GET A CAMERA MODULE FIRST WITH YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

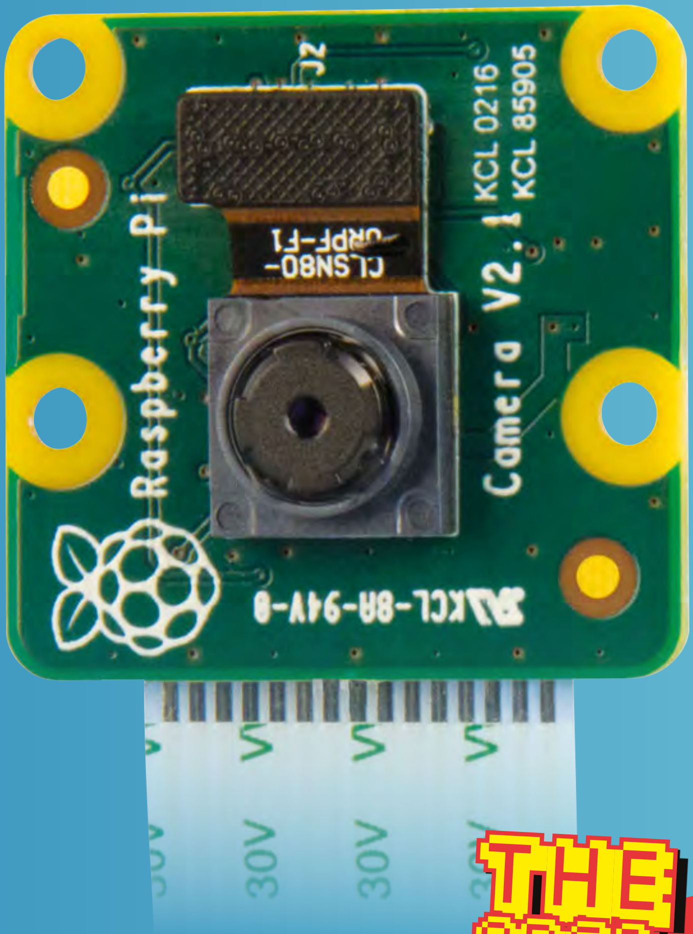
The official Raspberry Pi magazine

Issue 45 May 2016

raspberrypi.org/magpi

PICTURE PERFECT

The **NEW** Raspberry Pi Camera Module is here!



REPLICATE AN ASTRO PI EXPERIMENT

Make your own humidity sensor

MAKE NEOPIXEL ELECTRONICS

Build impressive costume effects

BUILD AN IOT THERMOMETER

And share your data online

A PI ZERO-POWERED RAILWAY

Pi takes a Dorset model railway to the next level

HACKING WITH DINOSAURS

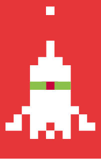
Jurassic Park, eat your heart out!

THREE NEW PI GAMES TO PLAY

From GameMaker: Studio

THE BEST PI ARCADE MACHINES

Our top 10 retro gaming arcade cabinets powered by Pi



Also inside:

- > CODER DOJO GETS A MASSIVE BOOST
- > TAKE AMAZING PICTURES OF THE MOON
- > THE ORACLE WEATHER STATION ARRIVES
- > THE LATEST GADGETS REVIEWED & RATED

Issue 45 • May 2016 • £5.99

05

9 772051 998001

THE ONLY PI MAGAZINE WRITTEN BY THE READERS, FOR THE READERS

WELCOME TO THE OFFICIAL PI MAGAZINE!

The Raspberry Pi Camera Module is one of the most popular add-ons for the Raspberry Pi. Released in 2013, it made use of the previously dormant CSI connector, spawned a whole new genre of amazing Raspberry Pi projects, and has since found itself everywhere from the depths of the ocean to the dizzy heights of the International Space Station.

Its 5MP sensor was equivalent to the kind of standard cameras you'd find on smartphones of the time, but since three years is an eternity in technology, Raspberry Pi have just released a shiny new update. You can learn all about the new camera starting on page 20, where we speak to its creators about its new specification, which includes a rather splendid 8MP Sony sensor. While the means of taking pictures and video remains unchanged from the original Camera Module, we've included everything you need to do to get started, as well as a number of fun and useful projects to get you snap-happy in no time at all.

If you're a subscriber, you'll already have received a special code that entitles you to skip to the front of the purchase queue at **ThePiHut.com**. Not yet a subscriber? It's not too late to make use of this fantastic offer – see page 34 for details!

Russell Barnes
Managing Editor

PROUD SUPPORTERS OF: NATIONAL STEM CENTRE

SUBSCRIBE & SAVE 25%
SEE PAGE 34 FOR DETAILS

THIS MONTH:

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

The MagPi

Available on the App Store

GET IT ON Google Play

CC BY NC SA

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

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Mike Kay**
Cover photography: **Gareth Halfacree**

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, Dan Aldred, Mike Cook, Lucy Hattersley, Richard Hayler, Phil King, Willem Koopman, Ross O'Neill, Matt Richardson, Ben Rogers, Jon Silvera, Richard Smedley & Francesco Vannini

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

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

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., 30 Station Road, Cambridge, CB1 2JH. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2051-9982.

The MagPi
PICTURE PERFECT
The NEW Raspberry Pi Camera Module is here!
THE BEST PI ARCADE MACHINES

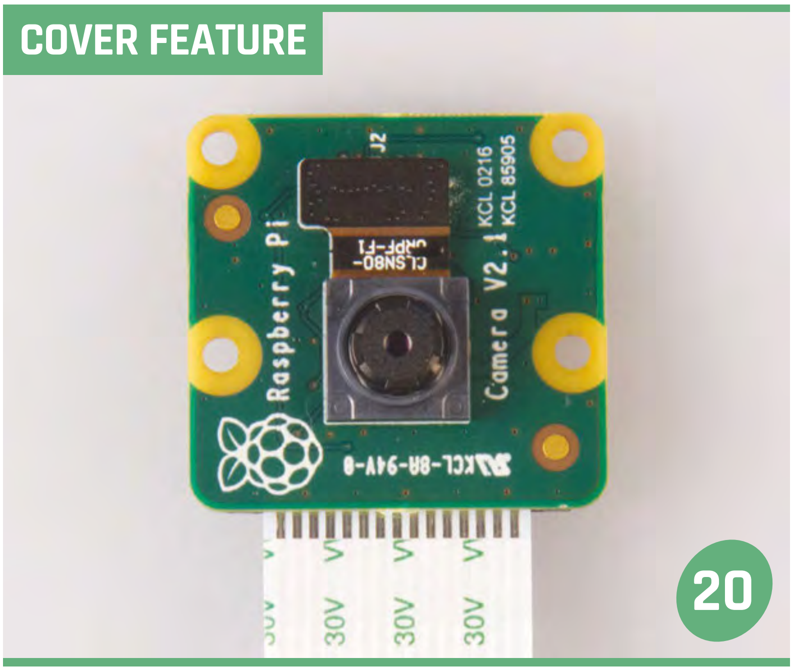
Contents

Issue 45 May 2016 raspberrypi.org/magpi

TUTORIALS

IN THE NEWS

COVER FEATURE



THE BIG FEATURE

YOUR PROJECTS

REGULARS

COMMUNITY

REVIEWS

ASTRO PI CODING CHALLENGE RESULTS ARE IN!

What music will Tim Peake and the rest of the ISS crew be listening to in space?

The Raspberry Pis may already be up in space performing experiments, but it doesn't mean the end for the coding portion of Astro Pi. Over the last few months, some intrepid young coders have been creating music and MP3 players to send up to the Astro Pis orbiting Earth, and recently it was finally time to judge the efforts of these amazing young people from around the country.

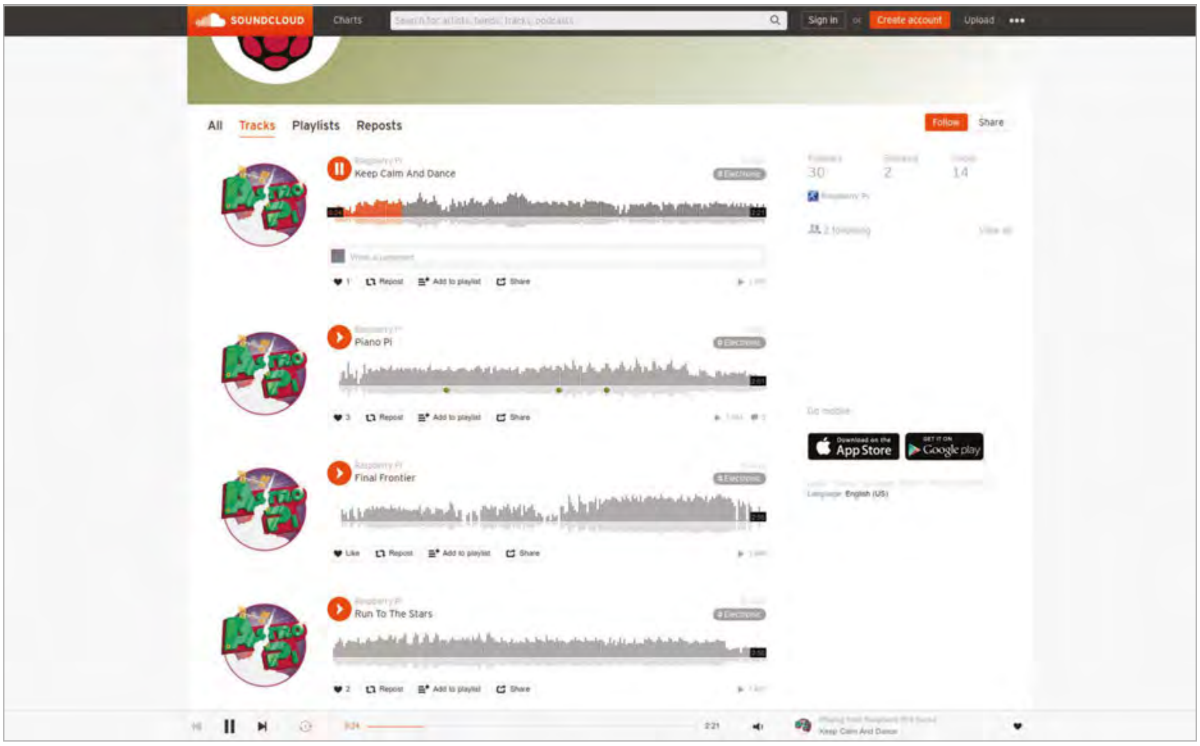
Judging was split over two days, with the MP3 players chosen first after an all-day session, and the Sonic Pi tracks compared on the second full day. Among the judges were members of the ESA, the Royal Institute, and even Andy McCluskey and Paul Humphreys from British new wave band Orchestral Manoeuvres in the Dark. The judges on both days were accompanied by Dave Honess of the Raspberry Pi Foundation, who has been at the core of Astro Pi from the very beginning.

"The judging was great fun," Dave tells us. "This competition differed from the previous, more open-ended one, in that students were given a specific brief to code to. The MP3 player challenge gave rise to lots of different approaches to the same task, which was really interesting to see. Where do you put the play/stop buttons? How does the heads-up display work? Do you put waveforms and VU meters in for fun? Some of them were really complex to operate, which shows that coding a program that's simple and elegant is not as straightforward as it sounds. The ones that won all nailed the simplicity and functionality in exactly the right place.

"The Sonic Pi tunes were really good, too. Many of them were simple beats and electronic sounds with a Kraftwerk-cum-Giorgio [Moroder] vibe. We had one song that was composed almost entirely of burp samples, which had us all in stitches! Shout out to Anchorsholme Academy Code Club for that one. My favourite track, though, was Piano Pi from Bartholomew Secondary, which had an old-school Ace of Base Nineties sound."

AND THE WINNER IS... CATEGORY: MP3 PLAYER

AGE GROUP:-	11 AND UNDER	11 TO 13	14 TO 16	17 TO 18
	Jude Young	**Lowena Hull**	**Joe Speers**	**Marcus Panchal**
	SCHOOL: Clifton College Preparatory School	SCHOOL: Portsmouth High School	SCHOOL: N/A, independent entry	SCHOOL: Lancaster Royal Grammar School
	SUPERVISOR: Andrew Young	SUPERVISOR: Gary Barrows	SUPERVISOR: Craig Speers	SUPERVISOR: Sanjay Panchal
	magpi.cc/1Vjumff	**magpi.cc/1VjvXlm**	**magpi.cc/1Vjw0xF**	**magpi.cc/1Vjw4gH**



Left Listen to the winning tracks on the Raspberry Pi SoundCloud account

Below The judging for the MP3 player lasted a whole day; same with the songs the following day

To ensure fairness, the entries were anonymous when presented to the judges, and a special scoring system was used that was then aggregated to reveal the winners. They were then contacted to let them know the good news.

> The MP3 player challenge gave rise to lots of different approaches to the same task

"We phoned them all before the announcement was made on our blog," Dave continues. "The best reaction I had was Jude Young from Clifton College Prep in Bristol, who sounded like he was doing a million fist pumps per minute. Having your code flown to the International Space Station and used by an actual astronaut is quite a profound thing for a young coder! We've seen, in both competitions, that this is the single biggest motivating factor for participation in Astro Pi. So it's basically gone down as a huge confidence boost for them, and rightly so."

The music from all the entrants is available on SoundCloud if you want to dance along to some space tunes (see links below). Dave assures us there will be more competitions to come in the future as well. So if you missed out this time, don't lose hope.

CATEGORY: SONIC PI MUSIC

AGE GROUP:-	11 AND UNDER	11 TO 13	14 TO 16	17 TO 18
	Iris and Joseph Mitchell	**Isaac Ingram**	**Jamie Andrews**	**Pinal Parmar**
	SCHOOL: Southend Raspberry Jam	SCHOOL: Knox Academy	SCHOOL: Bartholomew Secondary School	SCHOOL: Featherstone Sixth Form
	SUPERVISOR: Andy Melder	SUPERVISOR: Karl Ingram	SUPERVISOR: Jason Andrews	SUPERVISOR: Beena Parmar
	magpi.cc/1VjweVA	**magpi.cc/1Vjwl3c**	**magpi.cc/1VjwnrS**	**magpi.cc/1VjwoME**

Coolest Projects is an annual event which showcases digital projects, created in Dojos globally, in one unique space

CODERDOJO FOUNDATION BENEFITS FROM VMWARE FOUNDATION SUPPORT

The VMware Foundation has donated a total of 960 Raspberry Pis to CoderDojo to be distributed to Dojos around the world. **Ross O'Neill** tells us more…

MARY MOLONEY

GLOBAL CEO, CODERDOJO

"The very generous donation by VMware of over 900 Pis to kids in the CoderDojo community was very gratefully received. The Pis have been distributed to kids aged between 7 and 17 around the world, enabling them to explore, create, and build with a fun new device. Raspberry Pis are cool and easy to use; the kids have adopted them really easily, and have incorporated them into projects they're working on. Introducing kids to all possible forms of technology taps into their natural curiosity and creativity, and we have experienced just that with the sharing of the Pis funded by VMware!"

CoderDojo is a global movement of free, volunteer-led, community-based programming clubs for young people. At a Dojo, youngsters between the ages of 7 and 17 learn how to code; play with Pis; develop websites, apps, programs, and games; and explore technology in an informal and creative environment. In addition to learning to code, attendees meet like-minded people and are exposed to the possibilities of technology. Within the CoderDojo movement there is a focus on community, peer learning, youth mentoring, and self-led learning, with an emphasis on showing how coding is a force for change in the world. The global CoderDojo community is supported by the CoderDojo Foundation.

The CoderDojo movement was originally founded by James Whelton and Bill Liao. It all started in James Whelton's school in early 2011, when James (then an 18-year-old coder) received some publicity after hacking the iPod Nano. As a result, some younger students expressed an interest in learning how to code. James set up a computer club in his school, where he started teaching students basic HTML and CSS. Later that year he met Bill Liao, an entrepreneur and philanthropist, who was interested in growing the project into something bigger.

James Whelton co-founded the CoderDojo Foundation when he was just 18 years old

How did it begin?

Back in July 2011, the first CoderDojo was launched in the National Software Centre in Cork, Ireland. The Dojo was so popular that people frequently travelled over 250km from Dublin to attend sessions. Owing to this popularity, James and Bill open-sourced the model and encouraged growth. Soon afterwards, a Dojo in Dublin was launched, followed by more clubs across Ireland, before expanding outside the region. Through making the movement open-source, thousands of dedicated Champions (those who set up Dojos) and Mentors have created more Dojos across the world, making the CoderDojo movement a global phenomenon. As of April 2016, there are now over 900 verified Dojos in 65 countries, and the number is growing all the time. The CoderDojo Foundation consists of a core team based primarily in Dublin. Founded by the co-founder of the very first CoderDojo, James Whelton, the Foundation is focused on supporting new and existing Dojos through resource and community development, while also scaling CoderDojo through partnerships and raising awareness globally.

How will Pis help Dojos?

Raspberry Pi is a capable little device that enables people of all ages to explore how computers work and to learn how to program in languages like Scratch and Python, two of the most popular languages taught at Dojos. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video to playing games. What makes the Raspberry Pi so popular within a Dojo context, however, is that it has the ability to interact with the outside world, and has been used in a wide array of digital projects. On top of this, the CoderDojo Foundation has created content within its Sushi Card series specifically for Dojos using Raspberry Pis.

The VMware Foundation has donated a total of 960 Raspberry Pis to CoderDojo, to be distributed to active Dojos around the world. So far, the CoderDojo Foundation has sent out 480 of these to Dojos globally, with the rest being shipped this coming month. It's hoped that all these Raspberry Pis will encourage kids to program. They will also be used to create digital projects for the upcoming 'Coolest Projects' event at the Royal Dublin Society, where an expected 10,000 attendees will be visiting this June, all in the name of having fun with code.

> "Created content specifically for Dojos using Raspberry Pis"

Above The Docklands Dojo in full swing at Dogpatch Labs, a Dublin co-working space where the CoderDojo Foundation is based

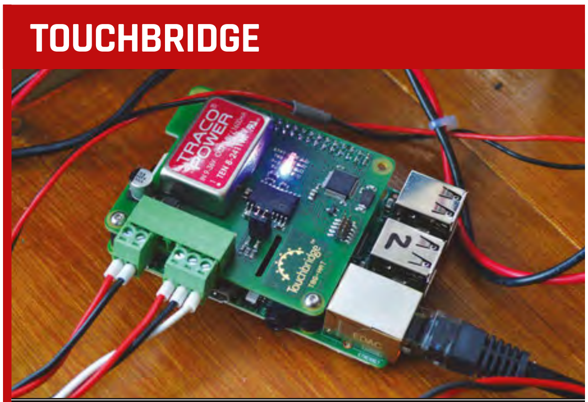
How do I join?

It's super-easy to either create your own Dojo or join an existing one near you. Just visit **CoderDojo.com** or email **info@coderdojo.com**

HACKING WITH DINOSAURS

British amusement park Blackgang Chine wanted to improve its Pi-powered dinos, so **Dr Lucy Rogers** put on a conference to do just that

TOUCHBRIDGE

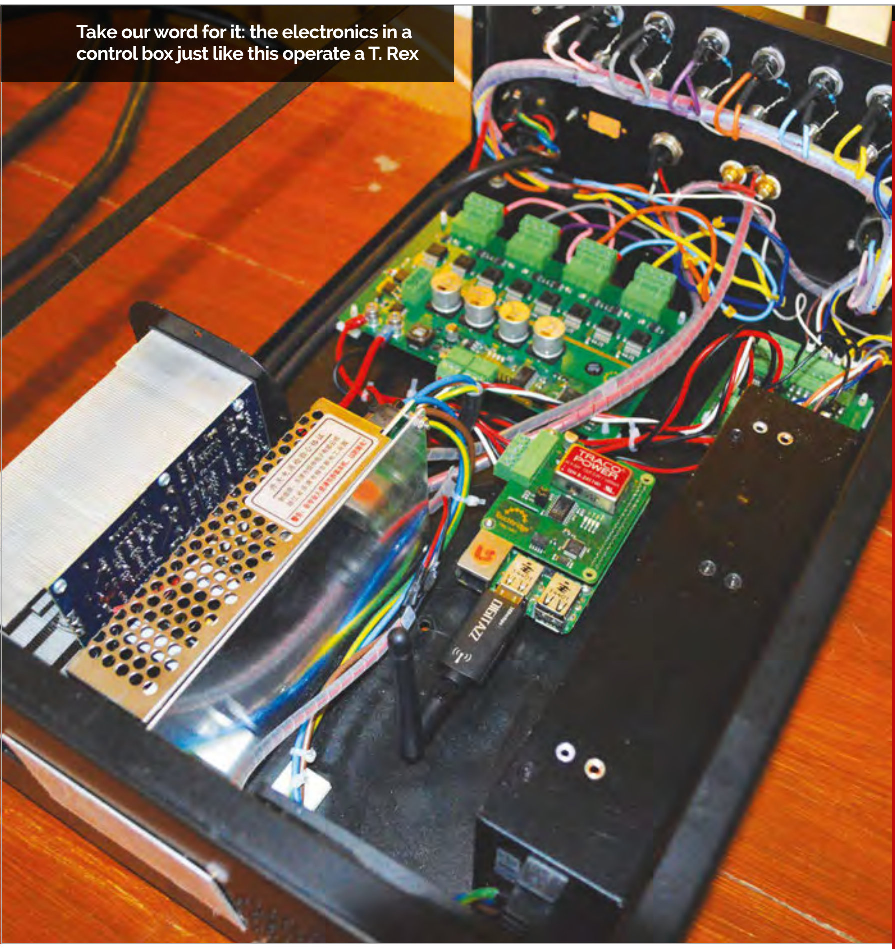


One of the pieces of tech that makes all of the animatronics and such compatible with the Raspberry Pi at Blackgang Chine is Airborne Engineering's Touchbridge system, a HAT for the Pi which can control additional hardware buses. This allows the Raspberry Pi to control 24-volt, 16-amp motors. Even though it only outputs 3V3 at 10 milliamps at the GPIO. 62 buses can be daisy-chained together, each with eight inputs or outputs, so you can control a fair number of motors with the right code and power requirements.

You may recall a video from a couple of years ago, of Dr Lucy Rogers showing you around Blackgang Chine's dinosaurs (**blackgangchine.com**), which had recently become powered by a Raspberry Pi or two. It was a great video that captured the imagination and showed just what the Raspberry Pi could do. Now, in 2016, the Raspberry Pis powering the dinos have received an upgrade; this is in no small part thanks to Lucy and a team of people she brought together for a conference-style Raspberry Jam, called Blackgang Pi, over the course of a weekend on the Isle of Wight.

We decided to visit and see just how the conference was going, so on a misty Sunday morning we made our way to Blackgang Chine to see the team in action. It's the UK's oldest amusement park and while it does have a couple of remarkable Victorian-era dinosaur statues, the dinosaurs that were being hacked today were much newer than that. A tea room at the park had been taken over by the team during Blackgang's off season and had been turned into a little workshop, with people working on electronics and animatronics, using Raspberry Pis interfaced with old and new components. The original Raspberry Pi upgrades that Lucy had done a couple of years before had worked fine. Due to weather damage, however, they'd begun to develop a couple of problems at the end of the last season and a further, more robust upgrade was in order.

It wasn't just dinosaurs that were being hacked either: one project was looking at how some of the electrics in Blackgang



Take our word for it: the electronics in a control box just like this operate a T. Rex

Chine, and the neighbouring Robin Hill adventure park, could be controlled via a Raspberry Pi. Sometimes when there's an issue at Robin Hill, it might require a 20-mile walk, so a Pi-controlled system would help to reduce these sorts of issues.

Another major part of the conference was the use of Node-RED (**nodered.org**). We've had a couple of tutorials for it in the magazine before, such as *Minecraft* in the Mashups issue and our hamster feeder from last issue; it's a visual programming language that, instead of writing code, has you building up blocks that create the program in a type of flow. It can be easier to understand for some people and it's useful for IoT or, apparently, dinosaurs.

By the time you read this, the park will have been reopened to visitors, including access to these upgraded dinosaurs. There's also going to be another Blackgang Pi event, a one-day Raspberry Jam this time, that's happening in early June; you can read all about it on the Events pages at the end of the magazine. As well as getting involved and maybe showing off your Raspberry Pi projects, like any other Jam, you might be able to take a quick break and enter the park itself to see the dinosaurs in person. Don't worry: they don't bite.

> "Electronics and animatronics, using Raspberry Pis interfaced with old and new components"

A DINO TOUR

Your guide to surviving the dinosaurs at Blackgang Chine

DEINONYCHUS SURPRISE

The deinonychus is the closest thing in the prehistoric world to the velociraptors from *Jurassic Park*. Raptors in real life were about the size of turkeys, while deinonychus were much more of the real deal. The ones in Blackgang Chine are quite tame, however, as you round a bend and see them lolling around an open space. Don't get distracted, though: while your attention is drawn by them, a motion sensor will reveal a hidden member of the pack right next to you.

T. REX ROAR

The tyrannosaurus rex has ingrained itself pretty firmly in popular culture as one of the greatest predators ever, and it's not hard to see why when you come face-to-face with a life-size adult at Blackgang Chine. The T. Rex here won't chase anyone holding a flare, but it will roar at you, shaking the viewing platform as it does (thanks to the speakers). A big, mean dinosaur powered by a little Raspberry Pi: it's not something you see every day.

TRICERATOPS TRIO

This family of triceratops is pretty cute, with parent and child moving and interacting with each other. Lucy always seems particularly pleased by this display, and after seeing some of the more ferocious predators Blackgang Chine has to offer, it's nice to see a calm family of herbivores just going about their business and having some lunch.

THE RASPBERRY PI ORACLE WEATHER STATION IS HERE

Not to be outshone by the camera, the weather stations for schools have been delivered. Time for science!

ON SALE YET?

The kits so far are just for the schools that signed up for the scheme. However, the Raspberry Pi Foundation is looking at a way to make them commercially available to other schools, and also to individuals. Keep an eye on this space for when we know more!

Starting all the way back in 2014, the Raspberry Pi Oracle Weather Station project was set up to give away 1,000 weather station kits to schools. Each one would be powered by a Raspberry Pi, and enable students to perform any number of weather-related experiments. Two years on, and the first batch of weather stations has just been shipped.

This first set of Oracle weather station kits has been supplied to 150 UK schools, with further stations to be sent out around the globe once feedback has begun to be processed. Now students can start learning code and meteorology through various subjects, and study how the weather works using much more interactive and exciting methods.

As feedback comes back and the rest of the units are shipped out worldwide, more and more learning resources will also be made available, so that schools and students have something to work towards while using their new weather station.

Whether you have a weather station or not, you can always join the community for it. There's a dedicated section of the Raspberry Pi forum for everything to do with the kit here: **magpi.cc/1quWtLg**

SPECS

CONTROLLER:
Raspberry Pi

FORMAT:
Hardware Attached on Top (HAT) with a real-time clock

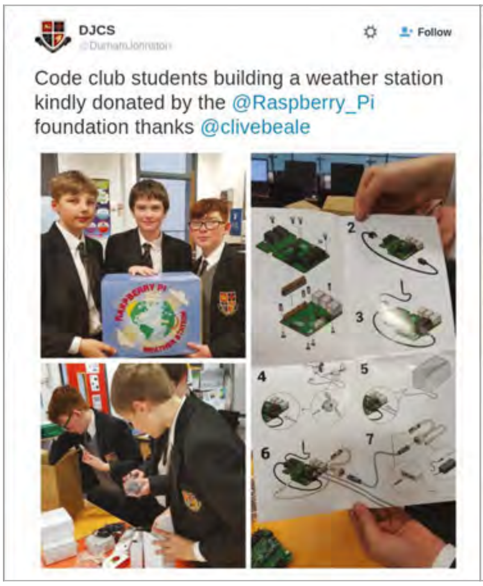
SENSORS:
Basic sensors: humidity, temperature, air quality, and barometric pressure

Rain gauge: measures the amount of rainfall in ~0.3mm increments

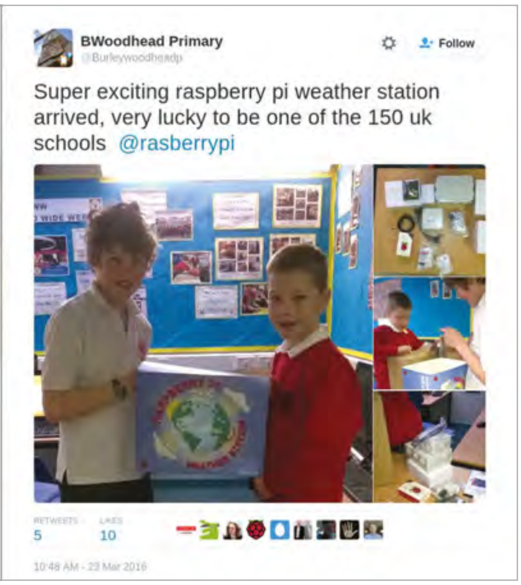
Anemometer: a wind speed gauge to keep track of whether you're in a gentle breeze or gale

Wind vane: it may not have a rooster on it, but it will still show you wind direction

Temperature probe: a basic probe that can measure ambient temperature in Celsius, accurate to three decimal places


DJCS
@DurhamJohnston

Code club students building a weather station kindly donated by the @Raspberry_Pi foundation thanks @clivebeale


BWoodhead Primary
@Burleywoodheadp

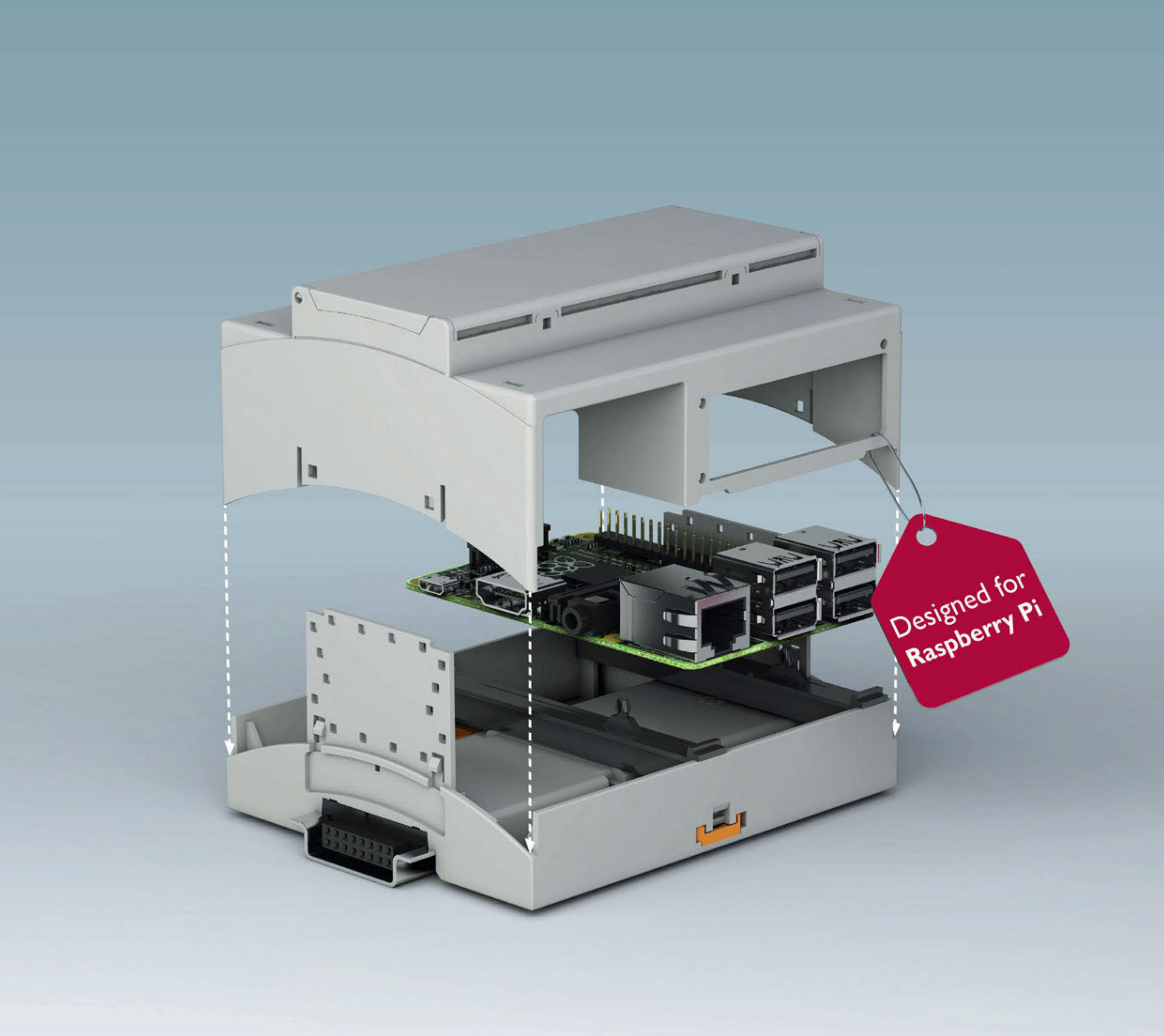
Super exciting raspberry pi weather station arrived, very lucky to be one of the 150 uk schools @rasberrypi

RETWEETS 5
LIKES 10

10:48 AM - 29 Mar 2016

Far left The students at Durham Johnston waste no time getting the kit set up for use

Left Burley Woodhead Primary School are also starting to get their new science experiments ready



Raspberry Pi gets a new home

Industrial automation with Raspberry Pi just got easier

The new RPI-BC enclosure is desgined specifically to accomodate Raspberry Pi computers. The DIN rail or wall mountable enclosure provides access to all ports, GPIO pins and SD card without removing the lid. Ideal for all industrial automation projects.

For additional information call 0845 881 2222 or visit **phoenixcontact.co.uk**

PHOENIX CONTACT
INSPIRING INNOVATIONS

XXX00-00.000.X0

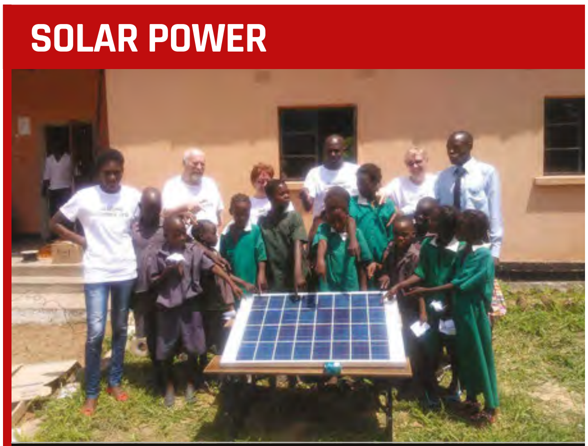
Howard helps to install new solar panels at Bbakasa primary school in the Siavonga district of Zambia

An enthusiastic response from pupils at Kabbila school, keen to learn all about the Raspberry Pi and Scratch programming

AN AFRICAN ADVENTURE

Introducing Raspberry Pi into rural Zambian schools with no electricity supply? We talk to the founders of the Siavonga Project to discover how it's done...

SOLAR POWER



Initially, an 80W Baird solar panel was chosen to supply power at each school, via a heavy-duty deep cycle battery (the sort used in caravans and motorhomes). Panels and batteries were purchased in Zambia, due to weight restrictions for air freight. A solar charge controller is used to prevent overcharging and enable the battery state to be monitored. Since the power supplied is 12V, a step-down regulator is used to convert it to 5V for the Pis. On later visits, 135W solar panels were installed, along with a 220Ah battery, enabling the system to also power LED lighting.

When asked to design a website for the Swansea Siavonga Partnership charity, Harold and Wendy Kirkman of Giakonda IT asked whether the rural schools in Southern Zambia had computers. "No, would you like to install them?" was the response, recalls Wendy. "Then they dropped the bombshell that many schools had no mains power!"

Wendy and Howard soon realised the Raspberry Pi would be ideal for the project, which enables teachers to deliver the Zambian ICT curriculum in schools where children have previously never even seen a computer or mouse. As well as being cheap and portable, the Pi requires low power and can run off batteries. It also offers a wealth of educational resources.

Since the first two primary schools set to benefit from the project had no mains power, Wendy and Howard chose to use solar panels. Before heading out to Zambia, they tested the setup at the rear of their premises: "[We] managed to run two Pis with two [portable] monitors, during the whole of December in Swansea!" They were naturally confident, then, that it would work in Zambia, where there's a little more sunshine!

Memorable school trip

During their first trip to Siavonga in January 2015, Wendy and Howard spent nearly a month setting up computers in two rural primary schools: Bbakasa and Matuwa. After installing the solar power system for each, they could get down to the business of demonstrating the Raspberry Pi in the classroom.

"As well as setting up equipment, we provide training for teachers and pupils," Wendy explains. "This is a vital part of the project in order to give them confidence in using Raspberry Pi. Howard and I are trained teachers... He taught computer science for many years."

Connor Smith solved many technical issues and installed more Raspberry Pis during his recent trip to schools in Siavonga

This process enables the Zambian teachers to teach ICT to children who are completely unfamiliar with computers. "They use LibreOffice on the Raspberry Pis for word processing and spreadsheets. The students particularly enjoy programming with Scratch."

While the original Raspberry Pi Model B was used initially, they have since installed the Pi 2 and, more recently, Pi 3: "The addition of wireless to this model saves money - and lost dongles." Since few of the schools have internet access, several Pis have been set up as servers to deliver the Khan Academy and RACHEL e-learning resources, on a 64GB card, to a hub of computers.

Technical troubles

While Wendy and Howard have each made two visits to Zambia, 21-year-old Connor Smith represented Giakonda on the latest trip in spring 2016. "Reinforcing their knowledge on how to use the Pi and the software provided was the first thing that needed to be done," he tells us. "Even the simplest things like corrupt Raspbian SD cards and faulty USB power supplies, which would be an easy fix in UK, provided real challenges in rural Zambia."

Fortunately, he was aided by local volunteer Bridget Muyuni, who was tasked with taking care of the systems put in place. "While I was there, I went over the most common issues with her. In many cases, we had to improvise and make do."

Solving the technical issues was all worth it when Connor witnessed the reactions of the schoolchildren: "Their enthusiasm for school and the chance for an education is what hit me the hardest." When he gave a lesson on the basics of the Raspberry Pi and RACHEL, there was silence at first, as the pupils wondered what on Earth he was talking about. "But as I showed everyone the Pi, I got some truly wonderful and wacky responses to what they thought [it] was. I then told them all that it was a fully functioning computer and their jaws dropped – I've never been in a room that was so quiet, as they sat in disbelief and then bombarded me with more questions."

While the project has worked with nine schools so far, supplying an average of four Raspberry Pis per school, there are plans for expansion: "There are 26 schools in the district which don't have mains power, and we plan to extend the programme to all of these on a rolling [basis]", says Wendy. She and Howard will be making another visit in June/July, possibly followed by another colleague in September.

SUPPORT THE PROJECT

The Giakonda IT Raspberry Pi project has been funded by Hub Cymru Africa, University of Wales Trinity St David, and S and C Electric Europe. The team work closely with the not-for-profit Swansea Siavonga Partnership (**swansea-siavonga.org**), whose Discovery student volunteering programme holds workshops in Siavonga and links its schools to local ones in Swansea. To find out more about the Giakonda Siavonga project, and possibly make a donation, visit: **giakonda.com/siavonga**.

PORTABLE MONITORS

Monitors have been an ongoing issue for the project. "On our first visit (January 2015), we used 5-volt, 9-inch Tontec screens and made our own enclosures," says Howard. "In July, we used 12-inch TFT screens from Beetronics, but at £145 each, they were more than we really wanted to pay. As we are looking to install much larger numbers, we are considering pi-topCEED."

Get the games now from
YOYOGAMES.COM/PI

GAMEMAKER GAMES ON RASPBERRY PI

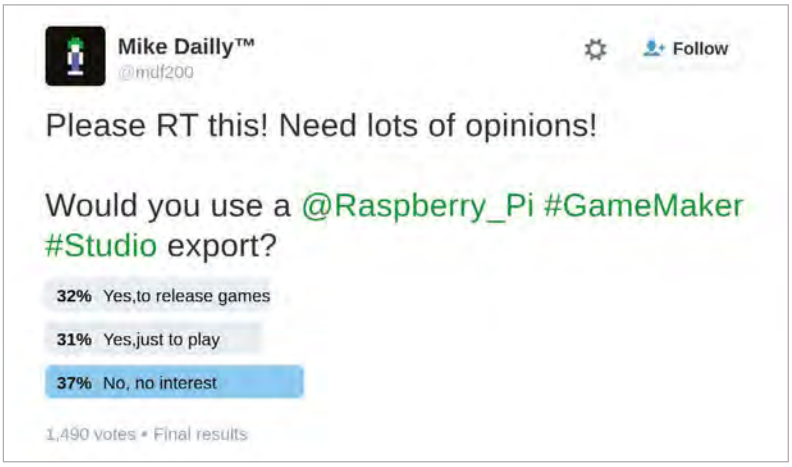
Three excellent video games make their way to the Raspberry Pi, and you can play them right now!

MAKE GAMES WITH GAMEMAKER

GameMaker: Studio is software which runs on Windows that makes game development easier. Developed by games industry veterans and based on a game design teaching tool, it's popular among indies and new devs thanks to a free version that's great to play around with, and a low price for a full version. It exports to a lot of major platforms and maybe soon it will be able to export to the Raspberry Pi! One day, you may even be able to use GameMaker: Studio on the Raspberry Pi as an educational tool. Check it out online here: **magpi.cc/1RMTmJp**

Video games on Raspberry Pi have been around since day one. From people creating or upgrading arcade machines with the Raspberry Pi and the help of some emulation software, to others programming their own games in Python, it's an established use of the Pi in the community. There haven't been very many games ported to or developed natively for the Raspberry Pi, though. That's all changed, however, thanks to the excellent folks at YoYo Games, who have put a handful of some of their GameMaker: Studio classics on the Raspberry Pi.

"We all know the Raspberry Pi is an amazing little machine, but it really needs to get some serious game developers on there to help get the great content that'll drive its use," Mike Dailly, head of engineering at YoYo Games, tells us. Mike is a British games industry veteran, being one of the original creators of *Lemmings* and *Grand Theft Auto*, so he knows his



Mike Daily™
@mdf200

Follow

Please RT this! Need lots of opinions!

Would you use a @Raspberry_Pi #GameMaker #Studio export?

32% Yes,to release games

31% Yes,just to play

37% No, no interest

1,490 votes • Final results

Above Mike's Twitter poll showed that over 60% of 1,500 respondents wanted to see games on Raspberry Pi – a hefty number, for sure

Above Indie darling *Hotline Miami* was made in GameMaker: Studio, showing the power of the dev tool

stuff. “So, in my (neverending!) quest to support the Pi, I went to some of the best indie developers out there and, with the promise of much cake, got them to let me use their games. These games are amazing in lots of ways, and each one pretty different from the next.”

It comes a long time after the original Raspberry Pi launch, but apparently this belated release is not because YoYo Games started on the ports recently, as Mike explains:

“I’ve been trying to get something released on the Pi for years, but as I could only do it in my spare time, it’s taken a *long* time. The GameMaker Linux port came about because I first had to port GameMaker to Linux before porting it to the Pi!” The Linux port was a big enough task on its own, according to Mike, and then he had to find the time to port it to the Pi. “But while it’s taken a long time, I’m overjoyed I’ve finally got something the Pi community can enjoy. If nothing else comes from this, there are now three amazing games on the Pi – and that’s worth it right there.”

