YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

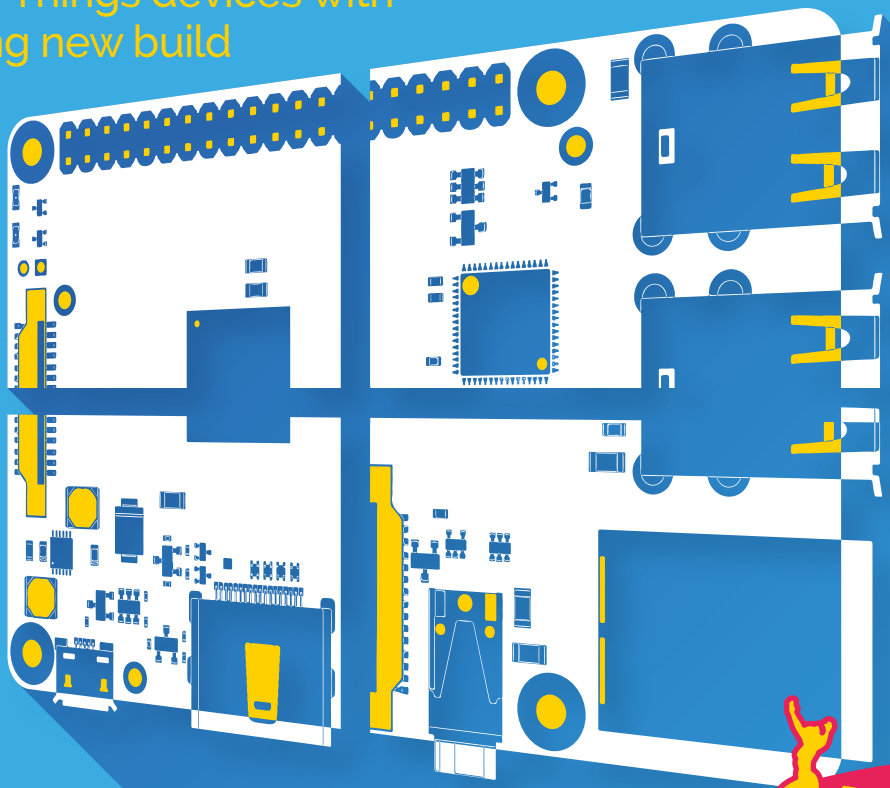
The official Raspberry Pi magazine

Issue 48 August 2016

raspberrypi.org/magpi

WINDOWS 10 IoT CORE ON RASPBERRY PI

Create Internet of Things devices with Microsoft's exciting new build



SET UP DROPBOX ON YOUR PI

Share your files in the cloud in easy steps

INTERNET-POWERED ROBOTS!

Battling bots controlled by Twitch.tv viewers

MAKE MUSIC WITH YOUR PI

Sonic Pi creator Sam Aaron shows you how

BUILD A TWEET-O-METER

Chart your social media success

GO FOR GOLD!

Four amazing Olympics-inspired games to make on your Raspberry pi

Also inside:

- BUILD THE CONTROLS FOR AN ARCADE MACHINE
- NATUREBYTES WILDLIFE CAMERA KIT TESTED
- NASA HELPS SEND CODE CLUB TO SPACE
- CODE IN C WITH OUR EXPERT SERIES

Issue 48 • Aug 2016 • £5.99

9 772051 998001 08

THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMUNITY

strato

Strato Pi Touch Display



Raspberry Pi 3
Raspberry Pi Touch Display
Strato Pi board

Power Supply 12-28VDC
Real Time Clock
Buzzer
Optional RS232/RS485 & UPS

Aluminium and steel rear chassis
Brushed aluminium front bezel
Optional back box for wall mount

SFERALABS

www.sferalabs.cc

WELCOME TO THE OFFICIAL PI MAGAZINE!

Windows is synonymous with computers. After decades of it being the most popular operating system in the home, it's been interesting to see Microsoft experimenting with expanding the uses of Windows. Recently, the firm got behind the maker movement, which led to Windows 10 being ported to the Raspberry Pi last year, in the form of Windows 10 IoT Core. There's been a great reaction from the maker community, and it's currently included on NOOBS for everyone to use.

Now with the Raspberry Pi 3 out, Windows 10 IoT Core has much more power to play with and the team at Microsoft are keen to show off what it can do. They've even made a new kit full of great project parts so you can make the most out of Windows 10 on Raspberry Pi. To celebrate this, we've got a whole feature on Windows 10 IoT Core – starting on page 14 – that shows you what's in the new kit, some of the amazing things you can do with IoT Core on Raspberry Pi, and some quick lessons to get started making with Windows.

We hope you enjoy the issue!

Rob Zwetsloot
Features Editor



THIS MONTH:

14 WINDOWS 10 IOT CORE
Create Raspberry Pi IoT devices with Microsoft's OS

36 MOTORISED SKATEBOARD
Ride to school/work on this Pi-powered electric board

44 RGB LED TWEET-O-METER
See how your Tweets are doing with this smart project

68 SCRATCH OLYMPICS 2016
Go for gold with these fun sporting simulations

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

The MagPi

Available on the App Store

GET IT ON Google Play

EDITORIAL
Managing Editor: **Russell Barnes** **russell@raspberrypi.org**
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**

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**

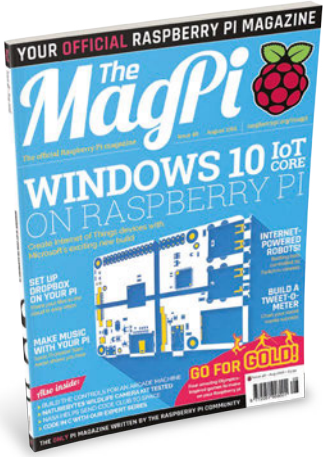
SUBSCRIPTIONS
Select Publisher Services Ltd
PO Box 6337
Bournemouth
BH1 9EH | **+44 (0)1202 586 848**

CONTRIBUTORS
Sam Aaron, Wes Archer, Alex Bate, Mike Cook, David Crookes, Lucy Hattersley, Richard Hayler, Phil King, Simon Long, Ben Nuttall, Matt Richardson & Richard Smedley

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

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

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.



Contents

Issue 48 August 2016 raspberrypi.org/magpi

TUTORIALS

> TWEET-O-METER **44**
Use an RGB LED to show the popularity of tweets

> INTRODUCING C **46**
Create variables and perform mathematical operations

> PI BAKERY **48**
Make an Olympic swimming simulator with Mike Cook

> DROPBOX ON PI **54**
Upload and download files from the cloud service

> SONIC PI **56**
Learn how to master its powerful slicing capabilities

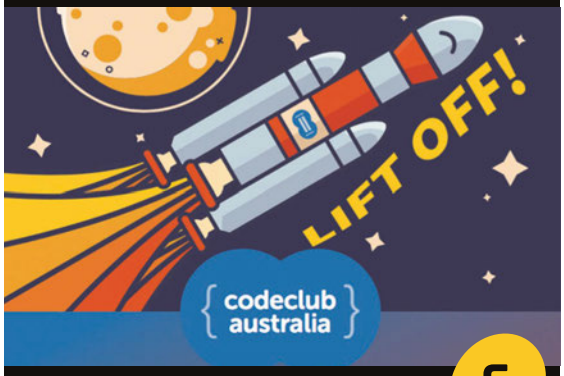
> SCRATCH BOAT RACE GAME **58**
Make a fun boat race game in Scratch on the Pi

> POETRY GENERATOR **60**
Use word lists in Scratch to generate random poems

> ARCADE MACHINE BUILD (PT 2) **62**
Create the controls for your RaspCade arcade machine

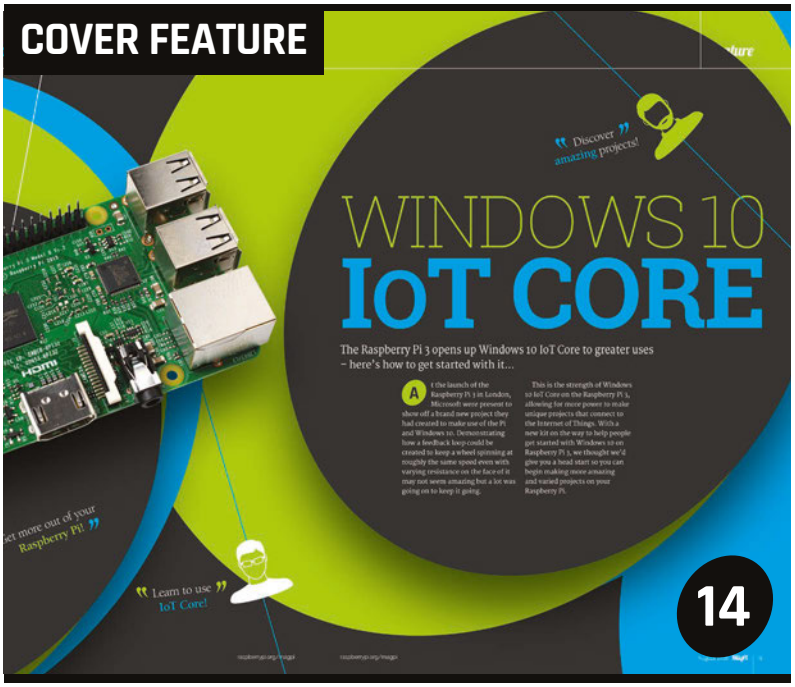
IN THE NEWS

One Giant Leap for Coding



Young coders in Australia were asked to aim for the moon in a world record attempt 6

COVER FEATURE



WINDOWS 10 IOT CORE

Learn to use IoT Core, make projects, and get more out of your Raspberry Pi

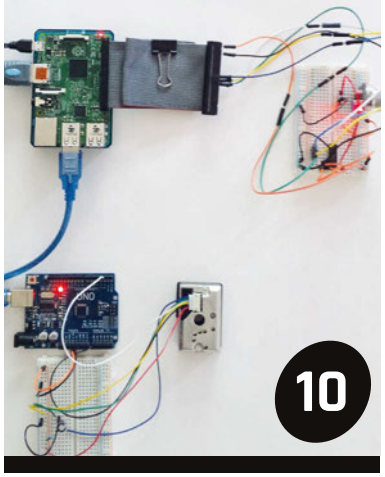
LET'S ROBOT 8



See what happens when a bunch of robots are controlled by web chat room users

Asthma Pi

Nine-year-old Arnav Sharma has built an award-winning air monitoring kit for asthma sufferers



Contents

THE BIG FEATURE

68



SCRATCH OLYMPICS 2016

Go for gold by programming four fun Olympic events in Scratch: archery, weightlifting, hurdles, and synchronised swimming

4TH BIRTHDAY BONANZA!

A treasure trove of Pimoroni HATs and Goodies MUST BE WON!

