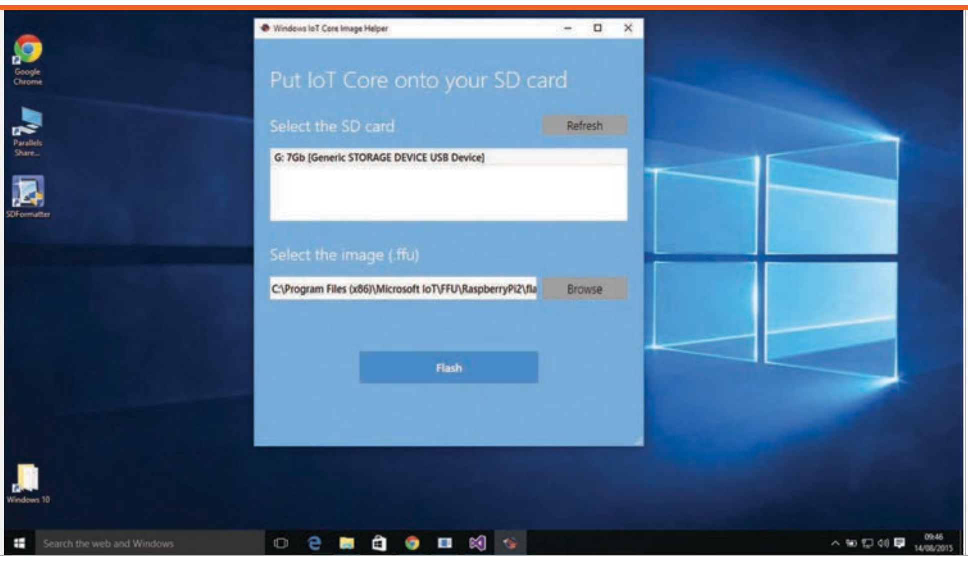
Last word

While it can take a little trial and error to best position the LEDs, once you've hit the sweet spot we found Lightberry a joy to use. It's not the cheapest Raspberry Pi project, but it's one that is sure to please the movie fans in your household.

★★★★☆

dev.windows.com/en-us/iot

FREE



Windows IoT Core Image Helper

Put IoT Core onto your SD card

Select the SD card

G: 7Gb [Generic STORAGE DEVICE USB Device]

Select the image (.ffu)

C:\Program Files (x86)\Microsoft IoT\FFU\RaspberryPi2\fla

Maker Says

"Rapidly prototype and build your Windows IoT solutions

Microsoft

WINDOWS 10 IOT CORE

What are we to make of Microsoft's first foray onto the Raspberry Pi? **Lucy Hattersley** investigates...

It's all too easy to get hung up on what Windows IoT Core isn't. It certainly isn't Windows desktop on your Raspberry Pi, or even a cut-down version such as Windows RT. There is no Start button here.

Windows IoT Core is also not a self-contained programming and development environment. There isn't a shell in the OS, so you can't type commands directly into your Raspberry Pi. In fact, there's no interface at all: plug it into a monitor and it simply displays the IP address and a few other bits of information.

Beyond that, there's the moral maze of Windows not being open-source, unlike Linux (which underpins Raspbian, Ubuntu, and most other Raspberry Pi OSes). As Linus Torvalds, principal developer of Linux, points out, open source isn't simply an ethical way to create software - it's a more efficient way to create better software. Open source enables users to pool efforts and build upon the work of each other. It's also more fun.

Welcoming Windows

Reaching beyond all those issues brings you to the good. Windows IoT Core is close in essence to Windows Embedded (or Windows CE). That's the software that sits behind information kiosks, photo booths, and countless advertising displays. IoT Core is ubiquitous in nature, too, which makes it worth investigating for its own sake.

Windows IoT Core offers real value for OEM developers to use Raspberry Pi devices for testing. Those people looking to learn Microsoft development will find the Raspberry Pi a great learning platform.

Installation and development

For the rest of us, Windows IoT Core is an odd beast. Setup is straightforward but long-winded. You will need a PC running Windows 10, and you need to have installed Microsoft Visual Studio (the Community version is free). You need to put Windows 10 into Developer mode and use the software to flash your SD card. Place this into the Raspberry Pi, boot up, and you'll see a welcome interface displaying the currently used IP address. You use this to connect from the Windows PowerShell and Visual Studio. All development takes place inside Microsoft Visual Studio.

Microsoft has integrated GPIO support, and sample code is available in C# and C++. Microsoft has already provided a great selection of online tutorials, and we counted over 60 projects to help you get started.

Related

SNAPPY UBUNTU CORE

Snappy Ubuntu Core is a stripped-down version of Ubuntu aimed at IoT development. It utilises a new 'Snappy' means of updating packages.

Free

developer.ubuntu.com/en/snappy

Last word

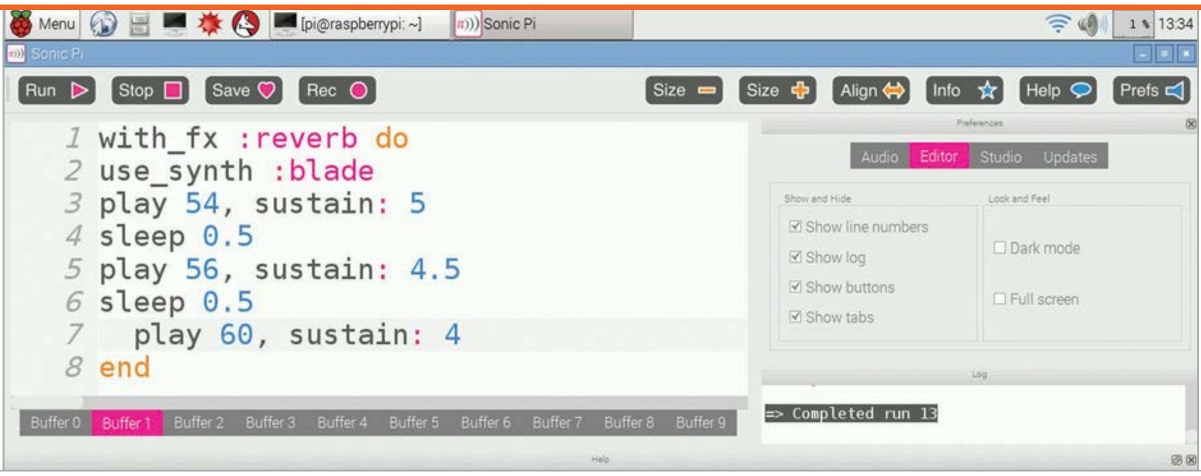
Equipment developers will find the Raspberry Pi a great low-cost development platform. If you want to be inside the Microsoft ecosystem, or just want to learn C# and Visual Studio, then this is a great way to start.