> “These games are amazing in lots of ways, and each one pretty different from the next”

Getting to this point may have taken a while, but it’s now a simple case of just exporting GameMaker: Studio projects to the Raspberry Pi, like the developers would for any other platform. There’s been a lot of interest from other GameMaker devs who want to port, whether themselves or straight through Studio, and a poll that Mike put on Twitter proved there was much wider interest in it.

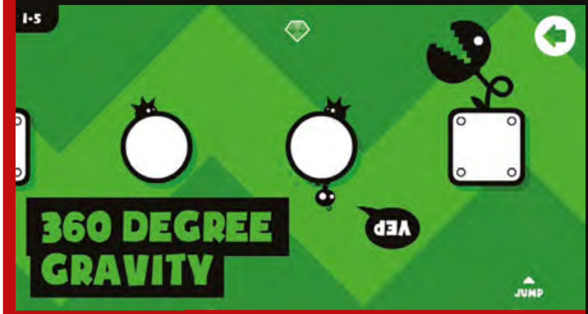
Why the Pi, though? Why has Mike put so much hard work into it?

“Because it’s an awesome little machine that reminds me of the freedom I used to have by having a ZX81 or Spectrum in my bedroom when I was growing up,” Mike recalls fondly. “The Pi has the chance to bring back this unique learning experience, and I for one will do all I can to help support the amazing work they’re doing!”

Now Mike wants to get the export option fully added into GameMaker: Studio. To do this, he needs people to download the games and check them out, which you can do by going to **yoyogames.com/pi**. Maybe one day you can create a game that works on the Raspberry Pi.

LET’S PLAY

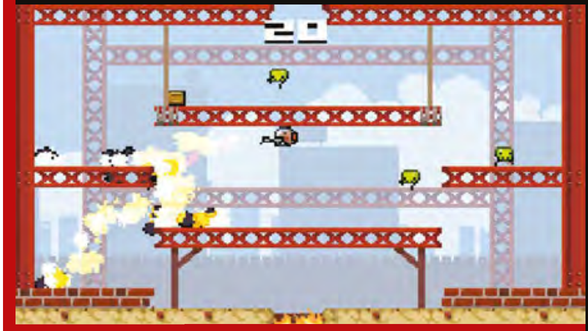
Here are the three games you can enjoy now on your Raspberry Pi – **yoyogames.com/pi**



THEY NEED TO BE FED

By: Jesse Venbrux

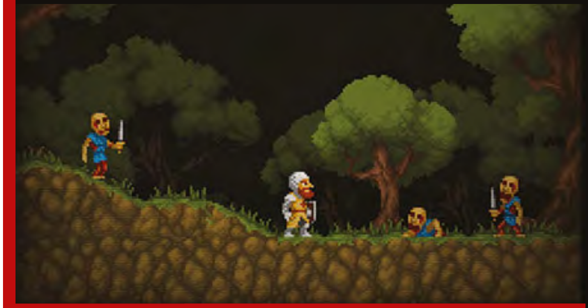
Run and jump through 11 crazy worlds to feed the monsters in this 360° gravity platformer. It doesn’t matter which way you go – up or down, left or right - you can’t fall off because of the 360° gravity! Jump from rotating planes to moving platforms, avoiding dangers and collecting diamonds.



SUPER CRATE BOX

By: Vlambeer

From the folks behind *Luftrausers*, grab your baseball cap and loosen your pants: it’s time to fight endless hordes of enemies and collect every weapon crate you can. Prepare for an arcade delight with tight controls, refreshing game mechanics, cracking retro art, and a hip chiptune soundtrack.



MALDITA CASTILLA

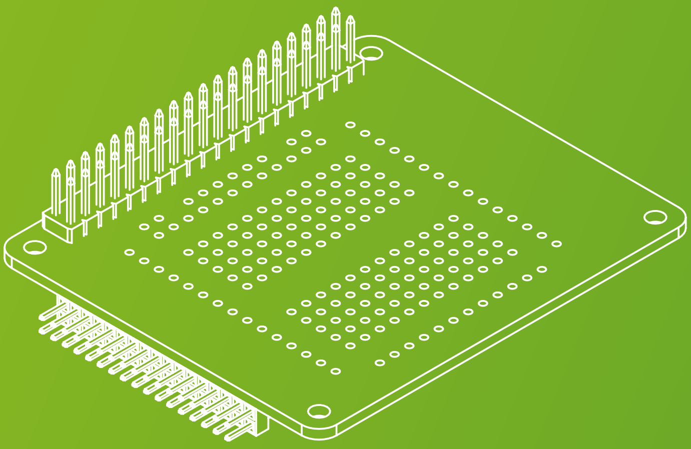
By: Locomalito

Maldita Castilla (‘cursed/damned Castile’) is an action arcade game full of myths from Spain and the rest of Europe. The graphical style is raw, made pixel by pixel with a limited colour palette like in the old days, and displayed through a dirty old monitor effect.

pi-topCEED
11:45
pi-top

pi-topPROTO

pi-topPROTO is an Add-on Board for your pi-top and **pi-top**CEED. It allows you to prototype circuit boards, create amazing IoT devices and more. Slide the **pi-top**PROTO into the modular rail to build any functionality into your device.



pi-top | Find out more at **www.pi-top.com**

THIS PART TO THE PI
ERO SHOT!
BARNEY
GOING I
HERE'S THE BACK!

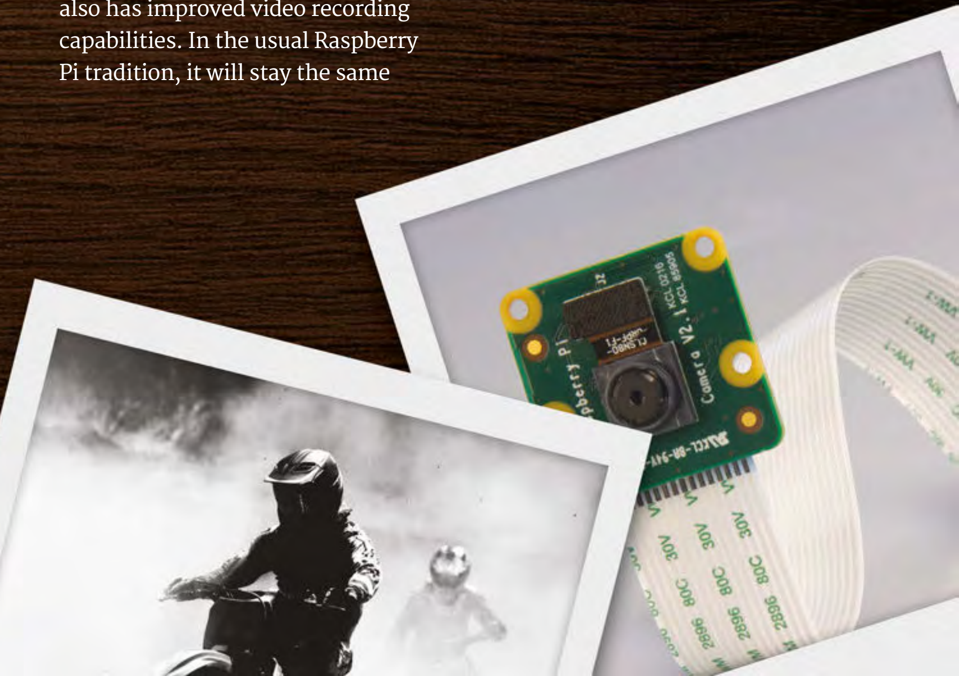
THE ALL-NEW RASPBERRY PI CAMERA

Take better pictures and video with the upgraded version of one of our favourite Raspberry Pi add-ons

The Raspberry Pi Camera Module has always been a fantastic add-on for the Pi. There's always a really fun project somewhere that makes use of it, and it results in a very gratifying visual aspect to your work.

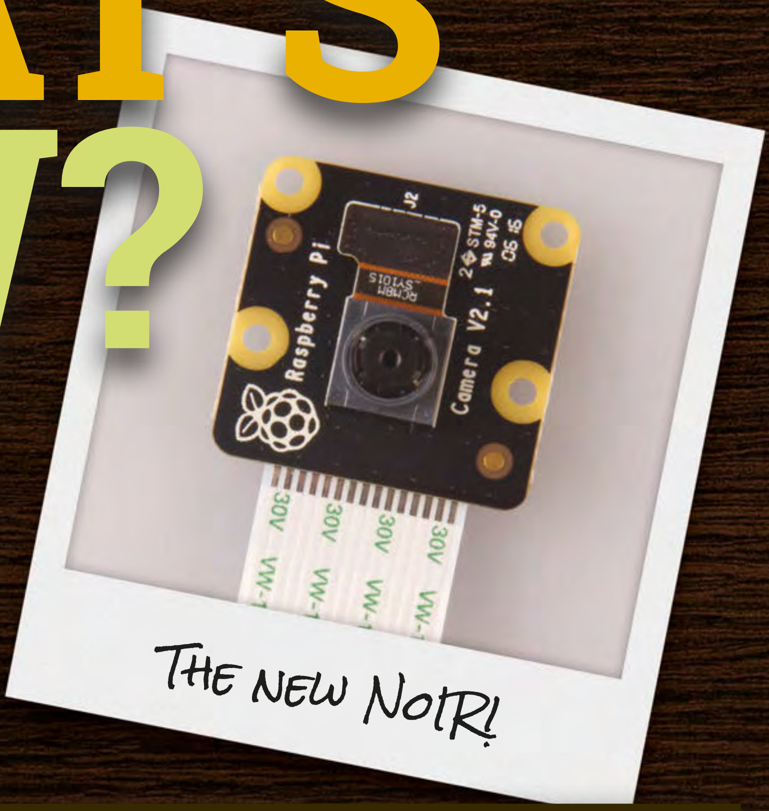
With the new Raspberry Pi Camera Module v2, these projects can only get better. At its most basic, it's a better camera that can take higher-resolution photos and also has improved video recording capabilities. In the usual Raspberry Pi tradition, it will stay the same price and work with every version of the Raspberry Pi, so you won't need to upgrade just to use it!

We're celebrating the new release with a load of cool projects to get you started using the new camera. They'll work with the old camera as well, but we think the new one will be well worth the upgrade; read up on why it's better over the next few pages, as we test it and interview the people behind it.



PI CAMERA V2 WHAT'S NEW?

We talk to the team behind the new camera to find out exactly what's new with the Camera Module v2



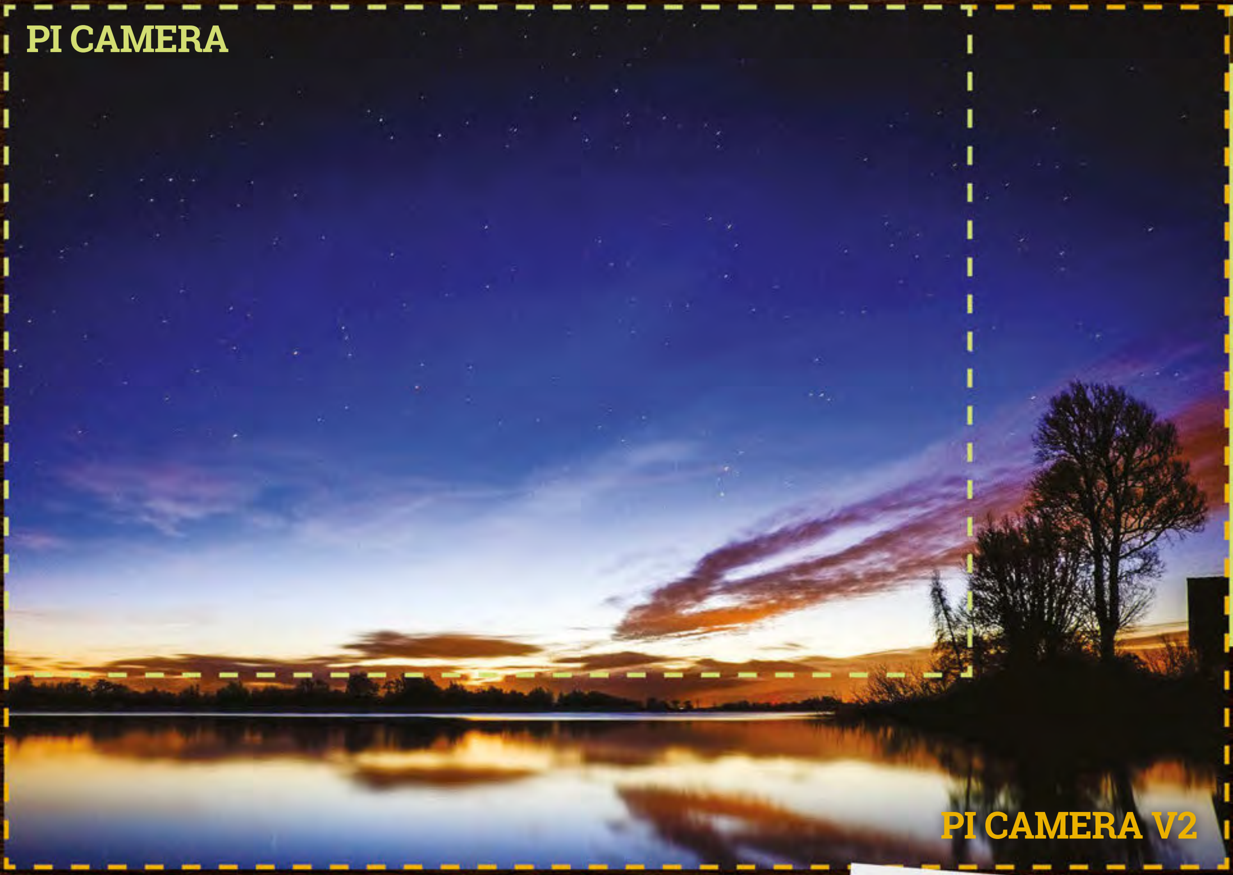
Almost exactly three years ago now (14 May 2013!), the Raspberry Pi Camera Module was released to the world. Making use of the CSI port on the Pi that had been sitting there relatively unused for over a year, the camera has been a small hit. We've always liked it, and so news of a new camera excited us. To find out all about the Camera Module v2, we spoke to James Adams, director of hardware at Raspberry Pi, and Gordon Hollingworth, director of software at Raspberry Pi, so we could cover both the hardware and software side of it.

Mother of invention

"The camera is designed to replace the original as closely as possible, as the original sensor is [End-of-life (EOL)]," James tells us. "So we had the same budget and form factor as the goal, with as much improvement in capability as that would permit… This new camera has been a background development task ever since we understood that the original sensor was EOL about a year ago."

SPECIFICATIONS

	PI CAMERA	PI CAMERA V2
SENSOR:	OmniVision OV5647	Sony IMX219
SENSOR RESOLUTION:	2592 × 1944 pixels (5 megapixel)	3280 × 2464 (8 megapixel)
SENSOR IMAGE AREA:	3.76 × 2.74 mm	3.69 × 2.81 mm
PIXEL SIZE:	1.4 µm × 1.4 µm	1.12 µm × 1.12 µm
OPTICAL SIZE:	1/4˝	1/4˝
VIDEO:	1920 × 1080 (1080p), 30fps 1280 × 720 (720p), 60fps 640 × 480 (480p), 90fps	1920 × 1080 (1080p), 30 fps 1280 × 720 (720p), 60fps 640 × 480 (480p), 90fps



RESOLUTION

With just pure numbers, it's difficult to comprehend just how much better the photos from the new Pi Camera are. Technically they're 60% larger, but that's tricky to visualise, so here's a little example of how much more picture that actually equates to. You get a lot more bang for your buck!

The need for a new camera, then, was somewhat out of necessity. However, with several years of new developments and changing chip prices since the first Camera Module, it did mean that improvements could be made to the original design.

"The major changes are a new sensor module based on a Sony IMX219," James explains. "In testing, it performs very well and is a good improvement on the V1 camera. It's also an 8 megapixel sensor rather than the 5MP of the original. And the board now has rounded corners!"

Looking forward

Not that the old board was sharp at all, but the new round-edged design fits the style of more recent Raspberry Pi releases, such as the Pi Zero and even the Sense HAT. With a strong focus on backwards compatibility with the Raspberry Pi, it's also worth noting that even with a higher-resolution sensor, it still works with all versions of the Raspberry Pi with a camera input.

With this backwards compatibility in mind, it's no surprise to learn the software side of it hasn't changed at all, with Gordon informing us that **raspistill** and **raspivid** still work the same as before. This also includes the lesser known timelapse and raspistillyuv (for RAW photos) functions. This means that you can replace the camera in your existing projects that use it, and the v2 will work just the same without needing to tweak any of the code.

The new camera is a lot better, then, and the team have done a good job making it, although James would like to see an addition to the hardware on the board if it were possible:

"The only real feature that would be nice on a camera is audio (i.e. a microphone), but currently we have no good technical solution for how to do this, as the camera connector does not support an audio interface."

Remember to bring a microphone along, then, when recording your next great Pi-powered short film.

THE HIGHER PIXEL DENSITY MEANS YOU'LL GET MORE DETAIL IN PHOTOS THAN BEFORE

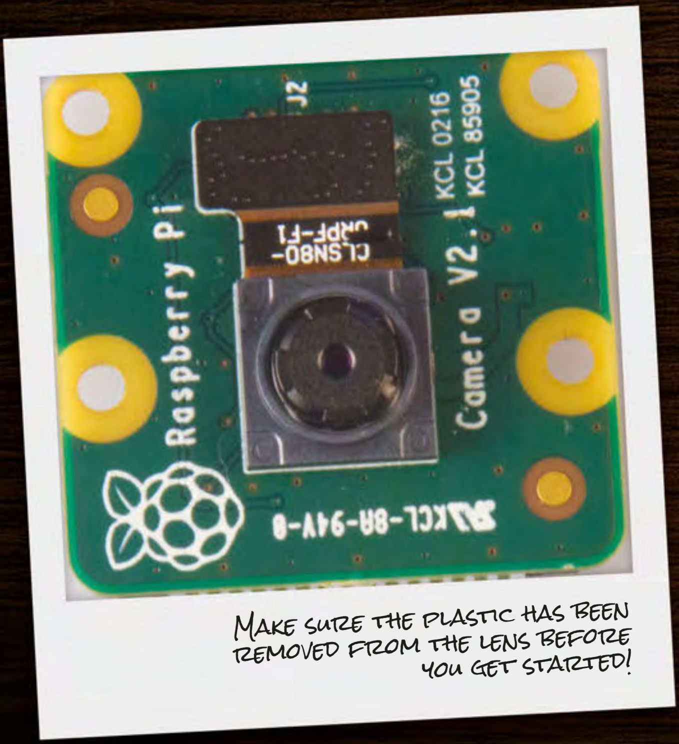
GETTING STARTED WITH THE NEW CAMERA

You'll Need

- Raspberry Pi running Raspbian Jessie **magpi.cc/1MYYTMo**
- Raspberry Pi Camera Module V2 **magpi.cc/1UaFuuW**

You know what's new with the camera – so how can you go about using it?

Higher pixel counts. Smaller pixel sensors. Rounded edges. You've seen the specs and stats for the new Camera Module, so how do you go about using it? Well, remarkably like the old camera, in fact. However, as we haven't done a very basic 'set up your camera' tutorial in quite some time, we thought we'd update it for 2016 and beyond.


Make sure the plastic has been removed from the lens before you get started!

>STEP-01

Connecting the camera

Make sure your Pi is switched off before connecting or disconnecting the camera. Pull out the power cable. Take hold of both ends of the plastic slider of the camera connector on the board, and pull it away from the Pi gently but firmly; it will move up a short distance, opening up the connector.

Insert the ribbon cable, with the connector side – the blue side – facing the HDMI port. Push the plastic slider back down, making sure to put pressure on both sides, so that it gently clicks into place.

>STEP-02

Camera software

Plug the Raspberry Pi back in and turn it on. Once it's booted into the desktop, click on the Program Menu and go down to Preferences. Here you'll find the Raspberry Pi Configuration menu, which you should now click on. On the tab called Interfaces, you'll find an option to enable the camera; if this isn't set to Enabled, do so now. Raspbian doesn't have the camera enabled by default, so this is required. Now reboot your Pi.

>STEP-03

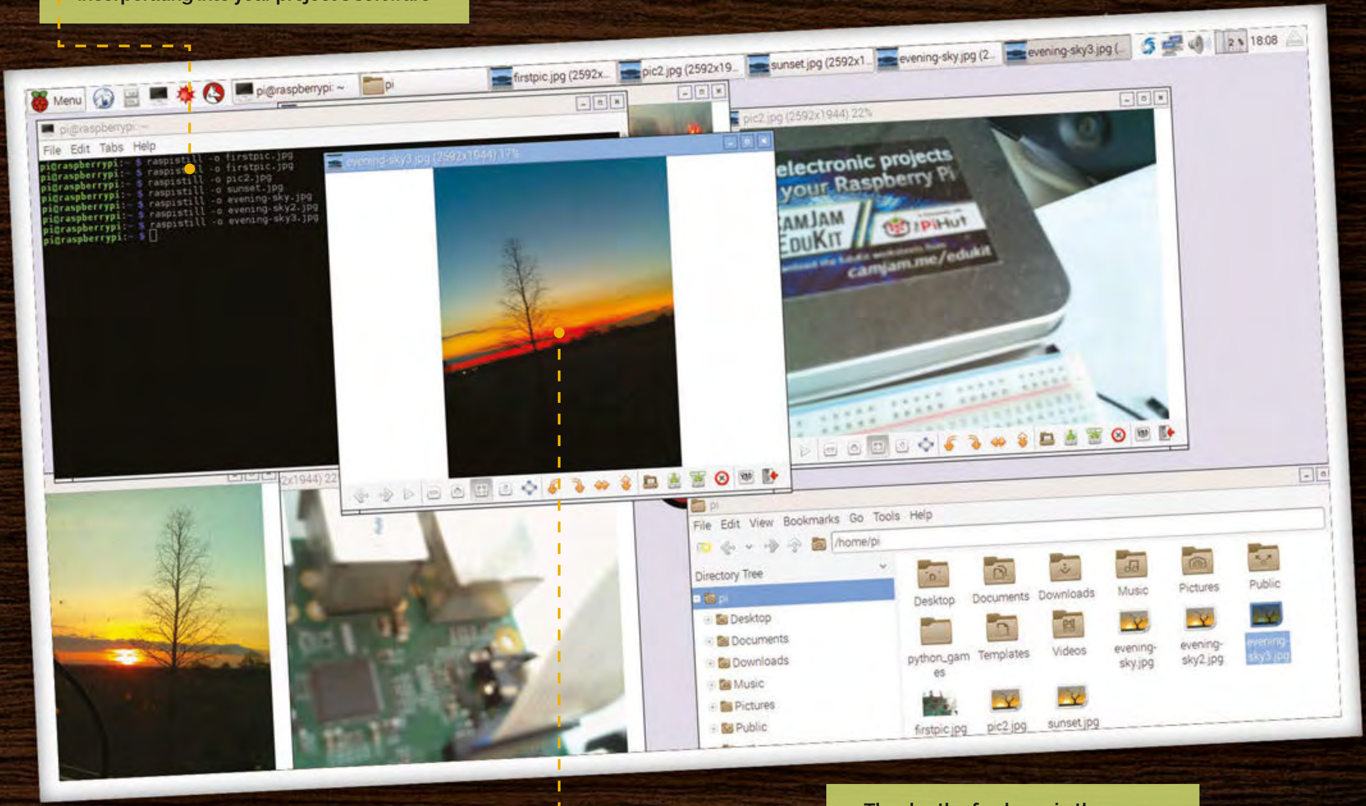
First shots

Point your camera at something interesting, then open a terminal and type in the following:

```
raspistill -o firstpic.jpg
```

You'll see a red light on the Camera Module, followed by the image of whatever the camera is pointing at appearing on the screen for a moment. If it's you, you can use the screen to arrange your best smile during the five-second pause before the picture is taken. Once snapped, the image can be found in the home directory – **/home/pi**.

Instead of 'point and click' photography, we give you 'point and command line' – not a backwards step, but much better for incorporating into your project's software



```
pi@raspberrypi:~ $ raspistill -o firstpic.jpg
pi@raspberrypi:~ $ raspistill -o firstpic.jpg
pi@raspberrypi:~ $ raspistill -o pic2.jpg
pi@raspberrypi:~ $ raspistill -o sunset.jpg
pi@raspberrypi:~ $ raspistill -o evening-sky.jpg
pi@raspberrypi:~ $ raspistill -o evening-sky2.jpg
pi@raspberrypi:~ $ raspistill -o evening-sky3.jpg
pi@raspberrypi:~ $ 
```

The depth of colours in the new Camera Module can make for gorgeous sunset shots – or improved intruder detection for your robot guard dog

>STEP-04

Find the image

You can open the picture and view it from the File Manager, but if you don't have a Pi 2 or Pi 3, you may want to avoid the unnecessary overhead of running that, and just enter this into the terminal:

```
gpicview firstpic.jpg
```

If it looks rather blurry, check that you remembered to peel the protective plastic from the Camera Module's lens!

>STEP-05

More advanced commands

The **raspistill** command has a list of options so long that it borders on the intimidating. Have no fear, though – you won't need to learn them all, but there are a few that might be useful to you, such as:

```
raspistill -t 15000 -o
newpic.jpg
```

The **-t** option changes the delay before the picture is taken, from the default five seconds to whatever time you give it in milliseconds – in this case, a full twenty-five seconds to get your shot arranged perfectly after you press ENTER. You can find the full list of options here: **magpi.cc/1SqBfuT**

>STEP-06

A quick fix

One of the problems with a camera at the end of a ribbon cable is getting it positioned properly. You may end up with the camera upside down or slightly askew. Upside-down Camera Modules can be commanded to flip the picture the right way up with **--vflip**, or **-vf** for short. **--hflip** handles horizontal flipping, should you need a mirror image. And if your camera is lying on its side, use **--rotation**, or **-rot**, followed by the number of degrees: 90 or 270.

SHOOTING VIDEO

For taking video, **raspivid** is what you need. It can record up to 1080p video at 30fps, a fast enough frame rate for cinema, and 720p at 60fps if you want something smoother. You can do this with:

```
raspivideo -t 10000 -o
testvideo.h264
```

This records a ten-second video (10,000 milliseconds) at the default 1920 × 1080 format. You can also shoot slow-mo video at 640 × 480 by using:

```
raspivid -w 640 -h 480
-fps 90 -t 10000 -o
test90fps.h264
```

Use **omxplayer** in the command line to play the videos back!

THE HAYLER-GOODALLS

Ozzy, Jasper, and Richard are mentors at CoderDojo Ham and gave a talk about their Astro Pi adventures at the Raspberry Pi birthday party.
@rdhayler / coderdojoham.org
richardhayler.blogspot.co.uk

ADD A BUTTON TO A PI CAMERA

Take wonderful pictures with your Pi Camera at a click of a button, perfect for time-lapse photography and stop-motion video.

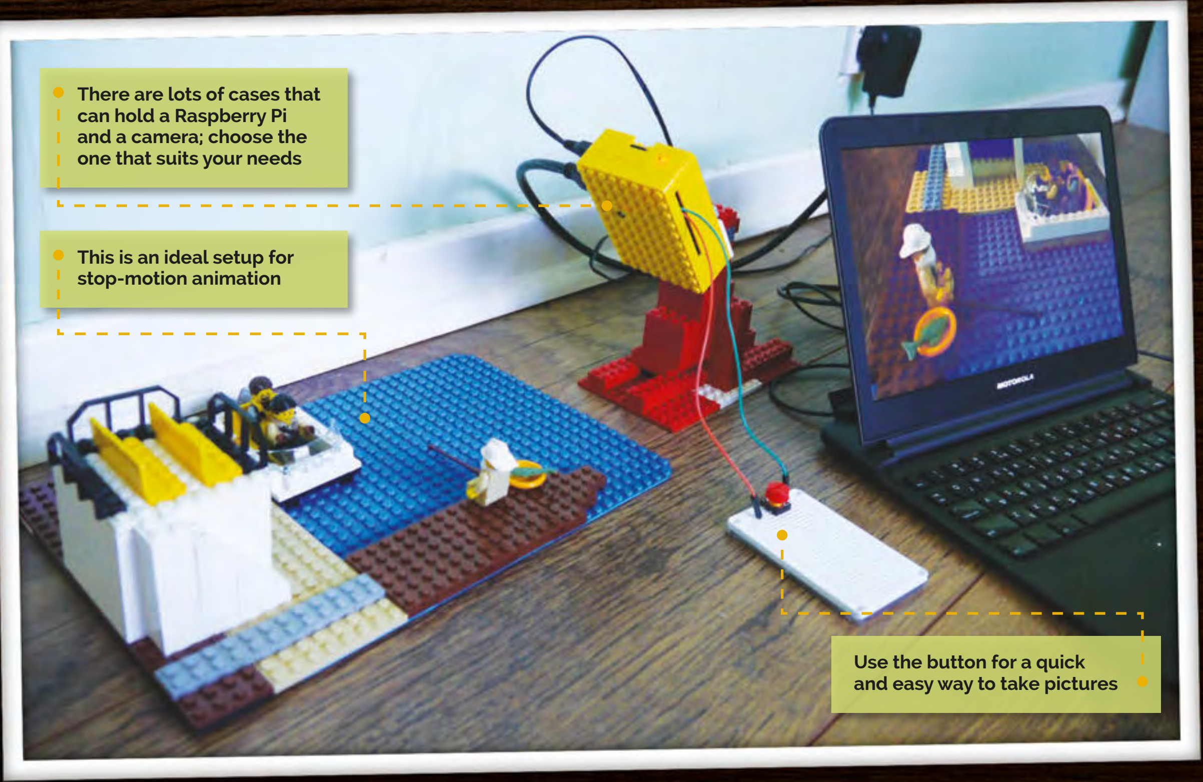
You'll Need

- Raspberry Pi camera
- Breadboard
- Jumper wires
- A button

Have you been reading the last few pages, thinking that you'd prefer to take a picture with a Raspberry Pi without all that typing on the command line? In this tutorial we'll show you how to take a picture with a click of a button, just like a real camera. This could be useful for many projects (for example, time-lapse photography), but in this tutorial we are focusing on stop-motion animation.

Before we start installing the camera and wiring the button, you need to make sure you've installed the camera on the Raspberry Pi, as shown on the previous page.

If your Raspberry Pi is not already switched off, do that now. Then, connect the button to the Pi via a jumper lead, as shown in **Fig 1**. One side of the button will be connected to ground; the other is connected to GPIO pin 14 (but you can choose your favourite pin). That's all the hardware done; now let's install all



There are lots of cases that can hold a Raspberry Pi and a camera; choose the one that suits your needs

This is an ideal setup for stop-motion animation

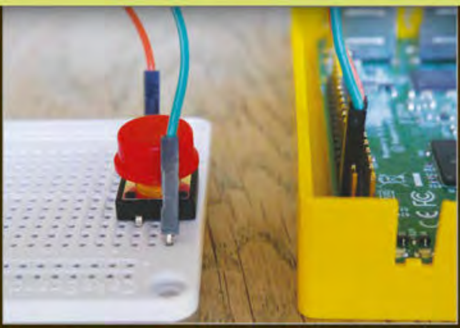
Use the button for a quick and easy way to take pictures

CREATE A BUTTON-OPERATED CAMERA

>STEP-01

Install the Pi camera

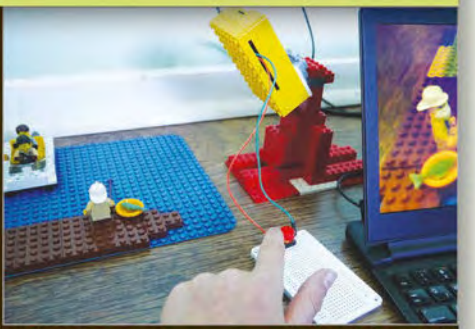
Make sure you add the camera to the correct socket: there are two on the Pi, which look identical. We want the one between the HDMI and audio ports.



>STEP-02

Wire up the button

You can use a breadboard for a small button. Alternatively, you can connect your jumper wires directly to the pins on a bigger one.



>STEP-03

Lights, camera, action!

Create your stop-motion scene and use the button to trigger the camera to take pictures and save them to timestamped files.

the necessary Python modules via the command line. Type:

```
sudo apt-get install python3-pip
sudo pip3 install gpiozero
picamera
```

Coding the button

Finally, we need to think about the code we'll be using. Because we're focusing on stop-motion, we've used the camera's preview mode so that we can set up our shot before we take it, to ensure everything is in the frame. Then, only when the button is pressed, we save an image file. Each image file will have a different name based on the date and time at which it is taken. This makes it easy to assemble all the images from the shoot for post-processing. The wonderful gpiozero library is used to capture the button activity; we simply define a function that is run whenever the button is pressed. This function uses the picamera Python library which allows us to control the camera through code, making all the normal command-line operations available.

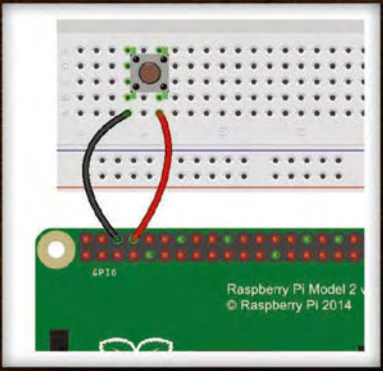


Fig 1 It's very simple to wire up a button to the Raspberry Pi

Download or type up the code from the listing and either run it through IDLE or the command line. Type **CTRL+C** to quit the program.

You should be able to use this code as a template to create a program for whatever photography project you have in mind. For example, you could alter the code so that the camera takes continuous photos while the button is held down. Or you could add extra buttons to make a variety of photography modes available.

With this sort of build, you can also start thinking about building a complete, portable, wirelessly connected Pi camera. For this, you can even use the Lego case – all you'd require is a portable mobile phone battery charger, along with a screen to attach to the Pi. With a bit of modification of the code, you can have it always show the preview of the camera on the screen. Want to record video? More modification of the code will allow for video capturing. The only issue you might have with both of these projects is a flash or light source, so a well-lit subject will be essential.

simple Camera.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1MFuVNG

```
#importing the necessary modules
from datetime import datetime
from gpiozero import Button
import picamera
import time

b=Button(14)
pc=picamera.PiCamera()
running = True
#pc.resolution = (1024, 768)
#use this to set the resolution if you dislike
the default values
timestamp=datetime.now()
def picture():
    pc.capture('pic'+str(timestamp)+'.jpg')
    #taking the picture

pc.start_preview() #running the preview
b.when_pressed=picture
try:
    while running:
        print('Active')#displaying 'active' to
the shell
            time.sleep(1)
#we detect Ctrl-C then quit the program
except KeyboardInterrupt:
    pc.stop_preview()
    running = False
```

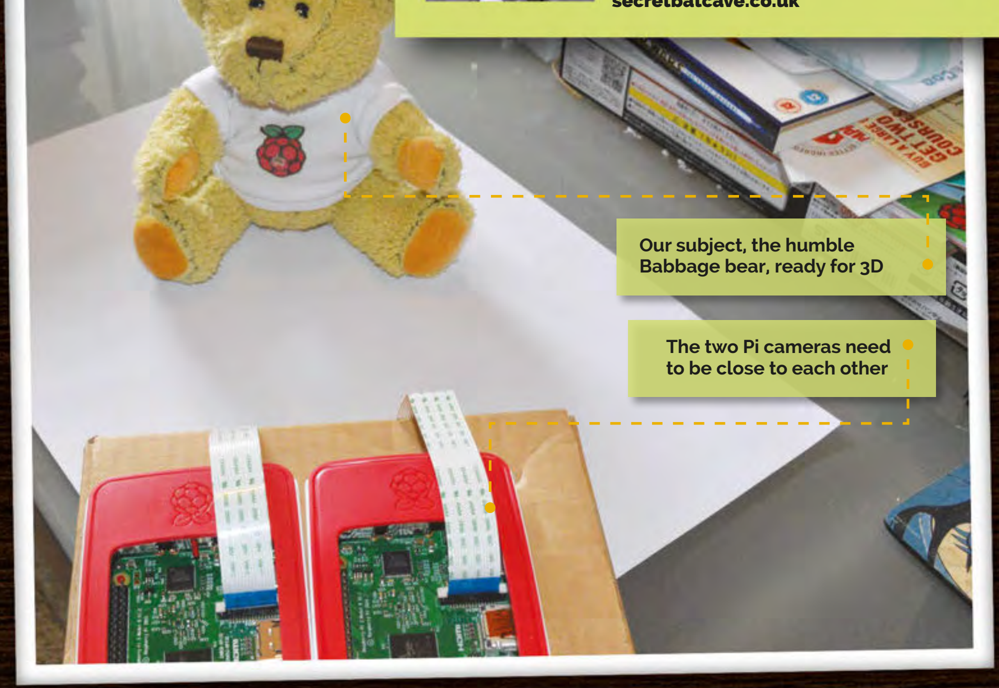
WILLEM KOOPMAN

Peripatetic sysadmin, providing grumpy solutions to grumpy problems. Stumbled upon the field of computer vision while working in the magical world of visual effects.

secretbatcave.co.uk

3D YOUR MOVIES

Technically, our setup can be used for making films as well. However, the process of turning the videos into a 3D movie is a bit more complex than a 3D photo. If you own a copy of Adobe Premiere, you can find ways to make it anaglyph red-and-blue 3D, like in the photos, but you'll need a bit more for modern movie-style 3D.



Our subject, the humble Babbage bear, ready for 3D

The two Pi cameras need to be close to each other

TAKE 3D PICTURES

Use two Raspberry Pis and camera modules to create 3D photos – with the help of a bit of software, of course

You'll Need

- 2× Pi camera modules **magpi.cc/1UaFuuW**
- 2× Raspberry Pis
- PC/laptop capable of SSH **magpi.cc/1Mm5Npi**

Making 3D video and film is actually not so tricky on the hardware side – you just need to make sure you have two camera sensors working simultaneously in the right orientation. The challenging bit is combining them and timing them. With the Raspberry Pi, or several of them, and the right bit of code, you too can take some 3D photos with the new camera module.

>STEP-01

A smidge of theory

If you fiddle with the internet long enough, you'll bump into sockets. Sockets are an API that allows one computer to talk to another. They work much like a telephone: you dial up a number (or IP address) and wait for the other side to answer. This is a one-to-one connection, or unicast.

But we need to talk to many Pis at once. This is where broadcast comes in. It's a mechanism that allows devices on the same network to talk to each other without needing a phone book.

>STEP-02

Preparing your Pis

You'll need to change the default hostname of both your Raspberry Pis. Let's make it simple and use the names 'left' and 'right'.

Power the first Pi on and type

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

in a terminal window. Replace **raspberrypi** with **left**. We also need to update the hosts file: open it with `sudo nano /etc/hosts` and again replace **raspberrypi** with **left**.

Finally, `sudo reboot`. You should be greeted by: `pi@left ~ $` when you log in or open a terminal. Repeat this process for the other Pi, using **right**.

>STEP-03

Tool installation

We need to install some tools. Power both Pis on, SSH into them from your other computer (`ssh left`, password **raspberry**) and then run:

```
sudo apt-get install git
python-picamera screen
```

Now, we check out the Python scripts on both Pis. We can do it quickly via SSH:

```
ssh left 'git clone https://
github.com/secretbatcave/
pipolypicture'
```

This will log into the **left** Pi and run

listener.py

```
#!/usr/bin/env python
'''
listens for a broadcast packet with the string "photo" in it.
then takes a picture, names it with the hostname and timestamp
'''
import time
import socket
import picamera


udp = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
udp.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
udp.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
udp.bind(('',8000))
print 'BEGIN!'
with picamera.PiCamera() as camera:
    camera.resolution = (2592,1944)
    time.sleep(1)
    while True:
        data, addr = udp.recvfrom(1024)
        print "Got me a packet"
        print data
        # This could be any word or string, allowing you to trigger
        # different things using the same system
        if "photo" in data:
            print "taking a photo"
            serial = data.split("-")
            serial = serial[1]
            camera.capture("{0}_{1}.jpg".format(
socket.gethostname(),serial))
```

sender.py

```
#!/usr/bin/env python
'''
This sends out a broadcast
packet with the word "photo"
and an incrementing number
'''
import sys
import time
import socket

MYPORT = 8000
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind(('', 0))
s.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

# Change this number to avoid accidentally
# overwriting previous images
count = 200
while 1:
    data = "photo-{0}".format(count)
    count = count + 1
    s.sendto(data, ('<broadcast>', MYPORT))
    time.sleep(2)
    message = raw_input(
"enter to take picture, q to quit:")
    if str(message) == 'q':
        sys.exit()
    print "woo"
```

Language
>PYTHON 2

DOWNLOAD:
magpi.cc/1N9d0z3

the **git clone** command. Do the same for the **right** Pi.

To test that everything is working:

```
ssh left
cd pipolypicture
sudo python listener.py
```

You should see something like:

```
pi@left ~ $ sudo python
listener.py
BEGIN!
```

>STEP-04
Listening and sending

Now that we have all the code and libraries installed, let's start taking some photos. You'll notice there are two programs: **sender.py** and **listener.py**. You need one listener.py for each of the devices that you want to control. Sender.py controls them all with its magic broadcast socket.

Let's get things set up:

```
ssh left
cd pipolypicture
screen python listener.py
```

Press **CTRL+A** then **CTRL+D** to disconnect the screen session but, crucially, leave listener.py running in the background. Press **CTRL+D** again to disconnect from **left**. We now need to do the same for **right**.

>STEP-05
Taking photos

We've primed the Pis to listen for a magic word, so let's start taking pictures! From your master computer, enter:

```
cd pipolypictures
python sender.py
```

You'll be greeted by this prompt:

```
enter to take picture, q to quit:
```

Hitting **ENTER** sends out a packet with the word 'photo' to every device on the local network. The listener programs on each Pi will pick up this packet and take a picture. Each packet we broadcast has a number attached. This number is used as the second half of the file name, so we can pair up the pictures easily later.

>STEP-06
Combine them into 3D

With both cameras placed right next to each other, you can use the resulting images to create a 3D image. The easiest way is to do this online using a website like **create3dphoto.com**; this makes old-school red-and-blue anaglyph stereoscopic photos, so you may need to dig out some old-fashioned 3D glasses for it. Experiment to see how to make the perfect 3D photo.

DAN ALDRED

A Raspberry Pi Certified Educator who writes and makes excellent Raspberry Pi projects, among many other things.

tecoed.co.uk

MAKE A MINECRAFT PHOTO BOOTH

Create a photo booth in Minecraft that takes photos of the real world. What will you see on your travels?