PIMORONI

95

MAKE YOUR OWN JAM

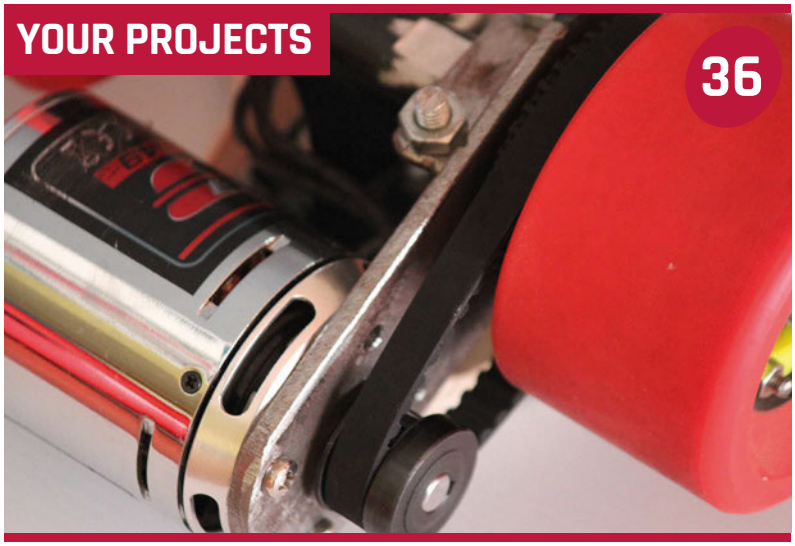
Practical tips for setting up your own Raspberry Jam from educator Cat Lamin



88

YOUR PROJECTS

36



PI SKATEBOARD

Forget hoverboards – commute in style with this Pi-powered electric deck

Discoverer 34

Search for treasure in the back yard with this metal-detecting robot



Internet of LEGO 38

Take a tour of this incredible internet-connected cityscape built from bricks



HAL 9000 40

This homage to 2001: A Space Odyssey makes for a stylish voice assistant



REGULARS

> NEWS 06
The biggest stories from the world of Raspberry Pi

> TECHNICAL FAQ 64
Got a problem? Maybe we can help...

> BOOK REVIEWS 84
The best reads for coders and hackers

> THE FINAL WORD 96
More musings from Raspberry Pi's Matt Richardson

COMMUNITY

> THIS MONTH IN PI 86
We round up other goings-on in the Pi community

> EVENTS 90
Get involved in a community gathering near you

> YOUR LETTERS 92
Talk to us about the magazine or Raspberry Pi

REVIEWS

> WILDLIFE CAM KIT 80

> ANALOG ZERO 82

> PICO-8 83

ONE GIANT LEAP FOR CODING

Young coders in Australia were asked to aim for the moon in a world record attempt

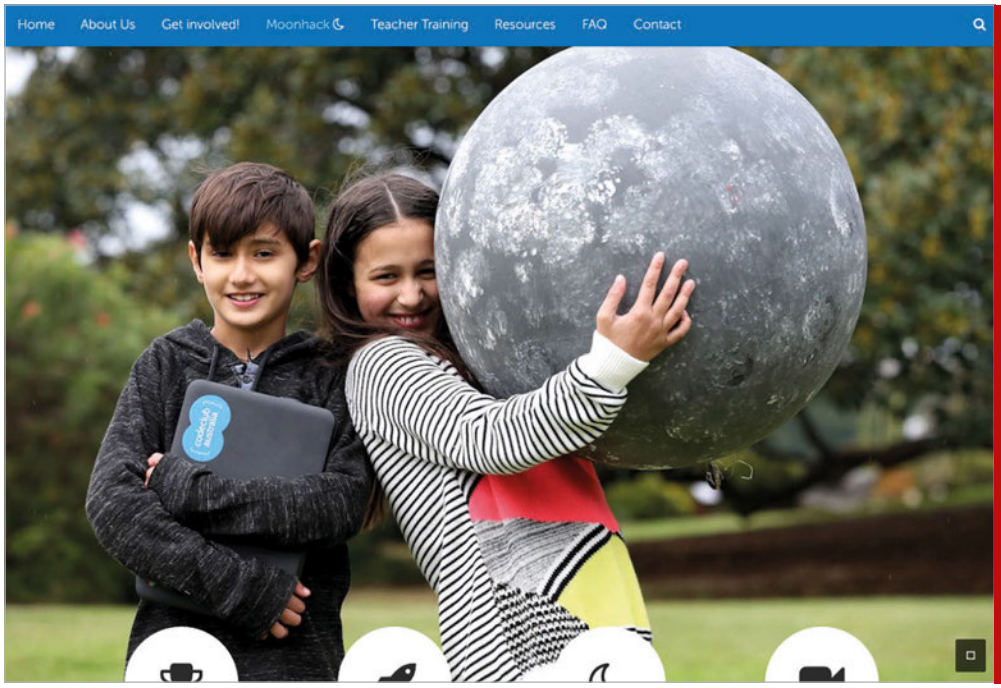
Regardless of the pursuit, there are few more exciting things in life than an attempt at a world record. Whether you're trying to grow an extra centimetre of fingernail in order to beat Shridar Chillai (how he types is anyone's guess), or whether you're looking to break the most toilet seats using only your head (46, as it stands), the will to be the best can take some dedication.

Code Club Australia (CCA) knows that more than most. On 20 July, the club's organisers looked to encourage 10,000 kids to take part in Moonhack, an online programming workshop that asked children to choose a moon-themed coding activity using Python, Scratch, or a combination of HTML and CSS. In doing so, it sought to set a world record for the largest number of children coding at the same time.

Much to the delight of CCA general manager Kelly Tagalan, children rose to the challenge, just as Annie Parker did in 2014 when she set up Code Club Australia for the benefit of 9 to 11-year-olds and watched its presence extend to 1,000 schools. "We had over 1,000 kids straight after the announcement," says Tagalan, who came up with the idea for setting a world record as a way of achieving something big with a simple concept. "Principals, teachers, and parents demonstrated a real zeal for what we were hoping to achieve."

An important anniversary

But it wasn't easy. Working with CCA board member Clive Dickens, the first task was to connect the idea to a purpose. "We discovered that the anniversary of the Apollo 11 moon landing was 20 July," Tagalan recalls. She added that it was a perfect tie-in: due to the position of the earth when Neil Armstrong walked on the lunar surface, the Parkes Dish in New South Wales received the first images of the US astronaut and put them out for the world. "It was live-streaming long before YouTube, and [with



Below Children aimed for the moon in their bid to set a world coding record

WALKING TALL



It was rather apt for Moonhack to take place in Australia on the anniversary of the Apollo 11 moon landing. "It was through the antenna dish now at this complex [in Canberra] that the historic images of Neil Armstrong stepping onto the surface of the moon were received and relayed to the entire world," Nagle explains.



Above The Code Club Australia website - **codeclubau.org** – had a dedicated Moonhack section



Right Posters could be printed and distributed, to drum up greater numbers of children

Moonhack] we were hoping to make a connection for Australian kids to a very important moment in world history that had a significant Australian contribution," she says.

It was a wise choice. The space-based theme and CCA's giant leap of ambition (10,000 children represents a quarter of so of the 45,000 Australian kids engaged in Code Clubs) soon brought high-profile backing, not least from the Canberra Deep Space Communication Complex (CDSCC), which maintains one of three stations in the Deep Space Network on behalf of NASA.

Glen Nagle, the NASA operations support officer for the CDSCC, contacted the organisers to discuss ways of supporting the activity. "When Glen reached out to us, we were – pardon the pun – over the moon," Tagalan says. "We were really grateful that the announcement had reached that far, and pleased to have had his expertise on the history of the moon landing. It's a wonderful thing to have such a great organisation stand behind getting kids ready for fantastic futures through education."

Aim for the moon

For the CDSCC, Moonhack presented a golden opportunity. Its deep space tracking station has its own education program which encourages more than 10,000 students each year to pursue the STEM subjects, and seeks to promote science and technology education to the next generation of space explorers. "The Moonhack program certainly fitted into our theme of promoting space science to schoolchildren, and showing how they don't have to wait years to get a degree to become a scientist or engineer," Nagle explains.

As such, the CDSCC worked with Moonhack to make more teachers aware of the event and encourage them to introduce it into their schools. "We included messages about Moonhack in our teachers' kits and spread the message through our social media channels," Nagle continues. To encourage them further, some of the best students from Moonhack are being invited to take a look behind the scenes at Australia's role in space exploration at the CDSCC. "It was really wonderful to have the invitation to visit the NASA CDSCC centre extended to some Code Club kids," says Tagalan.

It's certainly been a good year for tying coding to all things space. Astro Pi, which was taken on board the International Space Station and used by British astronaut Tim Peake, attracted lots of attention. "Having students being able to run experiments on the ISS is a brilliant way to engage and excite them," Nagle comments. "Australia is participating in these sorts of activities through the Quberider programme, which will see students coding small instrumentation packages to perform experiments remotely on ISS."

Such activity, he says, allows for a fresh approach to the promotion of coding. "Allowing students to do more than just the theory of science, and actually having them build and operate instruments of their own design in space, can only capture their imaginations and perhaps steer them on a path to future careers in this area. The only limit to what these children can do is their imagination."

RINGING THE CHANGES



Last December, Code Club Australia benefited from a cash injection of a million Australian dollars from the government and the Telstra Foundation, the philanthropic arm of the telecommunications giant. The Code Club and Telstra are intrinsically linked, though, since its founder Annie Parker worked for the Foundation's startup incubator programme, muru-D. Telstra chairman Catherine Livingstone (pictured) backs the club, saying: "So many of tomorrow's jobs are going to require some level of knowledge about computer coding."

Watch Let's Robot at letsrobot.tv!

TWITCH-CONTROLLED ROBOTS

What happens when you take the concept of Twitch Plays Pokémon and transpose it to stream-controlled robots doing arts and crafts?

Twitch has long since stopped just being a place to watch people play games. Twitch Plays Pokémon had the audience play Pokémon games. Bob Ross spent eleven days straight painting happy little trees to launch the creative channel. Now, you can control robots live via the chat with Let's Robot.

"Let's Robot began as an attempt to make the world's first live-interactive online show, by using robots," Jillian Ogle, founder and CEO of Let's Robot, tells us. "At the time I started this, I was an indie game developer and used to work for big companies like Disney.

"The way it works is the robot streams a live video feed to a web page (**letsrobot.tv**), where the audience can see the video and use a chat room to talk to each other and send the robot commands. The robot parses the commands and then passes the instructions to the microcontroller on board. We use a lot of Arduino-type controllers for the actual control of the robot."

The robots have got up to a lot of mischief in their time on Twitch. There have been story-based adventures through cardboard dungeons, harrowing pizza deliveries, artistic competitions, and even the robots feeding a man random items of food set out on a table for the robots to grab.

CONTROLLING A ROBOT ON TWITCH

Like a lot of Twitch-controlled projects, simply writing keywords in the chat causes the robot to perform various actions. The most basic commands are for movement:

forward, back, left, right

However, you can also control the LEDs on a robot using commands such as:

LED All (Colour); for example, LED All Blue

LED (Number) (Colour); for example, LED 7 Blue

You can also give them RGB numbers for colour. We understand that Easter eggs have been programmed into individual robots as well, so look for more interesting ways to control them.



The outdoor robot is much more robust than the others, with a few Mars Rover-like features to aid it



These robots are incredible artists, putting the likes of Jackson Pollock to shame

Mix in the Raspberry Pi

As well as Arduinos, the robots use a lot of Raspberry Pis, as Jillian explains to us:

"The Raspberry Pi is the most consistent component we have in all of our robots. It's used to capture and upload video in real-time using the camera. It also acts as a messenger between our web server and microcontrollers.

"We've actually tested a lot of boards besides the Pi that are much more powerful; however, none of them really have the supporting tools, hardware, or online support and community that makes the Raspberry Pi so easy to work with. We also like the Pi a lot, because we would actually love for people to be able to make robots like ours and join in the fun."

The robots have a few things in common; they all have cameras, wheels, and lights as well as the Twitch control aspect. They're mostly small so it's easier to make stuff for them, some have grippers and there's even one with a sword. For outdoor incursions, there's also a Mars Rover-esque machine. There's much more to come, though, with new robots being built all the time and future plans in place for robots after that.