★★★★☆

Maker Says

A free live coding synth for everyone

Sonic Pi



```
with_fx :reverb do
use_synth :blade
play 54, sustain: 5
sleep 0.5
play 56, sustain: 4.5
sleep 0.5
  play 60, sustain: 4
end
```

Preferences

Audio | Editor | Studio | Updates

Show and Hide

☑ Show line numbers

☑ Show log

☑ Show buttons

☑ Show tabs

Look and Feel

☐ Dark mode

☐ Full screen

Log

```
=> Completed run 13
```

SONIC PI V2.6

Les Pounder finds out how you can create music using your Raspberry Pi via an easy-to-learn language and great new interface

In 2012 we saw the release of the original Raspberry Pi, but behind the scenes in Cambridge, Sonic Pi was in the early stages of prototype. In fact, the first version was compiled using the original Raspberry Pi by Dr Sam Aaron, in consultation with the Raspberry Pi Foundation's future employee Carrie Anne Philbin. The original use-case for Sonic Pi was for a series of eight lessons in Carrie Anne's IT classes, where Sonic Pi would be used to teach computer science and music in a new and exciting way. Fast-forward to today and we find that with each new version, Sonic Pi has grown in features, with refinements and additions that excite and intrigue budding musicians and coders. But does it still meet its goal?

Algorave

With the latest version 2.6, code-named Algorave, the first thing that we notice is that the version 2 user interface has been improved. The new interface has a flatter look and uses the Sonic Pi branding, along with a hot pink colour for highlighting workspaces. Workspaces are now renamed 'buffers' to accommodate users with low-resolution displays and their limited screen real estate. The user interface retains its familiar layout, with buttons for quick access located to the top of the screen, a coding window to the left, a log/preferences section to the right, and the excellent help system at the bottom. Along with the new user interface, we also find a new 'dark mode' in the preferences section. This mode changes the familiar white interface to a dark and mysterious palette, suitable for live coding performances, something which Sonic Pi 2.6 has been geared towards. Further user interface configurations include options to show/hide aspects of the GUI, such as the buttons at the top of the screen.

Musically, Sonic Pi 2.6 comes with more synths, instruments which change the sound of notes; we are particularly fond of 'blade', which has a sound reminiscent of *Blade Runner*'s synth violins. There are also more effects that can be applied to instruments, such as panslicer which modulates the stereo panning of audio for some rather trippy results. Effects can be used to change a sound or instrument for

> We have seen it come of age thanks to a mature language and slick interface

Related

PIANO HAT

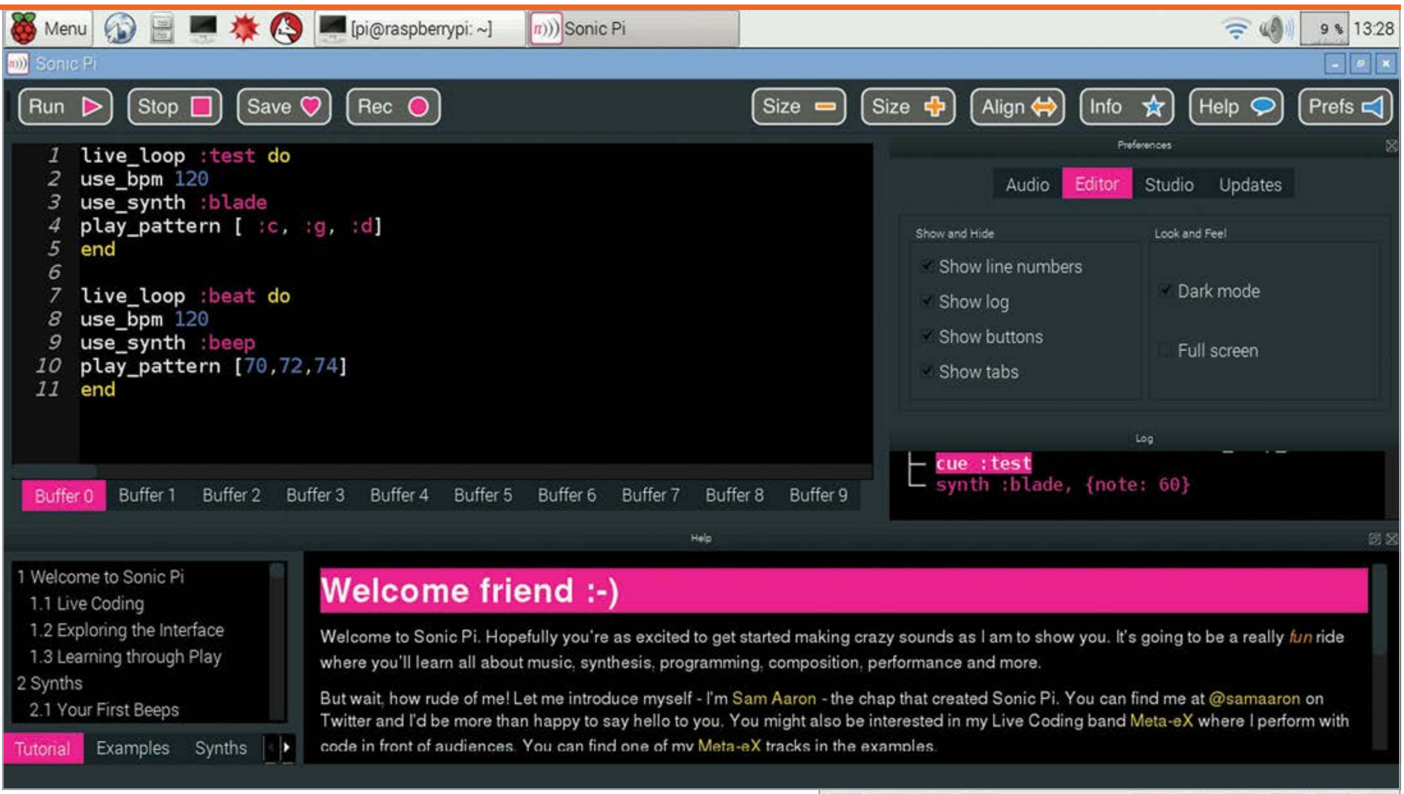
An elegant musical interface for the Raspberry Pi. Using Python and MIDI interfaces, it enables coders to hack tunes in many different ways.

£15 / $24

pimoroni.com

sonic-pi.net

FREE



```
live_loop :test do
use_bpm 120
use_synth :blade
play_pattern [ :c, :g, :d]
end

live_loop :beat do
use_bpm 120
use_synth :beep
play_pattern [70,72,74]
end
```

Show and Hide
☑ Show line numbers
☑ Show log
☑ Show buttons
☑ Show tabs

Look and Feel
☑ Dark mode
☐ Full screen

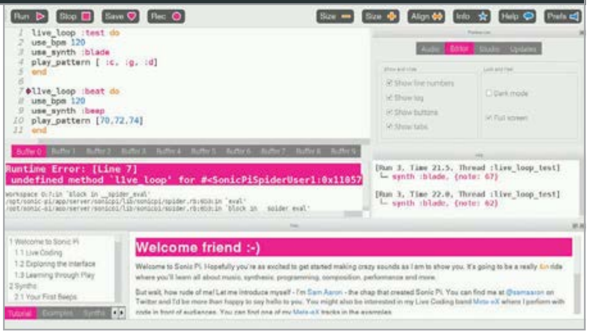
```
├ cue :test
└ synth :blade, {note: 60}
```

1 Welcome to Sonic Pi
1.1 Live Coding
1.2 Exploring the Interface
1.3 Learning through Play
2 Synths
2.1 Your First Beeps

Welcome friend :-)

Welcome to Sonic Pi. Hopefully you're as excited to get started making crazy sounds as I am to show you. It's going to be a really *fun* ride where you'll learn all about music, synthesis, programming, composition, performance and more.