You'll Need

- Raspberry Pi Camera Module **magpi.cc/1UaFuuW**

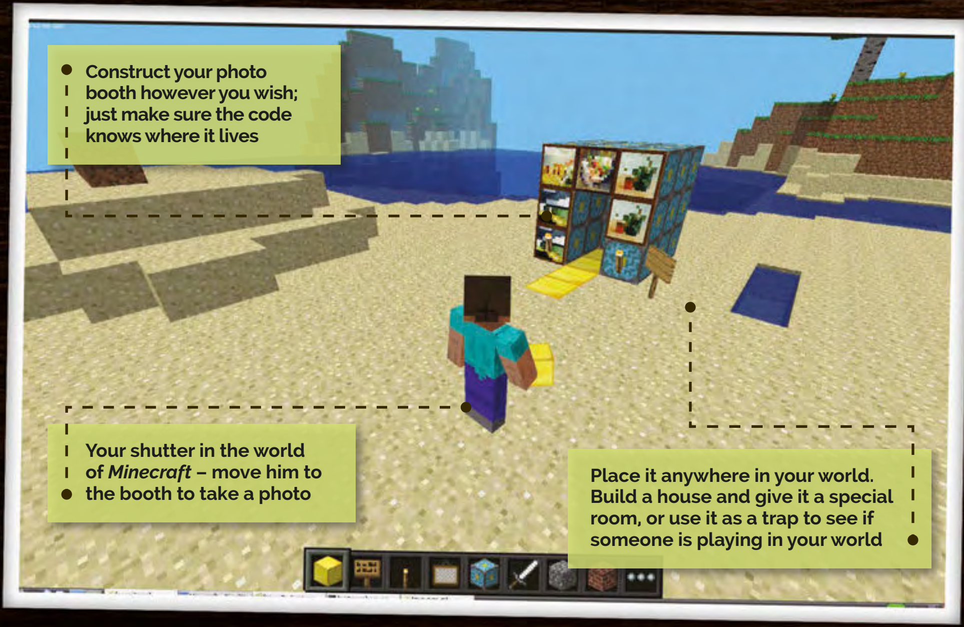
A few issues ago we did some *Minecraft* mash-ups, going beyond the normal *Minecraft* Python programming to have it interact with the real world. With this tutorial, we're bringing this concept back by having *Minecraft* activate the Pi camera with code. This tutorial can be found in the excellent Raspberry Pi resources (**magpi.cc/1qEg9Nh**).

The first thing you need to do is import the *Minecraft* API (application programming interface). This enables you to connect to *Minecraft* and use Python to code. You also need to import the **PiCamera** module to control the camera, and the **time** module to add a small delay between taking each photo.

Open *Minecraft* from the application menu, then enter an existing world or create a new one. Move the *Minecraft* window to one side of the screen. You'll need to use the **TAB** key to take your mouse's focus away from the *Minecraft* window to move it. This is needed later when you switch between the *Minecraft* and Python windows.

Open Python 3 from the applications menu. This will open up the Python IDLE code editor which you'll use to write the photo booth program; click **New>Window** to open a new window.

Enter the code in the listing (or you can download it and open it now). Save with **CTRL+S** and run the program with **F5**. You should see the message 'Find the photobooth' appear in the *Minecraft* world. This is the first part of the code. Stop it running using **CTRL+C** and we can explain the rest.



Construct your photo booth however you wish; just make sure the code knows where it lives

Your shutter in the world of *Minecraft* – move him to the booth to take a photo

Place it anywhere in your world. Build a house and give it a special room, or use it as a trap to see if someone is playing in your world

Camera tests

Next, we'll make sure the camera is set up. We've set the camera to show a two-second preview, so that you can strike your pose and smile before the picture is taken. The image is stored as a file called **selfie.jpg** in your home directory.

Now, you need to create a photo booth in the *Minecraft* environment. This is done manually, and the photo booth can be built wherever you want to locate it. Using any block type, build your photo booth. It can be any shape you like, but it should have at least one block width of free space inside so that the player can enter, like a door or gate.

Left Take selfies with the help of *Minecraft* Steve. If only he could join you in the photo as well…

Once you have created your photo booth, you need to be able to move your player inside and onto the trigger block. This is the block that the player stands on to run the function that you wrote in step one, which will trigger the camera. In the *Minecraft* environment, your position is given in reference to the x, y, and z axes. Look at the top-right of the window and you'll see the x, y, and z coordinates of your player – for example, 10.5, 9.0, -44.3. Assuming you're still in the photo booth, then these are also the x, y, and z coordinates of the 'trigger' block in your booth.

Walk into your photo booth

Note down all three coordinates of your camera 'trigger' block. When you're playing *Minecraft*, your program will need to verify that you're inside the photo booth. If you are, then it will trigger the **`take_the_pic`** function and take a picture with the camera. To do this, *Minecraft* needs to know where you are in the world.

In order to find your position, you use the code **`x, y, x = mc.player.getPos()`**. This saves the x, y, and z position of your player into the variables **`x`**, **`y`**, and **`z`**. You can then use **`print(x)`** to print the x value, or **`print(x, y, z)`** to see them all if you wish, by adding it to the code. Now you know the position of the player, you can test to see if they're in the photo booth.

At this point we have a photo booth, the coordinates of the trigger block, and code to control the Camera Module and take a picture. The next part of the code is to test whether the program knows when you're in the photo booth. To do this, we create a loop which checks if your player's coordinates match the trigger block coordinates. If they do, then you're standing in the photo booth. For this, we use a simple **`if`** statement, which we call a conditional.

Change the **`if`** line in the code to ensure the coordinates you enter are those of your photo booth. Save and run your code to test it: walk into your photo booth and you should see the message 'You are in the photobooth!' in the *Minecraft* window.

You will note that the **`if`** statement checks if the x value is greater than or equal to 10.5: this is to ensure that it picks up the block, as it could have a value of 10.6. Remember to replace the x, y, and z values with those from your photo booth. After the message is printed, the same preview and camera snap will happen as before the **`while`** loop, then the loop resets itself so you can enter it again and take another photo!

Language
>PYTHON

DOWNLOAD:
magpi.cc/MinecraftBooth

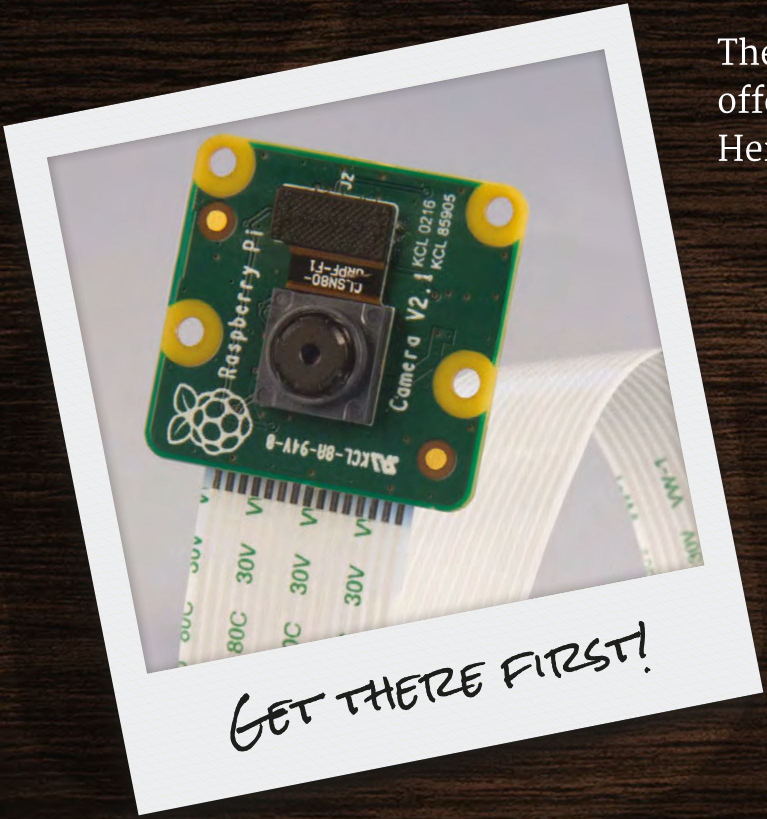
photobooth.py

```python
from mcpi.minecraft import Minecraft
from picamera import PiCamera
from time import sleep

mc = Minecraft.create()
camera = PiCamera()

mc.postToChat("Find the photo booth")

camera.start_preview()
sleep(2)
camera.capture('/home/pi/selfie.jpg')
camera.stop_preview()

while True:
    x, y, z = mc.player.getPos()

    sleep(3)

    if x >= 10.5 and y == 9.0 and z == -44.3:
        mc.postToChat("You're in the photo booth!")
        sleep(1)
            mc.postToChat("Smile!")
        sleep(1)
            camera.start_preview()
            sleep(2)
            camera.capture('/home/pi/selfie.jpg')
            camera.stop_preview()

    sleep(3) X
```

MAKE YOURSELF SNAP HAPPY WITH THE PI HUT

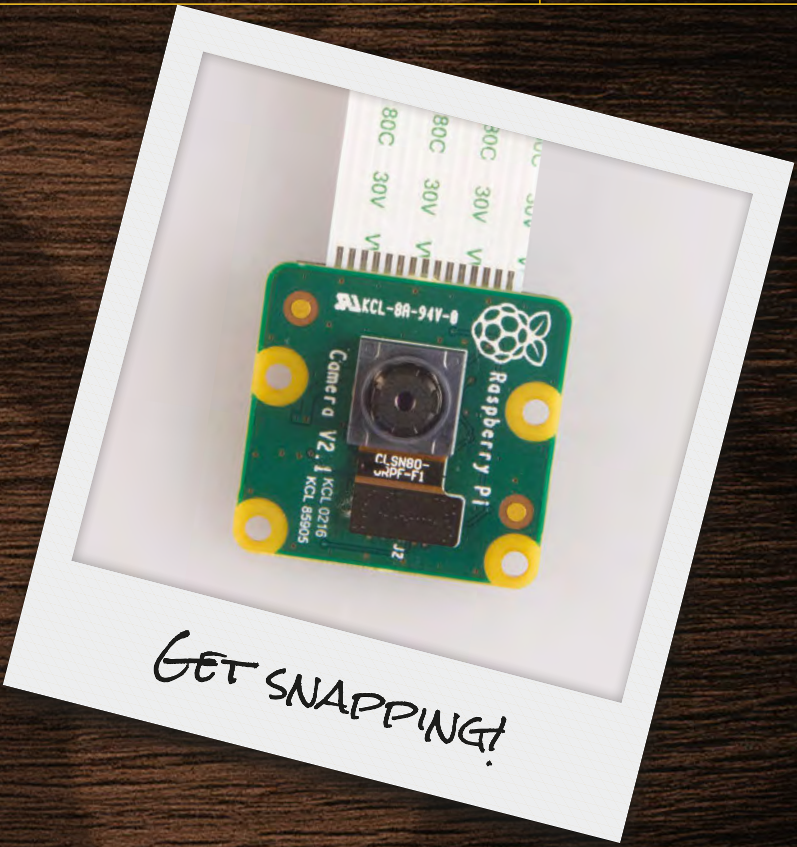
SUBSCRIBERS GET THE NEW CAMERA FIRST

There's much more than just savings on offer for print subscribers of the magazine. Here's how subscribers got their camera…



On the 25 April we sent all print subscribers a special code via email, for use at ThePiHut.com. The code enabled them to buy the new Camera Module and receive it before everyone else. If you didn't get a code or aren't yet a subscriber, you've got until 13 May to subscribe and get your code.

The PiHut
www.ThePiHut.com



Use your subscriber code at The Pi Hut to:

- Jump to the front of the queue
- Get priority handling of your camera order
- Be first in line if it's out of stock

NO CODE?

If you're an existing subscriber and you didn't receive an email, it might be because your address isn't in the subscriptions database. To rectify this, simply email magpi@selectps.com with your full name and postcode, and our subscriptions team will reply with your unique code within 24 hours.

Not a subscriber? It's not too late to use this offer. Simply subscribe (over the page) and email magpi@selectps.com requesting your unique subscriber code before 13 May, to make use of this fantastic offer.

HOW TO USE YOUR CODE

- **FIND YOUR EMAIL FROM THE MAGPI**
  The email was sent to all subscribers on 25 April from **magpi@raspberrypi.org**
- **GRAB YOUR SUBSCRIBER CODE**
  Make a note of your unique code or copy it from the email
- **VISIT THEPIHUT.COM**
  Visit **thepihut.com** and put a Camera Module in your basket
- **ENTER YOUR UNIQUE SUBSCRIBER CODE**
  Simply input your code at the checkout

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
- Subscriber-only offers
- Get it first (before stores)

PI ZERO #40 IS STILL ON OFFER
Call today!

The MagPi
PICTURE PERFECT
The NEW Raspberry Pi Camera Module
POWER UP YOUR LIFE WITH RASPBERRY PI
RASPBERRY PI 3

SAVE UP TO 25%

Pricing

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

Direct Debit

£12.99 (UK) (quarterly)

How to subscribe:

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

Available on the App Store

GET IT ON Google Play

SUBSCRIPTION FORM

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

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

Mag#45

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 □□/□□/□□

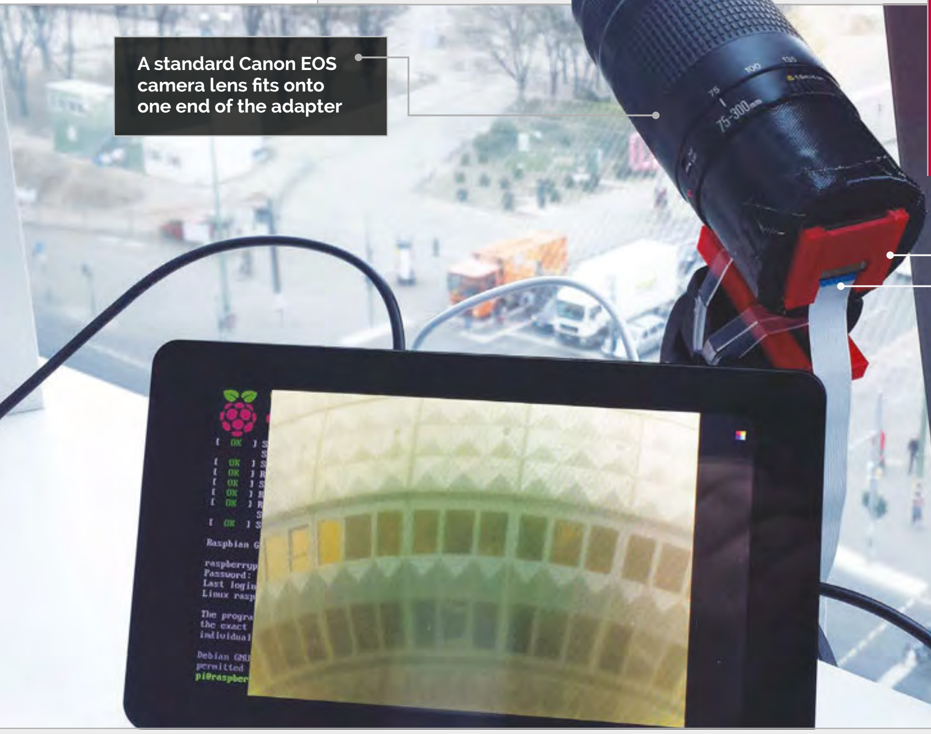
I would like my subscription to begin from issue .... (month + year)

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.



A standard Canon EOS camera lens fits onto one end of the adapter

The 3D printed lens adapter comprises three parts which slot together

The Pi camera sensor gathers light from a small field of view in great detail

JAMES MITCHELL

A software QA engineer at Oracle, James is also a Pi enthusiast who organises the regular Raspberry Jam Berlin event.

raspberryjamberlin.de / @monkeymademe

Quick Facts

- This particular adapter only works with Canon lenses
- It was 3D printed at Dimension Alley in Berlin
- You need to move the lens to keep the moon in frame
- James is hoping to use a longer lens for better photos
- He's also shot a time-lapse video with the adapter

PI MOON CAMERA

With the aid of a 3D printed lens adapter, it's possible to take detailed photos of the moon using the Raspberry Pi Camera Module!

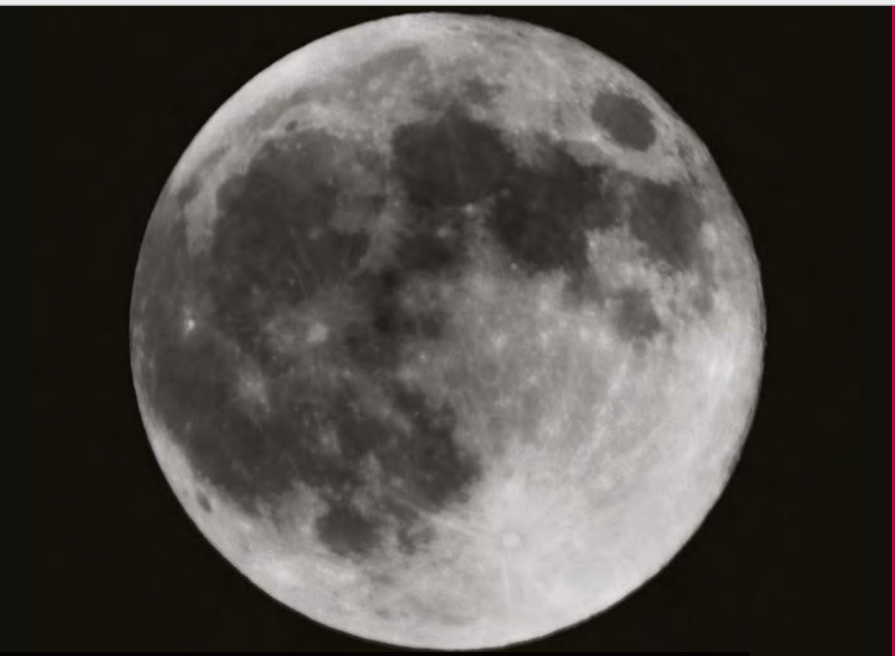
The pink tinge is caused by a reflection of light on the camera sensor, which James has since fixed

Shooting the moon has been an obsession of James Mitchell's for a very long time. After several failed attempts to get a clear photo from various cameras and lenses over the years, he decided "on a bit of a whim" to give it a go with a Raspberry Pi Camera Module (**magpi.cc/1Nd44Dx**).

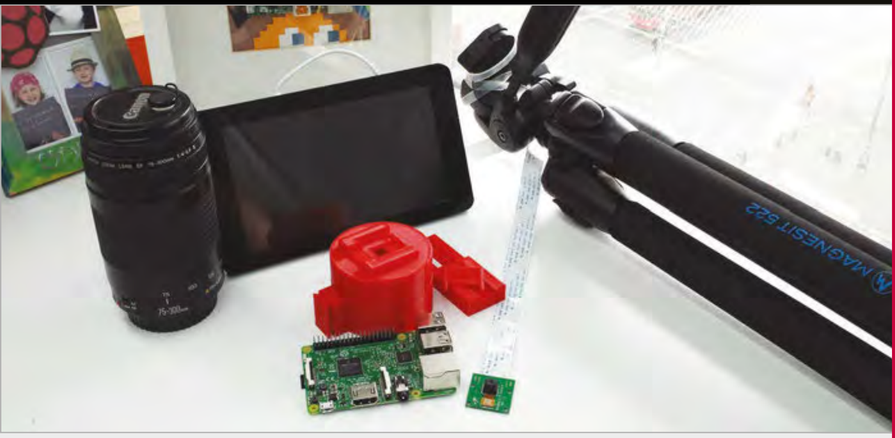
"I have been working on pushing the limits of the Pi camera for a while," James tells us. One of his first pictures was a long exposure of a Lego figure, which won him the runner-up prize in an Adafruit competition. "I wanted to use my lenses to improve on that."

After discovering a design for a Canon lens adapter for the Pi by Charmlee (**magpi.cc/1MhDNJl**), James got a 3D print made at his local Berlin printing cafe and was impressed: "Everything simply slots together. Actually quite a nice design, to be honest, but it only works with Canon EF (EOS) lenses." He soon realised that using the adapter with the Pi Camera Module requires the removal of the latter's tiny stock lens. "It really is simple… it just screws off. What might be considered difficult is that it has a little glue inside to stop it from moving. So I used a couple of pairs of pliers to remove it." While this did buff the plastic slightly, the lens can go back on to protect the sensor when storing the camera.

The Pi camera sensor can then be inserted into the 3D printed adapter, enabling it to be exposed directly to light coming through the attached Canon lens. Since the sensor only



The final image, post-Photoshop. The pink tinge on the original unedited image (bottom-left) was caused by a layer on the Pi camera sensor



Above All the components James used for the project, including the red 3D printed lens adapter in three parts, which he later wrapped in duct tape

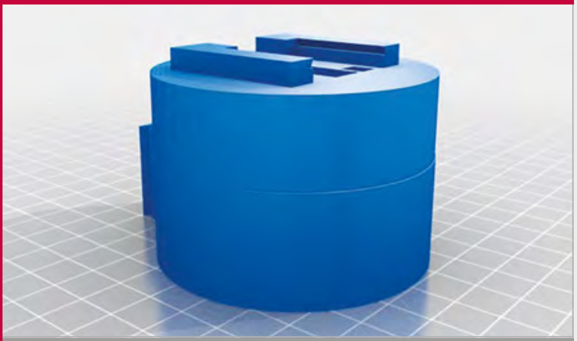
captures a small section of what the lens can see, it does so with 5 megapixels' worth of detail. This led James to try shooting a full moon from his Berlin balcony, using a 78-300mm lens, which resulted in a surprisingly detailed image of the cratered lunar surface.

Asked what camera settings he used, James replies, "I'm embarrassed to say I left it on auto. In fact, the command I used had no modifiers at all – `raspistill -o moon.jpg`. What I got really was a happy accident. Could I improve the picture with changing some settings? Absolutely!… I think I need to learn more about how the optics work. Maybe I could adjust the distance between the lens and the sensor to improve the sharpness."

James admits that a DSLR would be better for taking a picture like this, "but only with a longer lens, something larger than 300mm, or even a telescope." One peculiarity in his moon photo is the pink vignette around its edges, caused by light reflecting on a layer of the Pi camera sensor. He has since managed to reduce the effect by wrapping the camera adapter in black duct tape.

As well as being useful for astronomy images, James's setup is ideal for any type of photo or video requiring a high level of zoomed-in detail. "You could use [it] to monitor birds' nests that might be quite a distance away [or for] macro photography with an assortment of lenses and adapters."

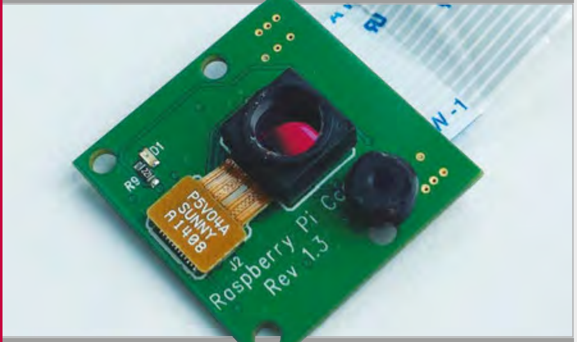
BUILDING A MOON CAMERA



>STEP-01

3D print the adapter

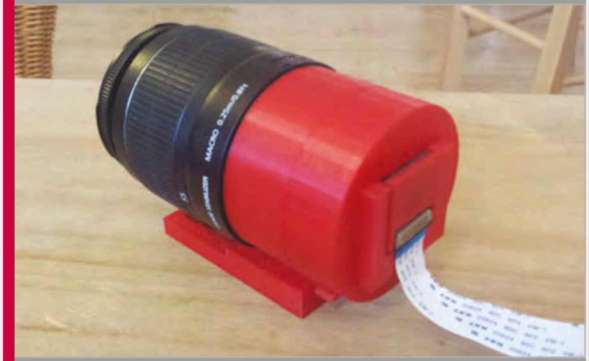
The Canon lens adapter was 3D printed from a design found on Thingiverse.com, by Charmlee (**magpi.cc/1MhDNJl**). Its three parts – the main body, base, and rear panel – simply slot together.



>STEP-02

Remove Pi camera lens

To enable the Pi Camera Module's sensor to be exposed directly, its tiny stock lens needs to be removed. It can be unscrewed using small pliers – do so at your own risk!



>STEP-03

Fit it all together

The Canon EOS lens screws onto the front of the adapter body, while the Pi camera sensor slots into the rear panel. You're now ready to shoot the moon, or any other subject!

MARCIO ROVAI

Marcio is a Brazilian engineer living in Chile, who recently rediscovered his passion for electronics. He dedicates his spare time to IoT and robotics.
magpi.cc/25ZqtzN

MARS ROVER

If you're going to build a robot, why not aim for the stars and build your own Mars Curiosity Rover?

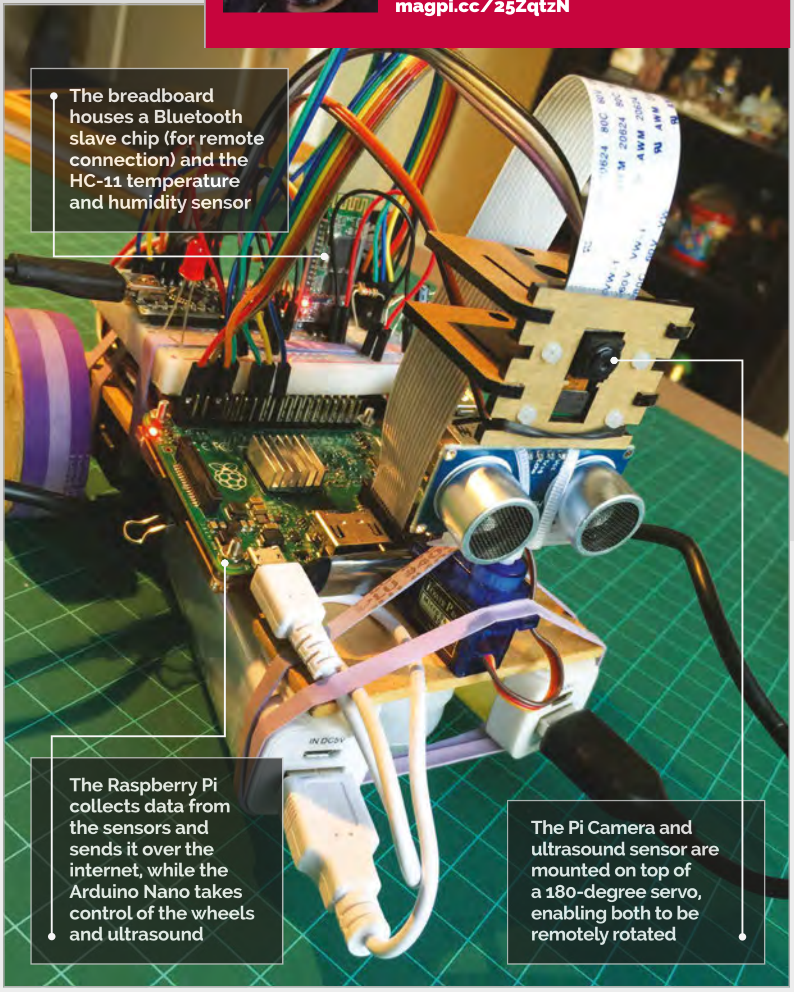
Quick Facts

- The total cost of parts was £120
- The total cost of Mars Curiosity was $2.5 billion
- The project uses three separate batteries
- The only sensors it's missing are wind and pressure
- Mars is the only known planet inhabited solely by robots

NASA's Curiosity is one of the most famous robots on... well it's technically on Mars, but it's more famous here on Earth. We think this Curiosity-based build is out of this world.

Marcio Rovai used Curiosity as inspiration for his final project for Coursera's 'Create Your Own Internet of Things (IoT) Device' (**magpi.cc/25ZpvUa**). His Mars Rover – or 'Mobile Station for Environmental Data Capture', to use its official name – uses a Raspberry Pi to form the brains of an environmental data capture robot.

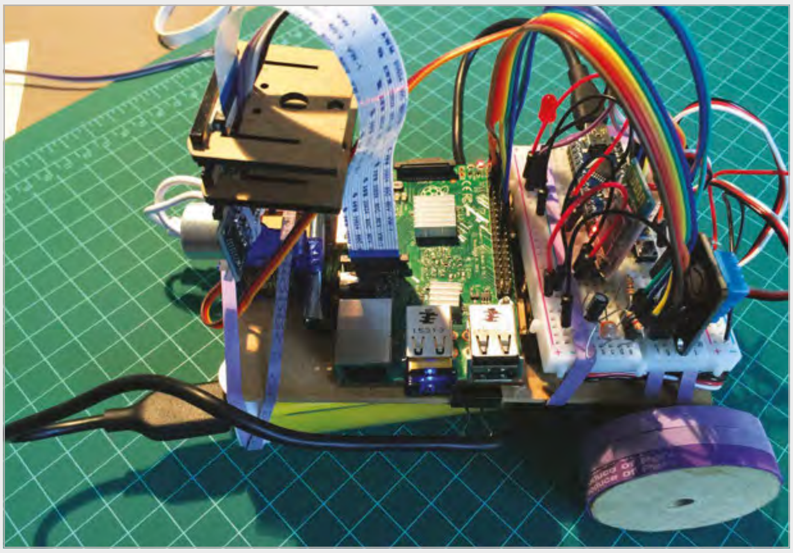
"It's a mobile station prototype responsible for environmental data capture, [such] as temperature, humidity, and luminosity," says Marcio. "The idea is to help students and hobbyists learn how a robot could be remotely controlled, and information captured and displayed over the internet."



The breadboard houses a Bluetooth slave chip (for remote connection) and the HC-11 temperature and humidity sensor

The Raspberry Pi collects data from the sensors and sends it over the internet, while the Arduino Nano takes control of the wheels and ultrasound

The Pi Camera and ultrasound sensor are mounted on top of a 180-degree servo, enabling both to be remotely rotated

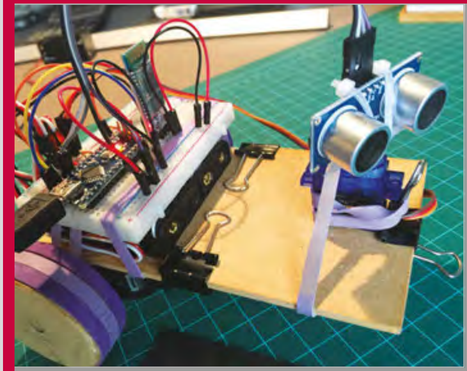


Right The breadboard enables all the sensors and components to be quickly attached to the prototype

"The Rover was based on the Curiosity," he tells us. "Both were developed for image and environment data collection." There are, of course, some physical differences (beyond one being slung to another planet). "The real Curiosity has six wheels that makes [it] all-terrain; my Mars Rover is a simple differential wheeled robot with two motors." Curiosity also has an articulated arm and drill for sample collection, which is missing on the Raspberry Pi Mars Rover. "But the basics are there," says Marcio.

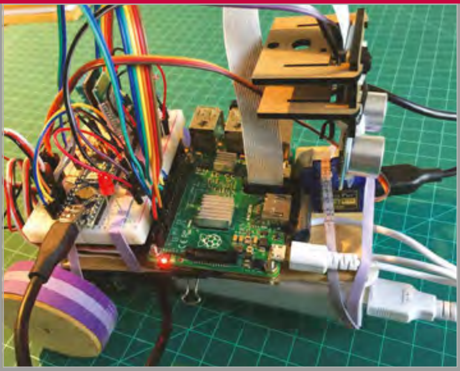
Mars Rover uses both a Raspberry Pi and Arduino. "The Pi is responsible for more complex tasks, such as web communication, sensor data capture, and live video streaming – leaving the core stuff like mobility, obstacle avoidance, and remote control with the Arduino."

BUILDING A MARS ROVER



>STEP-01
Arduino Nano

An Arduino Nano is used to control the servo motors and interact with the ultrasound range detector; this frees up the Raspberry Pi to focus on more important things.



>STEP-02
Taking charge

The Raspberry Pi handles the remote interaction and communication, and performs the data capture from a variety of on-board sensors. Data is sent directly to **ThingSpeak.com**.

>STEP-03
Mission control

An Android app is used to display the view from the Rover's on-board camera, and on-screen controls are used to issue commands to the Mars Rover.

"Controlling servo motors using PWM outputs directly from RPi GPIO isn't easy," he explains, "and it's best to have a HAT for that. So I decided to leave this job for my little friend, the Arduino Nano." The Nano's main tasks are to receive commands, control the servo motors, and monitor the ultrasound sensor for obstacle avoidance.

> The Mars Rover is controlled from an Android phone, which also provides a video stream

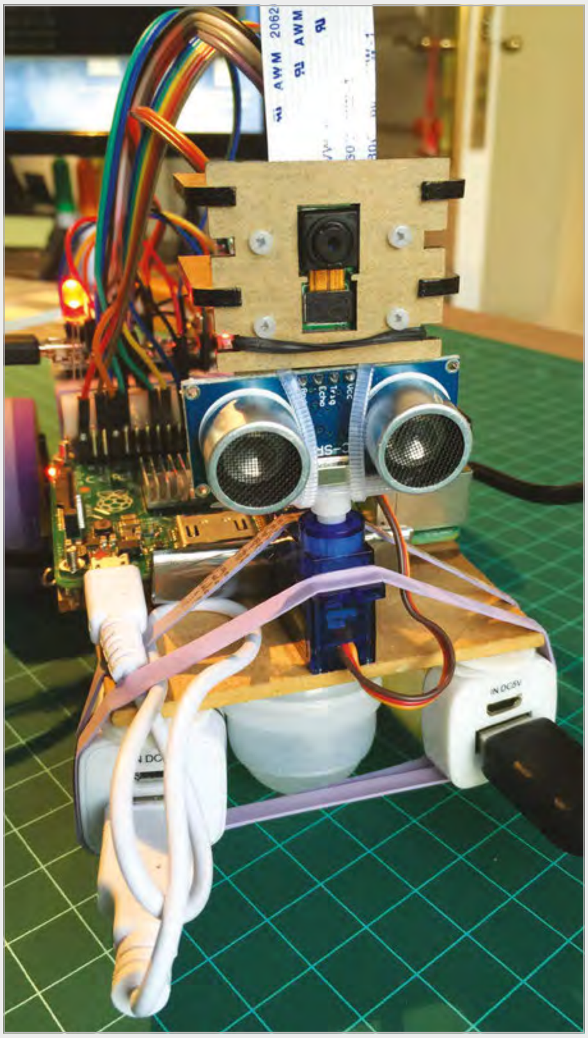
"Curiosity has a more advanced REMS (Rover Environmental Monitoring Station) system that incorporates the following sensors: atmospheric pressure, relative humidity, temperature (ground and air), ultraviolet radiation, and wind currents. So I included sensors for temperature, humidity, and luminosity (as a simulation of ultraviolet). My Rover doesn't capture wind and pressure data [like] the real Curiosity."

The Raspberry Pi Mars Rover is controlled from an Android phone, which also provides a video stream. Marcio used MIT AppInventor (**appinventor.mit.edu**) to create the app.

"The Rover isn't fully autonomous," admits Marcio, "as you must give [it] directions using the remote control. But I incorporated automatic obstacle avoidance capability, which will prevent a 'bad pilot' [from crashing it] on a rock, for example."

Data captured by the Mars Rover is sent from the Raspberry Pi to ThingSpeak (**thingspeak.com**), which turns it into visualisation. "One of the great advantages of IoT," says Marcio, "is the large range of public websites to record, analyse, and visualise captured data."

Marcio thinks it would be great to make it even more like NASA's Curiosity, including adding a robotic arm to do mechanical work and a more robust body. Solar panels are on the cards, too, although Curiosity is nuclear-powered; perhaps a step too far for realism.

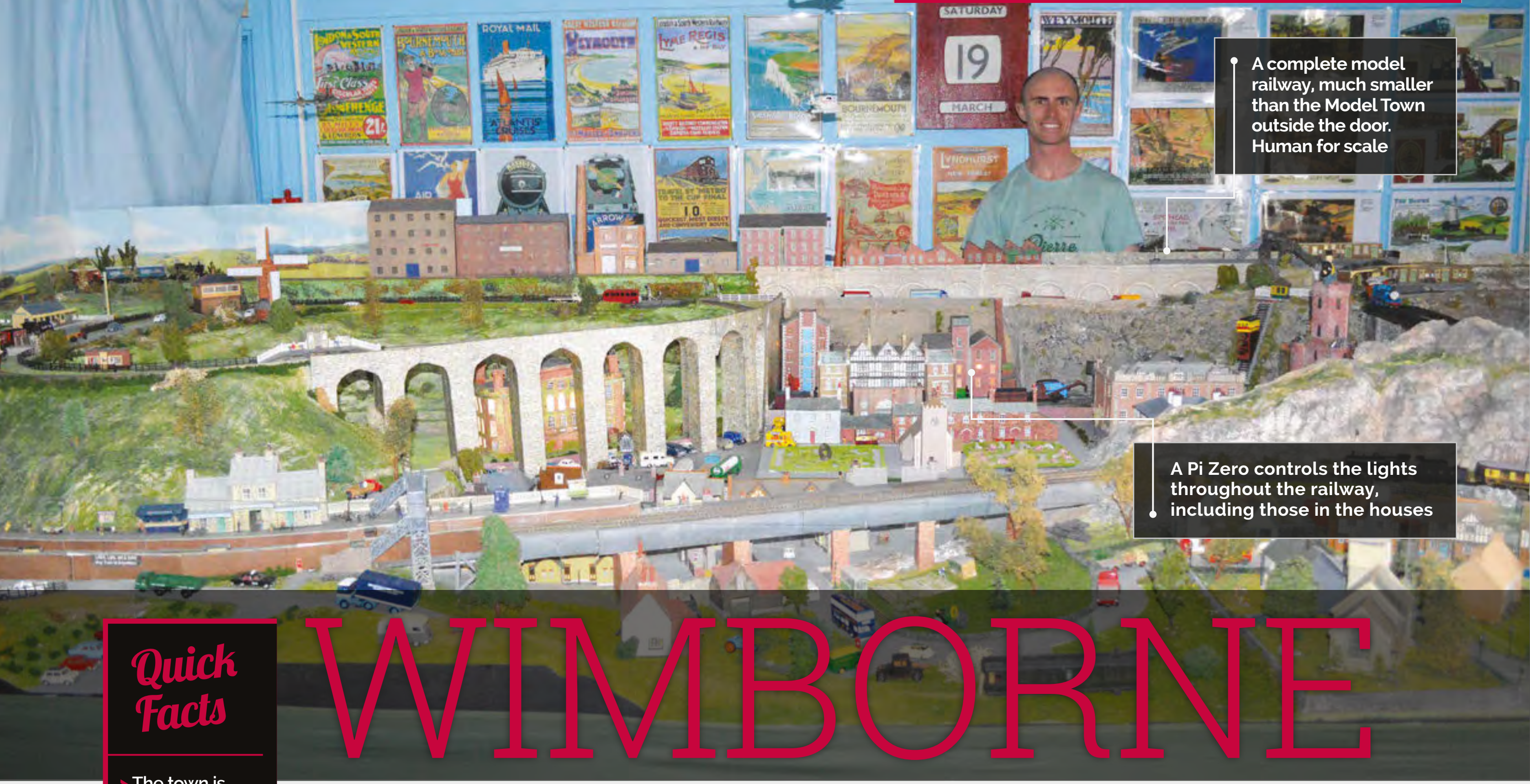


Above The finished robot looks ready for lift-off (or at least ready to explore the back garden)

WIMBORNE MODEL TOWN

An attraction in Wimborne, Dorset. A small team worked on the lights, with Terry Coles doing the coding.

magpi.cc/26eCNfu



Hidden from the eyes of the public is a strip of LEDs that simulate daytime for the residents of the model railway

A complete model railway, much smaller than the Model Town outside the door. Human for scale

A Pi Zero controls the lights throughout the railway, including those in the houses

WIMBORNE MODEL RAILWAY

A model town in Wimborne has a separate model railway attraction. When it came time to upgrade parts of it, a Raspberry Pi was the natural choice

Quick Facts

- The town is refurbished in the winter while it's closed
- Terry is a veteran coder
- The model railway itself is full of Thomas the Tank Engine figures
- More upgrades can be added in future
- Parts of the track are interactive for visitors

In the beautiful town of Wimborne, very close to where *The MagPi* is made, is a superb model town that has existed for almost 70 years. It's a lovely tourist spot and a wonderful attraction, of a type that's fairly rare these days. As part of the model town, there's also a great little model railway. During the off season just past, the lights that make up part of this miniature railway were upgraded using a Raspberry Pi, by a team led by Terry Coles, an engineering manager who helps out at Wimborne Model Town.

"The fundamental requirement for the lighting is simple," Terry tells us. "Simulate daylight, dim the lighting to simulate night-time, and then bring on lights in the trackside houses and streets. However, these were not the difficult goals; a bit more work was necessary because the finished system needed to be reliable, easy to use and maintain, and capable of being upgraded readily in the future."

After a lot of discussion, the team settled on a Raspberry Pi Zero for the task: "This allowed for future upgrades and was considered more accessible to the average volunteer than, say, using an Arduino."

The system isn't just made up of a Raspberry Pi, though: there's

The model town has been around since the 1950s, and offers an accurate view of the town from that era

Although Thomas the Tank Engine-themed, there are some other pop-culture cameos on the railway

much more to it. The lights being used weren't just a few LEDs stuck to a breadboard, as Terry explains:

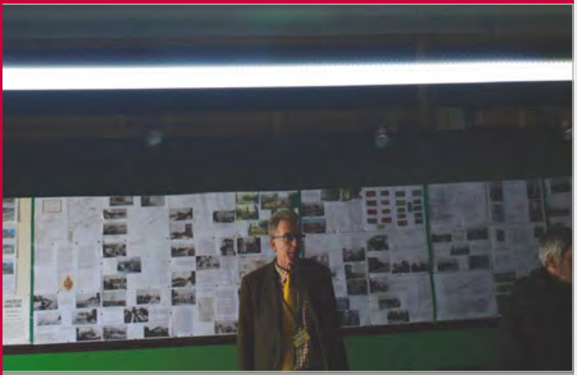
"The WiringPi2 library was used to allow programming of a pulse width modulation (PWM) signal. This dims and brightens a 48W LED strip on GPIO pin 18 (the hardware PWM pin) to simulate the 'Day' lights. A MOSFET driver was used to carry the current needed. The lighting in the houses and elsewhere was switched by four other GPIO pins via a relay board."

> "Reliable, easy to use and maintain, and capable of being upgraded readily in the future"

The system itself was developed on a Raspberry Pi with Raspbian, to make sure it was all working. Originally written in Python with the RPi.GPIO library, it doesn't quite support hardware PWM so was then changed to the aforementioned WiringPi2. The final code was then moved over to a piCore image for actual deployment. Terry says that as the volunteers will just turn the Pi on/off rather than do a software shutdown – it's a bit tricky with the way the system is built – piCore was the obvious choice, as the OS is loaded into memory and so there's no reading or writing going on from/to the SD card.

All the software written for the project is also open-source, with Terry maintaining very thorough documentation and sources that you can find here: **magpi.cc/1Nl7nhg**. If you ever find yourself on the south coast in or near Dorset, go give the model town and railway a look!

DAY & NIGHT CYCLE



>STEP-01

Daytime

During the 'day' portion of the cycle, a 48W LED strip is lit by the Pi (seen at the top of this photo) to light up the model railway. It's very bright, and illuminates the entire display for five minutes.

>STEP-02

Night-time

As the lights dim to signify the end of the day, lights in the village start to come on, one by one; first a few houses, then some streetlights.