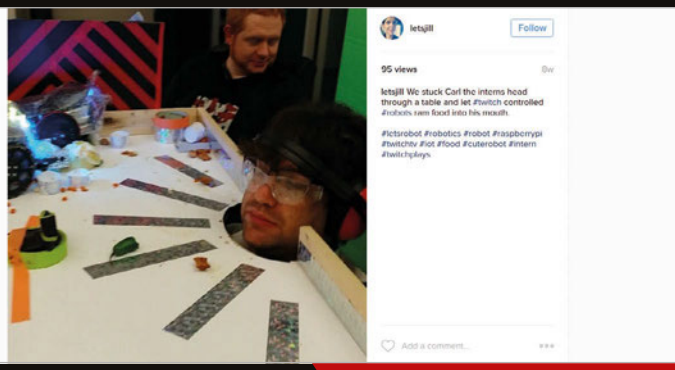
"We have many plans," Jillian mentions. "We're working to make a custom interface that's actually designed for many users to collaborate or compete for control of a single robot. We're trying to make the audience participation more fun and engaging, and working on ways to lower the latency."

It's all very exciting, and the cake baking challenge for the robots is coming up soon as well. We're sure it will be a masterpiece.

> "The robots have a few things in common; they all have cameras, wheels, and lights"

HIGHLIGHT REEL

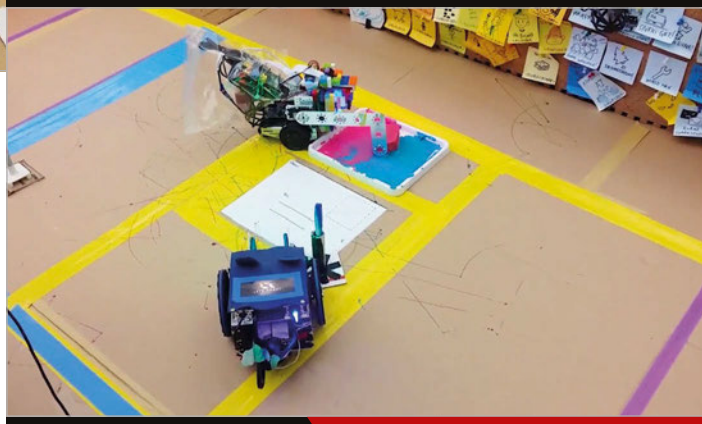
Catch up on some of the robot's best adventures so far...



Feed the human

magpi.cc/2a33DUG

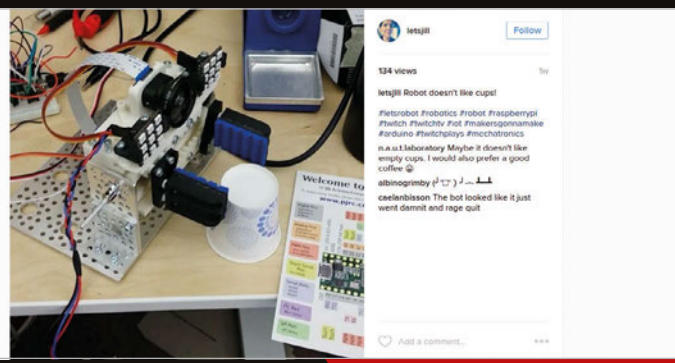
Poor intern Carl was used as a prop in a recent stream, tasking the chat to feed him via the robots. Apparently, they weren't so good at the task, as shown by the mess made around Carl's hole in the table.



Party time

youtu.be/nEbiwGP_lGo

To celebrate the one-year birthday of Aylobot, the robots decided to hold a party. This started with the robots writing out invitations, with one robot writing the address and another putting the stamps on the letters to send them out. They weren't very legible, unfortunately.



Cup-flipping robot

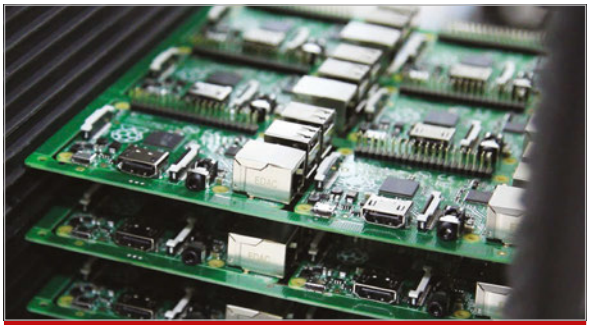
magpi.cc/2a1QcRV

This adorable little robot has one function and one function only: to grab and flip a cup. Or anything that can be grabbed from the sides and thrown into the air. It's a tough job, but some robot's got to do it.

PI PARTNER PREMIER FARNELL SOLD

A licensed Raspberry Pi manufacturer is acquired by Swiss firm Dätwyler

While some misleading newspaper headlines stated that the maker of Raspberry Pis had been sold, there is no need for concern! The reality is that Premier Farnell, which manufactures and distributes Pis under licence from the Raspberry Pi Foundation, is the subject of a takeover bid by Swiss conglomerate Dätwyler. With a recommended all-cash offer of 165p per share, this values the Leeds-based firm, which includes the Element14 group, at £792m. According to a statement from Dätwyler, "Premier Farnell and Dätwyler are two leading distributors of electronic components, with complementary product ranges and geographic market presence."



PI FACTORY

Most Raspberry Pi production, including 100% of Pi Zeros, occurs at the Sony factory in Pencoed, South Wales. Manufacture of all Pi models apart from the Zero is handled by two main licensed partners, Premier Farnell and RS Components, who decide how many should be made to meet demand. Official Raspberry Pi add-ons, including the Sense HAT and Camera Module, are also made at the Pencoed factory.

While Raspberry Pi sales have been booming, with 500,000 Pi 3s sold in the new model's launch month alone, the Premier Farnell business as a whole has been struggling with weak sales, resulting in a dividend cut and decline in the share price until the offer was announced on 14 June.

Premier Farnell is one of the two key licensed Raspberry Pi manufacturers, along with RS Components. "We license the technical designs for the Pi and the various trademarks to the partners, who then decide how many to make," explains Raspberry Pi Trading CEO Eben Upton. "They are required to keep the product in stock, and to keep the price at or below an agreed maximum ($35 in the case of the Model 3B), but beyond that the practical decision-making is up to them."

Eben doesn't believe Raspberry Pi production will be affected: "We've had a fantastic relationship with Premier Farnell over the last four and a bit years, and my expectation is that the acquisition won't do anything to change that. I've already had some contact with the team at Dätwyler and it's clear they understand the value of Pi to the combined business."

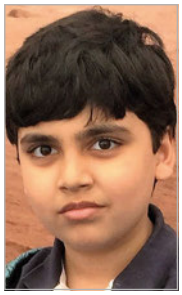
Indeed, Eben hopes the acquisition may have some benefits, possibly helping to increase Pi sales in other parts of Europe: "Dätwyler have strengths in B2C sales in some areas where the licensees have traditionally only had B2B channels. There is one Dätwyler business [Reichelt Elektronik] which is already a significant Raspberry Pi reseller, and I hope bringing the two businesses under the same umbrella will provide a boost."

COMBINED FORCE

Assuming the takeover – which is scheduled to close in the fourth quarter – goes ahead as planned, the combined group will boast a workforce of over 10,000. Based in Switzerland, the Dätwyler Group already comprises more than 50 operating companies, with sales in over 100 countries, generating an annual revenue of around CH1,200m (£947m). Its two main divisions are Technical Components (electronics, automation, and ICT) and Sealing Solutions (healthcare, civil engineering, automotive, and consumer goods).

AWARD-WINNING ASTHMA PI KIT

Nine-year-old Arnav Sharma's Pi-powered asthma management kit bags two awards



Arnav Sharma's AsthmaPi kit (**youtu.be/3Dniuy4-D3M**) has won two Tech4Good 2016 awards: People's and Winner of Winners. Based around a Raspberry Pi and Sense HAT, the kit can help parents of children suffering from asthma. Upon learning that asthma attacks can be triggered by dust, pollen, pollutants, and other environmental factors, keen coder Arnav set about researching the possibility of creating a detection device based on the Raspberry Pi. It took him two months to build a prototype that monitors all the main asthma triggers.

The Sense HAT on the Pi measures humidity and temperature. A gas sensor, wired up to an MCP3008 ADC converter on a breadboard, monitors hazardous atmospheric gases: ammonia, carbon monoxide, and CO_2. Finally, a dust sensor connected via an Arduino Uno detects particles and pollen in the air.

Arnav programmed the kit using Python and C++. "Coding was easier for the Sense HAT, but was very difficult for other sensors." As well as monitoring asthma triggers, the kit can even send email and SMS text alerts to prompt users to take their medication. Arnav has also created an accompanying booklet to make understanding asthma simple.

"I would love to produce a commercial version of the device," Arnav tells us. The Asthma UK charity has already shown an interest in the kit, so who knows?

tech4good 2016 Awards

TECH4GOOD AWARDS

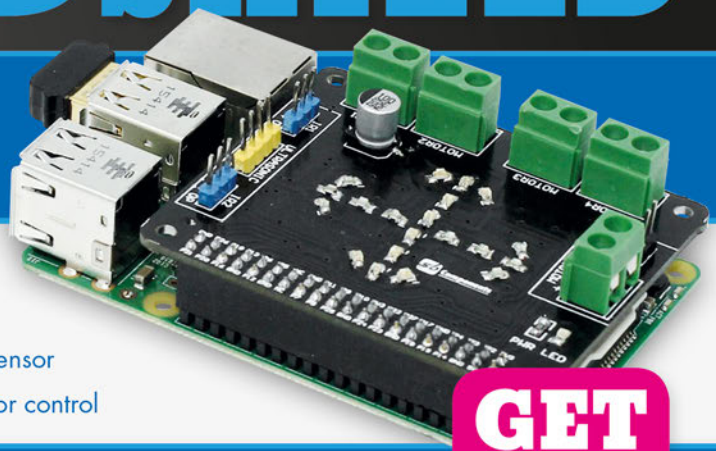
Held annually since 2011, the Tech4Good awards (**tech4goodawards.com**) recognise organisations and individuals who use digital technology to improve the lives of others. Previous winners include the Raspberry Pi itself. A finalist for the BT Young Pioneer Award, Arnav's AsthmaPi kit picked up the People's Award in a public vote of over 38,000 people. It also took home the Winner of Winners Award, decided by an audience vote at the ceremony, using glow sticks!

sb Components

BUILD RASPBERRY PI POWERED ROBOTS

WITH THE NEW SB MOTOR SHIELD

WITH YOUR **MOTOR SHIELD**, YOU CAN EASILY CREATE AND CONTROL YOUR OWN REMOTE CONTROLLED CAR OR OBSTACLE-SENSING ROBOT



THE SB MOTOR SHIELD FEATURES:

- Ability to power up to four motors with bi-directional control
- Connectors for 2 x IR sensors
- Directional LEDs
- Connectors for an ultrasonic sensor
- Python library simplifying motor control

www.motorshield.sb-components.co.uk Call: 0203 514 0914

At **SB Components** we strive to offer our customers the best prices for the best products. Our product team works tirelessly to source top quality affordable components from around the world.

Raspberry Pi is a trademark of the Raspberry Pi Foundation. Raspberry Pi not included. *Compatible with Raspberry Pi

GET 25% OFF when you pre-order before 19th August!