But wait, how rude of me! Let me introduce myself - I'm Sam Aaron - the chap that created Sonic Pi. You can find me at @samaaron on Twitter and I'd be more than happy to say hello to you. You might also be interested in my Live Coding band Meta-eX where I perform with code in front of audiences. You can find one of my Meta-eX tracks in the examples.



```
live_loop :test do
use_bpm 120
use_synth :blade
play_pattern [ :c, :g, :d]
end

live_loop :beat do
use_bpm 120
use_synth :beep
play_pattern [70,72,74]
end
```

☑ Show line numbers
☑ Show log
☑ Show buttons
☑ Show tabs
☐ Dark mode
☑ Full screen

```
Runtime Error: [Line 7]
 undefined method `live_loop' for #<SonicPiSpiderUser1:0x11057
workspace 0:7:in `block in __spider_eval'
/opt/sonic-pi/app/server/sonicpi/lib/sonicpi/spider.rb:604:in `eval'
/opt/sonic-pi/app/server/sonicpi/lib/sonicpi/spider.rb:604:in `block in __spider_eval'
```

```
[Run 3, Time 21.5, Thread :live_loop_test]
 └ synth :blade, {note: 67}

[Run 3, Time 22.0, Thread :live_loop_test]
 └ synth :blade, {note: 62}
```

1 Welcome to Sonic Pi
1.1 Live Coding
1.2 Exploring the Interface
1.3 Learning through Play
2 Synths
2.1 Your First Beeps

Welcome friend :-)

Welcome to Sonic Pi. Hopefully you're as excited to get started making crazy sounds as I am to show you. It's going to be a really fun ride where you'll learn all about music, synthesis, programming, composition, performance and more.

But wait, how rude of me! Let me introduce myself - I'm Sam Aaron - the chap that created Sonic Pi. You can find me at @samaaron on Twitter and I'd be more than happy to say hello to you. You might also be interested in my Live Coding band Meta-eX where I perform with code in front of audiences. You can find one of my Meta-eX tracks in the examples.

Above Error handling, along with user direction, has improved and now makes debugging a simple process

Top Live coding is at the heart of Sonic Pi, and the new dark mode creates a moody palette for performances

varied results; for example, reverb is often used to give an instrument a sense of acoustic space.

Sonic Pi has a built-in help system which equates to a full manual on how to use it. It's a great read that is constantly being updated with each new release, and this is the case for v2.6, with all of the new functionality, FX, and synths being listed with examples of their use.

Improved error handling

In previous versions of Sonic Pi, the error and syntax handling when encountering an issue has left some new users a little bamboozled. In Sonic Pi 2.6, however, we see the introduction of a new error message reporting system. This pinpoints the issues via an arrow in the left margin, along with an improved description of the error encountered in the output window, with syntax errors highlighted in blue and runtime errors in pink. These improvements provide an enjoyable and friendly experience when composing your first track.

Sonic Pi also boasts an impressive interface to *Minecraft*, which was introduced in v2.5 and originated during a conversation between Sam Aaron and Martin O'Hanlon. Now, using Sonic Pi, you can create music and alter a world in *Minecraft* to interact with the music, a truly epic experience for children and adults alike. In Sonic Pi 2.6 we see a bug fix for the mc_surface_teleport function, along with a few other logic fixes.

With the release of Sonic Pi 2.6, we can see it come of age thanks to a mature language and slick interface, and it has become clear that it is one of the key projects for the Raspberry Pi. It merges computer science, thanks to a solid base in the Ruby language, with music theory for impressive results in a short amount of time. Using Sonic Pi in live performances is now accessible to anyone, with refinements to the language and functions to handle complex tasks.

Coding as an art form has finally become reality, and that is all thanks to Sonic Pi.

Last word

Sonic Pi is sure-fire hit, adding yet another string to the bow of the Raspberry Pi. Merging music with code enables anyone to compose great-sounding audio using a simple language.

★★★★★

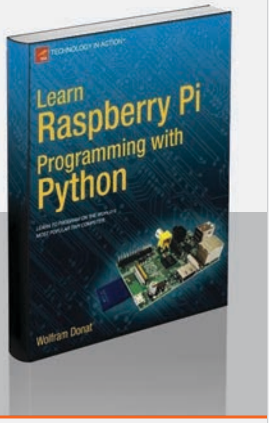
RASPBERRY PI BESTSELLERS

APRESS

Apress has several practical books for Pi users. Here's our pick from the most popular

LEARN RASPBERRY PI PROGRAMMING WITH PYTHON

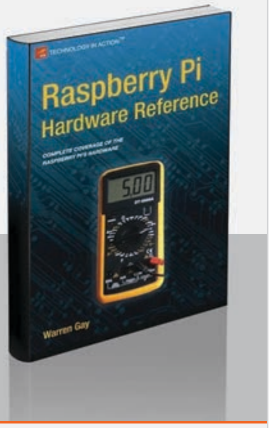
Author: Wolfram Donat
Publisher: Apress
Price: £17.50
ISBN: 978-1430264248
bit.ly/1LeMgN6



Project-based Python intro for the Pi: make a web spider, a weather station, a media server, a home security system, an underwater photography system, and a near-space weather balloon with a camera.

RASPBERRY PI HARDWARE REFERENCE

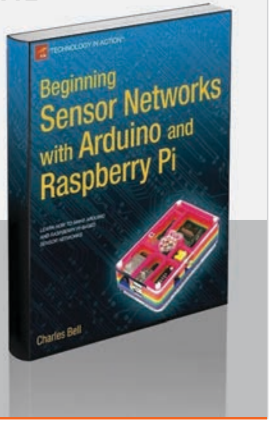
Authors: Warren Gay
Publisher: Apress
Price: £11.50
ISBN: 978-1484208007
bit.ly/1KWUipn



Delve deep into the Pi's workings for your sensor project: power, memory, USB, Ethernet, SD cards, I²C, GPIO, and the UART (serial) interface. With test code in C.

BEGINNING SENSOR NETWORKS WITH ARDUINO AND RASPBERRY PI

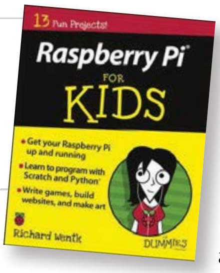
Authors: Charles Bell
Publisher: Apress
Price: £23.50
ISBN: 978-1430258247
bit.ly/1uu1HaB



Build a Raspberry Pi-based sensor node, then connect Arduino sensor nodes and XBee radio modules to the Pi, and collect data on a Pi-hosted MySQL database server.

RASPBERRY PI FOR KIDS

Author: Richard Wentk
Publisher: Wiley
Price: £21.99
ISBN: 978-111904951-7
bit.ly/1K7C41E



A well-paced introduction that goes from setting up to programming and debugging, embracing command line skills along the way. Fun software projects - like Turtles and *Minecraft* (controlling both with Python), simple and dynamic websites, and using a webcam - introduce many useful computing skills. Remembering the intended target audience, the inevitable setting-up chapters are both necessary and well judged, with details on cables and other information to enable younger makers to get going without adult help.