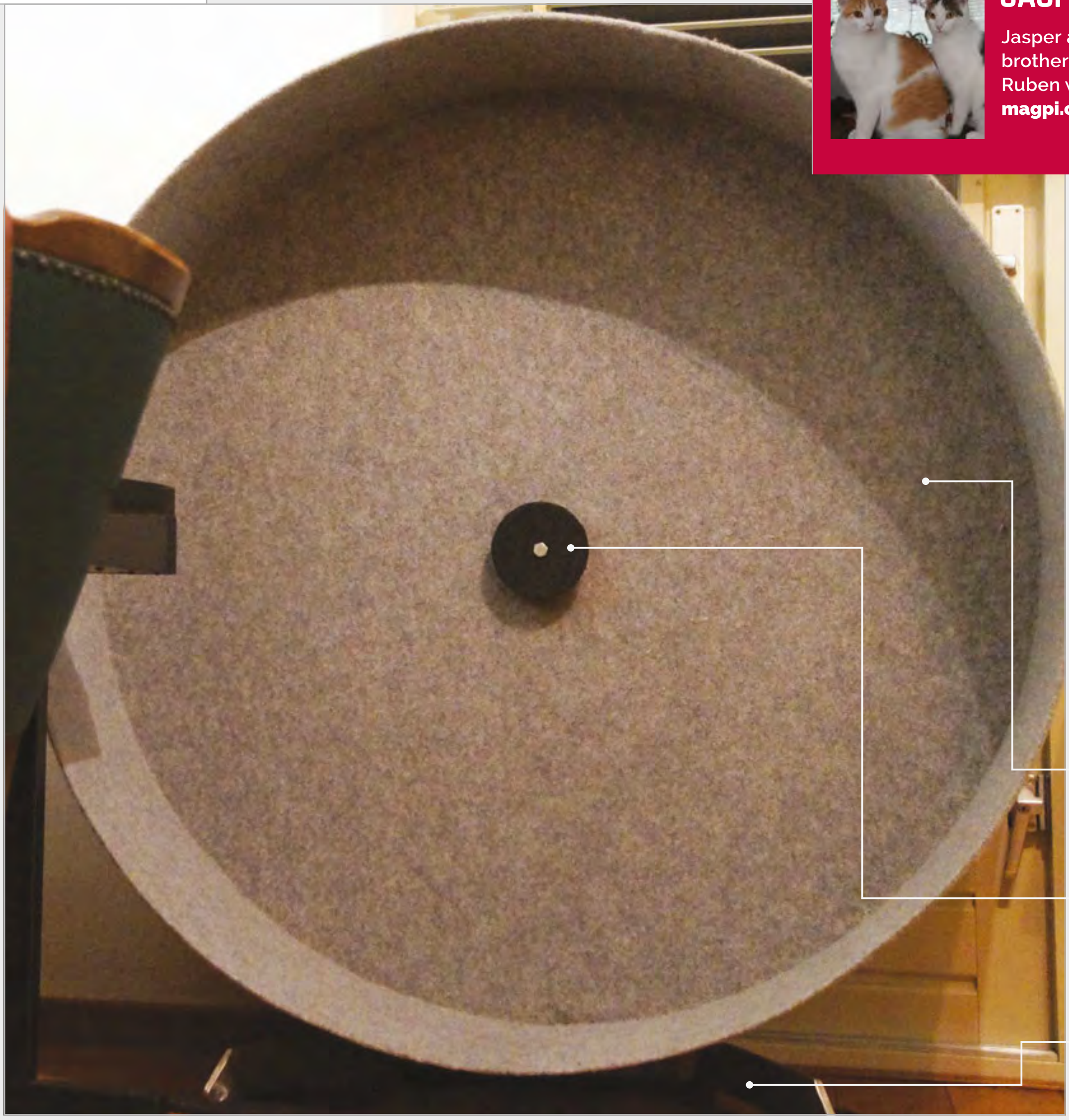
>STEP-03

When dawn comes

As the fake sun rises after three minutes, the lights in the town slowly start to turn off in order, as the town's diminutive population gets ready for the day.

JASPER AND RUBEN

Jasper and Ruben are six-year-old brothers. They like attention and Ruben will do anything for food.
magpi.cc/1Sz8J3T



Car felt is used to cover the wheel, so the cats are comfortable running on it

This disc (actually a CD) covers the nails and sensor used to detect the speed of circular motion

The wheel is held upright using a custom-built steel frame

CATWHEEL

Get your moggies moving with this giant hamster wheel for cats that monitors feline exercise

Quick Facts

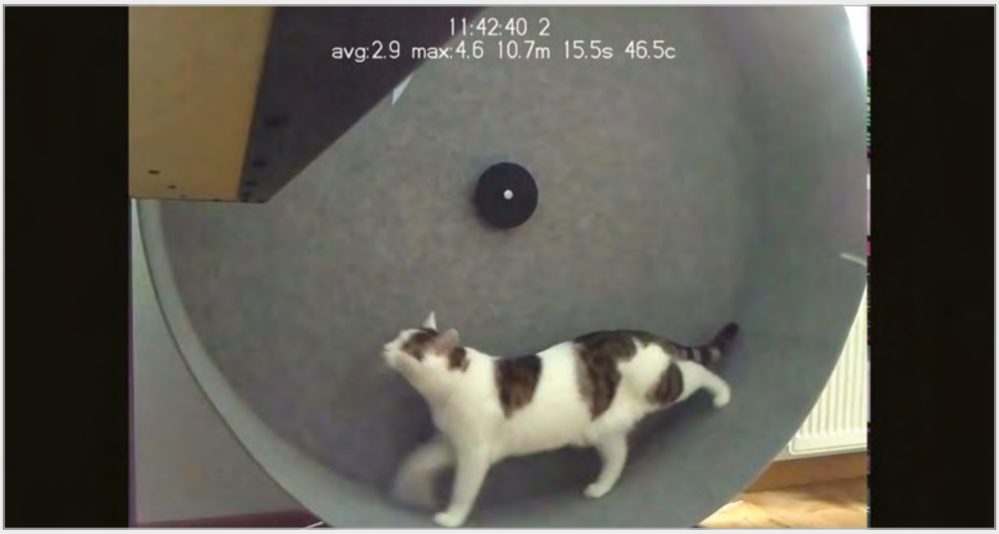
- The wheel's radius is 0.6m; the circumference is 3.77m
- The height, including the frame, is 1.35m
- The current speed record is 12.5km/h, set by Jasper
- The wheel has been in use for almost a year now
- Jasper and Ruben have travelled 42 kilometres

We love furballs here at *The MagPi*, and this giant hamster-style wheel for exercising cats is a strong contender for our 'pet project of the year' award.

Jasper and Ruben are two cool cats, with an even cooler owner (who doesn't want to be named). "[It's] the same as a hamster wheel," says Jasper's owner. "They can run and walk on it whenever they please." The proportions are a bit different: whereas most hamsters wheels are roughly twice the height of a hamster, this cat wheel is about three times a cat's height.

"The Raspberry Pi gets triggers from an electromagnetic sensor attached to the frame," he tells us. There are "six small nails under the lining on the back of the wheel, serving as actuators. Based on these triggers, activity on the wheel can be detected, and the speed is calculated."

The Raspberry Pi monitors the sensor using a GPIO port and on every trigger, the current speed is calculated. Jasper and Ruben were lured onto the wheel using cat treats. "We first let them get used to the wheel's movement, as standing on it causes it to swing. Then we just held the treats a little higher so they'd have to start walking to get near them. We'd reward them when walking by actually giving them the cat treat



Above A camera records the cat walking on the Catwheel and transmits the image, along with motion data

and some additional attention, which is what it's all about for them."

Both cats are now used to the wheel, and they've been using it for a long time. "They both do," says Jasper's owner, "at least twice a day, any time, night or day. Preferably when somebody is admiring them.

"They walk on it at about 3km/h and travel distances of over 250 metres," he continues, "often taking several short breaks, which are mostly caused by distraction. Sometimes they take a sprint, usually peaking at 8km/h. Overall, they perform equally – if there's a speed difference, it's too small to notice."

The build itself is also impressive. "The rear disc is a plywood plank, the cylinder is galvanised steel, and the surface of the entire wheel is covered in car lining. The entire frame is made of stainless steel, offering stability and necessary counterweight.

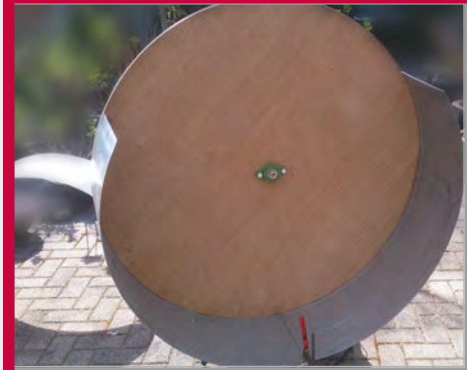
> "The Raspberry Pi gets triggers from an electromagnetic sensor"

He tells us it was a relatively easy project. "The only problem was moving it because it's large, bulky, and quite heavy. Having all the materials and necessary equipment in one place would have saved us a lot of time.

"I'd also like to be able to determine which cat is on the wheel," he adds, "maybe by analysing the colours in the picture. That way I'll be able to keep separate statistics."

If you're thinking of making a Catwheel, Jasper's owner has some good advice: "Make it as big as possible, so that the cats can walk comfortably. Use a central bearing system and high-quality bearings, so the wheel runs smoothly and quietly. Cats don't like noise, so a noisy wheel won't be very tempting to them. When it's finished and ready to use, be patient. Forcing a cat to do something will never work. Every cat is different, and it may just not be interested at all. It's a risk you should be willing to take."

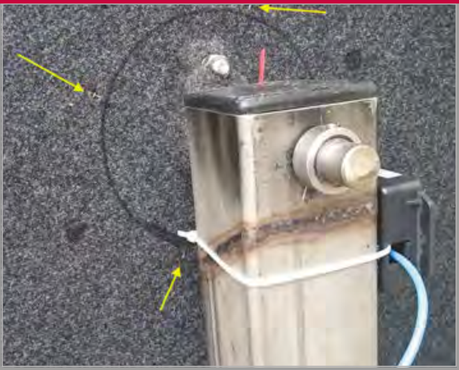
CATS ON WHEELS



>STEP-01

Building a wheel

Constructed from galvanised steel, the wheel offers enough weight to keep it stable when moving. It's curved into shape and covered in car lining, then placed into a stainless steel frame to hold it upright.



>STEP-02

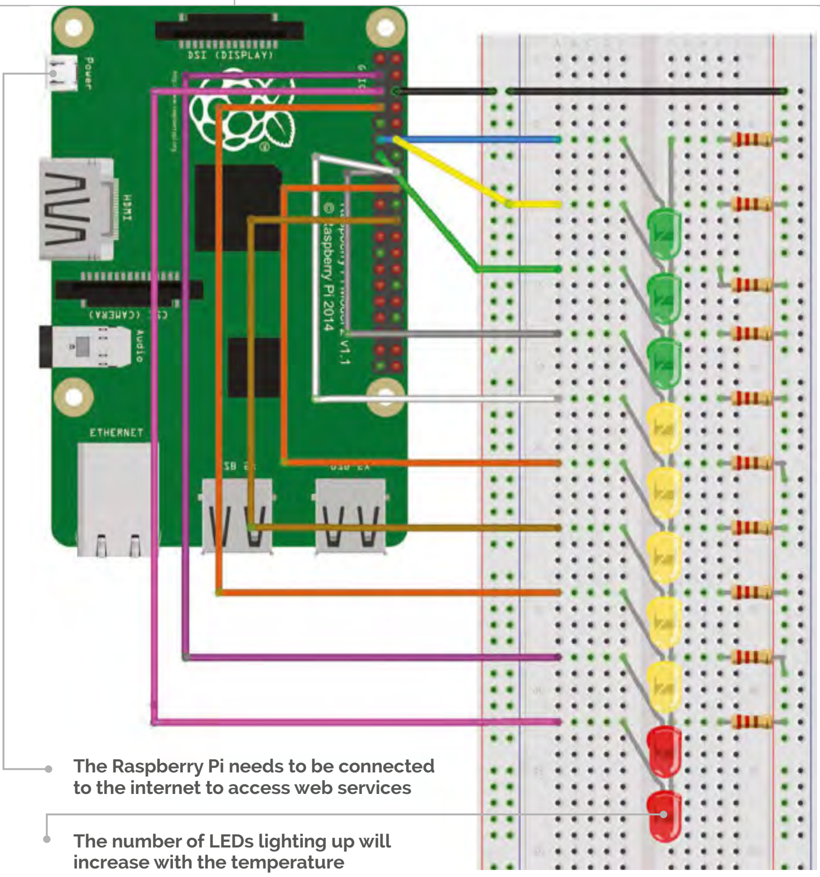
Nailed it

Six small nails serve as actuators. A sensor detects the movement of the nails. The Raspberry Pi monitors the sensor on a GPIO pin, and uses it to calculate motion and speed.

>STEP-03

Running around

The Catwheel automatically detects movement from the cats and starts to monitor the speed and distance. Motion is recorded to form an ongoing data capture of the amount of time they spend exercising.



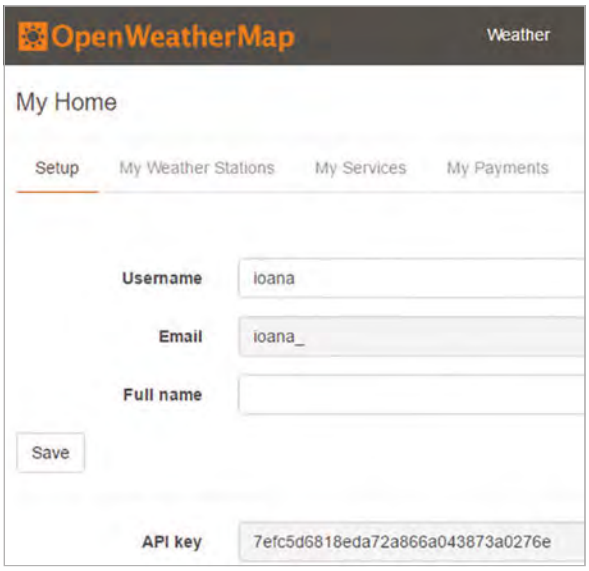
The Raspberry Pi needs to be connected to the internet to access web services

The number of LEDs lighting up will increase with the temperature

IOANA CULIC

Ioana is an IoT specialist, and is co-author of *Getting Started Guide for Raspberry Pi and Arduino Using Wyliodrin.*
wyliodrin.com

Below You need to copy the API key in order to use the OpenWeatherMap service



My Home

Username: ioana

Email: ioana_

Full name:

API key: 7efc5d6818eda72a866a043873a0276e

PI-THERMOMETER

Connect your Pi to existing web platforms and easily build your own internet-based thermometer out of LEDs

You'll Need

- Wyliodrin STUDIO **goo.gl/Sgj9HB**
- 3× Green LEDs
- 5× Yellow LEDs
- 2× Red LEDs
- 10× 220-ohm resistors
- Breadboard and jumper wires

How often do you check your browser or smartphone to see how cold it is outside? Nowadays, we read the temperature using applications that take data from the internet and display them in a nice interface. This reduces the interactivity and simplicity that old mercury thermometers had, while offering the great flexibility of reading the temperature anywhere on the globe. But what if you could have the best of both worlds? You can put your Raspberry Pi to use and replace the computer with a thermometer-like device, which has the capability of displaying the temperature at whichever location you wish.

>STEP-01

Set up the Raspberry Pi

Wyliodrin STUDIO is an IDE (integrated development environment) that makes it very easy to create complex applications for your Pi. First of all, you need to download it (**goo.gl/Sgj9HB**) and follow the first two steps of the Getting Started tutorial for the Raspberry Pi. Once you have an SD card with the extra service installed, you need to connect the Pi to the internet, in the same network as your computer. As soon as the board has booted, you will see it available under Port. Press Connect and you're ready to go.

>STEP-02

Create a new visual application

In order to create the thermometer, press the 'projects' button and create a new application. Name it 'pi-thermometer' and select 'visual' for the programming language. Open the newly created app and you'll see the frame in which you need to connect the visual elements. The blocks gallery is on the left. We have chosen this way of programming because it allows you to rapidly create the code, and it doesn't require you to know any specific programming language. In addition, if you're familiar with Python, you can press the 'Show code' button and see how the code gets generated as you drag and drop the visual elements.

>STEP-03

Build the thermometer

From the hardware point of view, you simply need to connect ten LEDs to the GPIO pins of the Pi. You can visualise the number assigned to each pin by selecting the ‘Pin layout’ tab in Wyliodrin STUDIO. We have chosen to connect the LEDs on pins 0 to 9. We will scale the read temperature and light up a variable number of LEDs accordingly.

>STEP-04

Weather API

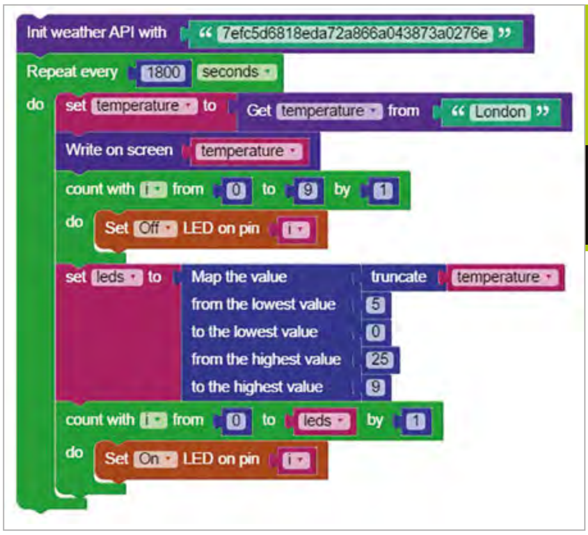
The website **openweathermap.org** offers a complex API that allows you to get weather details from cities around the world. To access the API, you need to sign up for a key at **openweathermap.org/appid**. You only need to provide your email and choose a password. Once you have created an account, you can use the API key available at **home.openweathermap.org**.

>STEP-05

Writing the application

The application reads the weather from the city of your choice and maps the value to the number of LEDs. The string used within the ‘Init weather API’ block is the key you got in the previous step.

In order to do the mapping, we specified the average minimum and maximum temperature values according to the current time of the year. It’s recommended that you use a reduced range, to make the differences in temperature visible on the thermometer.



```
Init weather API with " 7efc5d6818eda72a866a043873a0276e "
Repeat every 1800 seconds
do  set temperature to  Get temperature from " London "
    Write on screen temperature
    count with i from 0 to 9 by 1
    do  Set Off LED on pin i
    set leds to  Map the value              truncate temperature
                 from the lowest value      5
                 to the lowest value        0
                 from the highest value     25
                 to the highest value       9
    count with i from 0 to leds by 1
    do  Set On LED on pin i
```

If you would like to see the Python code that will get executed once you run the project, you can select ‘Show code’.

>STEP-06

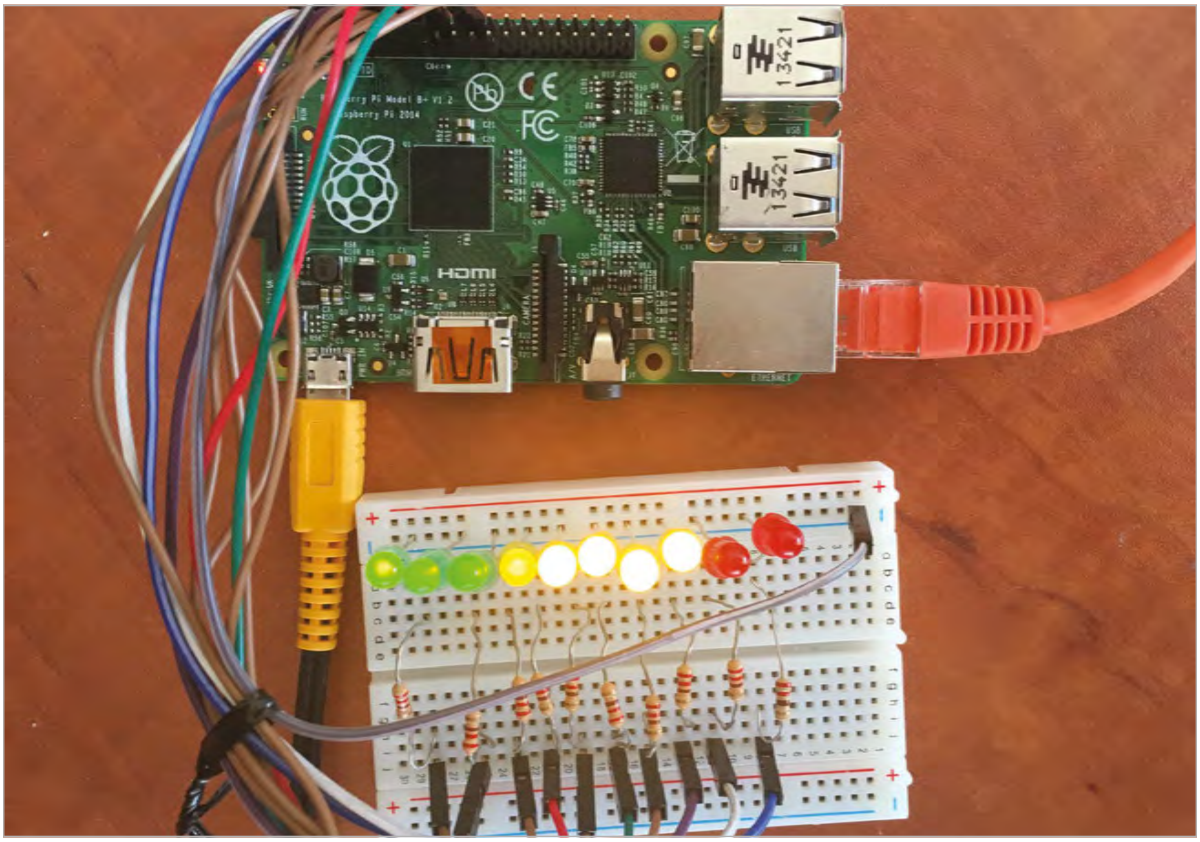
Check the thermometer

All that’s left to do is to run the application. As long as you don’t stop it, the device will read the temperature from the internet every 30 minutes and update accordingly. You could add more LEDs to make it suitable for a larger range of temperature values. We suggest you also put the device in a box so that it looks just like a thermometer. Now you’ll know how warm it is outside with just a turn of your head.

Language

>VISUAL PROGRAMMING

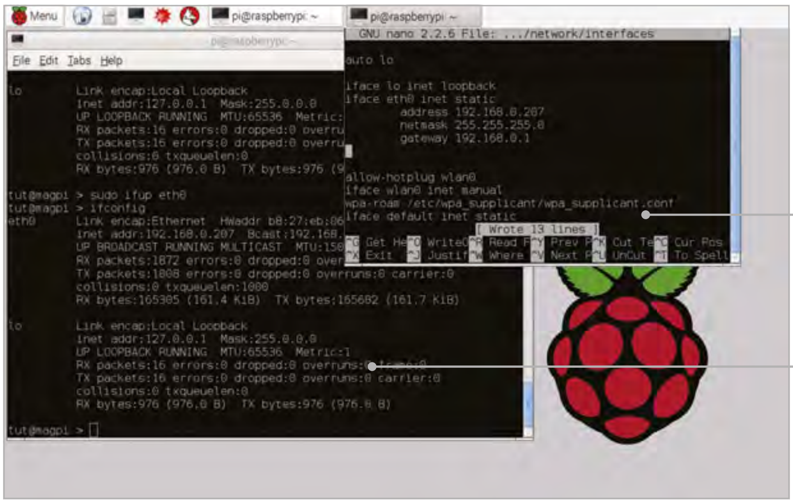
DOWNLOAD:
magpi.cc/1S8XD60



Left The thermometer displaying a temperature of 15° Celsius

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



```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:
          RX packets:16 errors:0 dropped:0 overru
          TX packets:16 errors:0 dropped:0 overru
          collisions:0 txqueuelen:0
          RX bytes:976 (976.0 B)  TX bytes:976 (9

tut@magpi > sudo ifup eth0
tut@magpi > ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:06
          inet addr:192.168.0.207  Bcast:192.168.
          UP BROADCAST RUNNING MULTICAST  MTU:150
          RX packets:1872 errors:0 dropped:0 over
          TX packets:1808 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:165305 (161.4 KiB)  TX bytes:165682 (161.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:976 (976.0 B)  TX bytes:976 (976.0 B)

tut@magpi >
```

```
  GNU nano 2.2.6 File: .../network/interfaces

auto lo

iface lo inet loopback
iface eth0 inet static
        address 192.168.0.207
        netmask 255.255.255.0
        gateway 192.168.0.1


allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet static
                    [ Wrote 13 lines ]
^G Get He^O WriteO^R Read F^Y Prev P^K Cut Te^C Cur Pos
^X Exit  ^J Justif^W Where  ^V Next P^U UnCut ^T To Spell
```

/etc/network/interfaces contains an entry telling the Pi to use DHCP to allocate an address. Change it to static, and give your board a permanent address

ifconfig gives you your internet address; unless your Pi is sitting in a data centre, this is likely to be a private address such as one in the range beginning 192.168.0.0

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

PREDICTABLE NETWORKING

Give your Raspberry Pi a permanent network address of its own…

They say that the network is the computer: something becoming more vital as connected devices integrate more and more real-world interactivity with data and processing across the internet. Raspbian takes care of automatically connecting in most situations, but sometimes you need to override automatic configurations, to ensure a consistent network setting for your Pi project: Raspbian has the tools, and we'll show you the essentials you need to stay connected.

Plug a network cable from your **ADSL** router to your Pi and, automagically, Raspbian knows where it is on the network, and can talk to the outside world. How? Because it's setting itself up the way your router tells it to. This is thanks to **DHCP** - Dynamic Host Configuration Protocol – which provides network configuration for every device connected into a network.

Check you're using it by issuing the command: **`cat /etc/network/interfaces`**. This should show, amongst others, a line like **`iface eth0 inet dhcp`**, which means you're set up to receive configuration via DHCP**.** The eth0 is the name of your Ethernet port into which you plug your network cable; a wlan0 here is for any WiFi adaptor you may connect to your board.

For many projects on your Pi – whether something cutting edge with sensors, or something as useful but mundane as a VPN to protect your mobile browsing from intrusive location-based advertising – you'll need an internet address that doesn't change, so that whatever remotely connects to you can always find you. We're using local addresses here, although it is possible to remotely access a Pi behind a router.

Static address

Network settings are found in **/etc/network/interfaces**; before changing them, make a backup copy of your current working setup with:

```
sudo cp /etc/network/interfaces /etc/
network/interfaces.bak
```

Open **/etc/network/interfaces** with your favourite editor (or nano) and change **iface eth0 inet dhcp** to **iface eth0 inet static** (and the later mention of **dhcp**, under default, if you have it in your config file). Then add the following information below it – or rather, a version appropriate for your network:

```
address 192.168.0.207
netmask 255.255.255.0
gateway 192.0.0.1
```

Here, 'address' is either the IP (version 4) address already allocated by DHCP to your Pi, or one with the last of the four numbers (together they're known as a dotted quad) changed to a higher number, which is unlikely to be allocated by your router's DHCP server. You can usually configure your router's range of addresses allocated, or reserve particular addresses.

IPV6

Despite the world of connected devices crying out for the billions of addresses of IPv6, IPv4 remains the norm. Adding IPv6 to the Pi simply involves putting **ipv6** on a line by itself at the end of **/etc/modules**.

The gateway address is the address of the device sitting between you and the internet: your ADSL router. You can discover this with **`sudo route -n`** where the **`-n`** is requesting the numeric address, rather than the name of the gateway. The address the Pi is currently using is shown with **`sudo ifconfig`**, as is netmask, which is listed as Mask.

If you look at example **/etc/network/interfaces** files online, or from other people's projects, you may also see entries for network and broadcast, showing the first three numbers of the dotted quad address of the device, followed by a 0 and a 255, respectively. These tell your computer a little more about the network that it is on, but aren't strictly necessary to getting things working. After editing the network configuration you need to restart the service:

```
sudo ifdown eth0 && sudo ifup eth0
```

Check that all is now well, both with ifconfig and trying a ping from another computer on the network.

Ping!

Ping is the most basic tool in the network testing armoury, but one which is often called upon. It sends **ICMP** (the Internet Control Message Protocol) **ECHO_REQUEST** to a device on the network. ICMP is built into every connected device and used for diagnostics and error messages: a ping will produce a reply from the pinged machine, which tells you it is on, and connected, and that the network is working between you and it. Information about packets lost, and time taken, also helps with fault diagnosis.

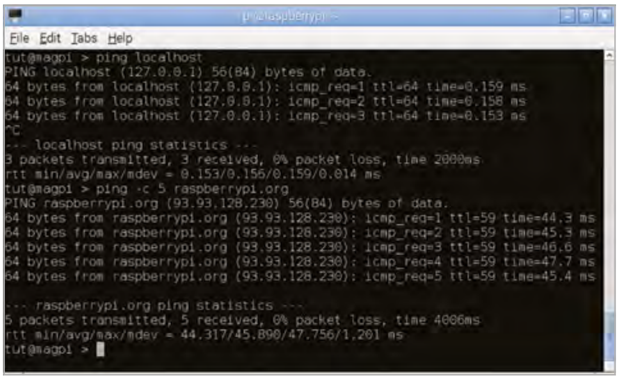
A successful ping localhost from the Pi tells you not just that the local loopback interface is working, but that localhost resolves to **127.0.0.1**, the local loopback address. Name resolution is the cause of many computing problems, so we'll come back to that further down the page.

Now ping the Pi from another machine on your local network: **`ping 192.168.0.207`** – you'll need to use the static IPv4 address that you set, rather than ours, of course. If you're doing this from a Windows machine, ping defaults to five attempts; from another Unix machine (another Pi, a Mac, or Ubuntu or other GNU/Linux), it will carry on until you stop it with **CTRL+C**, unless you set a number of **ECHO_REQUEST** sends with **`-c`** like so: **`ping -c 5 raspberrypi.org`**

What's in a name?

Numeric addresses are how connected devices see the internet, but humans prefer to remember names, so **DNS** – the Domain Name System – exists to translate (resolve) addresses like raspberrypi.org into something your PC can find on the internet, such as **93.93.130.214**.

The file **/etc/hosts** contains mappings of host (computer) names to IP addresses for local network computers without publicly registered addresses; typically, it will only have localhost mapping to **127.0.0.1**, unless you add further addresses. For the rest of the internet, a DNS server is queried: if the address of your ADSL router is listed in **/etc/resolv.conf**, then your router is getting DNS details from your ISP and passing them on with DHCP, having told the Pi to ask the router.



```
tut@magpi > ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.159 ms
64 bytes from localhost (127.0.0.1): icmp_req=2 ttl=64 time=0.158 ms
64 bytes from localhost (127.0.0.1): icmp_req=3 ttl=64 time=0.153 ms
^C
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.153/0.156/0.159/0.014 ms
tut@magpi > ping -c 5 raspberrypi.org
PING raspberrypi.org (93.93.128.230) 56(84) bytes of data.
64 bytes from raspberrypi.org (93.93.128.230): icmp_req=1 ttl=59 time=44.3 ms
64 bytes from raspberrypi.org (93.93.128.230): icmp_req=2 ttl=59 time=45.3 ms
64 bytes from raspberrypi.org (93.93.128.230): icmp_req=3 ttl=59 time=46.6 ms
64 bytes from raspberrypi.org (93.93.128.230): icmp_req=4 ttl=59 time=47.7 ms
64 bytes from raspberrypi.org (93.93.128.230): icmp_req=5 ttl=59 time=45.4 ms

--- raspberrypi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 44.317/45.890/47.756/1.201 ms
tut@magpi > 
```

Left The humble ping is a useful diagnostic tool, telling you if a connection is working, a device is connected, and DNS is resolving

Without DHCP to tell the Pi this, it's a good idea to add an entry for DNS servers to **/etc/network/interfaces** below the information we added earlier:

```
dns-nameservers 8.8.8.8 8.8.4.4
```

Here we've added Google's public DNS servers, which work reliably from anywhere. Other public DNS servers are available, some of which may guarantee not to log your IP address when you use them, or get your ISP's DNS server addresses from their FAQ or help website.

MyPiName

The line **127.0.1.1 raspberrypi** found in **/etc/hosts** is there for compatibility with some software which expects the hostname to resolve. If you changed to a static IP address, then you can change **127.0.1.1** to that new static address, too.

Should you wish to have your Pi change its name, then change the raspberrypi side of this entry too, as well as changing from raspberrypi in **/etc/hostname**. After this change of identity, a system restart would be a good idea. **`sudo reboot`** is essentially a pseudonym for **`sudo shutdown -r 0`**, which will securely reboot your system from the command line. Replacing the **`-r`** with an **`-h`** will just shut the Pi down, as will **`sudo halt`**. However, **`sudo /etc/init.d/hostname.sh`** will tell Raspbian's Linux kernel about the new name without the need to reboot the Pi.

Lastly, to add a **.local** domain to your devices on the local network, use apt-get to install avahi-daemon for mDNS (multicast Domain Name Servicing), using Apple's Bonjour, which works on most platforms. You should now be able to connect to **<yournewPiname>.local** from other Bonjour-enabled computers on your network without even needing to use the static IP address you allocated.

COMMAND LINE WIFI

If you add WiFi to your Pi, Raspbian is ready, but the Wicd Wireless Network Connection Manager also has command-line clients for both interactive and scripting use: **`apt-cache search wicd`**

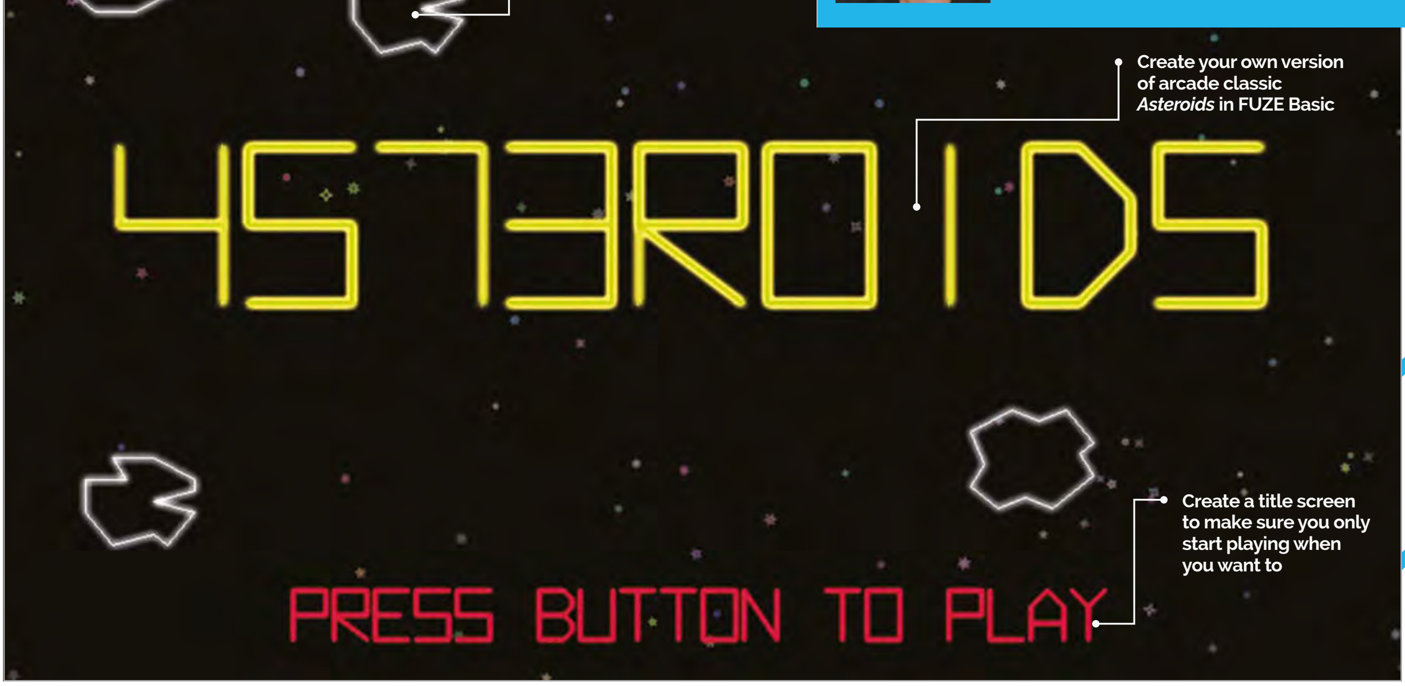
FREE / PUBLIC DNS

As well as dynamic DNS providers, some of those found listed at **FreeDNS.com** offer public DNS servers. For a wider listing of alternatives to Google's DNS servers, have a search on Google itself.

JON SILVERA

CEO and founder of FUZE Technologies, Jon Silvera is keen to help kids learn how to code.

fuze.co.uk



You can shoot up the asteroids in this version, but there's no collision detection yet

Create your own version of arcade classic *Asteroids* in FUZE Basic

Create a title screen to make sure you only start playing when you want to

MAKE AN ASTEROIDS CLONE IN FUZE BASIC

PART 01

Learn how to start making a full game in FUZE BASIC in part one of this Asteroids tutorial by the founder of FUZE himself, **Jon Silvera**

You'll Need

- Raspberry Pi with updated Raspbian
- FUZE BASIC **magpi.cc/1SbgYsY**

Asteroids is an arcade classic, one of the first really popular video games that has endured to this day. It's a very common game to make to show off the capabilities of a programming language, and FUZE BASIC is no different.

In the first part of this tutorial, we'll show you how to start making your own *Asteroids* clone in FUZE BASIC for Raspberry Pi, with a ship that moves and fires as you want it to.

The first thing you'll need to do is download the code so we can work through and explain how it works: **magpi.cc/1Sbhd7m**.

If you haven't already done so, download and install FUZE BASIC on your Raspberry Pi. Visit **magpi.cc/1SbgYsY** for the files and instructions. Next, start FUZE BASIC and either load (with **F8**) **4573R01D5part1.fuze**, or copy and paste the code straight into the editor – **F2** switches between the editor and immediate mode.

Run the program by pressing **F3** or typing **RUN** in immediate mode. This way, we can see how the end product looks before examining the code.

Control the player ship using the **Z** and **X** keys to rotate, the **RETURN** key to fire and the **RIGHT SHIFT** key to apply thrust. At this early stage, there's no collision, enemy ships, scoring or sound.

Press **ESC** to stop the program, then **F2** to bring up the editor. Follow along with the code to get an understanding of how it works.

Behind the scenes

After a few setup procedures, we have a nice and tidy **LOOP** at the beginning, a **REPEAT**, and then an **END**. It would be great if that was it, but there's much more!

The first **PROC** is **playerControls**. Procedures are a bit like objects or classes, but they are embedded in the main program. Here, we check for key presses; if they are pressed we rotate, apply thrust, and fire bullets.

There's a little bit of maths to explain in the thrusting section. If you're new to **COS** and **SIN**, then you have some work to do, but basically these allow you to calculate a position around the circumference of a circle. In this case, we add the **COS** and **SIN** to **newpX** and **newpY** using the current angle the player is facing.

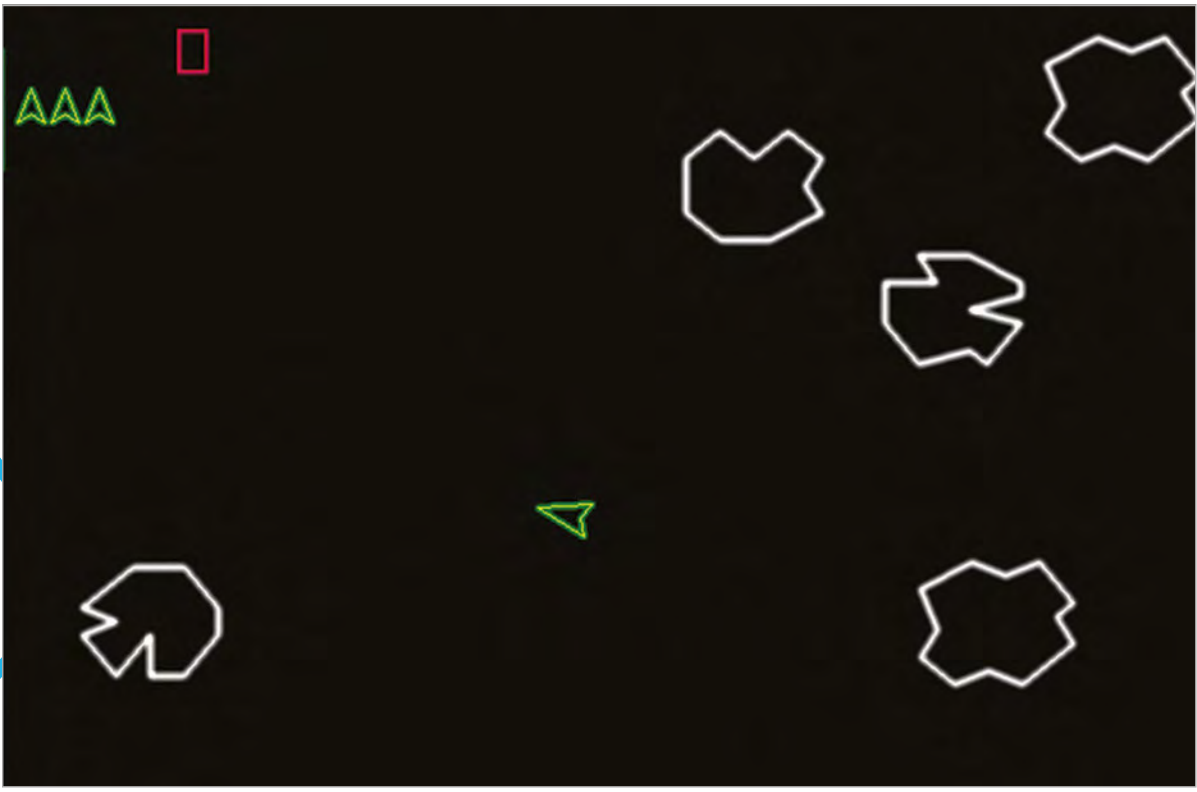
NO ASSETS

For the moment, FUZE draws all the graphics rather than using pictures, just like the original version of *Asteroids*.

Language

>FUZE

DOWNLOAD:
magpi.cc/1Sbhd7m



Left The code takes your ship's orientation into account using trigonometry, so that bullets can be fired in the correct direction

After this, we need to apply some damping to the movement, otherwise it will continue moving forever. This is, of course, correct in 'real space', but it makes for a bad game control mechanism!

Now, we add the new calculated offsets to every point used in the player ship – if rotating, this must be applied too. It looks complex, but we just work out the new x and y position for each point in the **`playerData`** array.

The fire button routine makes sure we have only five (**`maxPlayerBullets`**) bullets in use at once, and uses **`TRUE`** or **`FALSE`** to block the player from just holding the button down.

> Procedures are a bit like objects or classes, but they are embedded in the main program

Next, we check to see if the player has gone off the edge of the screen and if so, wrap it round to the other side.

Rest of the stage

The **`resetRocks PROC`** sets up the rocks for each new level. The number of rocks starts at five and increases by one per level. No more than 15 rocks are allowed. **`rDirX`** and **`rDirY`** determine the speed, and **`rAngle`** the direction. The speed increases with each level.

`PROC initialisePlayer` is particularly useful, as it allows the rocks to move around without showing them. It calculates a safe area around the player ship, so that we always start with time to act and not with a random rock dumped on top of us before we even get a chance. **`PROC resetPlayer`** sets up the player variables at the start of a new level, so they always start in the middle, are not firing, and are facing in the right direction. **`PROC rocks`** checks to make sure a rock is active, calculates its position, checks if it's off screen, and then draws them. Some of this won't make sense right now as we can't collide with them, but next month we'll have different sizes and types to work out.

The **`drawPlayer PROC`** simply plots lines between all the coordinates in the **`playerData`** array. The next routine, **`PROC drawScreen`**, displays the score and the number of lives remaining, using a line plot routine to display the text in vectors.

Next are some routines to set up new bullets and draw them. Again, there is a section to check to see if we need to wrap them around the screen.

The **`setup PROC`** prepares everything, sets system variables like the screen resolution, and contains the data for drawing the four different-shaped rocks and the player ship.

The function **`DEF FN distance`** is used to return a distance between two points. The last two **`PROCs`** draw and set up the vector font.