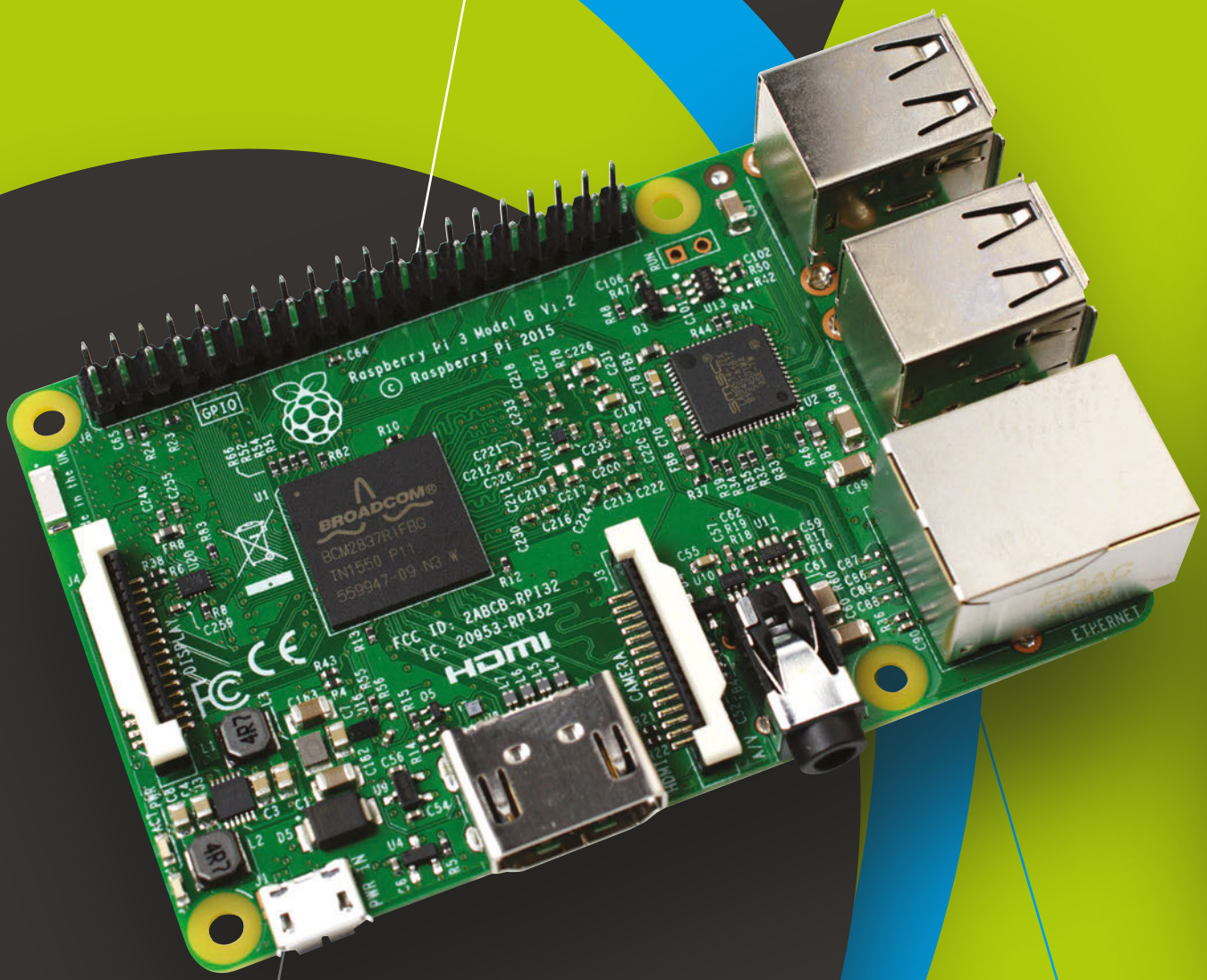
ENTER TO WIN

THE INTERNET OF VOICE CHALLENGE

Think you got what it takes? Make the best project using Amazon Alexa and Raspberry Pi and win Amazon Echos, Amazon Taps, Echo Dots, and $11,000 worth of gift cards! Contest ends August 31, 2016.

Go to **bit.ly/Alexa-Pi** for more details.

Only on hackster.io





"Get more out of your Raspberry Pi!"

"Learn to use IoT Core!"





"Discover amazing projects!"

WINDOWS 10 IoT CORE

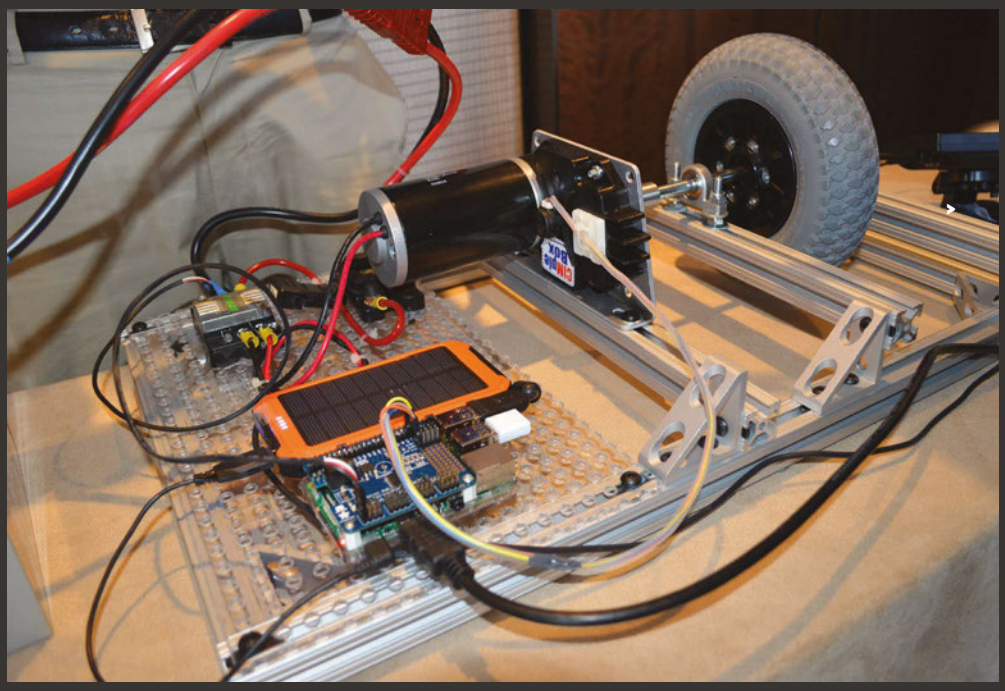
The Raspberry Pi 3 opens up Windows 10 IoT Core to greater uses – here's how to get started with it...

At the launch of the Raspberry Pi 3 in London, the Microsoft team were present to show off a new project they'd created to make use of the Pi and Windows 10. On the face of it, demonstrating how a feedback loop could be created to keep a wheel spinning at roughly the same speed, even with varying resistance, may not seem amazing, but a lot was going on to keep it turning.

This is the strength of Windows 10 IoT Core on the Raspberry Pi 3, allowing for more power to make unique projects that connect to the Internet of Things. With a new kit on the way to help people get started with Windows 10 on Raspberry Pi 3, we thought we'd give you a head start so you can begin making more amazing and varied projects on your Raspberry Pi.

RASPBERRY PI AND WINDOWS

How Microsoft is keen to support the makers of the future with Windows 10 IoT Core



Microsoft was at the Raspberry Pi 3 launch to show off its brand new wheel project. Find out the inner workings of it here: **magpi.cc/2a8rak9**

Microsoft is launching a new kit for the Raspberry Pi that allows you to make some amazing projects with Windows 10 IoT Core. We caught up with Asobo Mongwa and Dan Rosenstein from the Microsoft team behind the kit, to find out how we got here.

The MagPi: How did Windows 10 IoT Core on Raspberry Pi start?

Dan Rosenstein: Raspberry Pi was taking off in the market and at the same time we were working on Windows 10 IoT Core. Since the very beginning, our belief has been that makers are the creation engine and game changer for IoT, and that the maker movement was going to be the catalyst for that change. Our design goal for Windows 10 IoT Core was to make the programming model easy and accessible to makers, and work with the large set of democratised hardware already in the maker ecosystem. Our goal was to join makers where they were. The Raspberry Pi Foundation provided the perfect mix to help launch Windows 10 IoT Core: it offered the hardware, market share, partnership, and community that we wanted to be part of.

WHAT'S IN THE BOX

The new kit has a wealth of great components to get you started doing some amazing things with Windows 10 IoT Core:

- 5-inch display
- Grove shield (to plug components into)
- Ultrasonic range finder
- Sound sensor
- Temperature and humidity sensor
- Light sensor
- Rotary angle sensor
- LCD RGB backlight
- LED bar
- Relay
- Button
- Buzzer
- HDMI cable



The new Microsoft Grove kit is a specially put together package that lets you make the most of Windows 10 IoT Core on Raspberry Pi



MEET THE TEAM

Dan Rosenstein (left) is a Principal Technical Program Manager and Asobo Mongwa (right) is a Program Manager; they both work at Microsoft's Operating System Group in the Internet of Things Maker Program Management team. Here they are at the National Maker Faire in DC in June, one of many maker events that the team attend to show off the power of Windows 10 IoT Core on Raspberry Pi to the maker community.



TM: What's the reaction been like in the community?

Asobo Mongwa: We're pleased to see the excitement among the maker community, as more and more become familiar with the investment Microsoft is making in IoT, and the continuous improvements we're bringing to Windows 10 on the Raspberry Pi. We've heard this excitement first-hand during the recent Maker Faires in the Bay Area and Washington DC. We can't wait to see how makers continue to create new innovations using Windows 10 IoT Core and the Pi 2 and 3, particularly now that we've shared information on our Maker To Market pathways: solutions we've put in place to enable makers to accelerate the commercialisation of their idea in the Marketplace.

TM: What was the idea behind the new kit?

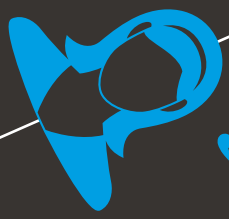
AM: The upcoming Microsoft Grove Kit is in collaboration with Seeed Studio. Seeed Studio took great interest in Microsoft's investments in the IoT and maker space, and approached us for a co-engineering engagement on the kit. We see it as an indicator that influencers in the industry recognise the impact in the work we're doing, and want to partner with us in helping the ecosystem to release the next generation of solutions on the Raspberry Pi with Windows 10 IoT Core. The Grove Kit is an enabler, and it's designed for hardware and software developers at all levels of technical expertise. What I personally like about it is the fact that it takes away complexities like soldering, jumper cables, and breadboards, and allows the user to focus on their creativity. It's modular, so you can use the components included in this kit as well as other additional Grove components.

TM: How much more can Pi 3 do versus previous iterations?

DR: There were a few fundamental changes with the Pi 3, all of which we enabled on Windows 10 IoT core: faster clock speed, upgraded SD controller, on-board WiFi, and on-board Bluetooth. Because of the faster clock speed, GPIO performance with our Lightning driver was measured at a higher throughput.

TM: How important is the relationship between Windows 10 and Raspberry Pi to Microsoft?

AM: Raspberry Pi has had a significant impact in education and the maker movement. We're excited to see the increased interest in the industrial market for designing end products. We partnered with Element14 to offer Pi 3 as one of Microsoft's supported silicon platforms for commercialisation with Windows 10 IoT Core. The Raspberry Pi has been showcased at some of Microsoft's major conferences like //build, and the Windows Hardware Engineering Community conference in China. These are events where we share Microsoft's current and future engineering investments with software and hardware partners.



"A mood ring for the modern age!"