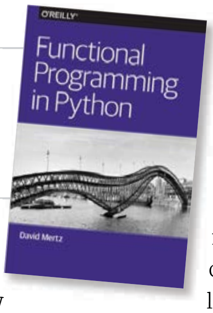
Ignoring the 'For Kids' branding, this book would be a valuable start for any nervous adult computer user keen to gain Pi enlightenment. From command line and nano, to Python, choosing a web server and learning programming skills, Wentk introduces and teaches using Raspbian. He does so in a clear style with gentle humour and an ability to make each learning challenge approachable.

PHP may be losing its lustre but it gives the quickest start in working with data, and is subsequently used for running shell and Python commands. Running webcam.py within PHP then leads on to debugging web problems, fixing Unix file permissions, and testing your software – all essential skills for growing your knowledge after finishing the book.

Score ★★★★★

FUNCTIONAL PROGRAMMING IN PYTHON

Author: David Mertz
Publisher: O'Reilly
Price: Free PDF download
oreil.ly/1KcIhgo



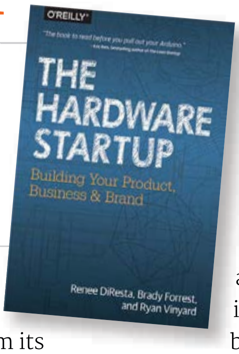
Functional programming "worries about what is to be computed rather than how it is to be computed," preferring the evaluation of expressions over statements, and treating functions as first-class objects (you can pass a function to another function as if it were a piece of data). Python, despite not being a functional programming language, is "a multiparadigm language that makes functional programming easy to do when desired, and easy to mix with other programming styles."

Right from the first chapter, '(Avoiding) Flow Control', the author (a Python Software Foundation director and sometime philosopher) brings you code that is subtly altered from imperative samples, with changes that won't alarm you but will work softly on your brain, cumulatively helping to shift focus from the 'how' to the 'what'.

"Simply changing the form of expression can often make a surprisingly large difference in how we reason about code and how easy it is to understand," says Mertz, as he uses list comprehensions, generators, and dictionaries and sets to abstract the 'how' and reorder our thinking on data, working up to lazy evaluation and higher-order functions (those that call or generate other functions). A good, quick introduction for all Pythonistas looking beyond OOP.

Score ★★★★☆

THE HARDWARE STARTUP


O'REILLY
THE HARDWARE STARTUP
Building Your Product, Business & Brand
Renee DiResta, Brady Forrest, and Ryan Vinyard

Author: Renee DiResta, Brady Forrest & Ryan Vinyard
Publisher: O'Reilly
Price: £23.50
ISBN: 978-1449371036
oreil.ly/1LRSDmb

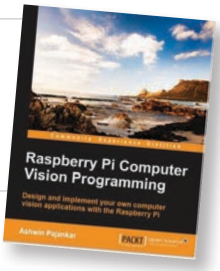
The hardware startup company of today is a very different beast from its antecedents, thanks to the very low cost of prototyping with boards like the Pi and Arduino, a new ecosystem for hardware manufacturing, including small manufacturing runs and 3D-printed parts, and the transfer of lean techniques from software to hardware businesses.

In *The Hardware Startup*, DiResta, Forrest and Vinyard provide a roadmap for anyone enticed by the lower cost of entry to transform their side project into a hardware business. From prototyping and manufacturing, to community engagement and legal concerns, and every aspect of developing and producing a product then getting it to customers, is considered in appropriate depth for those thinking seriously of turning a product idea into a business.

Open-source hardware is only touched on briefly, and selling your users' data is almost taken for granted, but the focus on a narrow, Silicon Valley view of what a startup is all about is unsurprising, given the smaller number of community and cooperative ventures in hardware than in software. However, the Raspberry Pi itself stands as a shining example of a social enterprise approach to running a successful hardware business.

Score ★★★★☆

RASPBERRY PI COMPUTER VISION PROGRAMMING


Raspberry Pi Computer Vision Programming
Design and implement your own computer vision applications with the Raspberry Pi
Ashwin Pajankar
PACKT

Author: Ashwin Pajankar
Publisher: Packt
Price: £16.99
ISBN: 978-1784398286
bit.ly/1Pdzlbf

Learning computer vision is best tackled with a practical approach: using the BSD-licensed OpenCV library, which enables image and video processing in C/C++, Java, and Python. Pajankar introduces OpenCV in a specific Pi/Python context but assumes no previous Raspberry Pi knowledge - fortunately, his Pi introduction is admirably brief. However, the OpenCV installation section is strangely lacking in refinement.

Python knowledge is assumed, but only the basics: NumPy is introduced early, for the mathematical functions needed in using OpenCV. Practical examples on images and webcam, using the Pi Camera Module, carry the less mathematical reader through. Noise removal and edge detection, useful tasks for your own Pi camera projects, are followed by motion detection and barcode reading. More real-life computer vision techniques are covered; there is a longer final chapter on the SimpleCV library, a framework to make computer vision tasks easier.

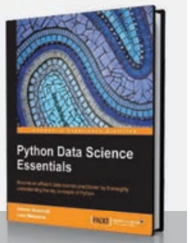
While many OpenCV books are now available, others focus on C++, Java, and cross-platform Python programming. We could be picky about the coding style but overall, for a Pi user with a little Python experience looking to get started with OpenCV and learn the basics of computer vision, this curate's egg of a book has much to teach.

Score ★★★☆☆

ESSENTIAL READING: APPLIED PYTHON

Not just for teaching & Pi projects, Python's a great language for real-world tasks of all sorts

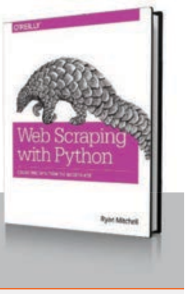
Python Data Science Essentials



Author: Alberto Boschetti & Luca Massaron
Publisher: Packt
Price: £26.99
ISBN: 978-1785280429
bit.ly/1bCAZEA

Direct and practical introduction to data analysis with Python, with good code examples and a problem-solving focus.

Web Scraping with Python



Author: Ryan Mitchell
Publisher: O'Reilly
Price: £21.50
ISBN: 978-1491910290
oreil.ly/1NkXYBu

Thorough grounding in the basics of web scraping, with balanced and concise coverage of a topic that continues to grow.

Mastering Python for Finance

Author: James Ma Weiming
Publisher: Packt
Price: £32.99
ISBN: 978-1784394516
bit.ly/1f32g4U

Comprehensive collection of financial theory and mathematical models applied to lucrative problems, with OOP and FP. Educational and interesting.

Black Hat Python

Author: Justin Seitz
Publisher: No Starch
Price: £23.50
ISBN: 978-1593275907
nostarch.com/blackhatpython

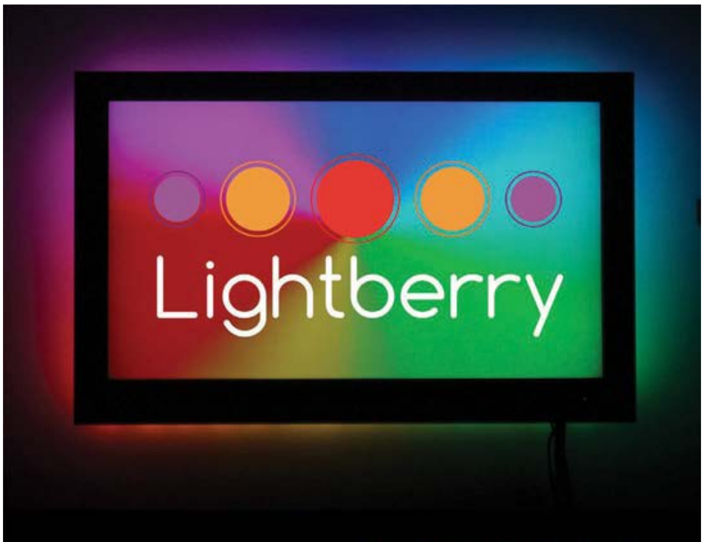
Penetration testing and security analysis, writing network sniffers, manipulating packets, infecting virtual machines, and creating stealthy Trojans in Python.