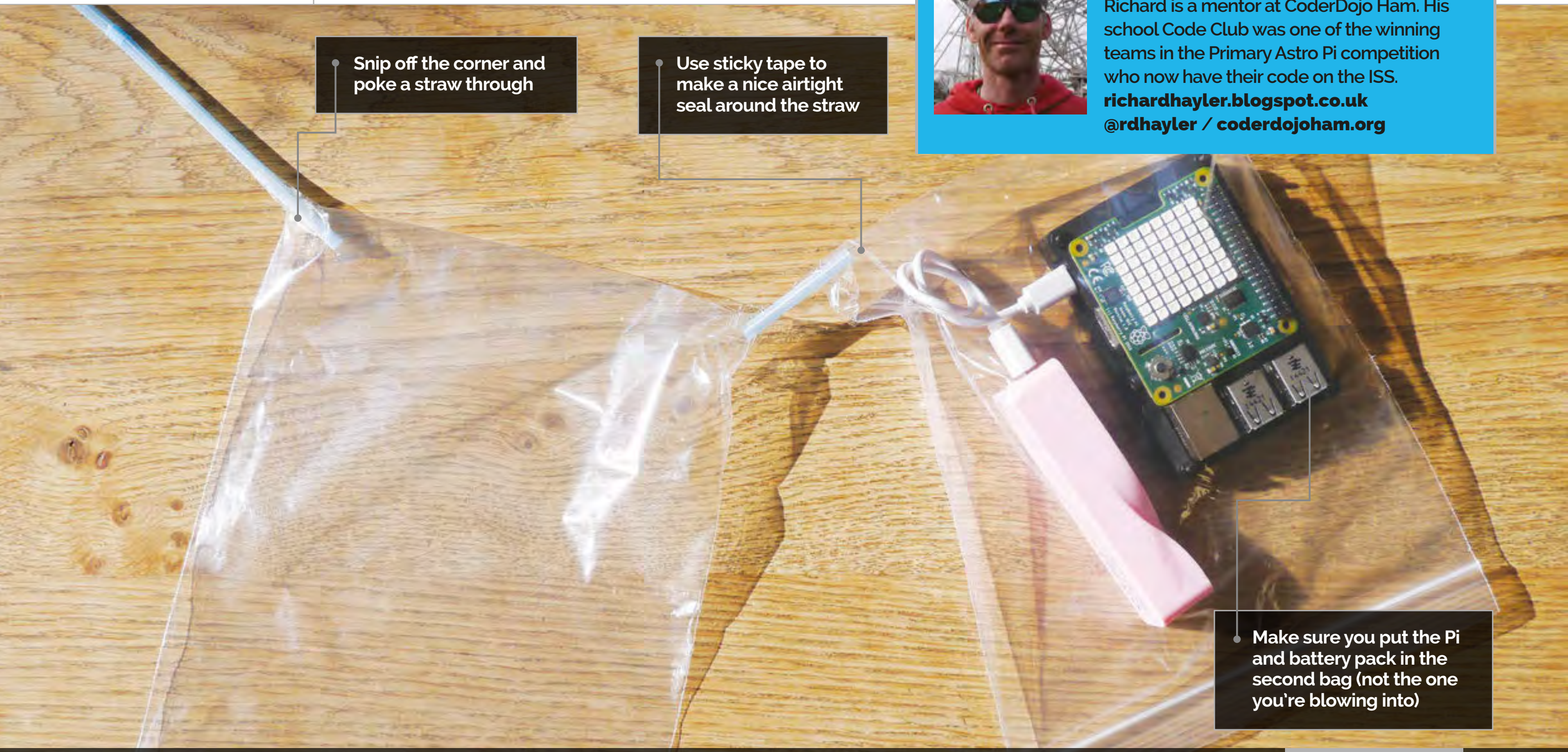
Next month, we'll finish the game by adding many of the trappings you'd expect from an old-school arcade.

SIMILAR GAMES

Can you figure out how to use the code to make a simple version of *Pong* in FUZE BASIC?

RICHARD HAYLER

Richard is a mentor at CoderDojo Ham. His school Code Club was one of the winning teams in the Primary Astro Pi competition who now have their code on the ISS.
richardhayler.blogspot.co.uk
@rdhayler / coderdojoham.org



Snip off the corner and poke a straw through

Use sticky tape to make a nice airtight seal around the straw

Make sure you put the Pi and battery pack in the second bag (not the one you're blowing into)

SENSE HAT SCIENCE: PART 03 HUMIDITY

Create a humidity sensor and duplicate the experiment **Tim Peake** has been running on the International Space Station

You'll Need

- Latest version of Raspbian **magpi.cc/1MYYTMo**
- A Sense HAT **magpi.cc/1TGGUt5**
- twoDigits Python library
- Some resealable sandwich bags, drinking straws, and sticky tape
- Crew Detector code **magpi.cc/1RC6AIF**
- Raspberry Pi Camera Module

In the last two issues, we used the Sense HAT to investigate some of the properties of light and measure atmospheric pressure. This time, we're going to experiment with another of the environmental sensors, and take a look at humidity and the things that can influence it.

Attach the Sense HAT to your Pi and power it up. Open IDLE3 or just type **python3** in a terminal window, and then import the **SenseHat** library and connect to the board:

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

As with pressure measurements, we can take a humidity reading with a single line:

```
sh.get_humidity()
```

This should produce a number like 32.87158203125 (if you just get 0 back, run the command again). What does this number mean?

A simple way to think of humidity is that it's the amount of water vapour in the air. However, there are different ways of measuring humidity. Absolute humidity is the water content of air at a certain temperature and is normally measured in grams/m^3 , whereas relative humidity is the amount of moisture in the air compared to what the air can 'hold' at that temperature, given as a percentage. The Sense HAT reports relative humidity; this is useful for weather forecasting as the higher the value, the greater chance of precipitation (rain, dew, or fog).

To make it easier to move around and measure differences in humidity, let's make our Pi portable.

Making a sensor

From last issue, you'll probably remember that the standard units for pressure are millibars, and normal values tend to be at least three or four digits long. Therefore, when displaying our measurements, we had to scroll them across the LED matrix. This time, we can easily round our humidity values to the nearest integer percent, because we should normally expect the readings to be 0-99%. Two digits is an ideal size for displaying a number on the LED matrix, and we've written a handy Python module that makes it simple. You can install it by typing this at the command line:

```
sudo pip-3.2 install twoDigit
```

Once that's done, download or type up the code from the listing, then start the program running by typing:

```
python3 humidity.py
```

Now disconnect the mouse, keyboard, and monitor. The humidity reading should be displayed on the Sense HAT LED matrix every half a second. Just as with the pressure measurements, red text is used if the value is higher than the previous one, and green text if lower.

Take your portable sensor for a walk and explore how humidity changes. You might notice a big change if you take the Pi outside. What differences do you see moving into a steamy kitchen or a bathroom?

Controlled environment

Obviously, a shower or a kettle can add lots of moisture to the air, but our bodies are also a great source of humidity. Let's see how much effect we can have on our environment just through breathing. Take two sealable sandwich bags and snip the tip off one of the bottom corners on each. Cut off a short length of straw and poke an end through each of the corner holes, then use sticky tape to seal up the bags around the straw. Now pick one of the bags, snip off the other bottom corner and seal it round the remainder of the straw. Put the Pi and its battery into the first bag (the one with a single straw). Finally, seal both of the bags up, making sure that they're not too inflated. Blow into the long straw and watch the readings increase. Can you cause a rise to 100% relative humidity?

You should quickly see why we used two bags rather than just one. The internal surfaces of the first bag will rapidly become covered in condensation, while the second – with the Pi inside – should stay drier. Always remember: electronics and water don't go well together!

humidity.py

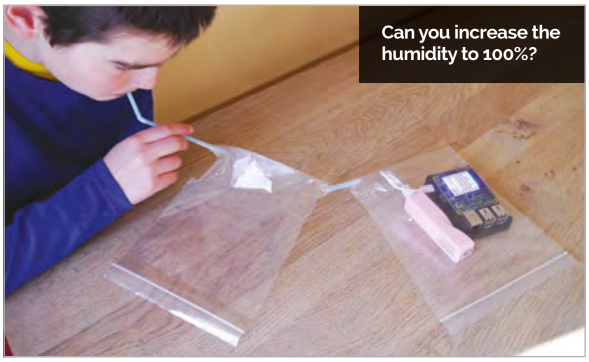
Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1VlUJks

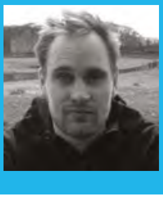
```
import logging
from twoDigit.twoDigit import numToMatrix
from sense_hat import SenseHat
from datetime import datetime
from time import sleep

# Set up the logfile name based on date/time
logfile = "humidity-"+str(datetime.now().strftime(
"%Y%m%d-%H%M"))+".csv"
# Logging settings and format for CSV
logging.basicConfig(filename=logfile, level=logging.DEBUG,
    format='%(asctime)s %(message)s',
    datefmt='%Y-%m-%d, %H:%M:%S,')

sh = SenseHat() # Connect to Sense HAT
h_old = 0
col = [0,0,255]

while True: # Main loop

   h = sh.get_humidity() # Take humidity reading
   logging.info(str(h)) # Log value to file
   if h > h_old: # If humidity has increased...
       col = [255,0,0] # Set to red
   else: # If humidity has decreased...
       col = [0,255,0] # Set to green

   h = int(round(h,0)) # Round to a whole percent
   i = numToMatrix(h,col) # Create 2-digit image for matrix
   sh.set_pixels(i) # Display reading
   sleep(0.5)
   h_old = h # Set previous value
```

Our Code Club's winning entry for the Astro Pi competition used the humidity sensor to try to detect astronauts on the International Space Station. You can download this code to see how well it works on Earth. Attach the Pi Camera Module and connect it back to mains power. Then type:

```
  git clone https://github.com/topshed/
SweatyAstronautCode
  cd
  python3 sweaty_astronaut.py
```


Can you increase the humidity to 100%?

The first thing you'll see is a slightly self-indulgent display of animations on the LED matrix. While this is in progress, set up the Pi where you're going to run the experiment. Once the animations have finished, you'll see a message saying that the program is starting. Keep away from the Pi for five minutes while it takes its baseline measurements. Once it has finished, you'll see a message reporting that the experiment has started. See how near you need to get to the Pi to trigger a reaction; once the code thinks there has been a significant rise in humidity, it will ask if anyone is there and take a picture with the camera. You may need to tweak some of the settings in the code depending on your environment; see the README.md file for more information.



SAM AARON

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

π))) Sonic Pi PART 10

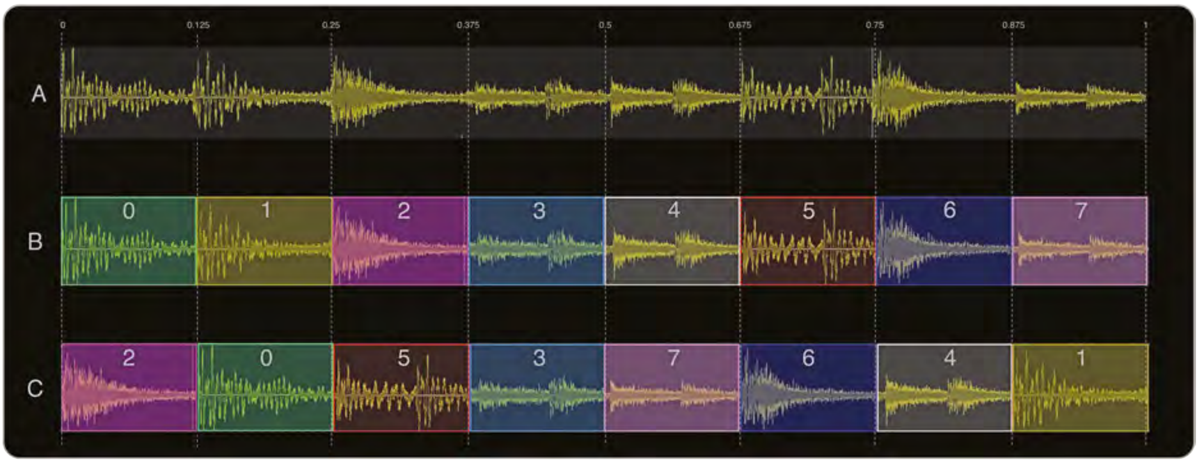


Fig 1 We can slice up a sample and play the parts back in a different order

SAMPLE SLICING

In his latest musical coding piece, Sonic Pi creator **Sam Aaron** shows us how to slice and dice audio samples

You'll Need

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

Way back in episode three of this Sonic Pi series, we looked at how to loop, stretch, and filter one of the most famous drum breaks of all time: the Amen Break. In this tutorial, we're going to take this one step further and learn how to slice it up, shuffle the slices, and glue it back together in a completely new order. If that sounds a bit crazy to you, don't worry: it will all become clear and you'll soon master a powerful new tool for your live-coded sets.

Sound as data

Before we get started, let's just take a brief moment to understand how to work with samples. By now, you have all hopefully played with Sonic Pi's powerful sampler. If not, there's no time like the present! Boot up your Raspberry Pi, launch Sonic Pi from the Programming menu, type the following line into a fresh buffer, and then hit the Run button to hear a pre-recorded drum beat:

```
sample :loop_amen
```

A recording of a sound is simply represented as data: lots of numbers between -1 and 1, which represent the peaks and troughs of the sound wave (see the boxout to the right for more information). If we play those numbers back in order, we get the original sound. However, what's to stop us from playing them back in a different order and creating a new sound?

How are samples actually recorded?

It's actually pretty simple once you understand the basic physics of sound. When you make a sound – for example, by hitting a drum – the noise travels through the air in a similar fashion to the way the surface of a lake ripples when you throw a pebble into it. When those ripples reach your ears, your eardrum moves sympathetically and converts those movements into the sound you hear. If we wish to record and play back the sound, we therefore need a way of capturing, storing, and reproducing those ripples. One way is to use a microphone, which acts like an eardrum and moves back and forth as the sound ripples hit it. The microphone then converts its position into a tiny electric signal, which is then measured many times a second. These measurements are then represented as a series of numbers between -1 and 1.

If we were to plot a visualisation of the sound, it would be a simple graph of data with time on the x axis and microphone/speaker position as a value between -1 and 1 on the y axis. You can see an example of such a graph at the top of the **Fig 1** diagram.

Playing part of a sample

So, how do we code Sonic Pi to play a sample back in a different order? To answer this question, we need to take a look at the **start:** and **finish:** opts for **sample**. These let us control the start and finish positions of our playback of the numbers which represent the sound.

The values for both of these opts are represented as a number between **0** and **1**, where **0** represents the start of the sample and **1** is the end.So, to play the first half of the Amen Break, we just need to specify a **finish:** of **0.5**:

```
sample :loop_amen, finish: 0.5
```

We can add in a **start:** value to play an even smaller section of the sample:

```
sample :loop_amen, start: 0.25, finish: 0.5
```

For fun, you can even have the **finish:** opt's value set *before* **start:**, in which case it will play the section backwards:

```
sample :loop_amen, start: 0.5, finish: 0.25
```

Reordering sample playback

Now that we know that a sample is simply a list of numbers that can be played back in any order, and also how to play a specific part of a sample, we can start having fun playing a sample back in the 'wrong' order.

Let's take our Amen Break and chop it up into eight equally sized slices, and then shuffle those pieces around to create a new drum break.

Take another look at the **Fig 1** diagram on the previous page: at the top, A) represents the graph of our original sample data. Chopping it into eight slices gives us B) – notice that we've given each slice a different colour to help distinguish them. You can see each slice's start and finish values at the top. Finally, C) is one possible reordering of the slices. We can then play this back to create a new beat. Take a look at the code to do this:

```
live_loop :beat_slicer do
  slice_idx = rand_i(8)
  slice_size = 0.125
  s = slice_idx * slice_size
  f = s + slice_size
  sample :loop_amen, start: s, finish: f
  sleep sample_duration :loop_amen, start: s,
finish: f
end
```

Let's analyse this piece of code to see precisely what is going on at each stage...

1. We choose a random slice to play, which should be a random number between 0 and 7 (remember that we start counting at 0). Sonic Pi has a handy function for exactly this: **rand_i(8)**. We then store this random slice index in the variable **slice_idx**.

2. We define our **slice_size**, which is 1 divided by the number of slices (8), or **0.125**. We need this to convert our **slice_idx** into a slice starting position, as a value between 0 and 1. For example, the slice with index 3 starts three slices into the sample: 3 * 0.125, or 0.375.

3. We calculate the start position **s** by multiplying the **slice_idx** by the **slice_size**.

4. We calculate the finish position **f** by adding the **slice_size** to the start position **s**.

5. We can now play the slice by plugging in the **s** and **f** values into the **start:** and **finish:** opts for **sample**.

6. Before we play the next slice, we need to know how long to **sleep**, which should be the duration of the sample slice. Luckily, Sonic Pi has us covered with **sample_duration**, which accepts all the same opts as **sample** and simply returns the duration. So, by passing **sample_duration** to our **start:** and **finish:** opts, we can find out the duration of a single slice.

7. We wrap all of this code in a **live_loop** so that we continue to pick new random slices to play.

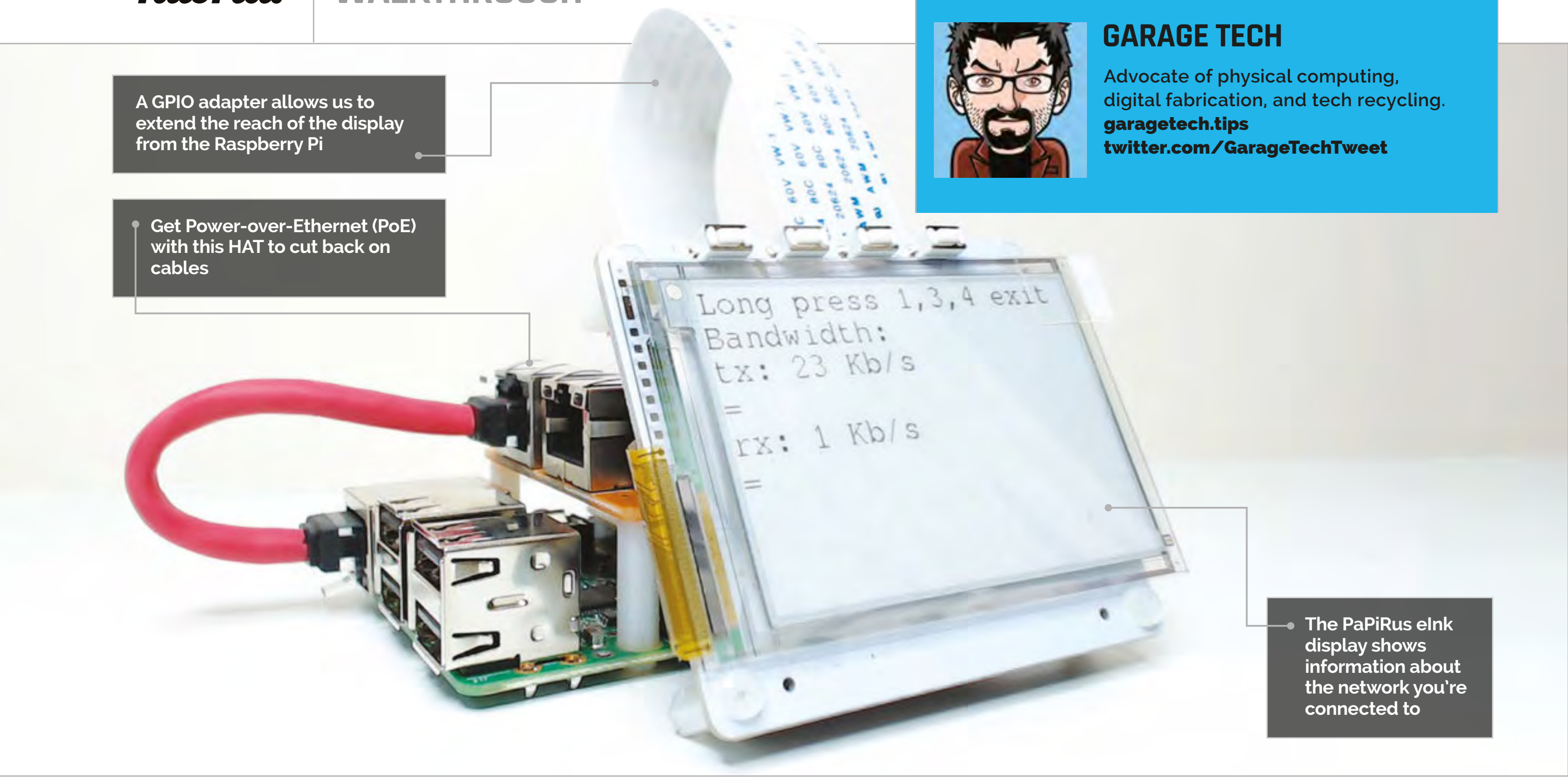
Bringing it all together

Let's combine everything we've seen so far into a final example which demonstrates how we can take a similar approach to combine randomly sliced beats with some bass to create the start of an interesting track. Now it's your turn: take this code as a starting point and see if you can create something new...

```
live_loop :sliced_amen do
  n = 8
  s = line(0, 1, steps: n).choose
  f = s + (1.0 / n)
  sample :loop_amen, beat_stretch: 2, start: s, finish: f
  sleep 2.0 / n
end

live_loop :acid_bass do
  with_fx :reverb, room: 1, reps: 32, amp: 0.6 do
    tick
    n = (octs :e0, 3).look - (knit 0, 3 * 8, -4, 3 * 8).look
    co = rrand(70, 110)
    synth :beep, note: n + 36, release: 0.1, wave: 0, cutoff: co
    synth :tb303, note: n, release: 0.2, wave: 0, cutoff: co
    sleep (ring 0.125, 0.25).look
  end
end
```

GARAGE TECH

Advocate of physical computing, digital fabrication, and tech recycling.
garagetech.tips
twitter.com/GarageTechTweet



A GPIO adapter allows us to extend the reach of the display from the Raspberry Pi

Get Power-over-Ethernet (PoE) with this HAT to cut back on cables

The PaPiRus eInk display shows information about the network you're connected to

BUILD A RASPBERRY PI ACCESS POINT WITH A DISPLAY

Build a professional-looking access point with a Raspberry Pi, that gives you relevant info on a PaPiRus screen

You'll Need

- PaPiRus ePaper / eInk Screen HAT **magpi.cc/1OAObXV**
- Pi PoE Switch HAT **magpi.cc/1T3WnnB**
- TP-LINK PoE Injector (optional) **magpi.cc/1T3Wq2N**
- 40-pin header **magpi.cc/1VV5zx2**
- GPIO adapter for RPi-Display B+ – female **magpi.cc/1VV5ERe**
- PuTTY **magpi.cc/1Mm5Npi**

This article features the PaPiRus, the Pi PoE, and the Raspberry Pi 3 to build a portable access point with a screen. We will be working with the Pi PoE, but everything in this tutorial can be achieved without it, too. This tutorial assumes your version of Raspbian Jessie is up-to-date.

Putting everything together

Building this access point is extremely easy. It requires minimal soldering and for the rest, it's really just a question of stacking together the various components.

The additional header will allow us to install the PaPiRus on top of the Pi PoE, but we did this via the GPIO adapter.

In some cases, your infrastructure will already be able to deliver Power over Ethernet; if not, you'll need to use a PoE injector like the TP-LINK TL-POE150S.

Connect everything up, making sure that the flat cable for the PaPiRus is connected the right way around.

Once all is connected and ready to use, a long press on the switch of the Pi PoE module will power up the system.

Preparing the access point

For the initial preparation, we need to enable SPI. Go to the program menu and find the Raspberry Pi Configuration menu to make sure it's enabled in the Interfaces tab. Reboot the Raspberry Pi. We're now ready to install and configure hostapd, which we will use to create the AP part of our system.

```
sudo apt-get install hostapd
```

Then we need to create the configuration file as follows. Note that the file doesn't exist and that we're creating a brand new one.

This is the right time to download the supporting code from GitHub, which also contains the templates for the configuration files.

```
git clone https://github.com/francesco-
vannini/PaPiRus-AP.git
```

Open the **hostapd.conf** template file in the terminal with:

```
sudo nano PaPiRus-AP/hostapd.conf
```

Replace the values for ssid and wpa_passphrase in this conf file with details for your network, then copy the file **hostapd.conf** to the correct directory:

```
sudo cp PaPiRus-AP/hostapd.conf /etc/hostapd
```

At this point, we need to edit the following file to make hostapd use the above configuration at startup. Open the file with **`sudo nano /etc/default/hostapd`** and change it by adding:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Lastly, we start the service with **`sudo systemctl start hostapd`**. We really want this configuration not to depend on services that will automatically alter our network settings, so we'll disable dhcpcd:

```
sudo systemctl disable dhcpcd
```

Now we need to install bridge-utils, which will allow us to merge the Ethernet and wireless interfaces:

```
sudo apt-get install bridge-utils
```

Once that's done, we can configure our networking using the template provided. Open it with **`sudo nano PaPiRus-AP/interfaces`** and edit the details in the 'Bridge setup' section.

Now back up your existing file:

```
sudo mv /etc/network/interfaces /etc/
network/interfaces.bck
```

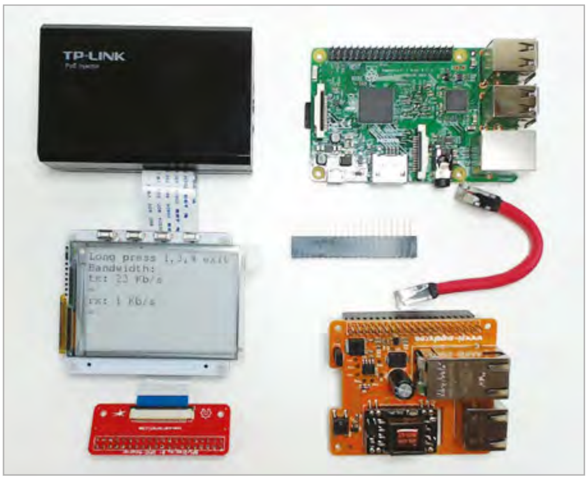
Copy the new one in:

```
sudo cp PaPiRus-AP/interfaces /etc/network/
```

At this point, you can reboot your Pi and if all is good, you should find a new AP with the SSID you choose! You can now connect via the bro IP that you specified. The access point is ready. Devices connecting to it should get their IP from the DHCP on your home/office router and be able to browse the internet. Great!

Get the PaPiRus ready

Whereas for the Pi PoE there are no software steps involved, for the PaPiRus we need to install drivers and packages. These will also provide some ready-made examples we could use to test all is working, before turning to the code that will allow us to manage the AP.



Left It doesn't look all that much, but it's all you need to build the PaPiRus access point

To install the PaPiRus software, execute the following command from your home folder – if you're unsure where you are, just enter **`cd~`** to get back to your home directory.

```
curl -sSL https://goo.gl/NzzRXa | sudo bash
```

The installation should be fairly quick, and resolve various dependencies for the packages required to use the PaPiRus screen.

Once the installation is finished, we need to tell the system which screen we're going to use. We've created the scripts to manage the AP for a 2.7-inch screen (the biggest). In this case, we need to run:

```
sudo papirus-set 2.7
```

If all goes well, you can test it with:

```
sudo papirus-clock
```

You should see a nice clock on your PaPiRus. We're now nearly done, and one step away from getting our AP ready for business.

Run the management script

We've created some demo code to display various information about the AP on the PaPiRus. By using the four switches, you'll be able to access network stats and basic information about the AP configuration. Run:

```
sudo python PaPiRus-AP/APInfo.py &
```

Your PaPiRus screen should spring to life and display your SSID and AP IP address. By pressing switch 2, you get to see some bandwidth info; by pressing switch 3, you get a detailed view of the AP networking setup.

You now have a very nice, portable access point to carry with you at Raspberry Pi Jams or hackathons, or wherever your Pi will take you next.

ACCESS POINT DIFFERENCES

There are a couple of methods that can be used to create an AP at the OS level. In this tutorial, we decided to implement the AP as an ISO layer 2 device, which is a fancy way to say that there won't be any routing, firewall, or NAT involved. The device will 'merge' the Ethernet network card and the WiFi into a bridge; it will only have one IP address and will, in fact, behave as a typical AP and not as a home broadband router.

DHCP REQUIRED

The AP won't serve as a DHCP, so it 's essential that you have one enabled on your router. The first part of the setup also requires that the Pi receives an IP from the router, in order to get all the packages required off the internet.

MIKE'S PI BAKERY

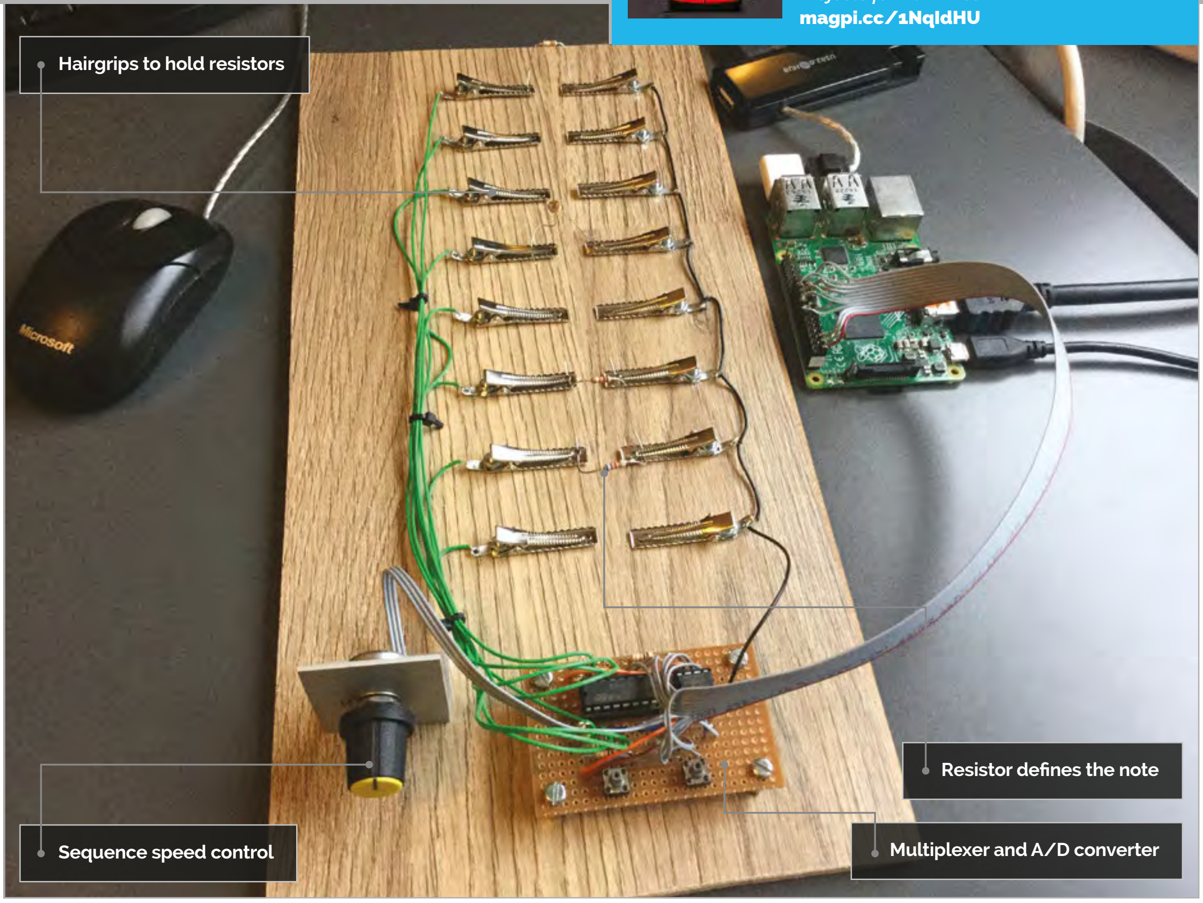
MIKE COOK

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

magpi.cc/1NqldHU

You'll Need

- 16× 33mm spring hairgrips
- 240 × 140 × 6mm board
- 74HC4051 analogue switch
- MCP3002 A/D chip
- 0.1uF ceramic capacitor
- 8× 10K resistors
- 1K to 22K pot, bracket and knob
- 2× push buttons
- 16-pin DIL IC socket
- 8-pin DIL IC socket
- Stripboard and mounting pillars
- 16× solder tags
- 16× M3 10mm countersunk screws & nuts
- GPIO connector socket



Hairgrips to hold resistors
Resistor defines the note
Sequence speed control
Multiplexer and A/D converter

GET A GRIP: PHYSICAL SEQUENCER

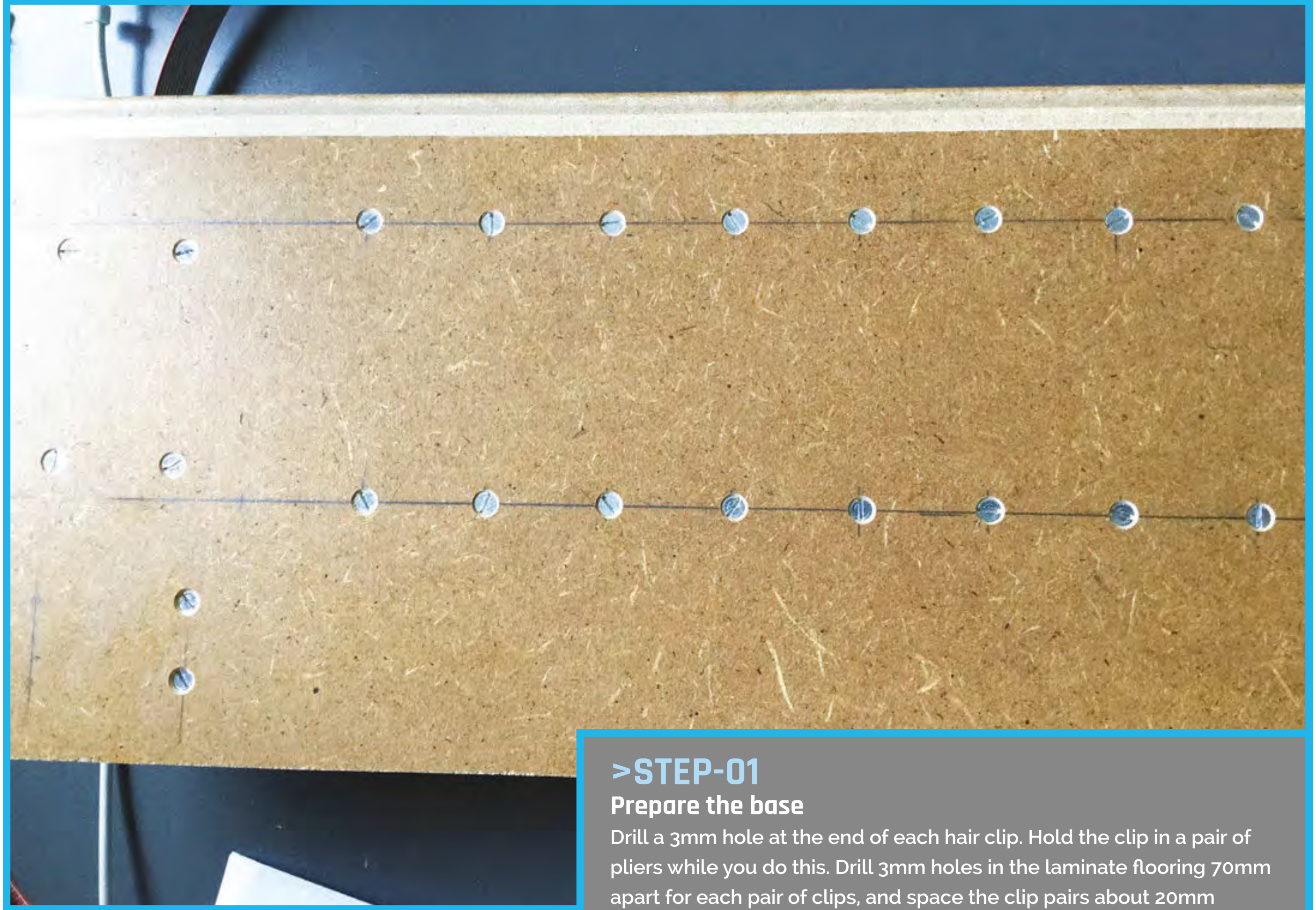
Hairgrips hold the key to a good music sequence. Build your own in easy steps...

Some months back, we were doing some work with Drake Music, who specialise in musical applications for the disabled. We were asked to come up with a stand-alone physical sequencer. This proved to be so popular that we thought we would do a Raspberry Pi version to take advantage of the sample-playing capabilities of the Pi.

The project

In a physical sequencer, some object represents a note, and where it's placed represents where that note appears in the sequence. Here, a note or sample is represented by a resistor, and its position by what pair of clips it's fastened to. After some research, the cheapest form of clip we found was a hairgrip or clip, hence the name of this project. The concept is quite simple: a multiplexer switches one of eight resistors to an A/D (analogue-to-digital) converter, which is measured and turned into a sample number. A second A/D channel converts a potentiometer output into a number that's used to determine the time between each step of the sequencer. The block diagram of the project is shown in **Fig 1** (overleaf). Note that for each pair of clips, a pull up resistor and note-determining resistor are shown: once in detail in the dotted box, and seven other times as a block to make the diagram simpler. Remember, a multiplexer is simply a big eight-way switch controlled from the Pi's GPIO lines. While the Pi doesn't have

MAKING THE SEQUENCER



>STEP-01

Prepare the base

Drill a 3mm hole at the end of each hair clip. Hold the clip in a pair of pliers while you do this. Drill 3mm holes in the laminate flooring 70mm apart for each pair of clips, and space the clip pairs about 20mm apart to give good room to access them. Countersink the underside of the flooring, so the 3M screws poke up a minimum amount – if too much thread shows, the clips won't open fully when pressed. Take a small piece of angle aluminium and drill it to accommodate the pot; remember to drill the second hole/slot for the pot's anti-rotation lug as well. In the other side of the angle, drill two 3mm holes to mount it to the board, and drill and countersink holes in the board to match.

a built-in A/D converter, adding an external one is straightforward. We chose the low-cost, two-channel MCP3002 10-bit A/D. Even better, our normal interface library, WiringPi, has a driver for this chip already included, so talking to it is easy.

The hardware

The schematic of the project is shown in **Fig 2** (overleaf). The data select chip is a 74HC4051, and the whole circuit is run from the 3V3 voltage rail. This way, you don't have to do any level shifting on the signal lines. Note that in the contents of the block diagram there are two push buttons labelled **start/stop** and **mode** – more on this later. Details of construction are in the 'Making the sequencer' step-by-step section. The A/D measurements you get from a resistor will not be precise due to the 5% tolerance of each resistor, so there is a range of values you could get. In order to identify what resistor you have, a sequence of threshold values was calculated using a spreadsheet – this is available in the GitHub repository, along with all the files needed for this project.

The software

As well as having WiringPi installed, you need to enable the SPI drivers. There are many ways to do this, but perhaps the simplest is to open a terminal window and type **`sudo raspi-config`**. Then, from the menu, pick 'Option 9 – Advanced Options', choose 'A6 SPI' and enable the SPI, then reboot. The code is in the **hard_time.py** listing (on page 62) and it needs two sets of samples to play. These are in OGG format and were created using Audacity; the individual samples are named s0 to s14 and cover two octaves in the key of C. The samples are stored in a directory at the same level as the program, and are called **piano_sounds** and **marimber_sounds** – can you guess what they are?

>STEP-02

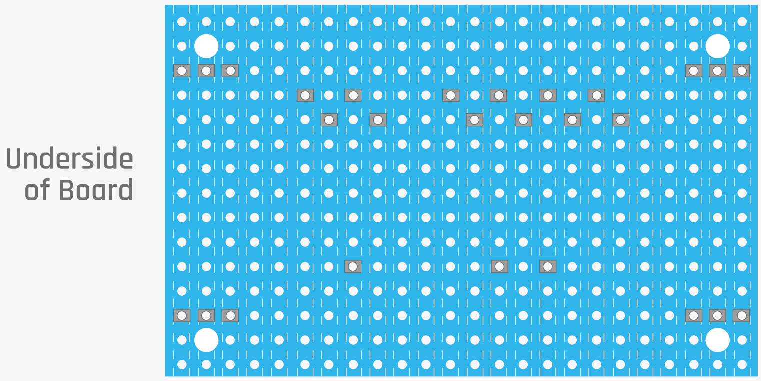
Mount the clips

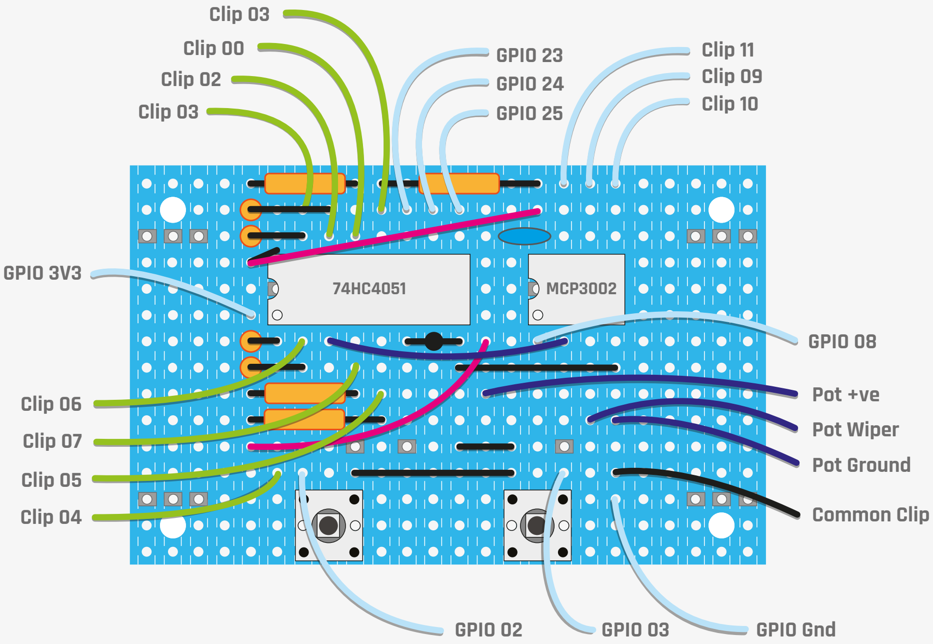
Fix the hair clips to the board. Make a sandwich with solder tag, clip and nut, then tighten the nut up very tight so it won't rotate. Use some glue if you want, but we didn't need any. If the clip is bent and doesn't sit flat against the board, then place a small screwdriver under the clip close to the screw, and push the clip flat against the board. Wire and solder one side of the clips together, and connect to ground on the circuit board. The other clip in each pair should be wired to the appropriate pin of the multiplexer chip.

>STEP-03

Cut the tracks

Prepare a 15-by-25-hole piece of stripboard. Drill 3mm holes in each of the corners, as shown, and fit to four 10mm pillars. Drill and countersink the base board to suit. Then flip the board over and cut the tracks on the rear side of the board with a spot face cutter or scalpel, at the places shown in the diagram.





>STEP-04

Make the circuit

Now flip the board over and fit the components, the internal board wires, and solid copper wires. The diagram shows the strips as seen through the board. Note the wire connecting pin 16 of the multiplexer chip underneath the board is also attached to the adjacent track where the wire enters the board. Once the board is complete, wire up the external connections. These are the wires to the Raspberry Pi's GPIO port connector, along with the connections to the hair clips and the speed control pot. The tactile switches used were the two-contact type; a small change is needed to accommodate the four-legged type.