DAVIDE LONGO

Davide is a .NET software developer at the IT department of an airline company. He lives on the beautiful island of Sardinia, is a graduate in physics, and is passionate about learning new technology.

magpi.cc/2a30GRI

CHROMOTHERAPY

This futuristic project uses Windows 10 IoT on a Raspberry Pi to monitor your happiness. It then adjusts the lighting in a room to match your mood

You'll Need

- Arduino Uno **arduino.cc**
- Strip LED RGB **adafruit.com/product/306**
- Bluetooth HC-05
- Generic breadboard **uk.farnell.com/prototyping-boards**
- Webcam **uk.farnell.com/webcams**

"Microsoft has a new service that works with facial recognition," explains .NET developer Davide Longo. "It processes and returns information in the form of a score, rating happiness, anger, contempt, disgust, fear, surprise, and sadness.

The service, part of Microsoft's Cognitive Services called 'Emotion API', inspired Davide to build this remarkable Windows 10 IoT project. Chromotherapy monitors a user's facial expressions and adjusts mood lighting to match.

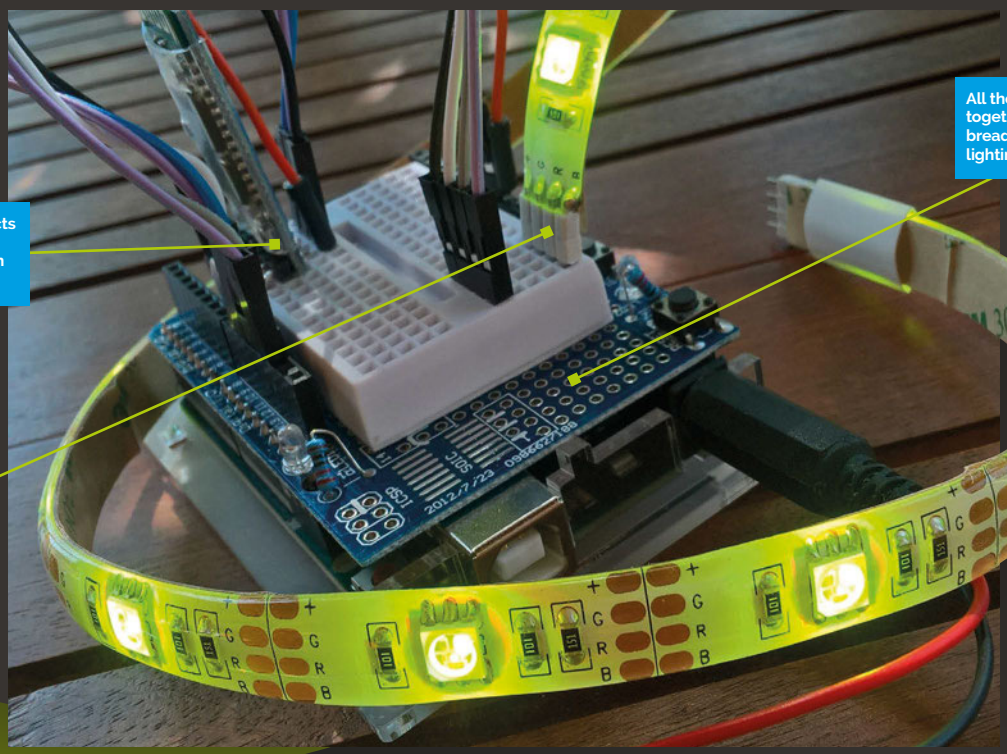
The project has a Raspberry Pi running Windows IoT 10 Core at its heart. The Raspberry Pi is hooked up to a webcam, which sends captured images of the user's face to the Emotion API.

This sends a message via Bluetooth to the in-home lighting system, simulated using an Arduino hooked up to an RGB LED strip.

"Each time I start a new project, I try to create something new and innovative that uses the latest technologies," says Davide. "I immediately thought of the importance of light in a workplace.

"In chromotherapy, there are simple colour rules that are reflected in daily experience," he explains, "Blue is a calming and refreshing colour; green symbolises balance, peace, and renewal."

All code is written in C# and available on GitHub (**magpi.cc/2a31fLp**). "Sharing ideas is the cradle of future innovation," says Davide. "Despite what you might think, all the Microsoft programs and Microsoft Azure services used here are free."





PHILIPPE LIBIOULLE

Originally from Belgium, Philippe is now an R&D group manager working in Canada. He creates a wide range of home automation projects, and is currently investigating Thread technology using Silicon Labs products.

hackster.io/ThereIsNoTry

PERSONAL HOME SAFETY AGENT

Fire and water are a risk to any home, but one maker uses his Raspberry Pi with Windows 10 IoT Core to monitor his home and get instant alerts to any problems

You'll Need

- Water valve **magpi.cc/2a3uP2B**
- Water sensor **magpi.cc/2a3v1io**
- Fire bell **magpi.cc/2a3ukpg**
- SSR relay **magpi.cc/2a3uZXY**
- Smoke alarm relay module **magpi.cc/2a3v79J**
- 12V SPDT relay (duo and single) **magpi.cc/2a3vlxR**

Canada is a lovely place to live, but it's a challenging environment. In winter, the temperature can go down to -30C, which means heating has to be kept on and frozen pipes can burst.

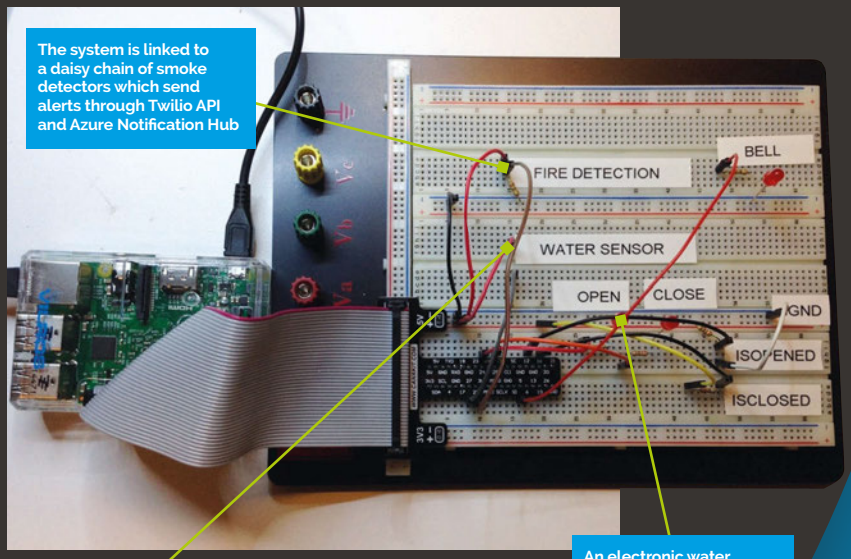
"I was talking to a friend," says Philippe Libioulle. "He had to live far from home for months, just because a dishwasher had failed. During a chat with another friend, I realised water leakages are a big problem as well. My friend had to replace all the furniture in his basement, just because a pipe broke while he was at work.

"Smoke detectors are mandatory, as per the regulations, but if there's nobody home to react, the alert is not forwarded." So Philippe put his maker mind to work and built this Internet of Things home safety system.

A Raspberry Pi runs Windows 10 IoT Core, and monitors smoke and fire alarms and water leakages. If it detects a fault, the cloud system triggers actions. These include turning on the fire bell, or closing the main water entry to the house. It can even send messages to Philippe's mobile phone, or his neighbours. "If the system is stable enough, I would consider notifying 911 too," says Philippe.

"I get alerts through SMS, using Twilio API (**twilio.com**). I also used Azure Notification Hub (**azure.microsoft.com**) to send alerts to laptops, smartphones, and tablets.

"I'm inspired by real challenges, expressed by real people," adds Philippe. "I have a new version of this concept, more modular and with a cleaner look."



The system is linked to a daisy chain of smoke detectors which send alerts through Twilio API and Azure Notification Hub

A water sensor at ground level is used to detect any leaks in the home

An electronic water valve is hooked up to the system. If a leak is detected, it automatically shuts off the flow of water to the house



"Keep your home safe!"



JENS MARCHEWKA

A senior C# software developer from Dortmund, Jens is very experienced with Windows, so it was the obvious choice for this project.
magpi.cc/29ysoSK

MAGIC MIRROR

Mirror, mirror on the wall, show me the latest news and all! This Pi-powered magic mirror delivers much more than your reflection

You'll Need

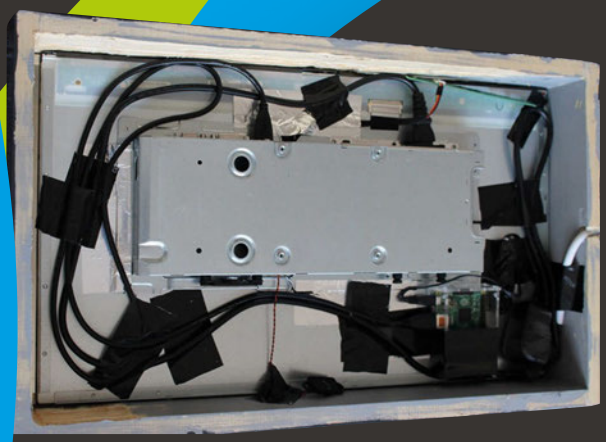
- GG Mirastar observation mirror **magpi.cc/29Fv0Co**
- LCD monitor
- Wood to make the frame
- Cables and connectors

When attempting to purchase one-way observation mirror glass for his latest project, Jens Marchewka got some strange reactions from the suppliers he phoned, such as "What do you want to do with that? No, I don't want to know, bye."

He needed it to create a 'magic mirror' that displays information on an LCD screen behind the reflective glass panel. Inspired by an earlier project by Michael Teeuw (**magpi.cc/29y06Yk**), Jens wanted a mirror that could show articles from his favourite news sources, along with weather and calendar info.

Not wanting to place ugly buttons down the side, he opted instead to use one-word voice commands to control the mirror's display. For this, Windows' built-in speech recognition system came in useful, although Jens experienced a few teething troubles. "I was using recognition without a grammar and that didn't work properly… it [kept] recognising words [that] no one





The interior of the fully installed mirror; at the bottom-right you can see the Pi

was speaking and even activated the radio. This is easy to avoid if a grammar is used."

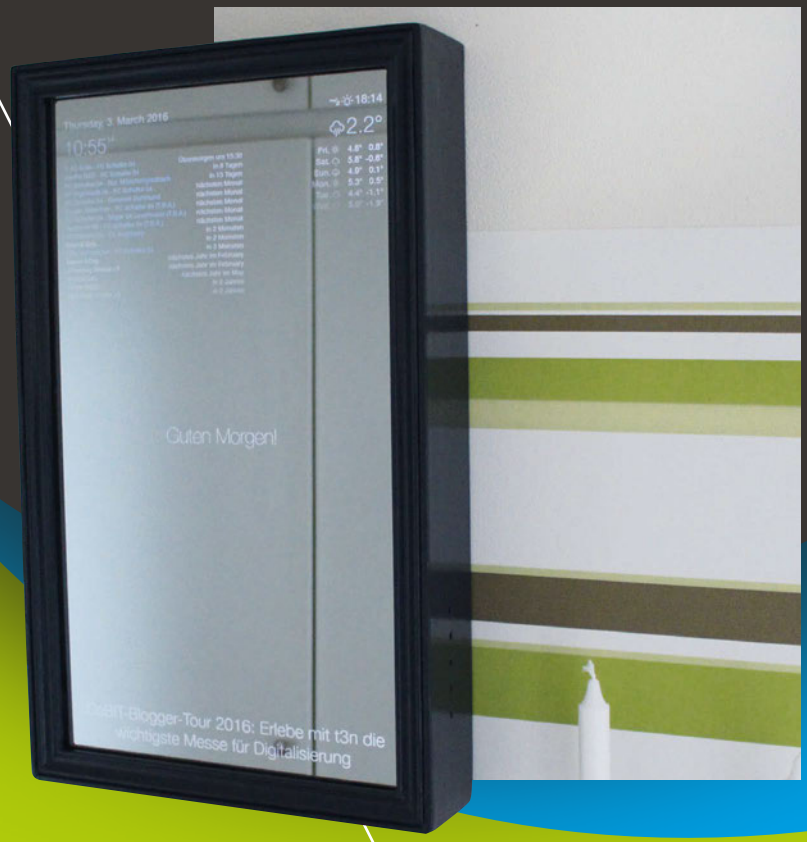
The whole build process took Jens only ten hours, including the construction of a wooden case to frame the mirror glass and enclose the LCD monitor, Raspberry Pi 2, and cabling behind it. To help keep everything firmly in place, he added a toeboard border to the front of the frame. He then painted the case black: "While constructing the frame, you should remember that the paint will take [up] space, so you need to keep that in mind and not build it too close."