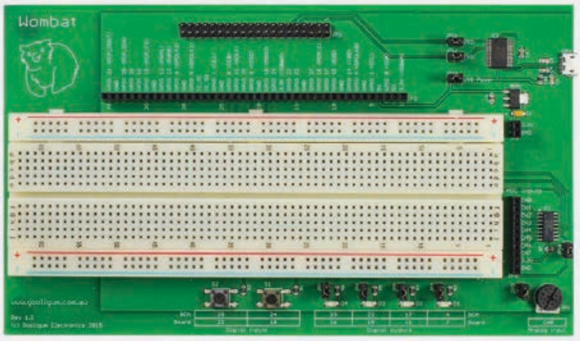
Learning Robotics Using Python

Author: Lentin Joseph
Publisher: Packt
Price: £29.99
ISBN: 978-1783287536
bit.ly/1IrlaeZ

"Robotics is an art." Python, ROS, and OpenCV, for building interactive autonomous mobile robots from scratch.

THE OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
AMAZING PI PROJECTS
10 imaginative creations that innovate and inspire
Win! £200 of robotics equipment
WINDOWS ON RASPBERRY PI
RASPBERRY PI DESKTOP SUPER
70 PAGES OF ESSENTIAL RASPBERRY PI
70 PAGES OF EXPERT ADVICE IN THE OFFICIAL RASPBERRY PI MAGAZINE


Lightberry


Wombat

MEDIA PI PLUS

RASPBERRY JAM EVENT CALENDAR

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

8 RASPBERRY PI DC MEETUP
1528 Connecticut Ave NW, Washington DC, USA
6 RASPBERRY JAM SILICON VALLEY
Computer History Museum, Mountain View, USA

PUT YOUR EVENT ON THE MAP

Want to add your get-together? List it here:

raspberrypi.org/jam/add

1 PRESTON RASPBERRY JAM

When: Monday 7 September

Where: Media Innovation Studio, Media Factory Building, Preston, UK

bit.ly/1KvamMh

Join in for a day of fun lightning talks, demonstrations, and hands-on time with the Pi.

2 CAMBRIDGE RASPBERRY JAM

When: Saturday 12 September

Where: Institute of Astronomy, Cambridge, UK

camjam.me

One of the biggest regular Jams in the home of Raspberry Pi. Excellent facilities for talks, demos, and workshops.

3 NORTHERN IRELAND RASPBERRY JAM

When: Saturday 12 September

Where: Farset Labs, Belfast, UK

bit.ly/1K5ihnJ

A regular gathering for everyone from complete beginners to advanced enthusiasts. All equipment provided.

4 TORBAY TECH JAM

When: Saturday 12 September

Where: Paignton Library and Information Centre, Paignton, UK

torbaytechjam.org.uk

Come along for a fun, informal day of hacking with your own kit, or use the equipment provided.

5 CODERDOJO HAM MINI-JAM

When: Saturday 12 September

Where: Ham Youth Centre, Ham, UK

coderdojoham.org

Aimed at 7 to 17-year-olds, varied activities include Raspberry Pi projects, Python programming, and Scratch.

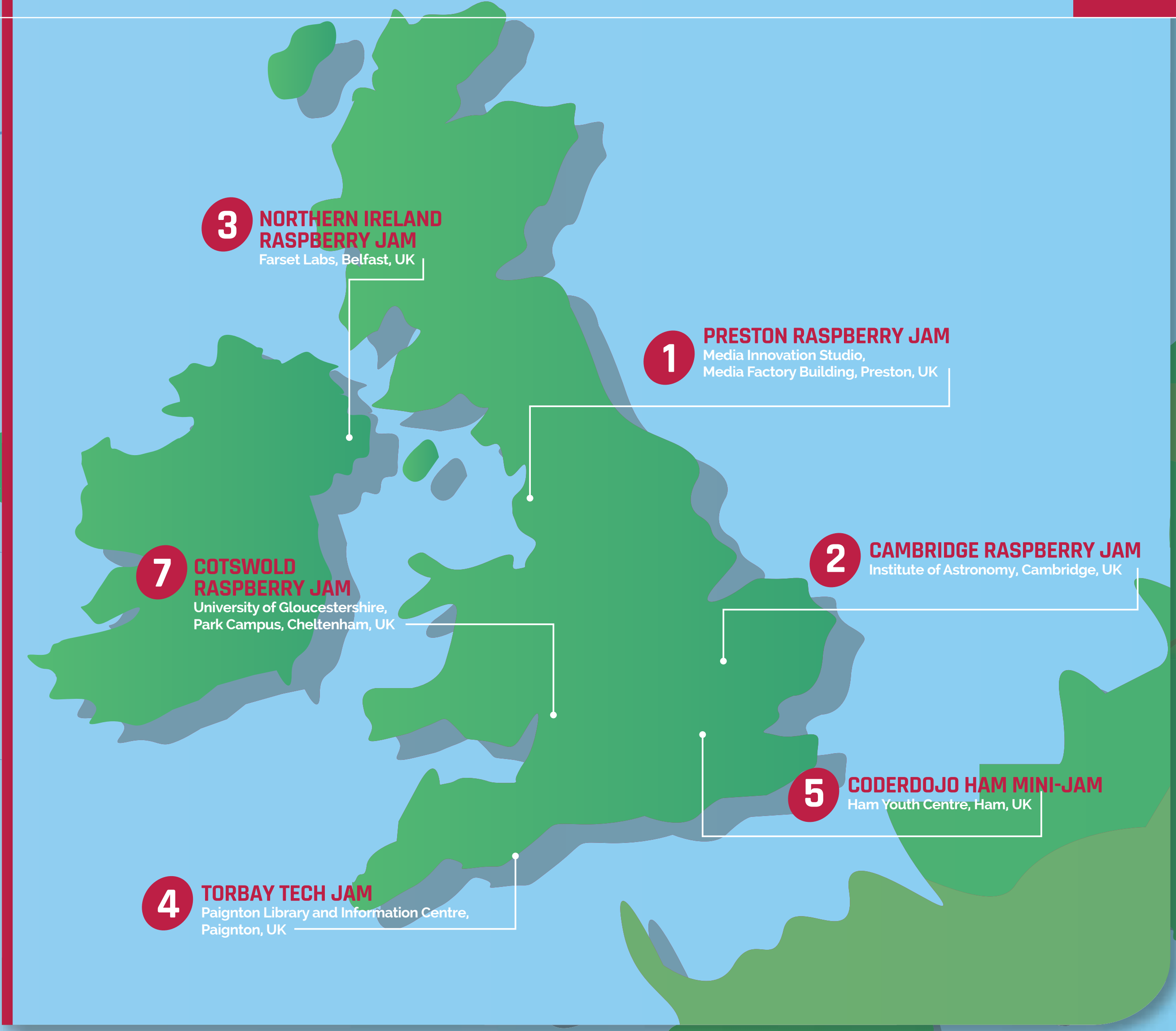
6 RASPBERRY JAM SILICON VALLEY

When: Saturday 19 September

Where: Computer History Museum, Mountain View, USA

bit.ly/1IKPPZ1

This regular monthly Jam welcomes Pi enthusiasts, educators, students, curious onlookers, and anyone else!