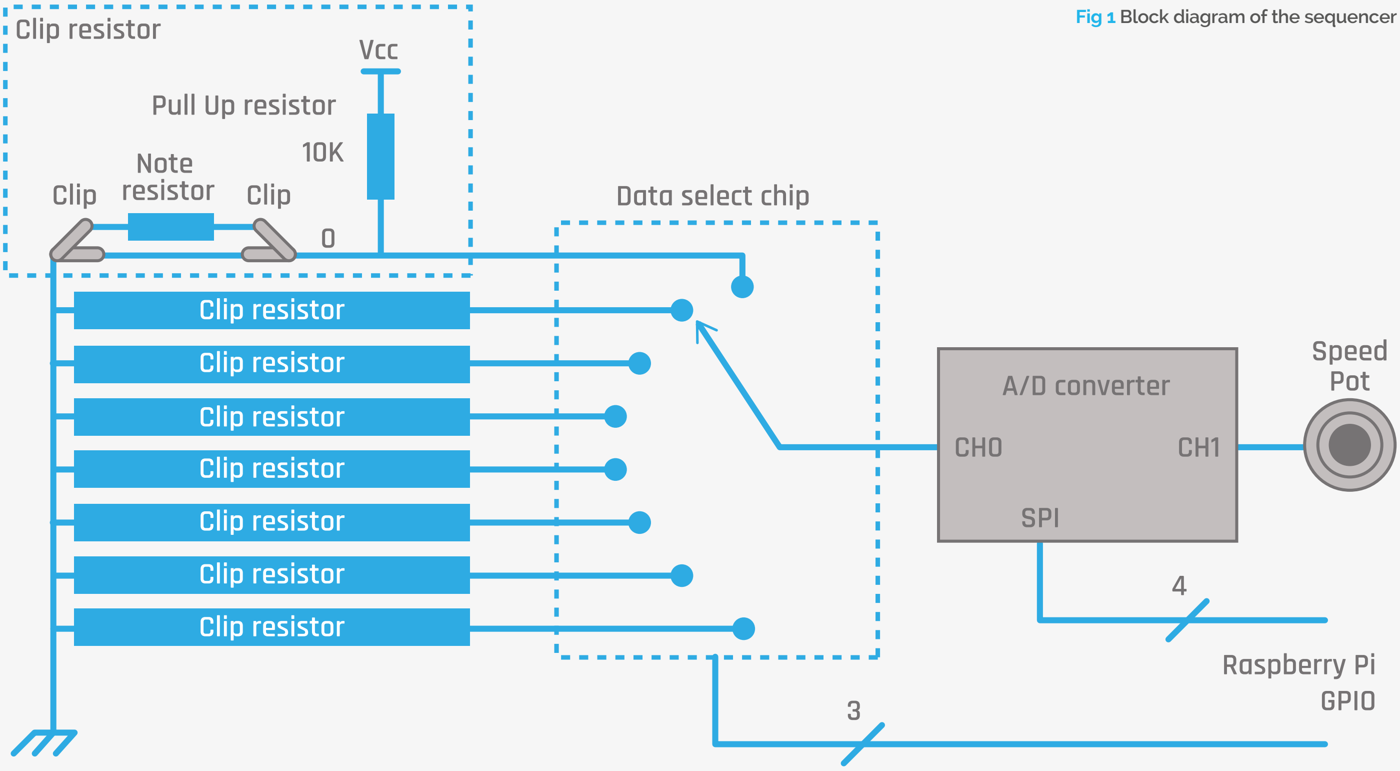


Fig 1 Block diagram of the sequencer

The **setMux** function outputs the multiplexer address on the appropriate pins; the **readAD** function reads the value from channel 0 of the A/D, works out what resistor is placed, if any, and plays the appropriate sample. Working out the sample number is easy: just search through the threshold list until the sample is greater than the threshold. The entry when this happens is the sample number to play. Well, nearly: in order to have nothing played when there is a short circuit, the sample is shifted by one place in the threshold list. The resistor values needed to produce a note are shown in the listing, underneath the threshold list.

> One of the most effective alterations you can make is to change the samples to percussion sounds

Operation

In operation, pressing the ‘start’ button begins the sequence; it will repeat indefinitely until the button is pushed again, whereupon it will stop at the end of the next complete cycle of the sequence. You will see a **stopping** message followed by a **stopped** message in the Python console window. When the sequencer is running, you can press the ‘mode’ button, whereupon the sequence will be played in reverse order on a different instrument.

Taking it further

The simplest thing you can do is to change the samples; perhaps one of the most effective things is changing them to percussion sounds to have a drum sample player. There are two other variants of the code on GitHub: one where the sound doesn’t change when the sequence plays backwards, and another where the sequence is repeated at a slightly different speed, so the two drift in and out of phase with each other in a Steve Reich sort of way.

One fun thing to do is to replace one or more of the resistors with an LDR (light-dependent resistor), and then the sequence can change as you cover it up with your hand. You can also change the threshold list and have more than just two octaves of samples.

For the adventurous, you can easily extend the project to 16 or more inputs. Pin 6 on the 74HC4051 is an enable pin; this is set to be permanently enabled. However, if you take this pin to a GPIO line and do the same with another 74HC4051, you can double the number of inputs. All the other pins on this second multiplexer are wired in common with the first. By repeating this with more multiplexer chips, you can add as many steps as you like to the sequencer.

hard_time.py

```
#!/usr/bin/env python
# Get a Grip - physical sequencer
# Version 1 reverse mode switch alt sound
# on reverse

import time, pygame
import os, sys
import wiringpi2 as io

pygame.init()     # initialise pygame
pygame.mixer.quit()
pygame.mixer.init(frequency=22050,
size=-16, channels=8, buffer=512)

pygame.event.set_allowed(None)
seqSound = [ pygame.mixer.Sound(
"marimber_sounds/s"+str(sound)+".ogg")
                  for sound in range(0,15)]
altSound = [ pygame.mixer.Sound(
"piano_sounds/s"+str(sound)+".ogg")
                  for sound in range(0,15)]

startPush = 2; modePush = 3
lastStart = 1; lastMode = 1
muxPins = [ 23,24,25]
threshold = [80,153,235,290,370,426,475,535,586,636,681,745,816,879,922,950,1024]
#note   C   D   E   F   G   A   B   C   D   E
# F   G   A   B   C
#       1K  2K7 3K3 4K7 6K8 7K5 10K 12K 15K 18K
# 22K 33K 47K 82K 100K

def main():
   global clip, stop, pendingStop, mode
   initGPIO()
```

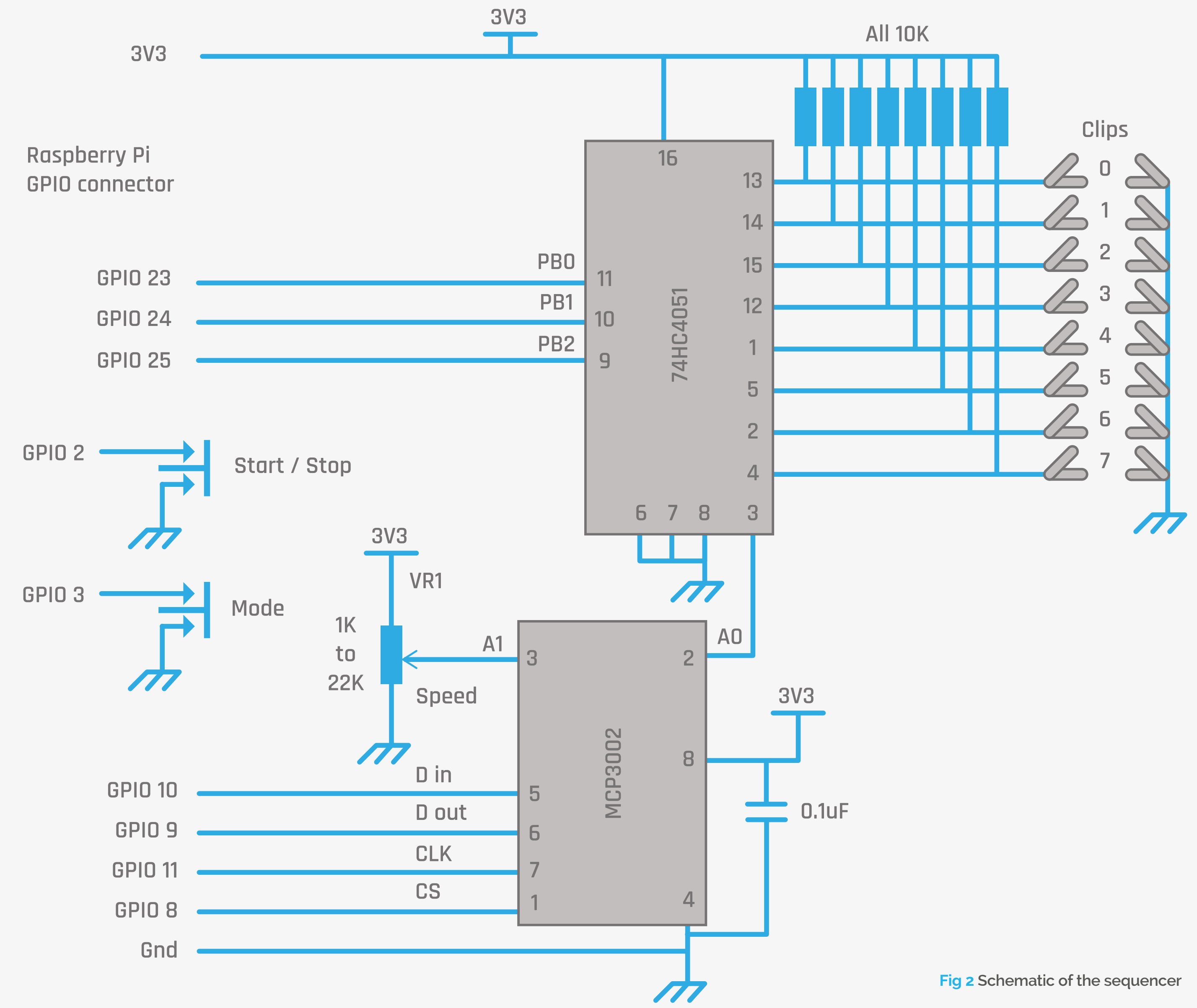


Fig 2 Schematic of the sequencer

```
   print"Get A Grip - sequencer"
   clip=0
   stop = False ; mode = True
   pendingStop = False
   while True:
      if io.digitalRead(modePush) == 0 and io.digitalRead(startPush) == 0 :
         terminate()
      if not stop :
         stepTime = 0.004 + (io.analogRead(71) / 2000.0)
         nextStep = time.time() + stepTime
         setMux(clip)
         while time.time() <= nextStep :
           readStart()
           readMode()
         readAD(clip)
         incrementClip()
      else :
         readStart()

def incrementClip():
   global clip, stop, pendingStop, mode
   if mode :
      clip += 1
      if clip>7 :
         clip= 0
         if pendingStop :
            stop = True
            pendingStop = False
            print"stopped"
   else :
      clip -= 1
      if clip < 0 :
         clip= 7
         if pendingStop :
            stop = True
            pendingStop = False
            print"stopped"

def readMode():
    global mode, lastMode, clip
    current = io.digitalRead(modePush)
    if current == 0 and lastMode == 1:
         time.sleep(0.02) # debounce delay
         mode = not mode
         if mode :
            print"forward mode"
            clip = 0
         else :
            print"reverse mode"
            clip = 7
    lastMode = current

def readStart(): # start / stop control
    global stop, pendingStop, lastStart
    current = io.digitalRead(startPush)
    if current == 0 and lastStart == 1:
       time.sleep(0.02) # debounce

       if stop :
          stop = False
          print"running"
       else :
          pendingStop = True
          print"stopping"
    lastStart = current

def readAD(step) :
   value = io.analogRead(70) # read multiplex
   look = 0
   while value >= threshold[look]:
      look +=1
   if look>0 and look <16 :
      if mode :
        seqSound[look-1].play()
      else:
        altSound[look-1].play()

def setMux(n):
   mask = 1;
   for pin in range(0,3):
     if (mask & n) != 0 :
        io.digitalWrite(muxPins[pin], 1)
     else :
        io.digitalWrite(muxPins[pin], 0)
     mask = mask << 1

def initGPIO():
   try :
      io.wiringPiSetupGpio()
   except :
      print"start IDLE with 'gksudo idle' from terminal"

      os._exit(1)
   for pin in range (0,3):
      io.pinMode(muxPins[pin],1) # mux pin to output
   io.pinMode(startPush,0) # start / stop input
   io.pinMode(modePush,0) # mode input
   io.pullUpDnControl(startPush,2) # input enable pull up
   io.pullUpDnControl(modePush,2) # input enable pull up
   io.mcp3002Setup(70,0)

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

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

Language

>PYTHON

DOWNLOAD:
magpi.cc/1NqJjmV

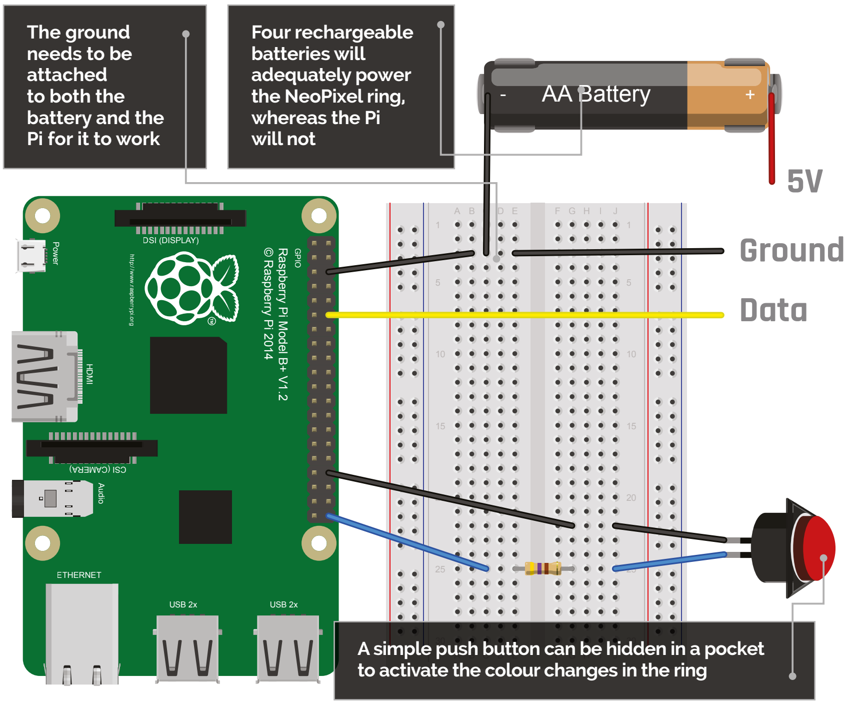
PROJECT VIDEOS

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

ROB ZWETSLOOT

Tinkerer, sometimes maker, other-times cosplayer, and all the time features editor of *The MagPi*.

magpi.cc



The ground needs to be attached to both the battery and the Pi for it to work

Four rechargeable batteries will adequately power the NeoPixel ring, whereas the Pi will not

A simple push button can be hidden in a pocket to activate the colour changes in the ring

The lights are attached behind black chiffon to slightly obscure them, and allow the cosplayer's face to remain hidden while still being able to see

You'll Need

- First-generation Raspberry Pi (A+, B+, Zero, etc.)
- NeoPixel ring
- 4× AA battery pack, preferably with a switch
- 4× rechargeable AA batteries
- Portable mobile phone power bank
- Push button
- Various wires and resistors

MAKE NEOPIXEL COSPLAY EYES WITH PI

Create your own powerful eyes for Sans from Undertale, by getting some LEDs and a NeoPixel ring. It'll be a skele-ton of fun!

NeoPixels are amazing, addressable RGB LEDs from Adafruit. You can get them in strips, rings or individually, and create incredible effects with them. However, it's not always clear how to control them on the Raspberry Pi.

Recently, friend of the mag Mel Nurdin (of Riddlerule Cosplay: **magpi.cc/1Q20QCJ**) expressed an interest in using NeoPixels in a cosplay of theirs. We decided to cut through the information and create a Pi-powered pair of eyes for their costume: skeleton pun-maker and hotdog salesman Sans from last year's excellent *Undertale* game. Follow along to make your own Megalovania eyes, or learn how to use NeoPixels in general for your own projects.

>STEP-01

Prepare your Pi

A Raspberry Pi Zero was used in the final build, so it would fit better within the confines of the costume. Update a version of Raspbian Jessie by going into the terminal and using **`sudo apt-get update`** then **`sudo apt-get upgrade`**, before installing the software needed for the NeoPixels:

```
  sudo apt-get install build-essential
python-dev git scons swig
```

Now download the library for the NeoPixels:

```
  git clone https://github.com/jgarff/
rpi_ws281x.git
  cd rpi_ws281x
  scons
```

Finally, we can install the module to the Pi:

```
cd python
sudo python setup.py install
```

>STEP-02

Wire up the NeoPixel

It's worth testing this out on a breadboard first, so you can understand the concept of the circuit. Basically, the NeoPixel needs to be powered separately by the four AA batteries, with a data cable coming from a PWM-enabled pin on the Pi to control the LEDs.

Language

>PYTHON 3

DOWNLOAD:
magpi.cc/SansEyes

eyes.py

```
#!/usr/bin/env python

import time

from gpiozero import Button

from neopixel import *

button = Button(21)

# LED strip configuration:
LED_COUNT   = 16      # Number of LED pixels
LED_PIN     = 18      # GPIO pin
LED_FREQ_HZ = 800000  # LED signal frequency in hertz
LED_DMA     = 5       # DMA channel to use for
# generating signal
#LED_BRIGHTNESS = 255  # LED brightness
LED_INVERT  = False   # True to invert the signal

# Function to control colour change

def Megalovania(strip, color):
    iterations = 0
    for i in range(strip.numPixels()):
        strip.setPixelColor(i, color)
        strip.show()

# Create NeoPixel object with
# appropriate configuration
strip = Adafruit_NeoPixel(LED_
COUNT, LED_PIN, LED_FREQ_HZ,
LED_DMA, LED_INVERT)
# Initalise the library (must be called once before
# other functions)
strip.begin()

Megalovania(strip, Color(255,255,255))
time.sleep(2)

while True:
    Megalovania(strip, Color(0,0,0)) # Eyes off
    time.sleep(0.5)
    button.wait_for_press()
    Megalovania(strip, Color(0,0,255)) # Blue eye
    time.sleep(0.5)
    button.wait_for_press()
    time.sleep(0.5)
    while button.is_pressed == False:
        Megalovania(strip, Color(255,255,0)) # Yellow eye
        time.sleep(0.1)
        Megalovania(strip, Color(0,0,255)) # Blue eye
        time.sleep(0.1)
```

You don't need to step up the 3V3 signal from the Pi to do this, so to keep it simple, we're not. The main thing to remember is that the ground of the NeoPixel needs to be connected to the negative on the battery pack and a ground on the Raspberry Pi. We've connected the data pin for the NeoPixel to pin 6 (GPIO 18).

>STEP-03

Wire up the button

The button is the easy one to wire up. It doesn't matter which way around it goes, but we have it connected to pin 40 right at the end and to ground on pin 34, to keep it away from the NeoPixel wires. You'll also need a resistor suitable for your button – ours uses a 470 ohm resistor – and it can be connected to either side of the button. This is controlled by GPIO Zero, so it's addressed at GPIO 21 in the code.

>STEP-04

Add the code

Type up or download the code to the Raspberry Pi. You can either put it in the home folder or in its own folder; either way, you can test it by running it in the terminal with:

```
sudo python eyes.py
```

It should light up the LEDs white for a couple of seconds before turning them off. A button press will make them turn blue, and another button press will make the LEDs flash blue and yellow quite quickly, emulating a specific scenario in the game *Undertale*. Add this command to the end of the file **/etc/profile** so it will run on boot.

>STEP-05

How the code works

Each NeoPixel in a series can be addressed individually. As the code only needs to have all the lights the same colour at one time, we've created a **for** loop in the main function that goes through and sets each LED to the same colour one by one. We've also told the code how many LEDs there are, where they're connected, and what frequency to use for refreshing them. The rest of the code is fairly straightforward, waiting for button presses and using delays to manage them.

>STEP-06

Finishing up

That's the pure circuit, but how is it used in the costume? Mel installed a ring of frosted acrylic into the skull she made and stuck the NeoPixel ring behind it: this served to diffuse the individual lights into a more coherent ring. Normal white LEDs were mounted in both eye sockets and controlled by a separate switch for the 'normal' eyes. The button from the Pi was on a long enough cable that it fit sneakily into her pocket – useful, as the character keeps his hands in them!

10 AMAZING ARCADE MACHINE PROJECTS

Building your arcade machine is an amazing idea, and there are projects galore from other makers. Follow in the footsteps of these builds to have your own coin-op at home

The Raspberry Pi has long been the beating heart of many an arcade machine project, and the souped-up Raspberry Pi 3 emulates coin-ops better than ever.

So if you've ever schemed and dreamed of your own personal arcade machine, now's the time to put your plan into action.

You might be pondering what type of arcade machine you'd like to build: a full-sized cabinet, a bartop device, or a cool tabletop machine? Whatever you choose, the Raspberry Pi community has your back. There are designs and concepts for every conceivable shape and size of arcade machine. Some are incredibly detailed, others purely practical, and most look plain fun to build (and play).

Most arcade projects run RetroPie (**magpi.cc/Retro-Pie**) or PiPlay (**piplay.org**), although there are other options like Lakka (**lakka.tv**). Ignition.io (**ignition.io**) is worth watching, too: it's in beta but adds modern features like friends lists, achievements, and online gaming to retro games.

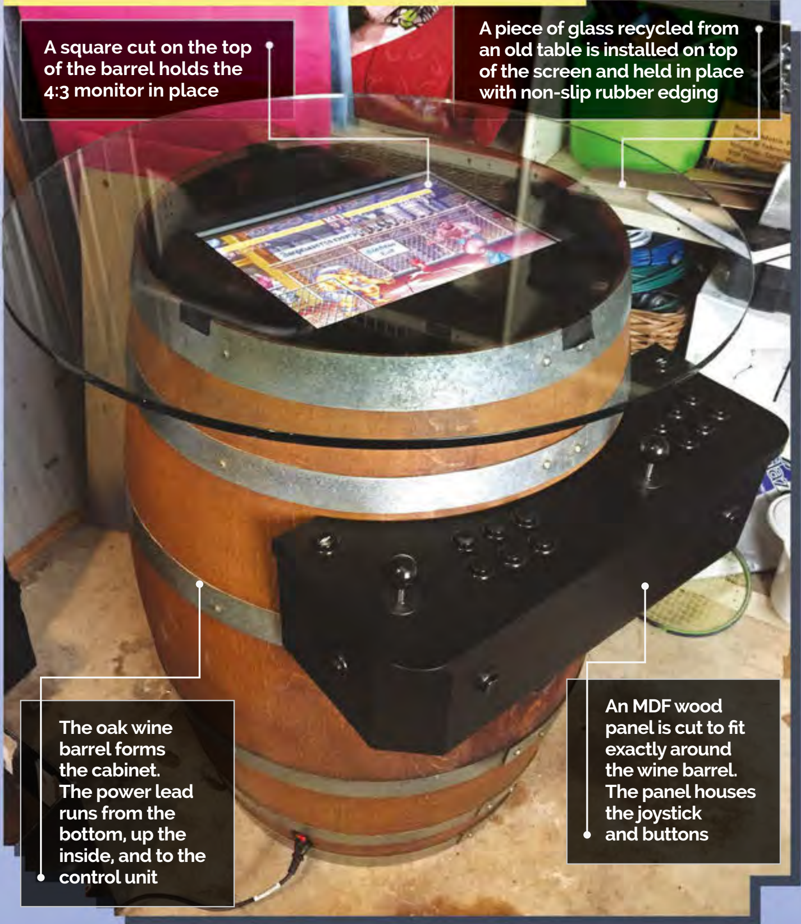
Building an arcade machine is an incredibly rewarding project, and no matter what type of machine you end up with, you're bound to enjoy using it and showing it off to your friends.

WINE BARREL ARCADE

Drain a barrel of wine and turn it into a Donkey Kong-style arcade table

MATT SHAW

Matt Shaw, is a keen gamer who lives in Adelaide, Australia. He was formerly a boilermaker and welder, and has always had a penchant for making things.
imgur.com/a/wzua5



A square cut on the top of the barrel holds the 4:3 monitor in place

A piece of glass recycled from an old table is installed on top of the screen and held in place with non-slip rubber edging

The oak wine barrel forms the cabinet. The power lead runs from the bottom, up the inside, and to the control unit

An MDF wood panel is cut to fit exactly around the wine barrel. The panel houses the joystick and buttons

We adore this grand *Donkey Kong*-style barrel transformed into a tabletop arcade machine.

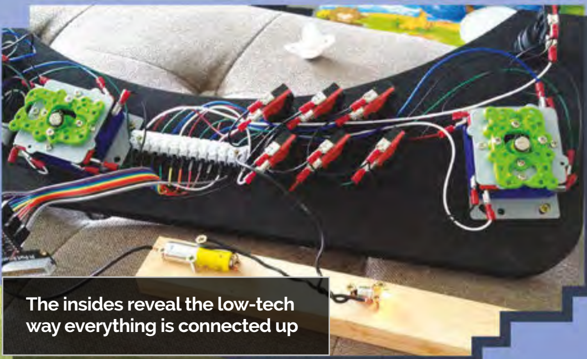
Built by Matt Smith, a gamer from Adelaide, Australia, the Wine Barrel Arcade (**magpi.cc/1S4DI8E**) is a surprisingly low-tech project. There's no soldering, and it uses crimps and block connectors to hold everything in place. All you'd need is a ludicrously oversized barrel and space to store it.

Matt based the idea on an earlier project called A Barrel of Kong (**magpi.cc/1VwrzPK**), although that project featured a JAMMA board (to accept original arcade machine boards). "With a mind to do something like that, I asked for a wine barrel for Christmas," he says. Sensibly, Matt switched the JAMMA arcade board approach of that project to a Raspberry Pi running PiPlay (**piplay.org**).

With a wine barrel rolled into his workshop, Matt sourced an unloved 4:3 monitor and table glass mounted on non-slip rubber. The only real cost was for the buttons, joysticks, and wiring. These are all held together with an MDF panel, cut to fit using an angle grinder, circular saw, and jigsaw. According to Matt, the whole build came in at around £90.

"The guys at the PiPlay forum were amazing, and still help me out (and anyone else that asks, it seems) today," Matt told Australian Kotaku (**kotaku.com.au**)."I love *Wonder Boy*, as it's such fun and a long game, but the *Street Fighter II* series [is] the best." Matt enjoys epic *Street Fighter* battles with his friends over beers.

It's visually impressive, but the best thing about the Wine Barrel Arcade machine is that despite the size, it's a relatively basic construction. We think this is one of the most impressive projects you can make – and if you've got the space, it's sure to impress.



The insides reveal the low-tech way everything is connected up

YOU'LL NEED

- Wine barrel
- 4:3 TFT monitor
- Joystick and buttons
- MDF board

MICRO PI

This desktop cabinet is a fully functioning arcade machine with a Raspberry Pi inside

A unique approach to building an arcade machine is this Micro Pi, designed by Marco Tan. Roughly the size of a can of Coke, it's still a perfectly formed arcade machine.

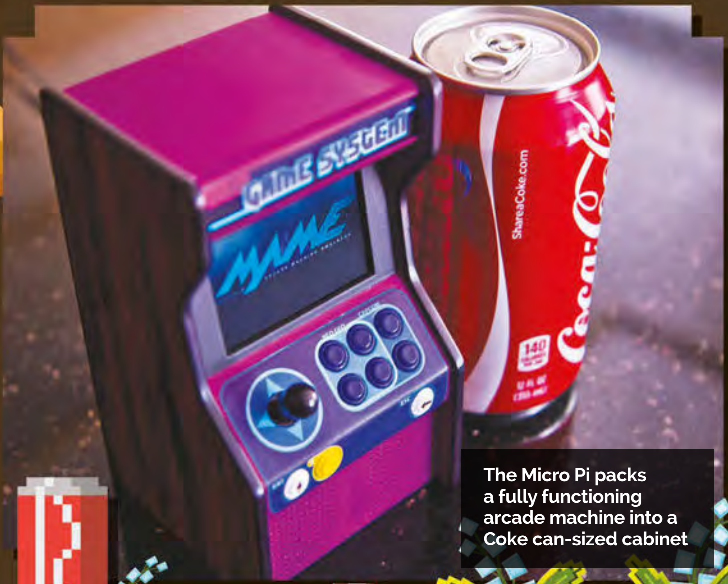
"I always wanted a tiny arcade machine for my desk," explains Marco, "[so] I decided to build it as small as possible. It may look simple from the outside, but a lot of engineering went into making it. After months of development and eight printed prototypes later, it's finally ready."

The cabinet is 3D printed with nylon SLS (Selective Laser Sintering). "It needs to be printed with nylon SLS, or else it won't work: it won't be strong enough," says Marco. You can buy a case directly from Shapeways (**magpi.cc/1S4FqXV**).

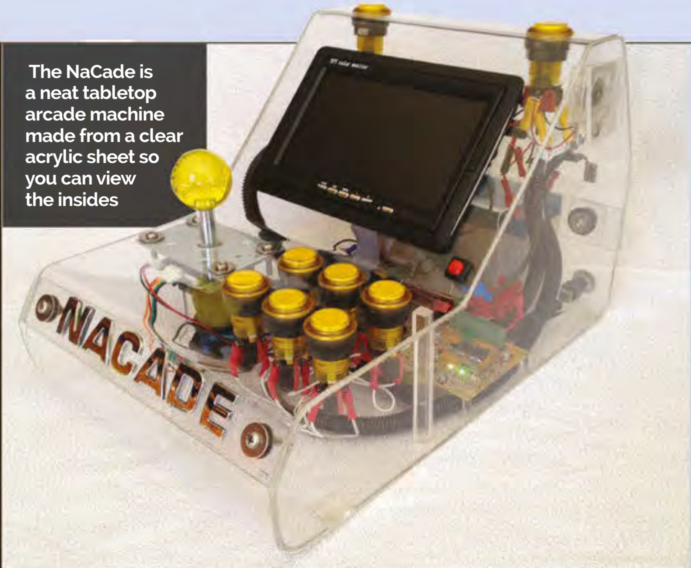
The controls are a GH7455-ND mini joystick and nine 679-2431-ND tactile push buttons. Epoxy is used to hold everything in place, and there's quite a bit of soldering in the project. A speaker is recycled from an old MP3 player to provide audio, and a 2.5-inch TFT forms the display.

It's a fiddly project, but it uses widely available parts, and the case can be pre-bought. There are full details on Marco's Instructables page (**magpi.cc/1S4Fw1I**).

We think this is one of the cutest projects we've ever seen, and it's a superb way to creating a dinky arcade machine that's fully functional.



The Micro Pi packs a fully functioning arcade machine into a Coke can-sized cabinet



The NaCade is a neat tabletop arcade machine made from a clear acrylic sheet so you can view the insides

NACADE

Don't be shy! Show off your Raspberry Pi with pride inside this see-through arcade machine

"Who doesn't like gaming?" asks Krimmy, creator of the NaCade. "Having grown up playing arcade machines, as a kid you could only dream of owning one. Now, with advances in technology, gaming is available to everybody."

NaCade (**magpi.cc/1S4HSgV**) is a naked arcade machine case, displaying the Raspberry Pi powered innards in all their glory. The display is a 7-inch LCD monitor recycled from a car reversing system. "[The display] is sufficient enough for these low-resolution games," says Krimmy, "although the menus can be a bit too small to read. I had to use a larger monitor when I set it up.

"I used an arcade-quality joystick and illuminated buttons, which are wired directly into the Pi," he adds.

Just when you think the NaCade couldn't get any cooler, Krimmy drops the solar panel on you. "The solar controller regulates voltage from an external source, which in this case is a solar panel. Yes, it's powered by the sun!

"Sure, there [are] plenty of consoles and handheld units, even smartphones to choose from, but what I wanted was the nostalgic feel of a stand-up arcade without the need for a large room to put it in. The portability is pretty handy, too."

TOM ROLFE

Tom Rolfe is a writer for TapSmart and *Swipe* magazine.
tapsmart.com

GALACTIC STARCADE

With clear instructions and a solid parts list, this superb arcade machine is a rewarding build

YOU'LL NEED

- MDF board
- LCD TFT monitor
- Joystick and buttons
- Heatsink for Raspberry Pi
- Plexiglas and flexible LED strip kit

The Galactic Starcade is a great bartop project that keeps the weight (and cost) down while providing two sets of controls and a large 19-inch TFT monitor.

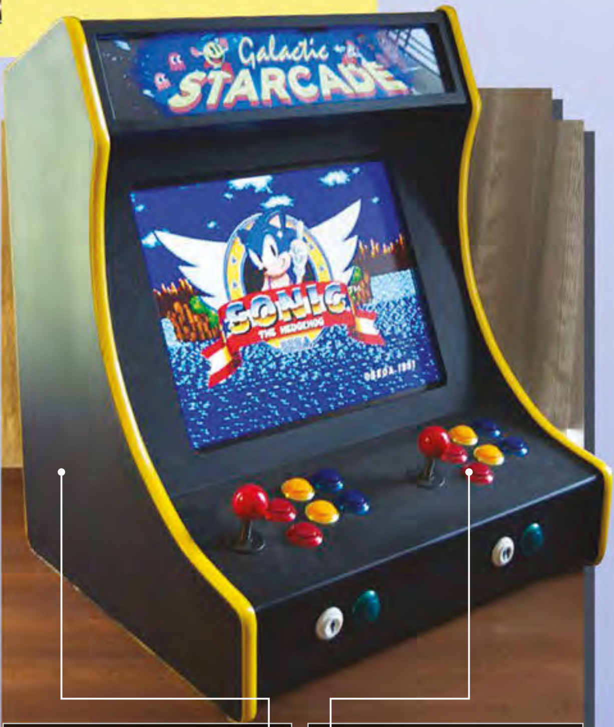
Built by Bristolian techie Tom Rolfe, the Starcade (**magpi.cc/1qOxaVh**) is our tip for a solid arcade cabinet with clear instructions. It's a challenging project, but you won't get lost during the build.

"I've always wanted an arcade machine for authentic retro gaming," reveals Tom, "but they take up a lot of space and cost a lot of money. Making a custom bartop cabinet like this one solves both of those problems. It also lets you play potentially thousands of games on a single machine. This project costs under £200 ($280) to make, whereas a pre-built custom cabinet can set you back four or five times that amount."

What we like most about the Galactic Starcade is the amount of detail in Tom's instructions. "I've knocked up full 1:1 scale printable guides for the side panels and the control deck, plus a reference sheet with dimensions and angles for the rest of the panels," says Tom. You'll find this project easier to follow than many others.

The cabinet is made from painted MDF, and the marquee is Plexiglas and a flexible LED strip kit. A Raspberry Pi is used along with a heatsink to keep things cool, and the controls are from Ultracabs (**ultracabs.co.uk**). It uses RetroPie as the software.

"There's a few things I would do differently, but overall I'm very happy with how this turned out," says Tom. "It proves that a little thing like the Raspberry Pi can happily power a near full-size arcade machine."

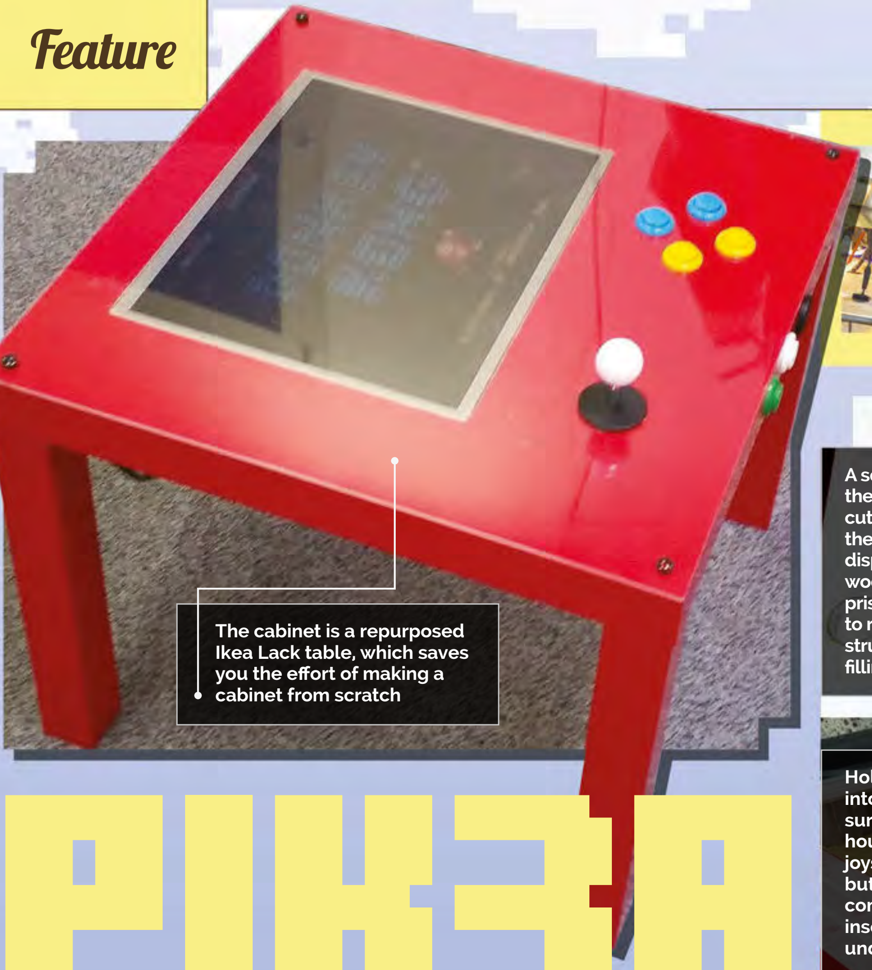


The cabinet is made from regular MDF (medium-density fibreboard) cut to shape and glued together

The controls are an all-in-one unit with a USB interface and pre-crimped wires

A 19-inch LCD TFT monitor with built-in speakers provides both video and audio output

A hinge on the rear of the cabinet provides quick access to the Raspberry Pi inside



The cabinet is a repurposed Ikea Lack table, which saves you the effort of making a cabinet from scratch

PIK3A

Some ideas are pure genius, like this Ikea coffee table turned into an arcade machine

YOU'LL NEED

- Ikea Lack coffee table
- Raspberry Pi and Arduino Leonardo
- Four-way ball-top joystick and buttons
- 17-inch LCD monitor
- USB computer speakers

We love getting ideas in Ikea, but clearly not as much as Element 14's community manager, Spanner Spencer, who had the genius to take apart an Ikea coffee table and turn it into an arcade table.

"It's an IKEA Lack coffee table with an LCD monitor cut into the top, arcade controls next to the monitor, and a Raspberry Pi 3 and accessories buried inside," explains Spanner.

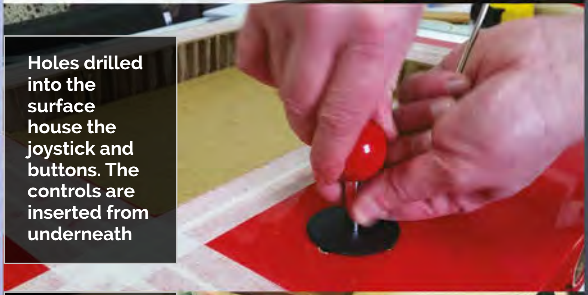
He describes it as a "minimalist, contemporary interpretation of the classic coin-op cocktail cabinet that uses an IKEA coffee table and a Raspberry Pi 3."

The display is an old 17-inch LCD monitor with a 4:3 ratio (this shape is better to match the square table). The chassis is taken out of the plastic casing, and the screen inside the shielding is the same depth as the Lack coffee table. "This means that once you've got the screen, all you need to do is drop it into the hole [you cut]," says Spanner.

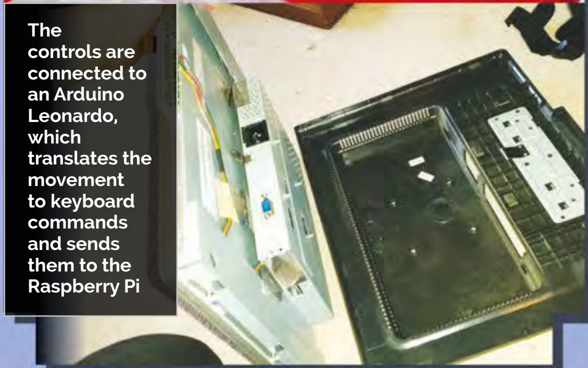
SPANNER SPENCER

Spanner is the community manager for Element14, an online community for engineers.
magpi.cc/1q0Dxb9

A square in the table is cut to contain the 17-inch display. The wood is then prised out to reveal the structural filling inside



Holes drilled into the surface house the joystick and buttons. The controls are inserted from underneath



The controls are connected to an Arduino Leonardo, which translates the movement to keyboard commands and sends them to the Raspberry Pi

There are clear instructions for joysticks and buttons on Element 14's website (**magpi.cc/1q0xwLG**). "Drill 28mm holes for each one," says Spanner. "This is the standard size for arcade buttons, and also gives the joystick plenty of room to move without the hole being visible around the round, flat cover that comes with the joystick."

The PIK3A uses an Arduino Leonardo (**magpi.cc/1q0xyTQ**) to interface the controls with the Raspberry Pi. It's an interesting way to hook up the controls, and a lot easier than other implementations we've seen.

MINI ARCADE

Looking for the perfect scale arcade machine? Look no further

We featured Tiburcio de la Carcova's Galaga Pi project in issue 44, but we couldn't do a feature on our favourite arcade projects and not mention the Mini Arcade.

Tiburcio's miniature arcade machine reproductions (of which *Galaga* is just the latest) remain the gold standard for scale reproductions (they even have small coin slots). Hand-built from plywood and acrylic, and with as many 3D-printed parts as possible, they are a labour of love. Reproduction is probably beyond all but the most dedicated of makers, but Tiburcio takes us through the build of *Galaga* on his YouTube channel (**magpi.cc/1V8XEvY**)

So far he's built perfect reproductions of *Space Invaders*, *Pac-Man*, and *Galaga*, with more to come. They're the most inspirational builds around.

ARCADE PI

This bare-bones arcade project is space-saving and great value

Arcade Pi is alone in our list of favourite arcade machines in that it isn't a complete build. Instead, it packs the Raspberry Pi inside a wooden base containing the arcade stick and buttons.

We think the Arcade Pi (**magpi.cc/1Q5gGw8**) is a great option for those looking to build an arcade machine, but not having the space, or money, for a full cabinet build. "I've always dreamed of having an arcade machine," says creator Sacha. "Man, it takes a lot of space, and it's expensive. I arrived at the conclusion that I had to build one myself."

This project enables you to experiment with arcade joysticks and buttons and build a working arcade console in a much smaller space, and for a much lower price.

Like all great arcade sticks, you should definitely customise the face plate yourself

LEGEND OF ZELDA BARTOP

There are lots of bartop projects around, but this *Legend Of Zelda* (**magpi.cc/1qOD3lx**) is one of the prettiest. "I've always wanted my arcade cabinet," says Phrazelle. "I'm glad I did because the final build exceeded my expectations by far."

COFFEE TABLE PI

Graham Gelding's Coffee Table Pi (**magpi.cc/1qODcFy**) is one of the neatest arcade tables around. It also has a huge 24-inch LCD screen and is sturdy and child-friendly (with rounded corners and Perspex mounted over the screen).

BUBBLE BOBBLE

We just adore Christopher Sadler's *Bubble Bobble* bartop arcade machine (**magpi.cc/1VwzpsM**), if for no other reason than its lovely decals. "All my plans were based around wanting to have artwork from my favourite game ever: *Bubble Bobble*," says Christopher.

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.

PI CAMERA MODULE V2

WHAT ARE THE OTHER FUNCTIONS?

The RAW function

The `raspistillyuv` command is a much more advanced option than `raspistill`, as it creates a RAW file. These can be edited using specific software, to take into account white balance and other colour levels much better than in a compressed JPG or PNG. You'll need to do this on another computer, though.

The time-lapse option

The option `--timelapse` when taking a photo can be used to create time-lapse photography – it's a massively popular thing to do with the camera, so being part of the software makes it easy. Set a time with `-t` for the desired duration of the time-lapse, then set intervals with the time-lapse option.

The keypress option

The option `--keypress` will allow you to take photos when you press the **ENTER** key for however long you've set the time for. If you give a time of 0, it will run indefinitely until you press **X** then **ENTER**. A slightly different way of shooting with the camera!

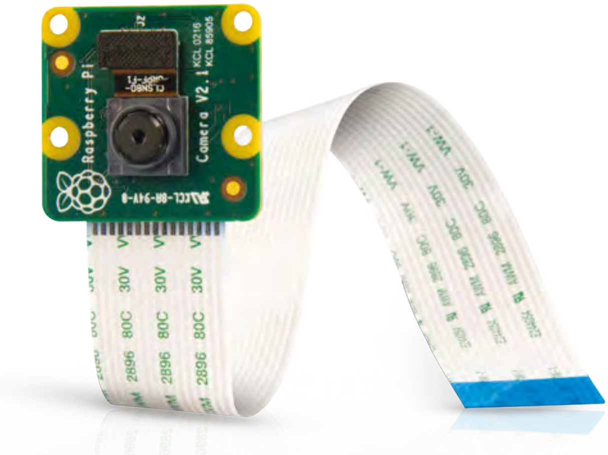
HOW CAN I VIEW MY IMAGES?

On the Pi

The Raspberry Pi comes with an image viewer that allows you to just open and view the photos you've taken. This works a little better on Raspberry Pi 2 and Pi 3, especially when you're using the maximum resolution of the camera for photos!

On a computer

You can easily plug an SD card into a PC or Mac and view the files on there. They'll be in the home folder, which can be found by going to **home**, then **pi** on the file system. Copy them off to make sure you don't lose them.



Online

You can easily upload the photos to your social media or cloud storage accounts via the web browser, either for safekeeping or to show them off. The files are normal image files, so will be recognised by any online service to upload or post.

HOW CAN I WATCH MY VIDEOS?

On the Pi

The Raspberry Pi, by default, has a video player called omxplayer, which you can access via the command line. While it does have a number of options, you can simply use `omxplayer [filename]` to play a file at maximum resolution.

On a computer

Alternatively, you can just grab the SD card from the Raspberry Pi and plug it into your PC. A video player like VLC will easily play the videos recorded on the Raspberry Pi camera – but remember, you won't have any sound in the file!

Playing slow-motion video

Not all the players will automatically play slow-mo video at slow-mo speed. At normal speed, it will play a real-time video with a very smooth frame rate, but with VLC you can turn down the playback speed to get the true slow-mo effect. Most video editors will let you slow down the video as well.

FROM THE RASPBERRY PI FAQ RASPBERRYPI.ORG/HELP

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

Where is the on/off switch on the Raspberry Pi?
There is no on/off switch! To switch on, just plug it in. To switch off, open the top-left menu in the Raspbian GUI and click Shutdown at the bottom, then press OK. Alternatively, from the Bash prompt or a terminal, enter `sudo halt -h` to shut it down. Wait until all the LEDs except the power LED are off, then wait an additional second to make sure the SD card can finish its wear levelling tasks and write actions. You can now safely unplug the Raspberry Pi. Failure to properly shut down the Raspberry Pi may corrupt your SD card, which would mean you would have to re-image it.

Why does nothing happen when I type in my password? Did my Raspberry Pi freeze?
Don't worry, it's still working. To protect your information, Linux doesn't display anything when typing in passwords in the Bash prompt or the terminal.

How do I connect a mouse and keyboard?
The Model A/A+ has one USB port, the Model B has two ports, and the Model B+, 2B and 3B have four ports. These can be used to connect most USB 2.0 devices. Additional USB devices such as mice, keyboards, network adapters, and external storage can be connected via a USB hub.

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 press '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. Just type in your search term to find the relevant results.

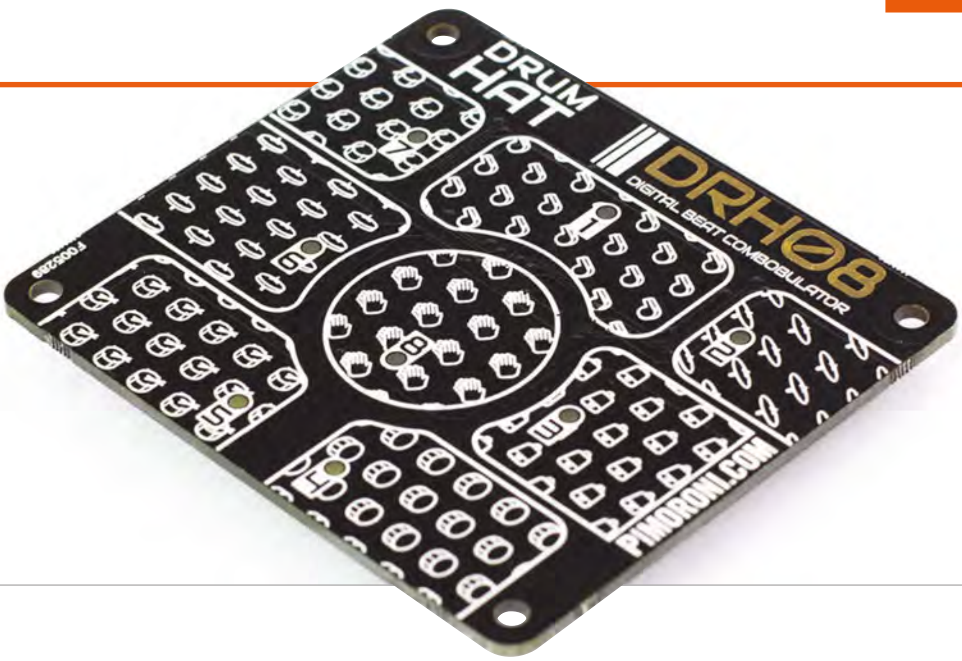
magpi.cc/1VwzNGn

£12 / $17

Maker Says

“Raspberry Pi drum-kit that lets your fingers think they're Stubblefield

Pimoroni



DRUM HAT

Hot on the heels of Pimoroni’s successful Piano HAT comes this hot drum machine. Discover the joy of code-based finger drumming

A while ago we came across the Piano HAT, a snazzy piece of hardware based on Zachary Igielman’s PiPiano and turned into a HAT (hardware on top) device by Pimoroni.

The Drum HAT is its funky brother, capable of quickly transforming a Raspberry Pi into a drum machine. On top of the board sit eight capacitive sensor pads; you tap the beat out with your fingers. Each pad also sports an LED that lights up when you tap (or can be programmed separately).

Installation is easy thanks to a command listed on the Drum HAT’s GitHub page (**magpi.cc/1VwzZWb**). Just enter **`curl -sS get.pimoroni.com/drumhat | bash`** to get started. This script installs the Python modules and downloads a bunch of sample WAV files and programs Enter **`python drums.py`** and you’ll quickly have a drum machine ready to play. A file called **direct.py** links the samples in the **drums2** folder, and you can edit the Python code to link to any folder you want. Then it’s just a case of creating some drum samples, or downloading sample files from a site like **looperman.com**.

Amen to that

We started by recreating ‘Boots and Cats’ (**youtu.be/NniorTLg5B8**) using our sampled voices, then grabbed a bunch of samples of the ‘Amen Break’ (**youtu.be/5SaFTm2bcac**) and set to turning a Raspberry Pi into a kick-ass drum-and-bass machine.

Taking apart the sample code (and reading the GitHub page) enabled us to figure out the Drum HAT code. You can set the pads to react when hit, or released, and you can get the pads to call a function that can do anything.

Building a drum machine is where it’s at, though, and we had an awesome amount of fun with the Drum HAT (much more than with the seemingly more complex Piano HAT). A good project to try out.

Related

PIANO HAT

The Piano HAT is more detailed, with 16 touch-sensitive buttons (forming 13 piano keys). You can use the Piano HAT and Drum HAT together using a Black HAT Hack3r device.

£15 / $21

magpi.cc/1OALwNT

Last word

Easy to set up and fun to bash around on, the Drum HAT turns a Raspberry Pi into a home-made 808 drum machine.

★★★★☆

MONK MAKES

Books & Other Bright Ideas from Simon Monk

Electronics Starter Kit for Raspberry Pi

This kit contains everything you need (apart from a Raspberry Pi) to create ten electronic projects.

£15 / $25

- 10 easy to follow project cards
- Quality 400 tie-point breadboard
- 20 jumper wires
- All Python code in one easy download
- Compatible with Pi B, B+, A+, Pi 2
- Includes RGB LED, red LEDs, switches, buzzer, light sensor, temperature sensor, and capacitor
- No soldering required

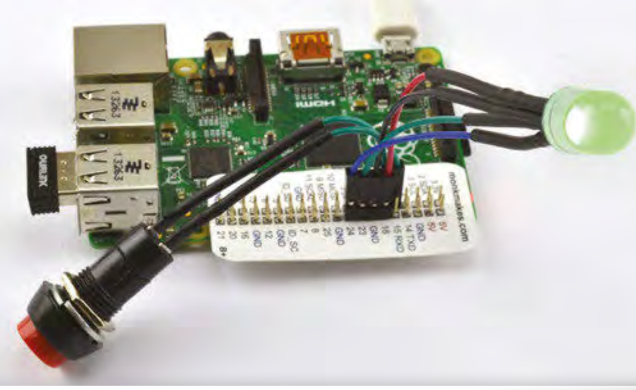
RasPiRobot Board v3

A simple to use motor controller for the Raspberry Pi. This board is designed to simplify the process of making small roving robots and controlling medium power loads using a Raspberry Pi.

£17.50 / $25 - $29.95

- Dual bi-directional motor control using TB6612FNG dual H-bridge
- Dual open drain 2A MOSFET outputs
- Socket for HC-SR04 Rangefinder
- Socket for 5V I2C Interface
- Provides regulated power to Raspberry Pi
- 2 x user controllable LEDs
- 2 x header pins for switches
- Reverse polarity protection
- Open Source Python Software library
- Screw terminals for motor and power connections
- Compatible with Pi, B+, A+, Pi2

Please note: Illustration of possible use only, chassis, Raspberry Pi, motors, rangefinder etc. not included



Raspberry Squid Combo Pack

Using the included Raspberry Leaf GPIO template, attach an RGB LED and push buttons directly to the GPIO pins of a Raspberry Pi. The switches are panel-mountable and perfect when putting your Raspberry Pi project into an enclosure.

£9.65 / $14.95

- Includes one RGB LED Squid and two Squid Buttons and a Raspberry Leaf GPIO template
- RGB LED and switches plug directly into Raspberry Pi GPIO pins
- Simple to use Python library for RGB LED control and switch debouncing

GET A 10% DISCOUNT

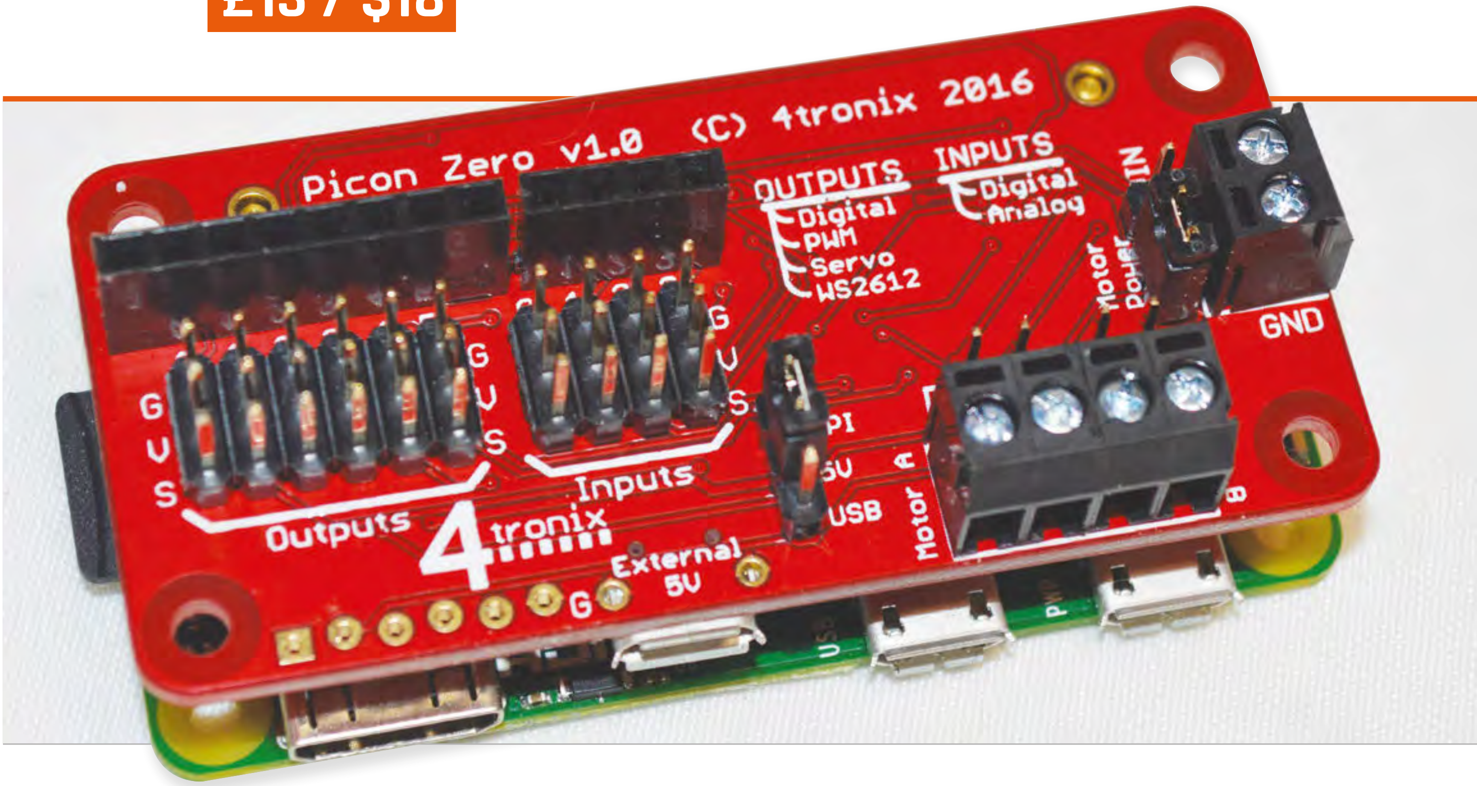
when you buy through amazon.com or amazon.co.uk

Enter promo code: **MAGPINOW**

www.monkmakes.com | twitter@monkmakes

magpi.cc/1p9wGaA

£13 / $18



PICON ZERO

A new robotics board from 4tronix that fits snugly on your Pi Zero for some miniature robotics

Pi Zero robots are quickly becoming popular. This was inevitable, given the nature of the Pi community and just how useful the Zero can be in such a situation. We've already got a new kit from the folks over at PiBorg coming soon after a successful Kickstarter, and plenty of little projects and builds that use it for diminutive automatons. Usually, these small robots use pre-existing small motor-controllers. However, we now also have the Picon Zero from Pi robot veterans 4tronix.

Its claim to fame is that it's a robot control board with the same size/form-factor as the Pi Zero. This means it slots right on top of a Pi Zero without poking over any of the edges. This is particularly handy, as it allows you to easily access the ports and SD card and such. It also means that you'll need to have a set of GPIO pins soldered onto the Pi Zero, which is really not a massive ordeal, although we know the idea of it is still a little scary for some.

It's certainly worth it, as the Picon Zero comes with a huge amount of benefits for robot building with the Pi. As you might expect, it can power two motors; unusually for a motorboard, it can draw power directly from the Raspberry Pi for this. This uses 5V, which the Pi can provide but which it is rarely used for due to the power draw. With two small motors, it's not so high; if it's a little too much for the Pi, however, the Picon can also take USB power. This can be a little easier to hook up and maintain than rechargeable batteries, so it's a neat addition.

The ins and outs

One of the best things about the board is the numbers of inputs and outputs it supports, especially for its size. Dotted around the PCB are inputs that accept digital and analogue signals, as well as having the ability to be configured to use popular temperature and humidity sensors. Output-wise, it's also fairly powerful, with configurations to allow digital output, the ability to use PWM, standard servo controls, and even NeoPixel support.

This is amazing for such a small device. Some full-size HATs or controllers for the Pi can handle motors and a HC-SR04 ultrasonic

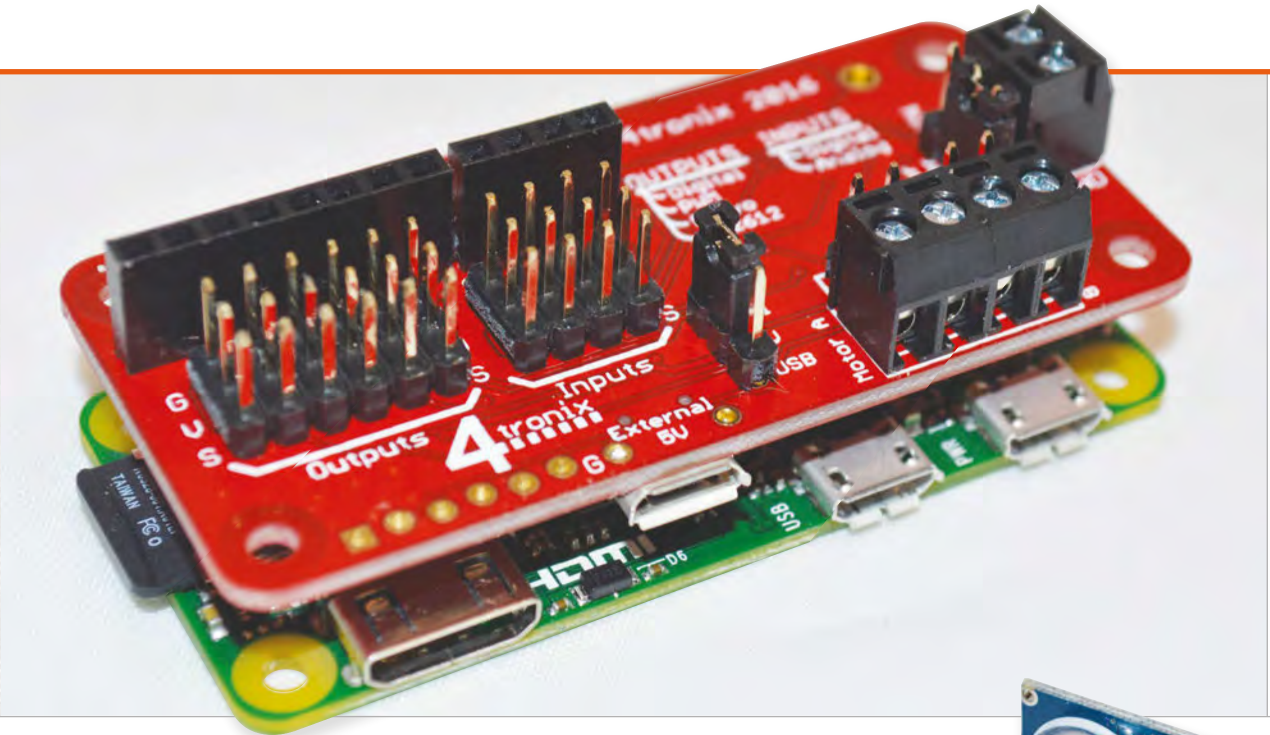
Related

RASPI ROBOT BOARD V3

Create slightly different robots with this board that allows access to all the GPIO pins for more manual construction.

£18 / $30

magpi.cc/1VBwoYG



Maker Says

“Intelligent robotic controller for Raspberry Pi”
4tronix

Left The board sits well above the Pi Zero, but not so high that it significantly decreases space efficiency

sensor input (which, by the way, the Picon also has a dedicated input for), but then you still might have to play around with some GPIOs to get the rest working. The Picon is incredibly comprehensive; it even opens up a small section of the GPIO port with five GPIO pins and a selection of Ground, 3V3, and 5V pins.

> One of the best things about the board is the number of inputs and outputs it supports

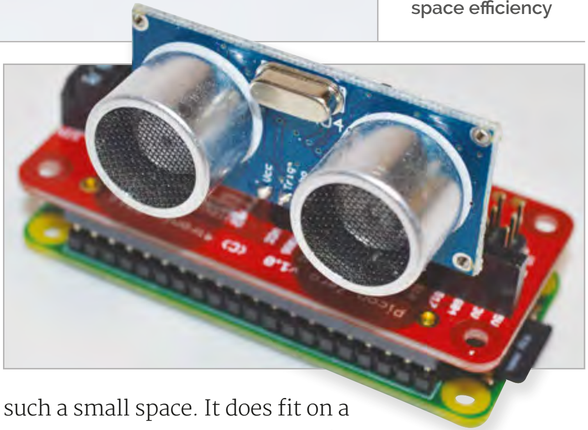
Code it

The whole system is powered by a Python library custom-made by 4tronix. It's incredibly simple to install, simply requiring you to download a script that will do all the work. It comes with some example Python scripts and the modules needed to operate the board with your own custom scripts. A full breakdown of the features can be found on the 4tronix blog for the Picon Zero (**magpi.cc/1SMc8N8**), but there's a huge number of functions that are very simple to program if you have a cursory understanding of Python. There are also some worksheets for teaching the ins and outs of using the library; at the time of writing, however, they only get so far as to teach you how to read the input of a reed switch, and cover nothing to do with motors.

They're likely to be on their way very soon and you shouldn't be discouraged by that, as reading up on the functions and looking over the test and example code that comes with the Picon libraries should be enough to get you making full use of the Picon.

The only real issue we have with the Picon is that things can get a little tight on the board if you're using a significant number of inputs and outputs, although that's the price you pay for it all being in such a small space. It does fit on a regular Raspberry Pi as well, so for the price it may well be something you could consider for any type of Pi robot project.

It's certainly one of the most comprehensive robot HATs we've come across for the Pi, and that low price makes it pretty special as well. Absolutely worth a look if you're wanting to make a more complex machine for your robotic needs.



Above An ultrasonic sensor can plug straight into the board if you want to keep things compact

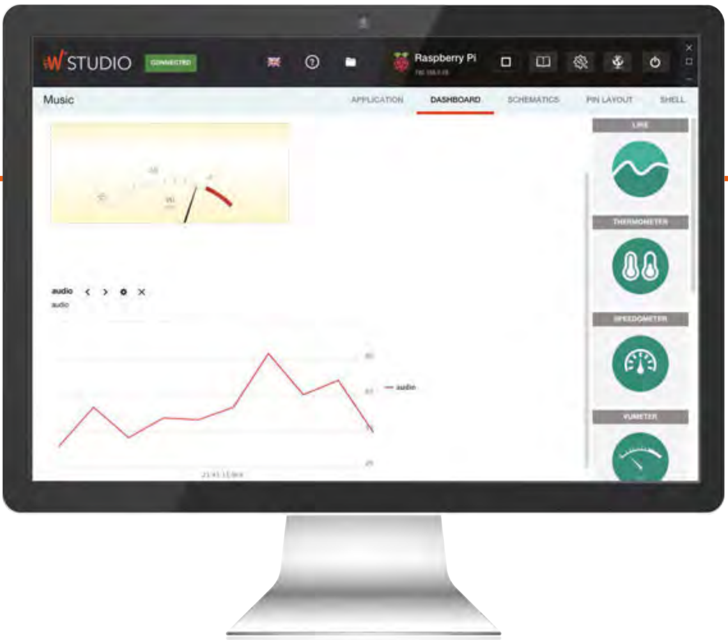
Last word

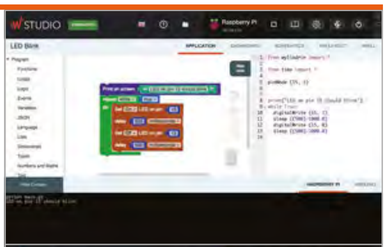
A wonderful robot controller at an amazingly low price. While designed for the Pi Zero, it can be used on any Raspberry Pi.

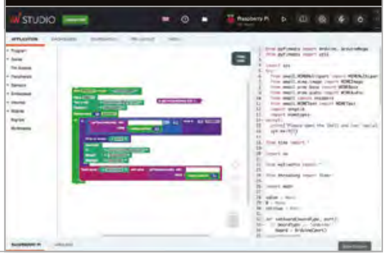
★★★★★

magpi.cc/1Q5i4il

Free







Maker Says

> Like a Web Control Panel on steroids - lots of steroids
>
> **Wyliodrin**

WYLIODRIN STUDIO

This Chrome App development tool makes it easy to create Internet of Things projects on your Raspberry Pi

Wyliodrin STUDIO is a Chrome App development suite that enables developers to take a board, like our beloved Raspberry Pi, and quickly prototype an IoT device.

Setup is a breeze: installation is via disk image and from that point on, everything takes place from Chrome on another computer. You can still use the Wyliodrin website (**projects.wyliodrin.com**), which has new features, but what we're really interested in is the Wyliodrin STUDIO Chrome Web Store app (**magpi.cc/23swO8o**).

Got a visual

Wyliodrin supports JavaScript and Python, but the entry point is Visual Programming, a beefed-up version of Google's Blockly. Programs are arranged, like Scratch, by dragging and dropping blocks together.

It's much more powerful than either Scratch or Blockly, with blocks for everything from social media to humidity sensors and accelerometers. There are blocks for GoPiGo, Lego Mindstorms, and Grove sensors. You can even set up a web server using blocks.

The blocks are converted to Python code on the fly, and you can move back and forth between Visual Programming and Python.

It's easy to be sniffy about using blocks when you've taken the time to learn Python or Java, but it's impossible to make a syntax error. Therefore, you can focus on the logic, and we find slotting blocks together a fun way to get inspiration. There are limitations (it doesn't scale up particularly well), but you can always move your project over to Python.

In a dash

The other side of Wyliodrin is the Dashboard, which features an array of graphs, charts, dials, buttons, and sliders – everything you need to visually interact with your Raspberry Pi. There is a great range of example programs included, and some powerful tutorials to follow.

Wyliodrin STUDIO doesn't have the limitations of the browser-based service, and you can add as many projects and Dashboard widgets as you want. Note that the contract states that it's for personal and educational use only.

We're impressed with Wyliodrin STUDIO. It's a solid IDE that's easy to set up, with no real restrictions. Visual Programming is a great way to demystify the often enigmatic nature of IoT prototyping, and there are some great tutorials available.

Related

WINDOWS 10 IOT CORE

Setup is completely different, but Windows 10 IoT Core is a rival platform targeting the same developer community.

Free

magpi.cc/1q0EEb1

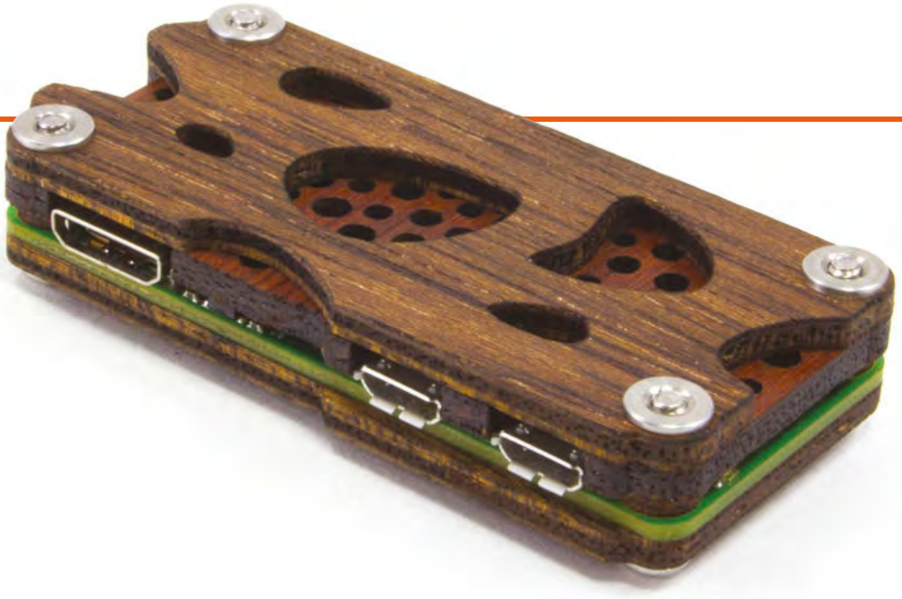
Last word

A solid springboard to start building your internet-based projects. Visual Programming is a fun way to get started, but you're not limited in scope. The Dashboard brings your projects to life with buttons and dials.

★★★★★

pihut.com

£5 / $5



Maker Says

> This eye-catching case is sure to get your friends' attention
>
> **C4 Labs**

NUCLEUS ZERO PI ZERO CASE

A case completely made of wood for the Raspberry Pi – it's absolutely not a fire hazard

The Zebra Zero case we reviewed a couple of issues ago made quite a splash when it was first unveiled. Having wooden parts is still fairly unusual for Raspberry Pi cases, although the Zebra Zero is wooden (albeit with a plastic centre). This C4 Labs case, though, is completely wooden, through and through. We reckon the screws are metal, but it even comes with wooden spacers. Very cool.

The design is very funky, with lots of asymmetrical cuts in the material that give an effect of naturally grown wood, but also allow for some airflow to the chips on the board. Very clever. There's easy access to the three main ports on the side; in fact, the way the wood is cut for these ports and the SD card slot, it's pretty easy to figure out how it's meant to be constructed. It's only three pieces, after all.

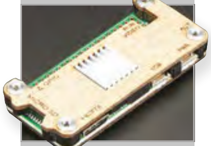
Access to the SD card is a little tricky compared to the bare board, and you'll curse cutting your nails short if you're in the habit of taking the card out regularly. Not everyone needs to do this often, though, and it's still quite possible. One thing unique about the Nucleus is that the GPIO holes are completely covered up by the case. For some this won't be a problem, and it's nice that this is an option if you don't plan to use them. There are plenty of other cases out there that open up the GPIO, such as the aforementioned Zebra Zero, so we're not too bothered about it. In fact, we like it as an alternative for certain Pi Zero projects.

This is a pretty sturdy and very protective case for the Zero, then. It's nice and cheap, too, especially as it's made of quite nice-looking wood (it's still laser-cut, of course). Really, it's just an eye-catching case and if a full-coverage case is what you're looking for, then maybe it's the one for you.

Related

ZEBRA ZERO CASE

Still slightly wooden, also with a plastic alternative, the Zebra Zero is slightly better for projects that need access to the GPIO, while still being a good case for the rest of the Zero.



£5 / $7

pihut.com

Last word

A nice alternative case, whether for the wooden construction or for the full coverage of the Pi Zero. It's well designed, while still looking great.

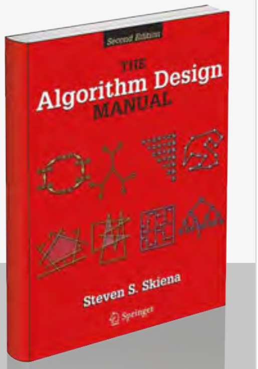
★★★★☆

RASPBERRY PI BESTSELLERS

ALGORITHMS

Improve your handling of computational problems with a thorough grounding in algorithms

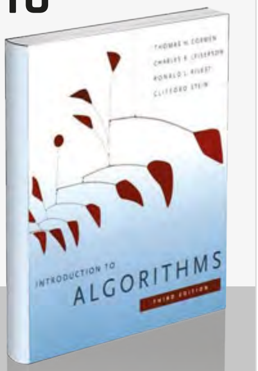
THE ALGORITHM DESIGN MANUAL



Author: Steven S Skiena
Publisher: Springer
Price: £55.07
ISBN: 978-1848000698
magpi.cc/1VDJchi

Skiena tackles algorithms with an engineer's mindset; lacking the mathematical rigour (and page count) of the MIT Press book (see below), it wins out on real-world examples, making a most practical introduction.

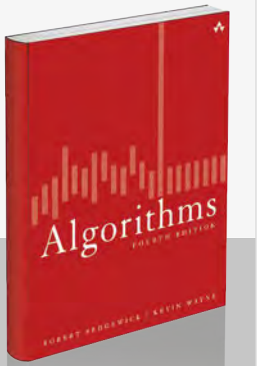
INTRODUCTION TO ALGORITHMS



Authors: Thomas H Cormen, Charles E Leiserson, Clifford Stein & Ronald L Rivest
Publisher: MIT Press
Price: £44.95
ISBN: 978-0262533058
magpi.cc/1VDJt3l

Perhaps the definitive reference work and tutorial on algorithms, but you'll need a good grasp of mathematics, as well as some programming experience, to get the most out of it.

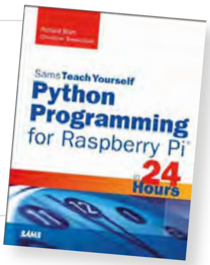
ALGORITHMS



Authors: Robert Sedgewick & Kevin Wayne
Publisher: Addison Wesley
Price: £55.99
ISBN: 978-0321573513
magpi.cc/1VDJrch

Sedgewick's popular book – which has its own two-part Coursera MOOC – has clear explanations and diagrams, and useful Java implementations which should translate to your preferred language (but check errata).

SAMS TEACH YOURSELF PYTHON PROGRAMMING FOR RASPBERRY PI IN 24 HRS



Author: Richard Blum & Christine Bresnahan
Publisher: SAMS
Price: £21.99
ISBN: 978-0672337642
magpi.cc/1VDJyV6

"Making Python the official programming language of the popular Raspberry Pi was genius," write the authors in the introduction. Some measure of astuteness must also be ascribed to Pearson's Sams imprint for realising a couple of dozen tutorial chapters of typical length made the equivalent of a day – albeit a somewhat demanding one – and creating the 'Teach Yourself ... in 24 Hours' series. Now Blum and Bresnahan, both experienced sysadmins, have updated and revised their fine introduction to programming in Python on the Pi.

Aimed squarely at those new to both Python and Pi, the first three 'hours' cover setup for both. That done, the next four hours are spent on programming fundamentals, building simple scripts, then advancing to logic flow control. Useful introductions along the way include the NumPy module, iterating over lists, and dealing with errors.

A ten-hour chunk of 'Advanced Python' covers topics like lists and tuples, creating functions, and object-oriented programming, as well as exception handling and an introduction to regular expressions. Next is GUI programming with Tkinter and Pygame, 'business programming' (network, database, and web server), and multimedia and GPIO projects round off a very useful guide. Recommended for any beginner with 24 hours to spare.

Score ★★★★☆

BEGINNING THE LINUX COMMAND LINE

Author: Sander van Vugt
Publisher: Apress
Price: £26.50
ISBN: 978-1430268307
magpi.cc/1VDJPHL

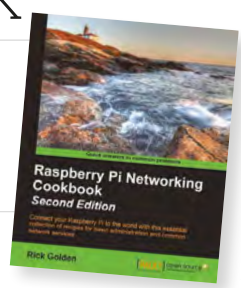
Every user of Raspbian, or any other GNU/Linux distribution, becomes in a small way a systems administrator. Most tasks are best accomplished via the command line, and van Vugt's excellent introduction is very much aimed at practical administration, whether the reader has a Pi, a server they're programming, or they're a Windows sysadmin taking on Linux tasks.

Providing plenty of background along the way – which should help new users understand the 'why' along with the 'how' of the command line – the author starts with several practical commands which give the user useful control of her system, with users, files, and disks all covered well in the first three chapters. The chapter on working with text files starts with mastering vi - the one editor found on all Unix computers. Further chapters are usefully task-oriented.

Steering a middle path between quick introduction and comprehensive reference, *Beginning the Linux Command Line* gives the reader practical information without overwhelming with detail, yet still fills in background where context is necessary for real understanding. The final chapter, on shell scripting, is a good example of the author's well-paced tutorial approach, and leaves the reader prepared for tackling many admin tasks.

Score ★★★★☆

RASPBERRY PI NETWORKING COOKBOOK



Author: Rick Golden
Publisher: Packt
Price: £28.99
ISBN: 978-1785280214
magpi.cc/1VDKs3U

Despite some bottlenecks in networking through Ethernet and USB-connected network interfaces, the Pi 2 is, as Golden points out in his preface, every bit as powerful as a network server from the late 1990s. This being so, despite 20 years of Moore's Law growth for modern network servers, the Pi 2 (and now the Pi 3) can be used for many network tasks to "share files, host websites, create internet access points, and analyse network traffic."

With chapters collecting recipes both by theme and by difficulty, there are entry points in this work for beginner and intermediate users. The early chapters also include enough information on setup, system administration, and maintenance, to enable those new to the GNU/Linux command line to learn the ropes.

Recipes for file sharing enable you set up the Pi as a network file server for home or office. The Advanced Networking chapter covers firewalls, web servers and wikis, wireless access points, and network analysis tools. Internet of Things goes from GPIO setup, through GrovePi, to web.py and controlling devices through a webpage. Lastly, there's computing over the network with clustering – and setting up your own Raspberry Pi "supercomputer". A valuable introduction and reference.

Score ★★★★☆

ADAPTIVE WEB DESIGN

Author: Aaron Gustafson
Publisher: New Riders
Price: £21.99
ISBN: 978-0134216140
adaptivewebdesign.info

It's a full house of second editions this month, and Gustafson's is a welcome update to this high-level yet practical guide to the why and how of adaptive web design. The approach here is more than adding responsive design for mobile browsers. Progressive enhancement is the mantra, building site features on a foundation of content that's accessible to all browsers and users.

From GitHub.io project pages to WordPress blogs, it's almost inevitable that makers end up responsible for websites, and thus for web design, so this could be the book to make you think more seriously about the subject. Aimed at developers, project managers, and web professionals, it's broad enough to be approachable for all, yet deep enough to leave you with a practical idea of what needs to be done to make your site(s) a pleasant and informative experience for all visitors.

Gustafson starts squarely with content – sadly an afterthought for many designers, but the only thing everyone needs from a site. Markup, visual styling, and interactions are then added to a common base of universally accessible content, ensuring each user gets the best site for her browser, device, and screen size. Practical, thoughtful, and a good read, too.

Score ★★★★☆

ESSENTIAL READING: ROBOT BUILDING

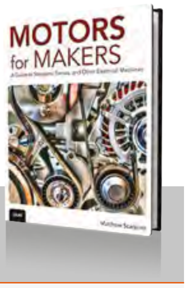
Classics and recent releases to help you get your robotic project moving

Raspberry Pi Robotics Essentials

Author: Richard Grimmett
Publisher: Packt
Price: £19.99
ISBN: 978-1785284847
magpi.cc/1VDKDw8

Build a walking robot and add six degrees of freedom, plus sensors, vision, and path planning.

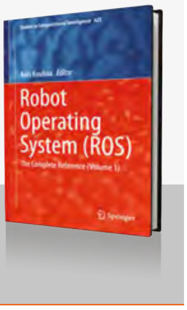
Motors for Makers



Author: Matthew Scarpino
Publisher: Que
Price: £21.99
ISBN: 978-0134032832
motorsformakers.com

Detailed and practical guide to using motors with the Pi and other boards, with code and vehicle projects.

Robot Operating System (ROS): The Complete Reference



Author: Anis Koubaa (Editor)
Publisher: Springer
Price: £149.00
ISBN: 978-3319260525
magpi.cc/1VDKR6k

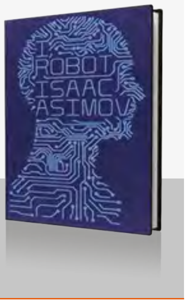
27 chapters, both practical and academic, giving comprehensive coverage of open-source middleware for autonomous robots.

OpenCV with Python By Example

Author: Prateek Joshi
Publisher: Packt
Price: £31.99
ISBN: 978-1785283932
magpi.cc/1VDKVmE

Well-regarded guide to the practical side of computer vision algorithms, that enables the reader to understand the problems – and solutions.

I Robot



Author: Isaac Asimov
Publisher: Harper Voyager
Price: £7.99
ISBN: 978-0007532278
magpi.cc/1VDL9uf

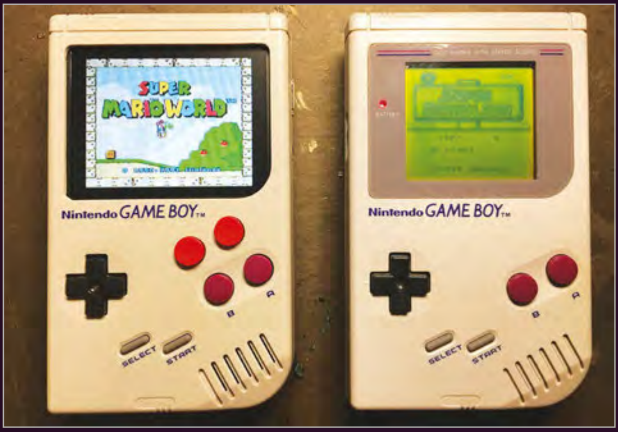
Robots that think like us are still science fiction, but Asimov's pioneering stories ask essential philosophical (and relevant) questions.

THE MONTH IN RASPBERRY PI

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

RASPBERRY PIS FOR GAMING

Over the last month, we've seen some amazing little projects that have made use of the Raspberry Pi for gaming, or just fitting one inside an old games console for the fun of it. It comes just in time as well; the software that usually powers all of this, RetroPie, has recently been updated to 3.6, and has a new beginner's guide for people getting their new Pi 3s. Here are some of the best gaming-based projects we've seen this month…



GAME BOY ZERO

imgur.com/a/jyjmG

This is a really cool one: Wermy has completely hacked an original Game Boy to be a portable retro-gaming machine. Well, the Game Boy was already one of those, but it couldn't play SNES games, and this one can. Powered by a Raspberry Pi Zero and with extra buttons added to handle more demanding titles, the Game Boy Zero is quite the feat of engineering. The SD card actually resides in a standard Game Boy cartridge, which needs to be plugged in so the Pi can actually have an operating system. It has incredible attention to detail, and adds to the masses of great 'Pi Zero in a small games thing' projects that have been appearing in the last few months. You can also check out a video of it all working on YouTube: **youtu.be/aMhTMgDHGaM**

RETRO GAMES IN AN UBER

magpi.cc/1SYOBsG

We like the idea of this, as sometimes you can have some really awkward conversations with a cabbie. It's not their fault: some people like to socialise. An easy way to find common ground for a chat, though, is a mixture of making and retro gaming, which is why dantheflipman on Reddit was proud to show off the Pi-powered retro gaming system in his own Uber. We like its simplicity – a SNES pad and a screen. That can get you through a lot of very good video games.

DS-SIZED POCKET COMPUTER

imgur.com/gallery/JoP5n

Made using off-the-shelf parts, this little computer acts like some of those other Nintendo DS-sized mini PCs you see on eBay for retro gaming. This one is a bit cooler, though, as it uses a Raspberry Pi. It's made as a fully functioning Linux computer, although it did require a fair bit of hacking (especially removing some bits from the Pi itself). It uses Raspbian to work, and it does fold up like a DS as well.

CROWDFUND THIS!

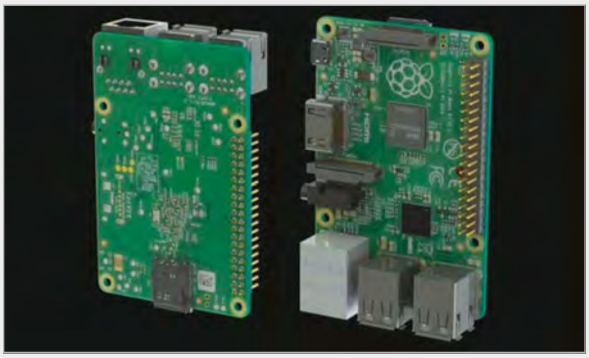
The best crowdfunding hits this month for you to check out…

The FUZEBOX - get more from those costly devices and be sure your kids are learning valuable skills for the future.
pads
FUZE BASIC
tablets
laptops
phones
The FUZEBOX

FUZEBOX

kck.st/22tVJSM

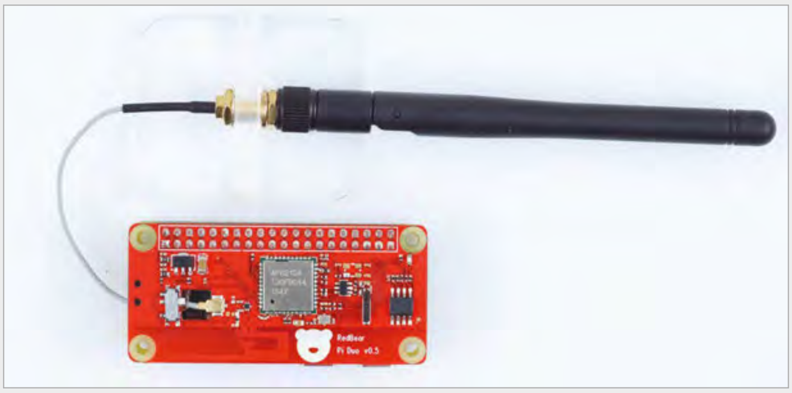
FUZE makes a device very much like the BBC Micro of old, but this one is powered by a Raspberry Pi, and has room for doing all sorts of physical computing and tinkering. Now the FUZE team are making the FUZEBOX, a device that connects (with a wire or wirelessly!) to a Raspberry Pi and other computers, so you can do some coding and physical computing on anything you own. It still needs some money to be funded, though, and could use your help!



RASPBERRY PI AIO SHIELD

magpi.cc/1qYEpKe

A new shield idea for the Raspberry Pi that allows for both analogue-to-digital conversion (ADC) and digital-to-analogue conversion (DAC), as well as 24-bit PWM support. It's not had much interest yet, but if you're willing to take a punt you could come away with an interesting little add-on for the Raspberry Pi that gives it some extra features. Great for making slightly more complex projects, if that's your thing.



IOT HAT FOR RASPBERRY PI

kck.st/1TOQFIt

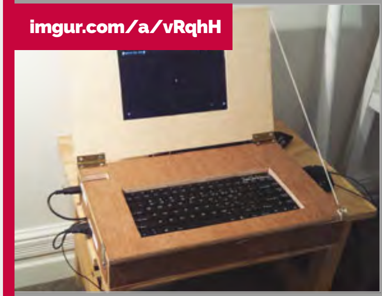
While the Raspberry Pi 3 is pretty amazing, in no small part due to its built-in wireless LAN and Bluetooth, these features are not on the older Raspberry Pi. Sure, a lot of people have been using dongles, but here's a project for a HAT with the same dimensions as the Pi Zero, that will allow for WiFi and Bluetooth on the Zero, and any other Pi that doesn't have it already. It's being marketed as a great thing for IoT projects, and it's already passed its funding goal, so give it a look.

BEST OF THE REST

Here are some other great things we saw this month

HOME-MADE LAPTOP

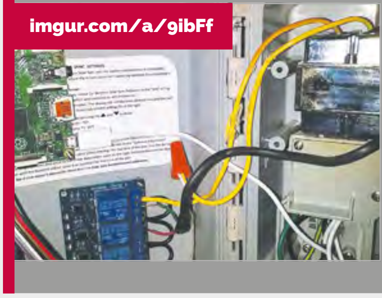
imgur.com/a/vRqhH



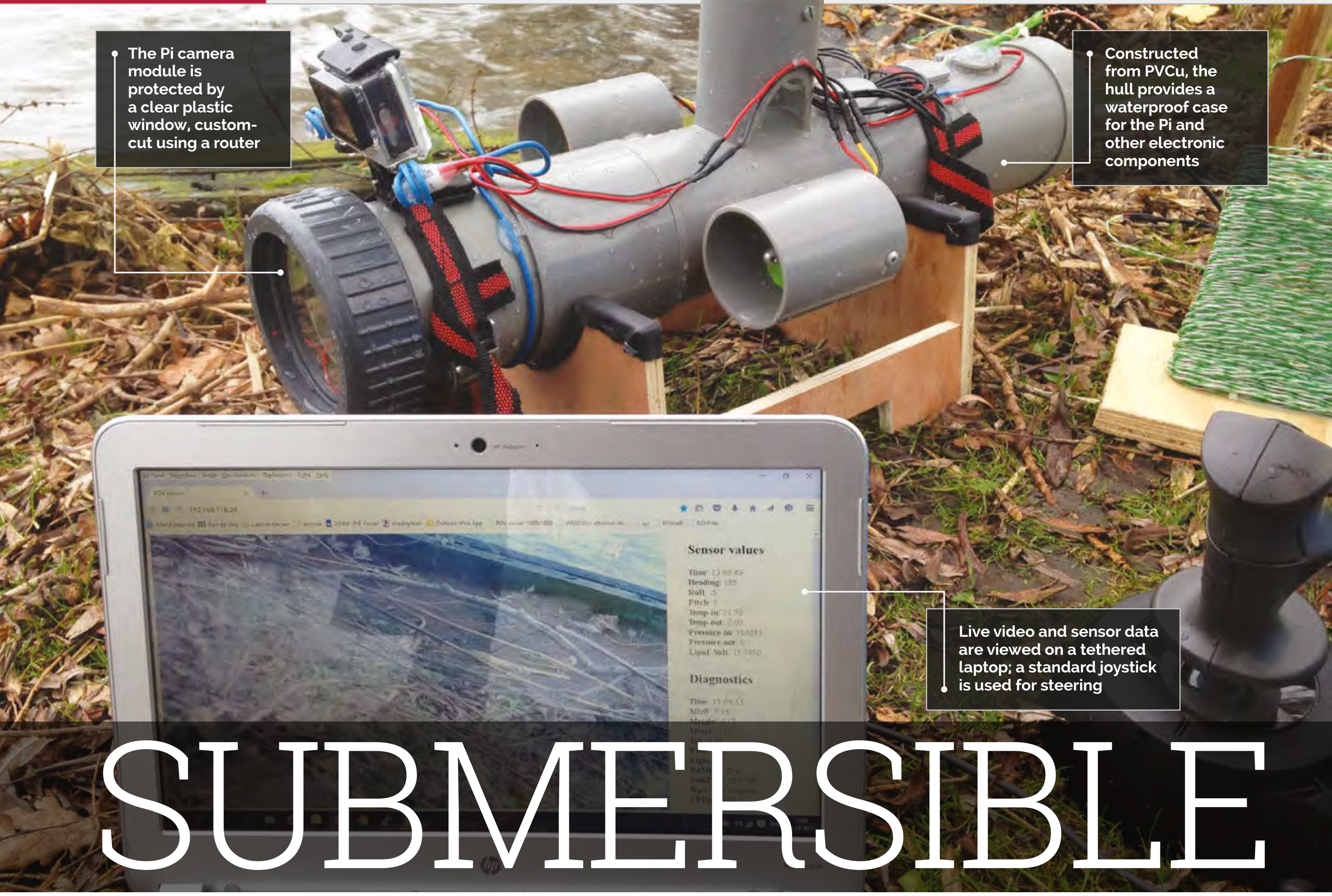
According to user gepset on Reddit, his school instituted a bring-your-own-device (BYOD) policy. Being a poor student, he made this wooden Raspberry Pi laptop from scratch. It folds, has access to all the necessary ports, and features a keyboard. A great little device!

RASPBERRY PI SPRINKLER SYSTEM

imgur.com/a/9ibFF



We think this one is ingenious – Reddit user pyr0t3chnician hacked a sprinkler system with a Raspberry Pi. Their sprinkler system is now part of the Internet of Things, so they can control it remotely and directly anywhere. We're sure the plants will thank them.



The Pi camera module is protected by a clear plastic window, custom-cut using a router

Constructed from PVCu, the hull provides a waterproof case for the Pi and other electronic components

Live video and sensor data are viewed on a tethered laptop; a standard joystick is used for steering

SUBMERSIBLE DRONE

Rather than taking to the skies like most Pi-powered drones, this ingenious ROV dives beneath the surface for some underwater exploration

A keen scuba diver and RC model maker, Niels Affourtit had always been fascinated by the possibility of building a remote-control submarine, but was deterred due to the expense and the potential for leakages. When he saw footage of a VideoRay underwater ROV (remotely operated vehicle) in 2014, however, his enthusiasm was rekindled and he set out to see if it was feasible to build a video ROV of his own (**magpi.cc/20TfFiE**).

"When I started Googling, it was the first time I read about the Raspberry Pi," Niels tells us."When I found out that the Pi can control servos and ESCs (electronic speed controllers) directly via PWM, has the option for a camera, and has a low power consumption, I realised it was a perfect platform for my ROV."

Naturally, a waterproof hull would also be required. Niels constructed his in just 15 hours using PVCu, similar to drainpipe material. Indeed, the closure is a standard connection for pressure piping, while Niels cut out its transparent screen with a router. The use of brushless motors also helped greatly: "Since the motors are running in water, there are no moving parts through the hull and leakage is almost impossible." The only parts that pass through the hull are wires for the motors, lights, echosounder, depth and temperature sensors, and communication with a laptop.

Since high-frequency wireless signals can't penetrate water, while low-frequency ones are too slow to send live video, the ROV is tethered to its laptop controller via a long cable. "From OpenROV (**openrov.com**), I learned to modify a specific Ethernet-over-power adaptor, which reduces the standard four-wire patch cable to only two and makes the wire very insensitive to interference and bad connections. The cable is now 100 metres long, but the maximum reach of the adaptors is in the order of 300m."

3D-printed parts

For the propellers and Pi camera tilt system, Niels drafted designs in AutoCAD and 3D-printed them with an Inno3D printer borrowed from a neighbour. "I have experience in 3D drawing via my work [as an engineer for offshore pipelines], so that is an advantage."

Niels's background in mechanical engineering also came in useful for assembling the electrical components, as did his experience with Python and Linux machines for programming the sensors. Even so, he found the process a real challenge, taking around 200 hours in total. "Every step in the design, you search for solutions and you have to select the best option." It wasn't without its setbacks, either: "I burned quite [a few] parts, too, and some stuff I learned by trial and error. I managed to solder a LiPo [12V battery] connector the wrong way and the connector vaporised when I connected it to another LiPo by accident… No injuries, luckily."

> "Every step in the design, you search for solutions and you have to select the best option"

Tucked safely inside the hull, the Raspberry Pi reads commands sent from the laptop via Ethernet, and controls the ROV's brushless motors via ESCs. A single servo is used to tilt the Pi camera up and down. All the sensors are connected via I^2C, except the depth sensor and MaxBotix echosounder (to determine the height above the seabed). "For these, I attached an Arduino via USB to the Pi."

Plumbing the depths

To take depth readings, Niels's original design featured a standard diving computer suspended in front of the ROV, to be viewed by the camera, but he has since added an inexpensive depth sensor. After testing the ROV in a swimming pool, Niels has taken it out to local lakes for dives of up to 14m deep (**youtu.be/-AUEswQq_vo**), although muddy water can make visibility tricky. "The lakes are dark, so I have powerful LEDs."

So, how deep could the ROV dive? According to Niels, a key limitation is the hull end-caps: "They will deflect too much when it goes deeper than 30m. I did an FE analysis at work to see the strains in the ROV, but I am not sure how it will behave in reality."

Niels's ultimate dream is to use his ROV for exploration at sea, "where I can do a quick survey of shipwrecks before we go down with divers." In the immediate future, he plans to use it to map the depth of lakes, using the echosounder and depth sensor while towing a surface GPS close behind. "I would also like to experiment with an autopilot, to make it into an AUV (automated unmanned vehicle) that can do a search pattern for lost items and come back to the surface to send the data to shore."

PROPELLER POWER

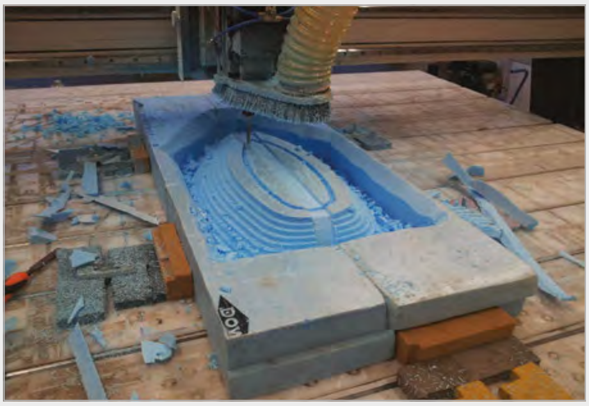
Steering is achieved by increasing power to one of the two forward propellers, while a vertical propeller controls depth. Niels 3D-printed his propellers, but you could buy Graupner M4 2308.45L+R ones instead. The 880KV 35 brushless motors produce plenty of thrust (for a maximum speed of around 10km/h), but their bearings corrode in water. "They are now worn out after ten runs," says Niels. "I could have improved their life if I properly greased them before and after use."

UNDERWATER CAMERA

Protected from the water by a transparent plastic window, the Pi camera module provides a live video feed from the ROV. It can also be tilted up and down in its 3D-printed mechanism connected to a servo. Before the depth sensor was added, Niels would tilt the camera up to read a diving computer suspended from the front of the ROV. To aid visibility in muddy water, two powerful LED 'headlights' are attached to the top of the hull.

A SEA OF ROBOTS

Above Final adjustments and tests, as one boat dreams of going out on the ocean



Above The boats and code were built from scratch, and the whole thing took over a year

Some Portuguese students have created a Pi-powered AI drone swarm that can map and search bodies of water very efficiently

Imagine you're out at sea, probably still near the coast. Possibly even in the harbour. Off to the side is a number of what look like model boats, racing around in a near-random pattern. At least, that's what the human eye sees; in fact, you may be looking at a swarm of AI boats that are searching a specific square of sea, and their pattern makes for the most efficient area coverage that a person couldn't calculate on the fly. This is a demonstration of swarm robotics, a field with not so much real-world research, according to one of the researchers on the team, PhD student Miguel Duarte.

"Right now, the research has only been done in labs and simulations, and our goal is to move that to the real world," Miguel explains to us. "So we have been trying for the past two years to make that happen, and this was the first instance of progress, or at least sizeable progress in that direction.

"Before, we had a lot of experience with robots in the lab [at ISCTE – University Institute of Lisbon] and wanted to make this technology useful for the real world. These are first proof of concepts, basically, just to demonstrate our artificial intelligence algorithms work in the real world, because there's been no demonstrations of these sort of algorithms that involve evolutionary computation and artificial neural networks."

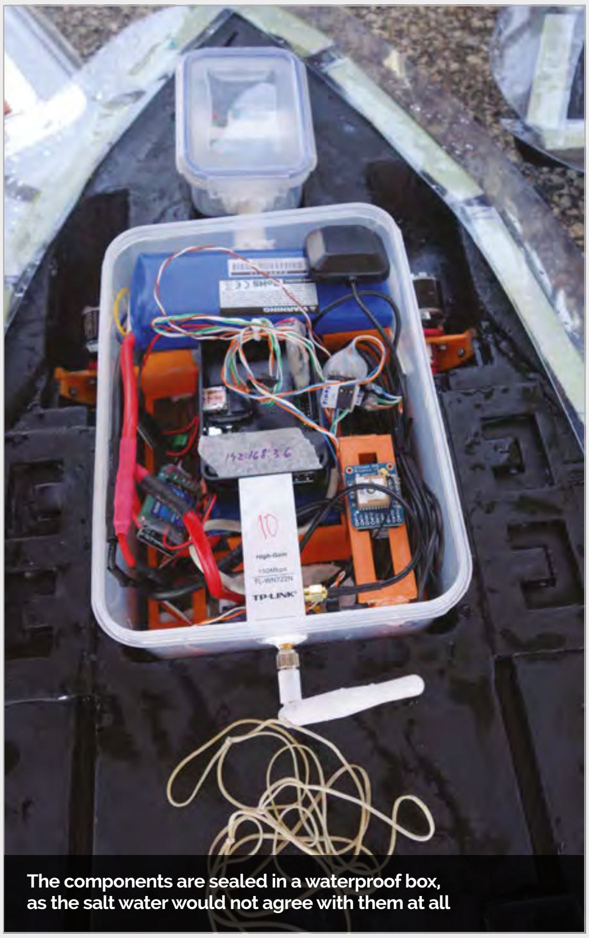
The drones are built from scratch and the controller for them is a Raspberry Pi – a deliberate choice, as Miguel

The boats launch and are then sent instructions on where to start; the AI handles the rest

A VIDEO OF ROBOTS

The title of this article comes from the original video the team made to show off the power of their drones in the Association for the Advancement of Artificial Intelligence (AAAI) 2016 competition. You can see the whole video presentation here: **youtu.be/JBrkszUnms8**

After a hard day's work, the boats return home to be collected and analysed by the team



The components are sealed in a waterproof box, as the salt water would not agree with them at all

tells us:"There was no alternative in our minds: it was the first thing we considered and the first thing we used. For everything we have to do – run the control, read from the sensors, take care of communication, [and] save the logs to the SD card – it can do that very easily, and we don't have to worry about the computing power. We had some experience with the Raspberry Pi before, and the fact we can actually interface with hardware through the IO ports, that was a winning combo… It simplifies our setup a lot."

It's certainly a very impressive project and the code is already being used in other robots, which you can find out about here: **magpi.cc/1VBi8ze**.

Hopefully, these autonomous Pi-powered boats will be able to improve the field of swarm robotics in the way the team want them to.

RASPBERRY JAM EVENT CALENDAR

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

PUT YOUR EVENT ON THE MAP

Want to add your get-together? List it here:

raspberrypi.org/jam/add

1 STAFFORD RASPBERRY JAM WITH CODE CLUB

When: Wednesday 11 May
Where: King Edward VI High School, Stafford, UK
magpi.cc/1Qk0kjF

This event is for Code Club volunteers, teachers, and members of the public interested in helping.

2 OKC RASPBERRY PI MEETUP

When: Saturday 14 May
Where: Edmond Bahá'í Center, Edmond, OK, USA
magpi.cc/1Qk0PKl

With featured topics on home automation and soldering, this is the sixth OKC Raspberry Pi meetup.

3 POTTON PI & PINTS : A RASPBERRY JAM DOUGHNUT

When: Saturday 14 May
Where: The Rising Sun pub, Potton, UK
magpi.cc/1VzzZq7

Fancy a chat with fellow geeks? Then 'Potton Pi & Pints' is for you.

4 SWANSEA JAM

When: Sunday 15 May
Where: TechHub Swansea, Swansea, UK
magpi.cc/1Qk2boy

PiCymru is back with another Raspberry Jam event in Swansea, this time at TechHub's new premises in the heart of the city.

5 TWICKENHAM CODING EVENING MAY

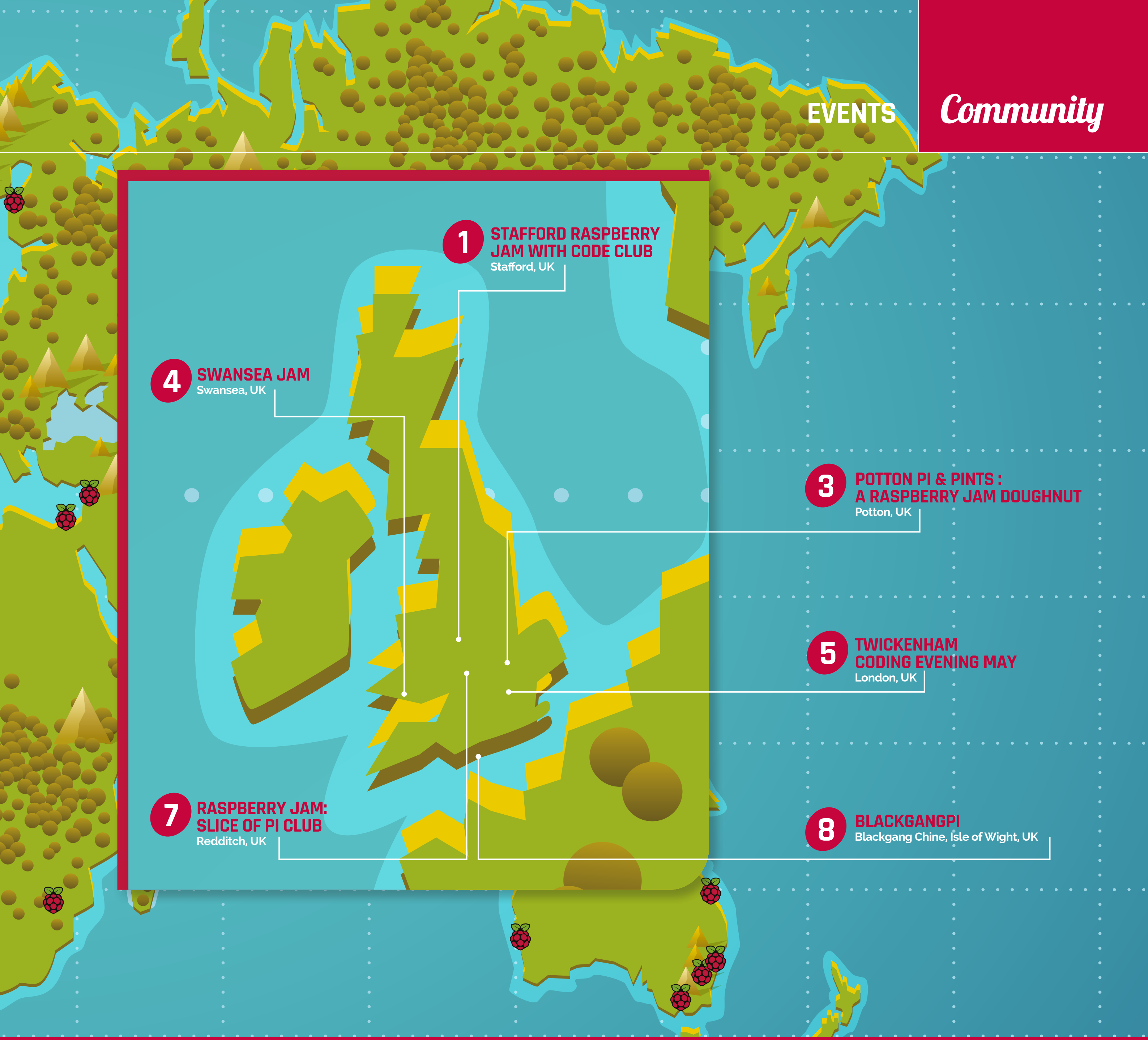
When: Thursday 19 May
Where: Stokes and Moncreiff, London, UK
magpi.cc/1VzBdSi

For those who want to help support the teaching of coding as part of the new computing curriculum.

6 RASPBERRY JAM: B&N HOLMDEL

When: Saturday 21 May
Where: Barnes and Noble Holmdel, Holmdel, NJ
magpi.cc/1qZHeL9

An informal Jam, looking to help beginners and those interested in the Pi to create and share ideas.


1 STAFFORD RASPBERRY JAM WITH CODE CLUB
Stafford, UK
4 SWANSEA JAM
Swansea, UK
3 POTTON PI & PINTS : A RASPBERRY JAM DOUGHNUT
Potton, UK
5 TWICKENHAM CODING EVENING MAY
London, UK
7 RASPBERRY JAM: SLICE OF PI CLUB
Redditch, UK
8 BLACKGANGPI
Blackgang Chine, Isle of Wight, UK

7

RASPBERRY JAM: SLICE OF PI CLUB

When: Tuesday 24 May

Where: Heart of Worcestershire College, Redditch, UK

magpi.cc/1VzBDIh

Slice of Pi Club is set up to bring like-minded people together to learn about the wonderful world of the Pi.

8

BLACKGANGPI

When: Sunday 5 June

Where: Ship Ashore Tea Room, Blackgang Chine, Isle of Wight, UK

magpi.cc/1VzBKnl

Show your projects and discuss with other Pi enthusiasts, and speak with the Blackgang and Robin Hill staff about how they use Pis in the theme park.

DON'T MISS: BLACKGANGPI

When: Sunday 5 June **Where:** Ship Ashore Tea Room, Blackgang Chine, Isle of Wight, UK

This is where the dinosaurs were being hacked from our news piece earlier in the magazine. Blackgang Chine is the UK's oldest theme park, but it keeps itself modern. This time, it's a general Jam where you can see demos and show off what you've done, all while enjoying a (hopefully) sunny day on the Isle of Wight. For more details, check out the event page: **magpi.cc/1VzBKnl**

blueberry®

Add a blueberry to your Raspberry Pi

Turn your Raspberry Pi into a Multi-Purpose IoT Controller Board

The Blueberry IoT Controller for the Raspberry Pi is a versatile device designed to interact with the real world by reading sensors, controlling electrical equipment and using popular communication channels.

It is supported by a range of other devices which can turn your Raspberry Pi into the nerve centre of a simple or complex monitoring and control system.

CONTROL

gates, garage doors, air-conditioning, irrigation systems and much more.

Use the power and connectivity of the Pi or the on-board GSM to control your system from anywhere in the world or receive reports and alerts via email or texts (SMS).

Rapid electronics

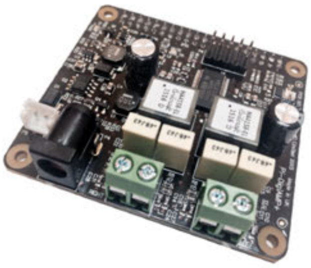
www.rapidonline.com

Rapid Electronics part of the CONRAD Group

Made in the UK

IQaudIO

Audiophile accessories for the Raspberry Pi



Pi-DAC+

- Raspberry Pi HAT, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments PCM5122
- Variable output to 2.1v RMS
- Headphone Amplifier / 3.5mm socket
- Out-of-the-box Raspbian support
- Integrated hardware volume control
- Access to Raspberry Pi GPIO
- Connect to your own Hi-Fi's line-in/aux
- Industry standard Phono (RCA) sockets
- Supports the Pi-AMP+

Pi-AMP+

- Pi-DAC+ accessory, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments TPA3118
- Up to 2x35w of stereo amplification
- Provides power to the Raspberry Pi
- Software mute on GPIO22
- Auto-Mute when using Pi-DAC+ headphones
- Input voltage 12-19v
- Supports speakers from 4-8ohm

Pi-DigiAMP+

- Raspberry Pi HAT, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments TAS5756M
- Up to 2x35w of stereo amplification
- Out-of-the-box Raspbian support
- Integrated hardware volume control
- Provides power to the Raspberry Pi
- Software mute on GPIO22
- I/O (i2c, 3v, 5v, 0v, GPIO22/23/24/25)
- Just add speakers for a complete Hi-Fi
- Input voltage 12-19v
- Supports speakers from 4-8ohm

M2P MAX2PLAY · rune audio · VOLUMIO · moOde audio player · PiMusicBox · openelec · piCorePlayer SQUEEZE AUDIO · Archphile

Twitter: @IQ_audio
Email: info@iqaudio.com

WWW.IQAUDIO.COM

IQaudio Limited,
Swindon, Wiltshire.
Company No.: 9461908

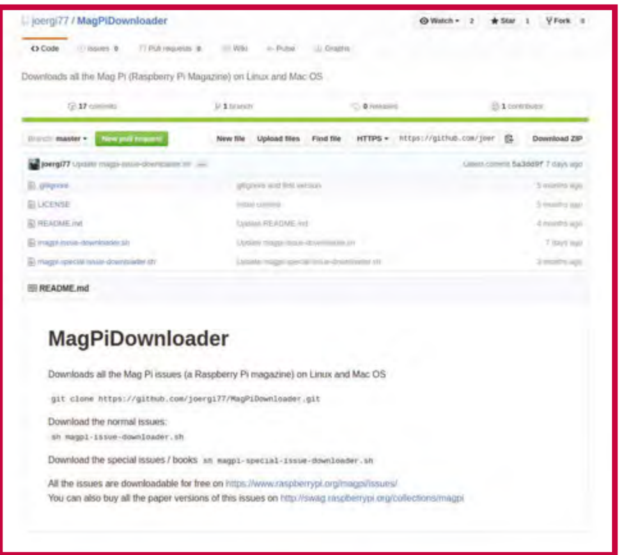
YOUR LETTERS

PDFs everywhere

I’ve been reading *The MagPi* PDFs since the magazine started, but a while ago I lost some files and my small collection of PDFs went with them. I didn’t bother re-downloading them all at first, as it seemed like it would be a big job and… well, it’s been a few years since then, and my problem has only grown. Do you have a way to download all the PDFs at once? Some sort of ZIP file on the website? I can probably figure out a little script otherwise, but if you do, it would make it a bit faster.
Ellie Denton

Thanks for your enquiry, Ellie. Unforunately, we don’t have a ZIP file or anything like that right now for *The MagPi*, although the path names of all the PDFs are kept fairly uniform to make it easier. So you could easily create a script to download them all.

Which is what one of our readers already thought of! They’ve put together a little GitHub repo with a script that allows you to download every issue of *The MagPi* in PDF format, even our Essentials and Projects books. It’s updated regularly and you can find it here for all your PDF needs: **magpi.cc/1VAqE1s**


joergi77 / MagPiDownloader

Downloads all the Mag Pi (Raspberry Pi Magazine) on Linux and Mac OS

https://github.com/joer

joergi77 Update magpi-issue-downloader.sh		Latest commit 5a3dd9f 7 days ago
.gitignore	gitignore and first version	5 months ago
LICENSE	Initial commit	5 months ago
README.md	Update README.md	4 months ago
magpi-issue-downloader.sh	Update magpi-issue-downloader.sh	7 days ago
magpi-special-issue-downloader.sh	Update magpi-special-issue-downloader.sh	3 months ago

README.md

MagPiDownloader

Downloads all the Mag Pi issues (a Raspberry Pi magazine) on Linux and Mac OS