Jens used a 24-inch LCD Iiyama monitor for his project, but any model or size could be substituted, so long as it has a good black level to achieve a nice mirror effect. Jens also notes that the monitor case needs to be disassembled to position the LCD panel as near as possible to the mirror. While his LCD screen is fully powered 24/7, he's working on a setup for scheduling its power via CEC.

Along with news and weather, Jens's magic mirror shows calendar events – including fixtures for his favourite football team, Schalke 04 – obtained via the iCal web calendar. "You can define more than one iCal source and assign a colour to each… that makes it possible to have multiple events shown in different colours." Jens has also added internet radio playback to the mirror, and is planning to use face recognition for displaying personalised events and news for him and his wife.



> "The future is here with the internet in your mirror!"



Calendar events are shown top-left, with weather top-right. Say 'news' to see the latest headlines

THE MAGIC OF WINDOWS 10 IOT

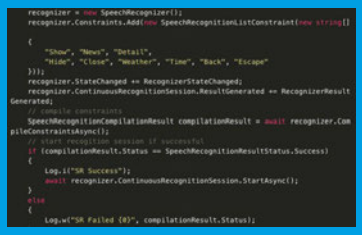
User interface design

One of the main reasons Jens chose to use Windows 10 IoT Core for this project, along with his experience as a Windows developer, is how easy it is to design custom user interfaces that look polished.



Speech recognition

Windows' built-in speech recognition comes in handy for the magic mirror. Jens simply created a SpeechRecognizer function with a list of commands. The mirror runs a continuous speech recognition session.



```
recognizer = new SpeechRecognizer();
recognizer.Constraints.Add(new SpeechRecognitionListConstraint(new string[]
{
    "Show", "News", "Detail",
    "Hide", "Close", "Weather", "Time", "Back", "Escape"
}));
recognizer.StateChanged += RecognizerStateChanged;
recognizer.ContinuousRecognitionSession.ResultGenerated += RecognizerResult
Generated;
// compile constraints
SpeechRecognitionCompilationResult compilationResult = await recognizer.Com
pileConstraintsAsync();
// start recognition session if successful
if (compilationResult.Status == SpeechRecognitionResultStatus.Success)
{
    Log.i("SR Success");
    await recognizer.ContinuousRecognitionSession.StartAsync();
}
else
{
    Log.w("SR Failed {0}", compilationResult.Status);
```

Syndication client

The detailed news is sourced from RSS feeds, using the SyndicationClient class within the Windows framework. "It's a very comfortable class for parsing RSS feeds," Jens tells us.



GETTING STARTED WITH WINDOWS 10 IOT CORE

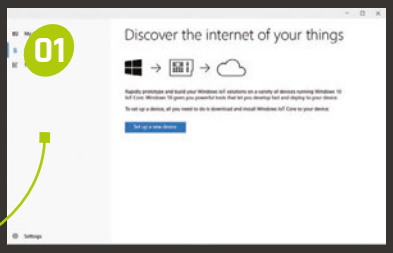
Prepare your Raspberry Pi 3 for working with Windows 10 IoT Core

Before we can start using Windows 10 IoT Core, we need to get it installed on the Raspberry Pi. The best and recommended way to do this is via the IoT Dashboard on Windows 10 systems; this then easily allows you to program a Windows 10 Raspberry Pi using Visual Studio. Let's get started!

>STEP-01
Prepare your PC

To start with, make sure your PC or laptop is updated to the latest version of Windows 10, or at least version 10.0.10240. You can check this by clicking the Start button and typing **winver**. Once that's checked and you're properly updated, you'll need to download and then install the IoT Dashboard from the link here:
magpi.cc/2ah13dt

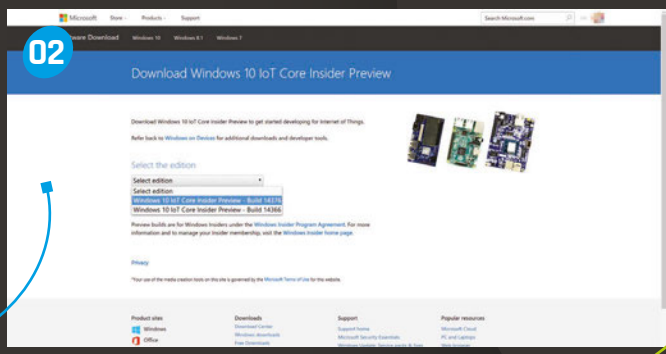
From the IoT Dashboard, you can also find sample projects and soon you'll be able to connect to the Azure IoT cloud service.



>STEP-02
Get the right image

For the Raspberry Pi 3, you need to get yourself signed up for the free Windows Insider Program (**insider.windows.com**). You'll then need to find and download the latest version of the Windows 10 IoT Core Insider Preview from here (pick the highest number):
magpi.cc/29Ugtoz

Go to where you downloaded it and double-click on the ISO file to enter it, then click on the installer inside to get your PC set up to write the SD card properly. Follow the on-screen instructions in the wizard.



>STEP-03
Write to the disk

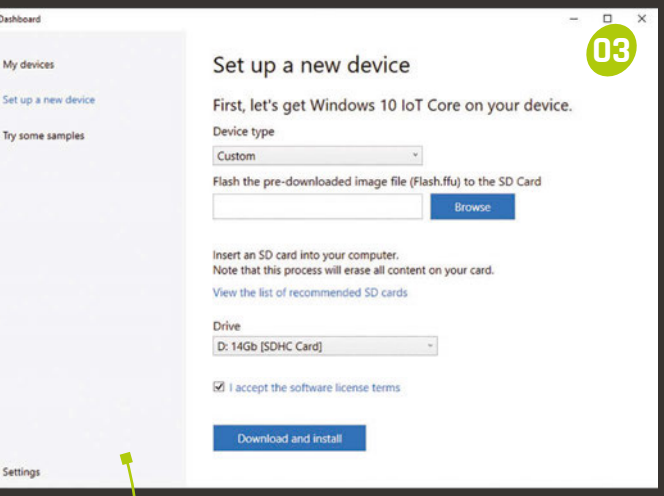
Once the installer is finished, launch the IoT Dashboard we downloaded in step 1. From here we can write to the SD card. First, click 'Set up a new device'. Select **Custom** from the 'Device type' menu and then click **Browse** right below that to find the image. The image lives in:

```
C:\Program Files (x86)\
Microsoft IoT\FFU\
```

WINDOWS 10 ON THE **RASPBERRY PI 2**

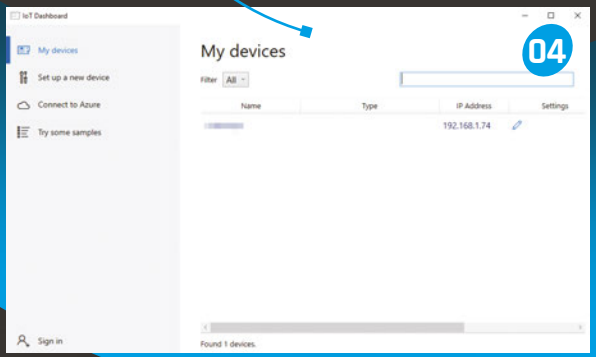
You can set up Windows 10 IoT Core very similarly on the Raspberry Pi 2; in fact, IoT Core is available in NOOBS already. Upgrading to a Raspberry Pi 3 is the best way to make use of Windows 10 on Pi, though, thanks to its extra power.

There will be a Raspberry Pi folder in there containing the file **Flash.ffu**. Select that and click **Install/Download and install**.



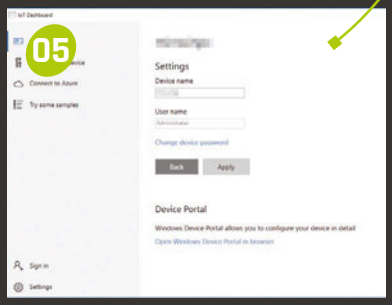
>STEP-04
Networking

For our tutorials, and in general, you need to be able to connect to the Raspberry Pi over the network to quickly and easily program it. It's recommended to connect it via Ethernet to start with if you can. Go to 'My devices' and you'll be able to find the Raspberry Pi, so you can configure it once it's turned on; this includes setting up access to your wireless network to use it in the future.



>STEP-05
Configure the board

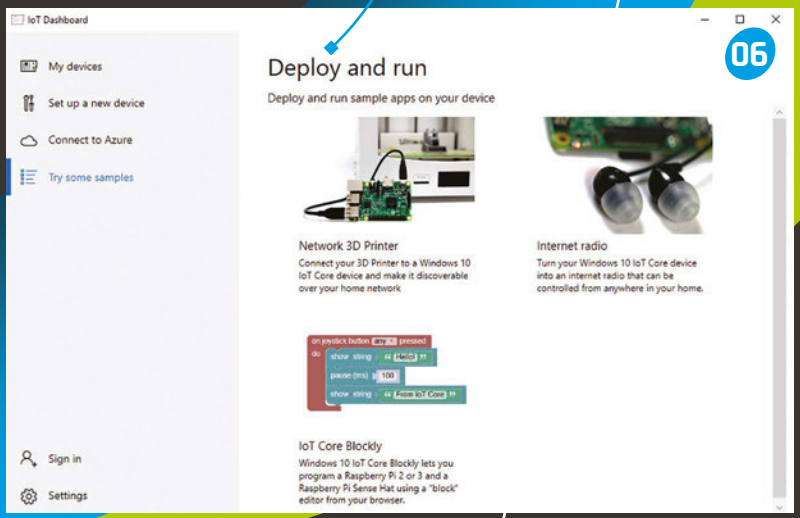
Once you've found your Pi, you can configure it by clicking the pencil icon next to it. From here you can launch Windows Device Portal, which has some basic settings you can edit. As well as setting up wireless networking if you need to, it's also a great idea to set a username and change the default password. The latter is **p@ssword**, which you'll need to know before changing it. It's just good InfoSec!



>STEP-06
Start making

You're ready to get going! You may have noticed there's an option to try out samples on the side column, so if you want to test out a basic project from the list, you can do so. Just select the Raspberry Pi from the drop-down list and you're off.

To program it further, we'll connect up Visual Studio, which we'll cover in our first tutorial over the page...