3 NORTHERN IRELAND RASPBERRY JAM
Farset Labs, Belfast, UK
1 PRESTON RASPBERRY JAM
Media Innovation Studio, Media Factory Building, Preston, UK
7 COTSWOLD RASPBERRY JAM
University of Gloucestershire, Park Campus, Cheltenham, UK
2 CAMBRIDGE RASPBERRY JAM
Institute of Astronomy, Cambridge, UK
5 CODERDOJO HAM MINI-JAM
Ham Youth Centre, Ham, UK
4 TORBAY TECH JAM
Paignton Library and Information Centre, Paignton, UK

7

COTSWOLD RASPBERRY JAM

When: Saturday 26 September
Where: University of Gloucestershire,
Park Campus,
Cheltenham, UK
cotswoldjam.org

A new venue will provide extra space for larger tutorial sessions and more attendees. Bring along your Pi.

8

RASPBERRY PI DC MEETUP

When: Monday 28 September
Where: 1528 Connecticut Ave NW,
Washington DC, USA
bit.ly/1J3kMTr

For everyone from masters to novices, talks will range from the most in-depth projects to intro-level concepts.

DON'T MISS: NORTHERN IRELAND RASPBERRY JAM

When: Saturday 12 September Where: Farset Labs, Belfast, UK

Hosted by PiNet creator Andrew Mulholland, this regular Jam features a wide range of fun activities to try out, including getting started with Raspberry Pi, electronics, Python, hacking Minecraft Pi, exploding balloons, and creating an automated parent detector! Andrew says, "We always have plenty of beginners at each Raspberry Jam, usually after parents buy them Raspberry Pis and want a hand getting them started." Doors open at 1pm. For details, visit **bit.ly/1K5ihnJ**

TAKE US ANYWHERE

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
GET STARTED WITH RASPBERRY PI
MINECRAFT SPLAT
THE OFFICIAL RASPBERRY PI MAGAZINE
AMAZING PI PROJECTS

SAVE 45% with a Newsstand subscription (limited time offer)

The MagPi Magazine

Available now for smartphones & tablets

Available on the App Store

Get it on Google play

From just

£2.29 rolling subscription or **£19.99** full year subscription

Why buy digitally?

- Get all copies back to issue 31
- Instant downloads every month
- Fast rendering performance
- Live links & interactivity

In association with:
raspberrypi.org

10 The MagPi DIGITAL SUBSCRIPTIONS MUST BE WON!

WHAT IS YOUR FAVOURITE TABLET DEVICE?

Tell us by 25 September for your chance to win!

70 PAGES OF ESSENTIAL RASPBERRY PI KNOWLEDGE & KNOW-HOW
The MagPi
Win! 10 RasPi Model A+s
The official Raspberry Pi magazine
Issue 32 Apr 2015
raspberrypi.org/magpi
BUILD A STROBE LIGHT
Get the party started with a Pi
ANIMATE GRAPHICS IN PYTHON
How to make your Pygame projects move
FACIAL RECOGNITION MADE EASY
Using OpenCV and the camera module
RASPBERRY PI WEATHER STATION
The Foundation & Oracle team up for giveaway
CROWDFUNDING'S GREATEST HITS
The biggest projects powered by Raspberry Pi & funded by you
Also inside:
UNICEF PUTS PIS IN LEBANON
CONQUER THE COMMAND LINE
MAKE AN ELECTRONIC DOOR LOCK
MORE OF YOUR BRILLIANT PROJECTS
NATURE BYTES!
Get closer to nature with this awesome wildlife camera kit
Plus SONIC PI: MAKE AMAZING MUSIC WITH OUR ESSENTIAL TIPS & TRICKS

How to enter:

All you have to do is email **competition@raspberrypi.org** letting us know what tablet device you use, or that you're considering buying in the next six months.

Terms & Conditions

Competition closes 25 September 2015. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don't like spam. Participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.

HOW DO YOU INSPIRE CHILDREN INTO STEM?

The Raspberry Pi Foundation's **Clive Beale** tackles this month's big question...

Once upon a time, someone in an education meeting got really bored of having to say 'science, technology, engineering, and maths' all the time. After 20 minutes of serious thinking, and many failed ideas (METS, TEMS, SMET...), they invented the acronym 'STEM'. If you Google 'stem', you don't get pages about plants but lots of serious chatter about how important STEM is and why more people (especially young people) should do it. And it *is* important – it tells us how the world works and allows us to change it. STEM underpins and informs modern life, from cancer cures to Facebook; from space exploration to robot vacuum cleaners.

So far, so functional, but it's not the best way to inspire most children. Horror tales of the olden days when the only social media was graffiti in public toilets, and your mum issuing you with an emergency 2p piece to call home just don't cut it. Fortunately, we have a secret weapon. STEM, as well as being important, is exciting, creative, challenging, and fun. So how *do* you get kids into STEM? All you need is some inspiration and a Raspberry Pi.

Below Buttons, LEDs, and even the Pi camera are easily controlled in Scratch

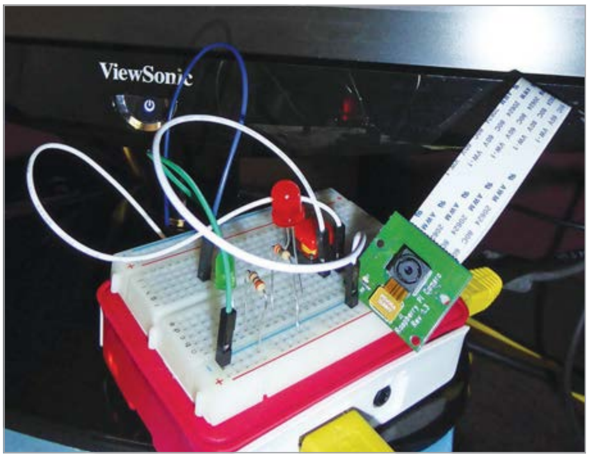


Image: Clive Beale

Hack your hobby

If you are not interested in something, then the bad news is that STEM won't make it more fun. Anything that you do enjoy, however, can be exponentially expanded, extended, and improved when you throw a little STEM into the mix. It doesn't matter whether that interest is LEGO, music or knitting: there's always a way to hack your hobby with a Raspberry Pi and inject some STEM into it. And if your hobby is already STEM-based (astronomy, robotics, cold fusion), then a Pi will make it bigger, better, and faster.

Start early

There's a saying in computing education: 'age eight is too late'. While the age is arbitrary (based as it is on rhyming with 'late'), the concept is sound and the sooner kids get into STEM the better. Realistically, this means starting at home. Once children see that they can make a computer do things there's no stopping them, and the ideal tool to foster this is the programming language Scratch.

That's the whole of STEM covered by a single program and accessible to preschool children. The physical computing aspect is particularly inspiring and a powerful way to learn. Which brings us to my favourite way to inspire kids in STEM...