```
git clone https://github.com/joergi77/MagPiDownloader.git
```

Download the normal issues:

```
sh magpi-issue-downloader.sh
```

Download the special issues / books `sh magpi-special-issue-downloader.sh`

All the issues are downloadable for free on https://www.raspberrypi.org/magpi/issues/
You can also buy all the paper versions of this issues on http://swag.raspberrypi.org/collections/magpi

Above This little script is excellent and can help you complete your *MagPi* collection very quickly

More Pi upgrades

A new Raspberry Pi, a new Compute Module, even a new case for the Raspberry Pi! I also hear there’s a newer version of the Pi Zero coming out. Are any of the other components expected to be updated this year? Is everything going to get a new version? Should I expect a Sense HAT 2 or a Camera 2 by the end of the year? Honestly, I love it: keeping it the same price, and having it work just the way things did before, makes it seem like you’re not missing out whether you upgrade or not!
Sam Z

Well, as you might have gathered from this issue of the magazine, the Pi Camera 2 did happen! Well done on guessing that. We don’t think loads of refreshes will be coming to some of the more ancillary Pi hardware; some of it is quite new, such as the Sense HAT you mentioned, and others wouldn’t really benefit from updates at the moment. This is probably just going to be a constant gradual process as tech comes and goes. Who knows where we’ll be in another four years!

Raspberry Pi 4?

I was quite amazed when the Raspberry Pi 3 was announced at the end of February, especially only a year after the Raspberry Pi 2 came out! I was actually looking to get another Raspberry Pi for some projects I had planned and the Pi 3 came along at just the right time for me, although the Pi 2 is still pretty great.

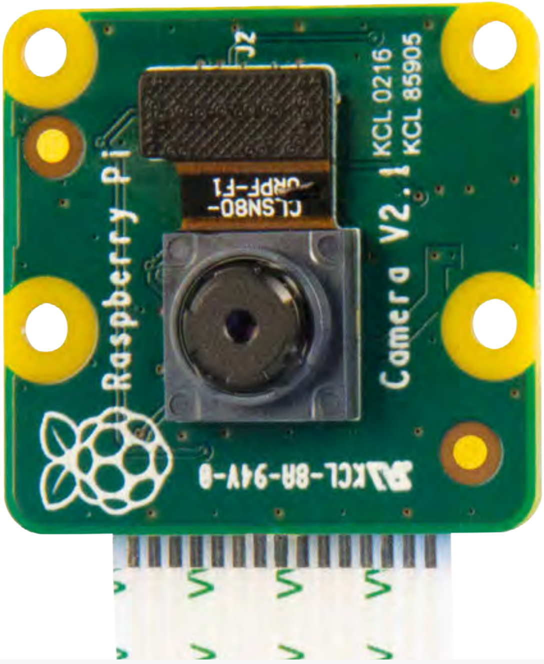
I’m wondering, though, does that mean we should expect a yearly Raspberry Pi upgrade? I mean, I’d probably get them either way, but it’s nice to know if that’s sort of the plan going forward. Thanks!
Edward Irons

The new Pi 3 took us a bit by surprise as well when we heard about it, so we understand your concerns. Eben has made it clear, though, that the Pi won’t be released on a yearly cycle; on one hand it makes sure Raspberry Pi doesn’t get predictable, and on the other it ensures some of the other Pi products get an upgrade. We’ve already seen the Compute Module get a jump straight to the new chip, so hopefully we’ll get a new Model A as well, with a bit more oomph.

Digital downloads

I've really liked reading your magazine over the past few months, so I finally decided to make the commitment and go for a full print subscription to the magazine. I do have a question, though: now that I have this print subscription, can I get access to the digital subscription as well? Hope you can help!
Robin S

Thanks for subscribing – we really appreciate it! Unfortunately, we don't have a system set up for people to get a digital subscription code with their print subscription, so we can't help you there. We do have an alternative, though – all issues of *The MagPi* are free to download from our website as a PDF. Get a PDF reader on your phone or tablet, and you can have them on the go as well. They don't take up much space either!



Left Yep, there's a new camera. If you've managed to skip the whole cover feature to read this, we applaud you, but also ask you go back and check it out.

FROM THE FORUM: EXPERTISE LEVEL

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

I just bought my first Raspberry Pi and I have zero programming experience. I like the idea of subscribing to *The MagPi* for the projects and tutorials from more experienced users, but I'm afraid it may be too advanced for me.

What kind of skill level is *The MagPi* aimed at? Is it appropriate for the inexperienced, or should I wait until I've advanced my knowledge and skills?
FlightMedic

While we definitely have a wide range of projects and tutorials in the magazine, we like to make them as understandable as possible. Some projects will definitely be for more advanced Pi people, but we also have a lot that assume very little to no knowledge. For example, issue 36 has our 'getting started' guide. We also have the Essentials series of books that cover a topic from beginning to end, so you can learn it as you go. Best of all, they're all free as PDFs, so you can give them a read and see if they're to your liking before grabbing a digital or physical subscription. You can learn all about the subs here: **magpi.cc/Subs1**

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**

READ IT ANYWHERE

GET A CAMERA MODULE FIRST WITH YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
PICTURE PERFECT
The NEW Raspberry Pi Camera Module is here!
YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
POWER UP YOUR LIFE WITH RASPBERRY PI
CHAPTER ONE
WHAT IS THE SENSE HAT?

SAVE 25% with a Newsstand subscription (limited time offer)

DO SCIENCE WITH THE SENSE HAT

WITH OUR **NEW ESSENTIALS E-BOOK**

AVAILABLE ON THE MAGPI APP!

ONLY £2.99 $3.99

FREE: DOWNLOAD ALL 30 ORIGINAL ISSUES

The MagPi Magazine

Available now

for smartphones & tablets

Available on the App Store

GET IT ON Google Play

Subscribe from

£2.29 or **£26.99**

rolling subscription

full year subscription

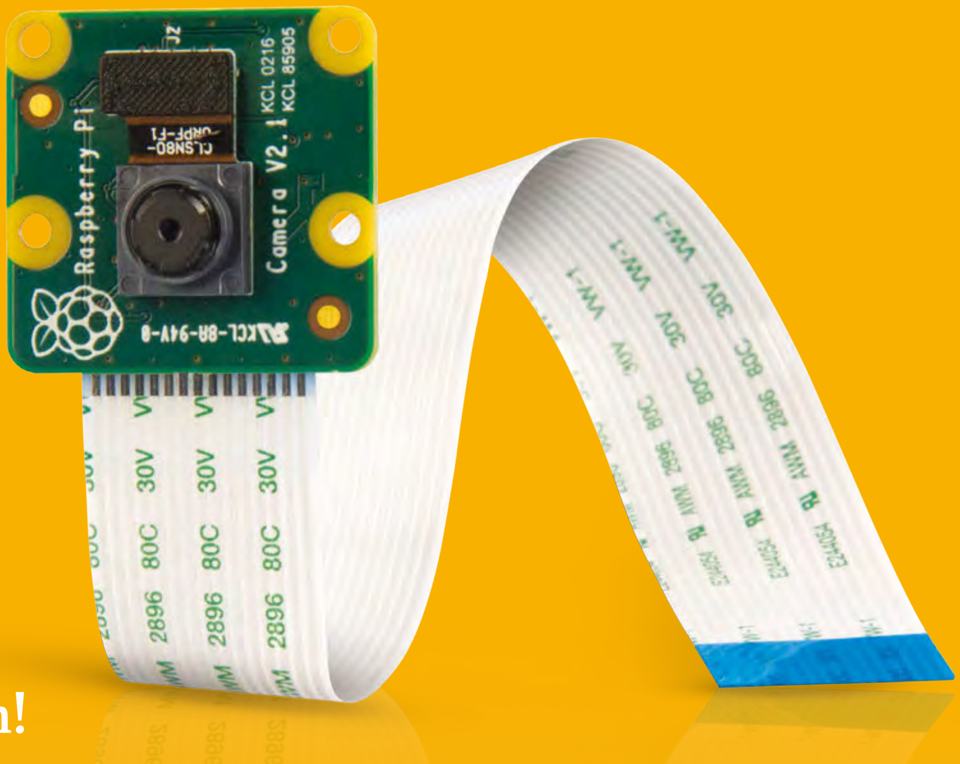
Download it today – it's free!

- Get all 30 legacy issues free
- Instant downloads every month
- Fast rendering performance
- Live links & interactivity

In association with The PiHut
www.ThePiHut.com

10 CAMERA MODULES MUST BE WON!

WHAT'S YOUR FAVOURITE RASPBERRY PI CAMERA PROJECT?



Tell us by 23 May for your chance to win!

Simply email **competition@raspberrypi.org** with details of your favourite project using the Pi Camera Module.

Competition closes 23 May 2016. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don't like spam. Participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

MEETING FACE TO FACE

For **Matt Richardson**, seeing people in person is the best part of the job

As I write this column, I'm on my way to Washington, DC for the USA Science and Engineering Festival, which is a three-day STEM-focused event for all ages. It happens every other year, and the organisers estimate that around 300,000 people course through the Walter E Washington Convention Center to get hands-on with new technology or learn a little bit of science. There, with a few other members of the Raspberry Pi team, we're going to share our passion for computing and digital making with as many of those people as we can. Events like this one are an important part of getting the word out about our mission and our computer.

Even though we're constantly engaging with the Raspberry Pi community through our blog, the forums, on Twitter, and with *The MagPi*, there's something special about seeing people face-to-face. Whether we're chatting with a long-time Raspberry Pi user or are introducing someone to Raspberry Pi for the first time, it's incredibly gratifying to speak to people about it in person. I feel so fortunate that I get to hear about so many projects, stories, and ideas. I especially love when I encounter someone who hasn't heard of Raspberry Pi before. Their amazed reaction to our $35 computer is, well, priceless.

At many events, we offer a hands-on activity so that people can try Raspberry Pi for the first time, or at least try something new with Raspberry Pi. We've had plenty of people come to our booth and write their first line of code, ever. It's so much fun to see people's reaction when they get their own 'hello world' message lighting up the LED matrix on the Sense HAT.

At the SX Create event in Austin a few weeks ago, our Raspberry Pi stations were often packed with multiple kids at each setup. Parents would sometimes have trouble getting their children to step away from the Pi and see other activities at the event. For us, not only does it affirm that learning about computing can be fun and engaging, but it's also a chance to observe the way young people interact with a Raspberry Pi.

Varied audiences

Every year, we try to reach lots of different audiences. At Maker Faires, we talk to families and hobbyists. At the BETT conference, we share our educational goals with British teachers. This summer, we'll be at ALA for the first time, where we're introducing Raspberry Pi to American librarians at their national conference.

Wherever we are, I encourage you to come out and see us if you can. Keep an eye on our events page (**raspberrypi.org/education/events**) or on our Twitter account for updates on where we'll be. And, as a small incentive, we usually keep a stash of Raspberry Pi stickers to give away to people who stop by to say hello.

Although we try to go to as many events as we possibly can, we can't be everywhere, but that doesn't mean you're out of luck. If you want to meet IRL with your fellow Raspberry Pi community members, check for Raspberry Jams in your area (**raspberrypi.org/jam**). If you don't see one, then that's a perfect excuse to start your own. Raspberry Jams don't have to be big affairs. In fact, they usually start off with just a few people.

As much as we absolutely love the emails, tweets, and forum posts from our community, nothing can beat a handshake, a smile, and a conversation. Eventually, we hope to meet every single one of you!

EXPERIMENT WITH THE SENSE HAT

Get started today for **just £2.99 / $3.99**

SENSE THE REAL WORLD WITH YOUR Raspberry Pi

The MagPi ESSENTIALS

From the makers of the official Raspberry Pi magazine

Available on the App Store

GET IT ON Google Play

Find it on The MagPi digital app

| EXPERIMENT WITH THE SENSE HAT |

[CHAPTER ONE]

WHAT IS THE SENSE HAT?

The special add-on to the Raspberry Pi that lets it interact more with the world around, as seen on the ISS!

The Raspberry Pi can do a lot of things thanks to its size, portability, and ability to connect to the internet easily. With the GPIO ports you can control electronics and interact with the world. One of the best ways of doing this is by using an add-on like the Sense HAT.

The Sense HAT is a very sophisticated add-on board for the Raspberry Pi. While HAT is an acronym (Hardware Attached on-Top), it does act in a way like a hat for your Raspberry Pi. The Sense HAT contains a suite of sensors that allows the Raspberry Pi to sense the world around it, along with an array of LEDs on top which can be used to display information on what the board can sense, and a little joystick.

The Sense HAT is a vital component of the Astro Pi, the specially adapted educational Raspberry Pis which were sent up to the International Space Station with British ESA astronaut Tim Peake to run code created by children. This wasn't what the HAT was originally designed for, though, as the Sense HAT's Project Lead Jonathan Bell explains:

"I sort of hijacked a pet project of James [Adams's] and turned it into a space-faring board," says Jonathan. James Adams is the Director of Hardware at Raspberry Pi, and along with Jonathan, was one of the main driving forces behind the Sense HAT.

magpi.cc/Sense-HAT-book