HAVE A DIFFERENT **SETUP?**

Follow the customisable Getting Started guide here: **magpi.cc/2abqzAW**





PHIL KING

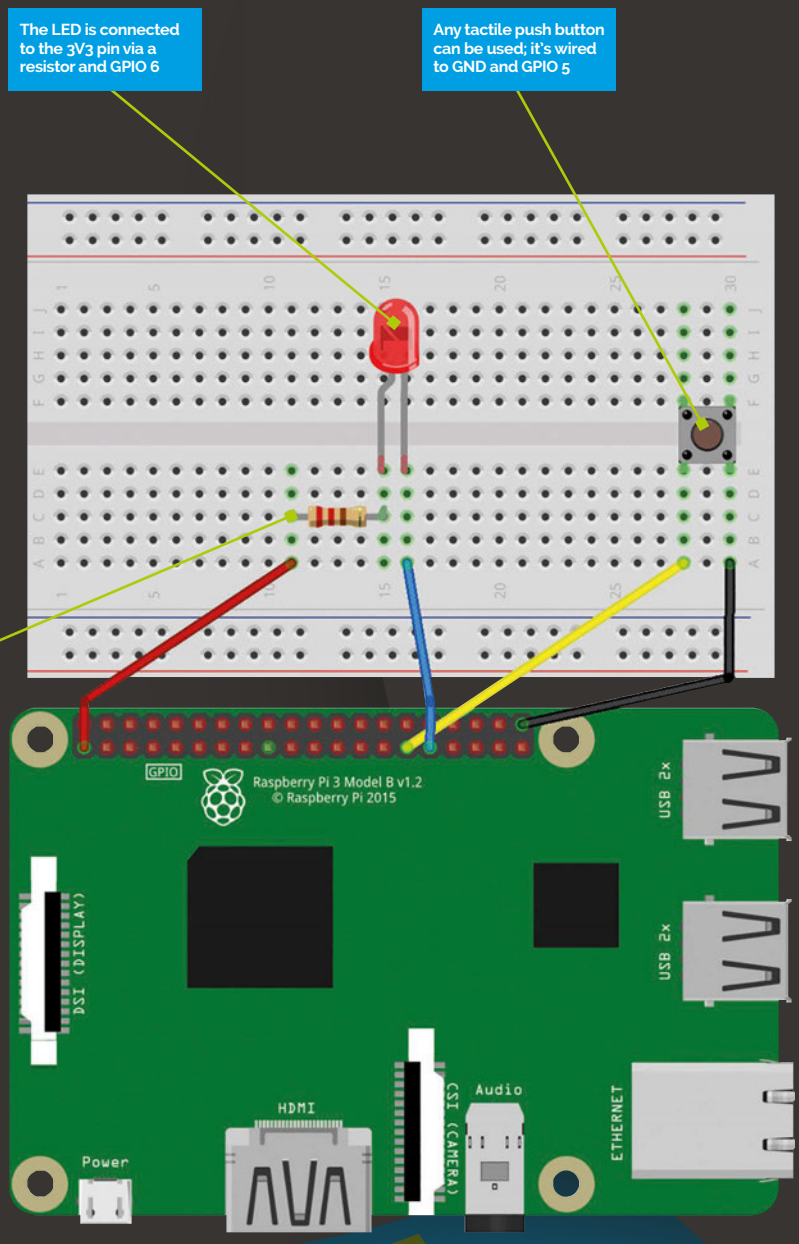
When not sub-editing The MagPi and writing articles, Phil loves to work on Pi projects, including his two-wheeled robot.
@philking68

BUTTON-ACTIVATED LED

Create a simple physical computing project using an LED and a push button

You'll Need

- 1× solderless breadboard
- 1× tactile push button
- 1× LED
- 1× 333Ω resistor
- 4× male-to-female jumper wires



You've downloaded Windows 10 IoT Core, installed it on an SD card, and set up your Raspberry Pi as an IoT Core device? It's time to start coding it, using Visual Studio 2015 on a Windows 10 PC. Let's start with a simple project that uses a push button to toggle an LED…

Connect up your Pi

Before commencing, you'll need the IP address of your Raspberry Pi, as shown in My Devices in Windows 10 IoT Core Dashboard. You'll also need Visual Studio 2015 Community installed, which you can find here: **magpi.cc/2adz7Hw**

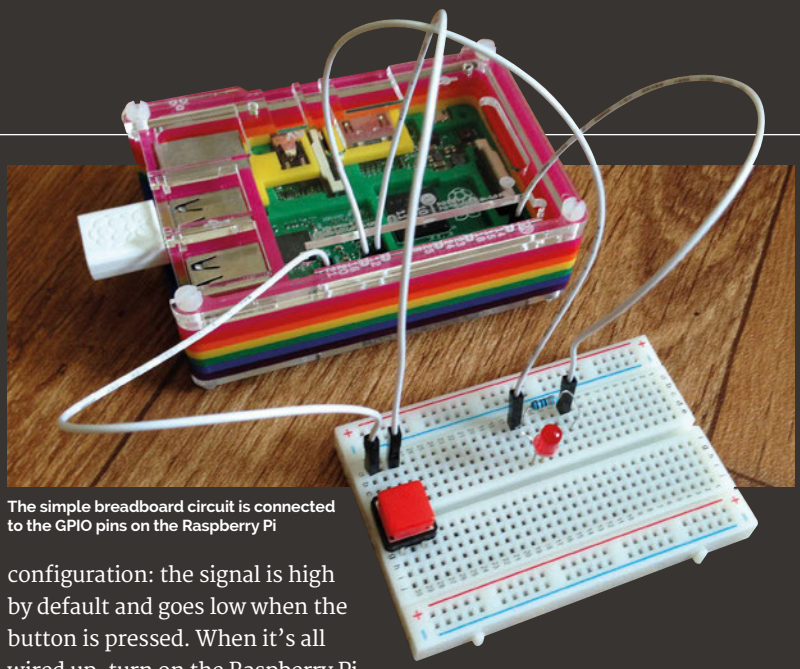
First, let's wire up our electronic components on a breadboard. It's best practice to turn the Raspberry Pi off while doing so. As shown in the diagram, we connect the shorter, negative leg of the LED to GPIO 6 on the Pi. We connect its longer, positive leg to a 333Ω resistor; the other end of the latter is connected to the Pi's 3V3 pin. This is what's known as an 'active low' configuration, as when the GPIO pin is set low, current will flow through the LED and light it; when the pin is set high, it will turn off.

Next, we'll wire up the tactile push button, which straddles the central groove of the breadboard. One pin is connected to GPIO 5, while the other (on the same side) is wired to GND. Again, this is an active low

Language
>C#

FILENAME:
PushButton.csproj

DOWNLOAD:
magpi.cc/29Qwut8



The simple breadboard circuit is connected to the GPIO pins on the Raspberry Pi

configuration: the signal is high by default and goes low when the button is pressed. When it's all wired up, turn on the Raspberry Pi. Note that the Pi should be in headed mode (which is the default) for this project. If not, you'll need to set it in PowerShell on your main Windows 10 PC – see **magpi.cc/29K4XWH**.

Now, on your Windows 10 PC, download the code examples, including PushButton for this project, from the GitHub repo: **magpi.cc/29Qwut8**. Right-click the zip folder and extract all files, then open **samples-develop\PushButton\CS\PushButton.csproj** (the Visual C# project file) in Visual Studio; if not done already, you'll be prompted to enable Developer Mode on your PC.

Select **ARM** for the target architecture. Go to **Build>Build Solution**. This process may take a while. When finished, select **Remote Machine** from the Device drop-down arrow, enter the IP address of your Raspberry Pi, and select **None** for the authentication type. Press **F5** to deploy and debug. Once it's successfully deployed, try pressing the push button on your circuit; the LED should light up. Each time you press the button, the LED should toggle its state.

Covering the code

Let's take a look at the code in the **MainPage()** element to see how this works. At the top, we open the GPIO pin resources we'll be using...

```
  buttonPin = gpio.
OpenPin(BUTTON_PIN);
  ledPin = gpio.OpenPin(LED_
PIN);ode>
```

We assign the LED pin a high value to make sure it's turned off to begin with. When we change the drive mode to Output, it will immediately drive the latched output value onto the pin:

```
  ledPin.Write(GpioPinValue.
High);
  ledPin.
SetDriveMode(GpioPinDriveMode.
Output);ode>
```

We then set up the button pin. Since the Raspberry Pi 2 and 3 have built-in pull up resistors that we can activate, we don't need to add an external one.

```
  if (buttonPin.IsDriveMode
Supported(GpioPinDriveMode.
InputPullUp))
    buttonPin.
SetDriveMode(GpioPinDriveMode.
InputPullUp)
  else
    buttonPin.
SetDriveMode(GpioPinDriveMode.
Input);ode>
```

Next, we connect the GPIO interrupt listener, which is called each time the button pin changes state. We also set the DebounceTimeout property to 50ms to filter out spurious events caused by electrical noise.

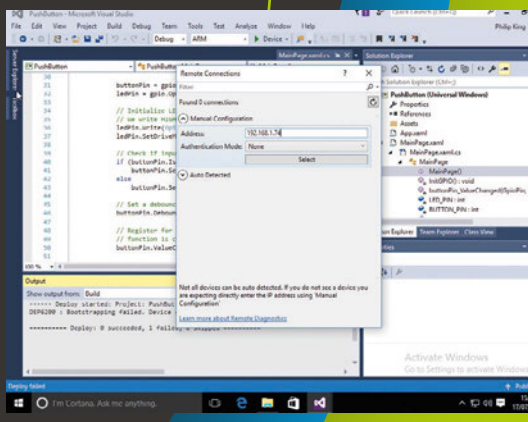
```
  buttonPin.DebounceTimeout =
TimeSpan.FromMilliseconds(50);
  buttonPin.ValueChanged +=
buttonPin_ValueChanged;ode>
```

In the button interrupt handler, we look at the edge of the GPIO signal to determine whether the button was pressed or released. If pressed, we toggle the state of the LED:

```
  private void buttonPin_
ValueChanged(GpioPin sender,
GpioPinValueChangedEventArgs e)
  {
    if (e.Edge == GpioPinEdge.
FallingEdge)
    {
    ledPinValue = (ledPinValue ==
GpioPinValue.Low) ?
    GpioPinValue.High :
GpioPinValue.Low;
    ledPin.Write(ledPinValue);
    }ode>
```

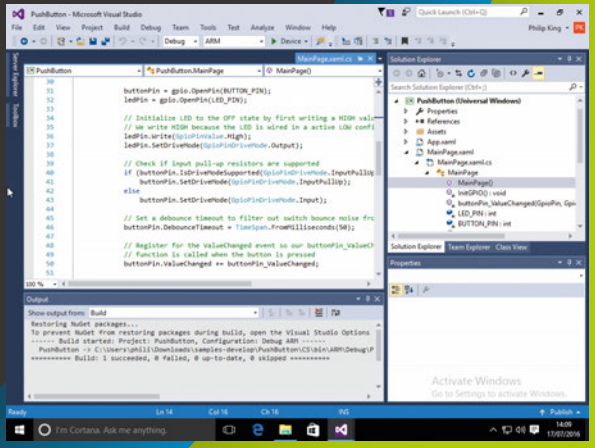
Finally, we want to update the user interface with the current state of the pin, so we invoke an update operation on the UI thread.

This tutorial has been adapted from Push Button Sample, from the Windows IoT team consisting of Daniel Kuo, Tycho's Nose, and Liz George, from the Microsoft Projects site: magpi.cc/29K5xne



Select 'Remote Machine' and enter the IP address or name of your Raspberry Pi

The code that controls the circuit can be found in the MainPage() element of the project





ROB ZWETSLOOT

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

INTERNET-CONNECTED LED

"Connect to the Internet of Things!"