Get physical

School tech departments once rang out to the happy sounds of children setting fire to their shoes with molten aluminium and welding the teacher's chair to the ceiling. Sadly, this is now frowned upon. But humans are curious and creative animals, and physically manipulating the world around us is compulsively engaging. STEM scratches that evolutionary itch and the Raspberry Pi makes it accessible.

Physical computing covers everything from flashing an LED to sending a Pi into near space. It's really cheap to get started and you can make surprisingly brilliant things for very little outlay. It's also a chance to repurpose things you've got lying about the house – those Wii Remotes in the kitchen drawer, or the old remote-control car under the bed. Break things and then rebuild them better. Break things and make something else. If you live in a weird minimalist box of glass and steel and have no junk, then ask your school or relatives. Or salvage stuff – just remember to ask the owner before you root around in their skip. And don a titanium exoskeleton for health and safety reasons.

Image: Michael Horne

Above Hacking *Minecraft* at the Cambridge Raspberry Jam

Join in

The advice here is simple: get involved with the community, as it's a terrific multiplier in terms of skills and confidence. Whether you need technical advice on a project, are looking for someone to collaborate with on your latest madcap scheme, or are just plain flummoxed, then you are in luck because the Raspberry Pi community is one of the most helpful and friendliest around. The official forum (**raspberrypi.org/forums**) has experts in all of the STEM fields and any questions are usually answered within minutes. If you have an idea for a STEM project but are not sure where to start, then this is the place to ask. Someone has probably tried it before and would be happy to share their experiences.

Of course, the community extends way beyond the forums to include Raspberry Jams, competitions, and community resources such as *The MagPi* that you are reading now (**raspberrypi.org/community**). A lot has been written about Raspberry Jams, but simply they are one of the best places for young people and families to learn, get involved, have a go, and collaborate on all things Pi. There's only one thing better than building a giant, Raspberry Pi-controlled, evil robot zombie and that's challenging other such robots to a duel. With death lasers.

Competitions are a great way to get schools involved in STEM and we've run several in the past. We like to give away cool things, so keep an eye on the blog (**raspberrypi.org/blog**). There's also the fabulous PA Consulting competition which showcases STEM in a variety of themes (**paconsulting.com/events/raspberry-pi-competition**).

Go and play

Although there's no single way to inspire children with STEM, there is a simple formula and that is to just get stuck in, mess around, and make things. STEM is an amazing, useful and empowering thing, but it can be daunting when you first set out. The Raspberry Pi is a great way to start to explore these subjects in creative, fun and exciting ways and when you do that, STEM happens by stealth. Go and play. When you do, things like this happen...

Image courtesy of: Dave Akerman and St Alban's School

NEW! YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Win! 10 RASPI 2s THANKS TO ELEMENT14.COM
100 pages of hacking & making
Issue 36 August 2015
raspberrypi.org/magpi
GET STARTED WITH RASPBERRY PI
How to unbox your credit card-sized PC and start hacking today!
RETRO CLASSICS
The very best games of yesteryear you can bring back to life...
SIMULATE PHYSICS WITH PYTHON
Create a whole solar system on your Raspberry Pi
3D PRINTING WITH RASPBERRY PI
Control and view your printer with OctoPrint
CREATE A START SCREEN IN SCRATCH
How to put the finishing touches to your latest creation
Also inside:
CONQUER THE COMMAND LINE
BUILD YOUR OWN MOOD LIGHT
MAKE A SIMON SAYS ELECTRONIC GAME
IQAUDIO: THE ULTIMATE AUDIO ADD-ONS
MINECRAFT SPLAT
Recreate Nintendo's brilliant Splatoon on your Raspberry Pi
THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY

Competition Winners

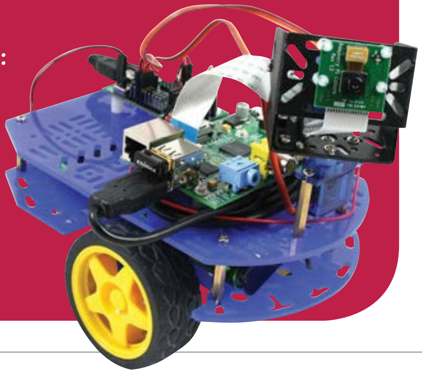
Back in issue 35, we asked you to you to let us know what you hack and make with your Raspberry Pi. In return for an emailed response to competition@raspberrypi.org, our friends at dawnrobotics.co.uk kindly offered to give away their top-of-the-range Raspberry Pi robot worth £150 to one winner, and four Dagu Mini Driver MKII boards worth £14 to the runners-up.

Here are the winners, who will all be hearing from us soon!

DAWN ROBOTICS

Star Prize winner:
Paul Smith

Runners-up:
Theo Frieling
Ivan Smerznak
Jason P
Paul Brown



Article submission

I have a Raspberry Pi project that I'd like to share. Briefly, we built a daughter board that plugs into the GPIO pins on the Raspberry Pi. The board is designed to charge and discharge a battery using a user-defined constant current (10 microamps to 10 milliamps) and a user-defined voltage range (e.g. 3-4.2V for Li-ion; 0-5V possible). The circuit records voltage, time, current and temperature into a text file, as the cell is cycled.

This circuit is very useful for people testing batteries (and a hundred times cheaper than a similar circuit), or hobbyists who want to know more about battery performance.

Would it be possible to submit an article to your magazine based on this circuit? If so, how would I go about doing that?

Shane Beattie

Thanks for getting in touch with details of your excellent project, Shane! We can imagine hundreds of applications for your add-on board and I'm sure lots of our readers can too. You'd be very welcome to submit an article for us to include in the magazine – looking forward to taking a look! If any other readers have ideas for tutorials, features or reviews to go in the magazine, all they need to do is email us at **magpi@raspberrypi.org** with their idea and a short example of any writing they've done in the past. Once we've had a brief conversation about the article (usually via email or VoIP), we'll give you a text document that outlines our requirements (images, text format and so forth). It really is that easy to feature in the magazine!

Subscription options

I am a Tunisian engineer who is very interested in this magazine. I just want to know that if I subscribe to your magazine for one year, will I get each issue monthly? Also, do you ship to this part of the world, and if so, do you ship directly to my door or just to post offices?

Oussema Hidri

Subscribing to the magazine is easily the cheapest and most convenient way to secure your issue every month. Not only can you save up to 25% on the cover price, we'll deliver your copy to your door pretty much anywhere in the world. We hope that answers your questions, Oussema. Just visit **bit.ly/MagPiSubs** or check out pages 26-27 for more details! The magazine is available for £12.99 quarterly (by Direct Debit in the UK), six-monthly, and yearly.

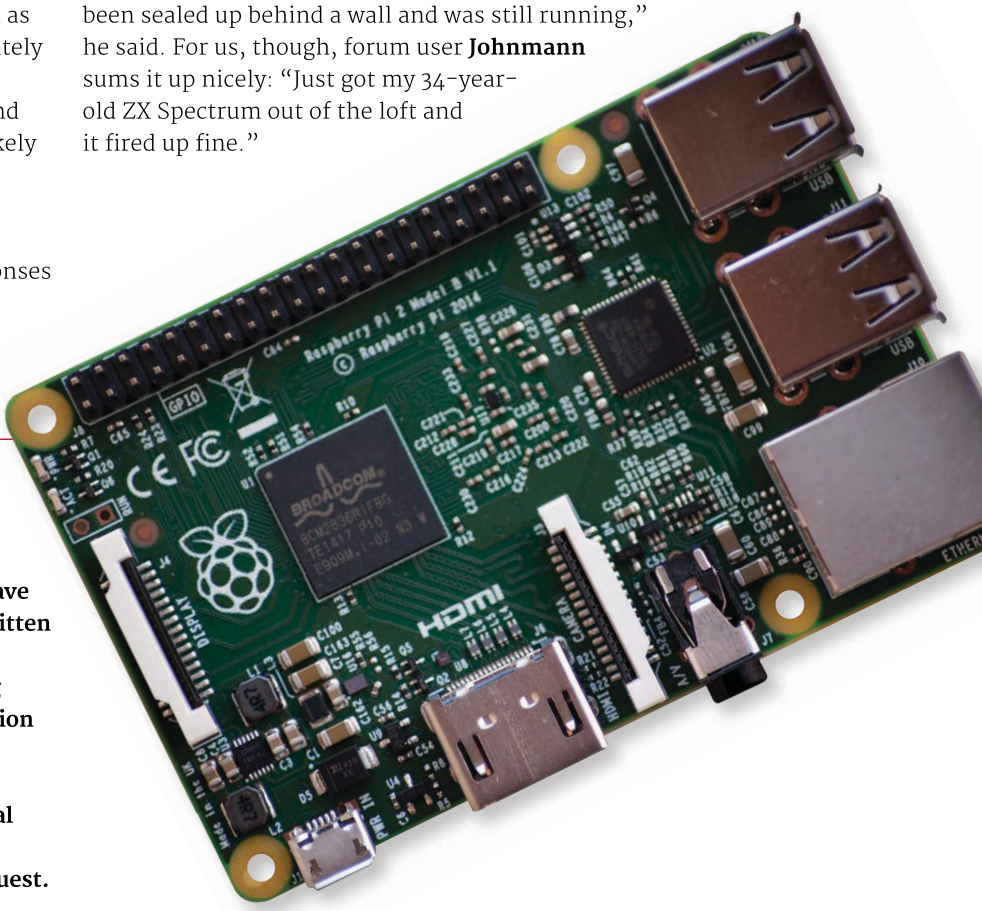
FROM THE FORUM: BLAST FROM THE PAST

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

My Raspberry Pi 2 has been running as a web server 24 hours a day, seven days a week for the past three months or so, and I was using a laptop for a month before that. So far, it has received 1,000 visits during that time. How much longer do you think it will last?
Gigapouch

It's generally Sod's law that dictates that as soon as you ask how long something will last, it immediately falls over, never to work again. That said, the Raspberry Pi is actually one of the most robust and long-lasting bits of kit in your house. It's very likely that it will outlast your mobile phone, laptop or classic platter-style hard drive.

Our opinion was very much reflected in the responses forum user Gigapouch received in the thread. **W H Heydt** very much expects his Rasperry Pi to outlast him, while **rpdom** says it's likely a Raspberry Pi 10 will be available long before the Pi 2 gives up the ghost. Taking it much further, **Bobstro** suggests that a well-designed system such as the Raspberry Pi should be able to run more or less indefinitely, as long as it has power. "There is a story about a Novell NetWare server (**bit.ly/1J3y3xl**) that ran for years with no downtime. After four years, they decided to track it down, only to find that it had been sealed up behind a wall and was still running," he said. For us, though, forum user **Johnmann** sums it up nicely: "Just got my 34-year-old ZX Spectrum out of the loft and it fired up fine."



External storage tutorials

Although I've had my Raspberry Pi for several months now, I'm still very much a newbie and have greatly appreciated the *MagPi* articles that are written primarily for beginners.

One article that I found particularly interesting was 'External Storage, Part I: File systems, partition tables and rsync' by W H Bell, which appeared in issue 29, December 2014. Please consider this note as a request for more articles in the 'External Storage' series.

Many thanks for your consideration of this request.
Gordon Atkins

Thanks for getting in touch, Gordon. We're glad you enjoyed Will Bell's piece from the end of last year. As you'll see from the tutorial on pages 64-67, Will is back creating content for the magazine. We'll have a word with him and see if we can convince him to do more tutorials on external storage for a forthcoming issue. Thanks for reading!

WRITE TO US

Have you got something you'd like to say?
Get in touch via **magpi@raspberrypi.org** or on The MagPi section of the forum at **raspberrypi.org/forums**

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

TAKING PI TO NEW HEIGHTS

Why is a Raspberry Pi in space so exciting? **Matt Richardson** explores the universe of possibilities

Orbiting earth at 27,600 kilometres per hour, the International Space Station is home to six crew members working on scientific research in a microgravity environment. By the end of this year, there will be two Raspberry Pis joining them on board. There are a lot of reasons why this is significant and exciting. In fact, sometimes when I'm reading an update about the mission on the official Raspberry Pi blog, I shake my head in disbelief.

On board the International Space Station, the Raspberry Pis will run programs designed and written by British primary and secondary school students. It's all a part of our Astro Pi competition, a partnership with the European Space Agency and the UK Space Agency. To help students with their on-board experimentation, each Raspberry Pi will be outfitted with a camera and a Sense HAT add-on board, which has sensors for position, movement, temperature, pressure, and humidity. In order for the crew to interact with the students' creations, there's also an 8 by 8 pixel RGB LED display, a joystick, and a few buttons.

It's exciting that Raspberry Pis are being launched to operate on the International Space Station. But, as I've learned in the past few months, getting a Raspberry Pi up there is no small feat. Long before it will be loaded onto the Soyuz rocket to be launched into space with astronaut Tim Peake, the Raspberry Pi and its accessories must undergo a battery of tests in order to receive its certification to be operated on board the space station. I'm excited to know that our own Raspberry Pis here on Earth are space-flight certified!

Even more exciting than seeing Raspberry Pis on the International Space Station is how much it can motivate and inspire young kids to learn about computer science. As I've written before, we shouldn't only teach coding for the sake of coding. We should think of coding as a means to accomplish something that interests students. The science of space is such a captivating subject matter for so many students and this is a great opportunity for them to code with purpose. I remember myself as a kid being incredibly interested in space travel. I could imagine the younger version of me being captivated and challenged by the Astro Pi competition.

Code in space

For the winners of the competition, it's also an incredible opportunity for them to run their own code on the ISS. According to **Space.com**, there are 3.3 million lines of code in ground computers and 1.8 million lines of flight software code, all of which must be carefully written and tested. Not just anyone can write code that runs on the ISS, because it's so critical that everything works properly. The Raspberry Pis on the ISS will be isolated from the rest of the life-supporting hardware on board, acting as a sandboxed environment for running code on the station. It's a unique opportunity for kids to have their code run safely in space.

It's a sandbox that can be replicated down here on Earth at the low cost of a Raspberry Pi. I'm sure that most of the other flight hardware on the International Space Station carries a hefty price tag. But now it's affordable to own the very same hardware that will be orbiting Earth by the end of the year.

For all these reasons, it's exciting to see a Raspberry Pi go to space. If you're anything like me, you can see each and every Raspberry Pi as a computer with enormous potential. It's incredible to see the potential of Pi now going beyond the Earth's horizon.