Take information from the internet to light an LED, connecting to the Internet of Things, and becoming part of the World Map of Makers

You'll Need

- 1× Solderless breadboard
- 1× LED
- 1× 560Ω resistor
- 2× Jumper wires

You now know how to light an LED by pressing a button on Windows 10 IoT. Let's now use the IoT part of the software to have an LED controlled by the internet!

Prepare your Pi

Turn your Raspberry Pi off and wire up the circuit as shown in the Fritzing diagram. It's a simple one so it shouldn't take you very long.

Next, you'll need to download some code to get the whole setup working: **MainPage.cs**, which is a small part of the project. Grab it from here: **magpi.cc/2a5G0rP**. Open up **Lesson_201.sln** and then open up **MainPage.xaml.cs**. This needs to be modified to be able to run the LED script we want to write, so replace all the code with the full version from here: **magpi.cc/2a7rNgL**.

Once that's done, we can create the class file. From the main menu, select **Project > Add Class...** The Add New Item dialogue will open and default to Visual C# Class. As mentioned before, we're calling it **InternetLED.cs**.

To begin with, we need to add 'using' lines at the top of the file so the code can reference the GPIO device, web interfaces, and the system diagnostics.

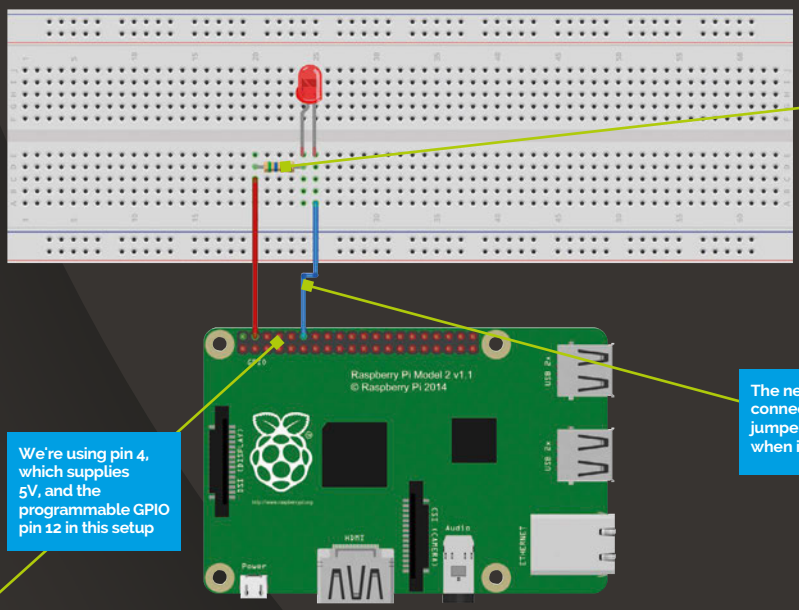
```
using System;
using System.Diagnostics;
using System.Net.Http;
using System.Threading.Tasks;
using Windows.Devices.Gpio;

namespace Lesson_201
{
```

Inside the class brackets, add the following lines. The first three lines are for controlling the GPIO, the next line allows us to add a delay if we don't get a response for the web API call, and the last two lines store the state of the LED (on or off).

```
  class InternetLed
    {
      private GpioController
gpio;
      private GpioPin
LedControlGPIOPin;
      private int
LedControlPin;
      private const int
DefaultBlinkDelay = 1000;
      public enum eLedState {
Off, On };
      private eLedState _
LedState;
```

Now, add the class constructor code which will store the value of the pin used to control the LED, interacting with the GPIO controller section of the class.



Language
>C#

FILENAME:
InternetLED.cs

DOWNLOAD:
magpi.cc/2a5Fqdp

Join 4,000 other makers on the World Map of Makers, and light up their LEDs too!




```
  public InternetLed(int
ledControlPin)
    {
      Debug.
WriteLine("InternetLed::New
InternetLed");
      LedControlPin =
ledControlPin;
    }
```

Now we add code that initialises communication with the Pi's GPIO. It first sets up the GPIO to be used for the LED, and then allows us to see the current pin value:

```
  public void InitalizeLed()
  {
    Debug.WriteLine("InternetLed
::InitalizeLed");
    gpio = GpioController.
GetDefault();
    LedControlGPIOPin = gpio.
OpenPin(LedControlPin);
    LedControlGPIOPin.
SetDriveMode(GpioPinDriveMode.
Output);
    GpioPinValue startingValue =
LedControlGPIOPin.Read();
    _LedState = (startingValue
== GpioPinValue.Low) ?
eLedState.On : eLedState.Off;
  }
```

This part allows us to interact with the LED, depending on its state:

```
  public eLedState LedState
  {
    get { return _LedState; }
    set
    {
      Debug.WriteLine("Interne
tLed::LedState::set " + value.
ToString());
      if (LedControlGPIOPin !=
null)
      {
        GpioPinValue newValue
= (value == eLedState.
On ? GpioPinValue.High :
GpioPinValue.Low);
        LedControlGPIOPin.
Write(newValue);
        _LedState = value;
      }
    }
  }
```

Now add a part that allows us to blink the LED on or off. If the LED is on, it's switched off. If it's off, it gets switched on. Simple!

```
  public void Blink()
    {
      if (LedState == eLedState.
On)
      {
        LedState = eLedState.
Off;
      }
      else
      {
        LedState = eLedState.On;
      }
    }
```

We can now actually start using information from the internet. Start out by setting up a way to read a page on the web and then execute it. Once it's been read, output the returned string to the debug channel so you can see it while running the project.

Finally, we determine what the value of the delay is and return it for the rest of the code to use.

```
  const string
WebAPIURL
= "http://
adafruitsample.
azurewebsites.net/
TimeApi";
  public async Task<int>
GetBlinkDelayFromWeb()
  {
    Debug.WriteLine("Interne
tLed::MakeWebApiCall");
    string responseString = "No
response";
    try
    {
      using (HttpClient client =
new HttpClient())
      {
        responseString
= await client.
GetStringAsync(WebAPIURL);
        Debug.WriteLine(String.
Format("Response string: [{0}]",
responseString));
      }
    }
    catch (Exception e)
    {
      Debug.WriteLine(e.
Message);
    }
    int delay;
    if (!int.
TryParse(responseString, out
delay))
    {
      delay = DefaultBlinkDelay;
    }
    return delay;
    }
  }
}
```

Once all the code is entered, you can build the solution and run the code on your Pi. If you're having issues, the code listing has the full solution to check against.

This tutorial has been adapted from World Map of Makers by the Windows IoT team consisting of David Shoemaker, Anthony Ngu, and Aparajita Dutta, from the Microsoft Projects site: magpi.cc/2a5GaPD



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

TALKING LIGHT SENSOR

You'll Need

- 1× Solderless breadboard
- 1× Light sensor (photoresistor)
- 1× 560Ω resistor
- Male-to-female jumper wires
- Headphones or speaker (optional)

Create a customisable, talking light sensor that lets you know when the amount of light hitting it is just right

In the last two tutorials, we learnt how to use a button to turn on an LED, and then how to use a web API call to blink another LED. Let's bring both of these concepts together and make a device that takes user inputs from two dials (variable resistors) and a light sensor, and then uses that information to tell you a result using an online text-to-speech API. Let's play it bright, or not!

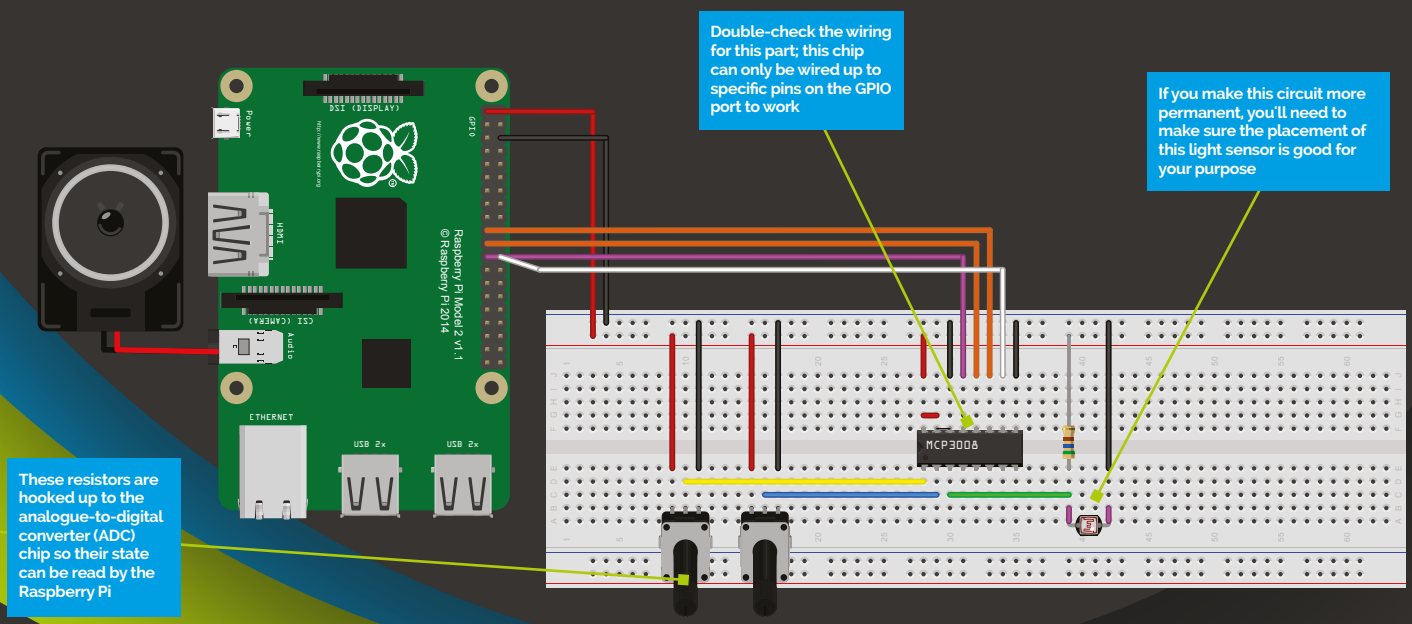
First of all, with your Raspberry Pi off, wire up the circuit as shown in the Fritzing diagram, paying careful attention to the orientation of the chip. Refer to **Fig 1** on the next page for notes on the orientation. Connect a speaker or headphones now if you want to use them.

The main code

As with the last tutorial, we'll download a project to work from. Once again, download everything from here: **magpi.cc/2a9OEW1**.

Open up **Lesson_204.sln** in Visual Studio and open the **mainpage.xaml.cs** file. We need to do a bit more editing to get it working, so we'll run over that first.

Add the following lines to the top of the MainPage class, right after the { bracket. First, we need to let the code know how much voltage the chip is getting:



Language
>C#

FILENAME:
MainPage.xaml.cs

DOWNLOAD:
magpi.cc/2ac38rr



Credit: Iainf

The variable resistor, or potentiometer, is a voltage divider that, when twisted, allows for an output that has a variable voltage

```
  const float ReferenceVoltage =
5.0F;
```

Then we tell it what the channels will be detecting. The third line is for the light sensor:

```
  const byte
LowPotentiometerADCChannel = 0;
  const byte
HighPotentiometerADCChannel =
1;
  const byte CDSADCChannel = 2;
```

Here's the text strings we want to use for when the Pi talks to us:

```
  const string
JustRightLightString = "Ah,
just right";
  const string LowLightString =
"I need a light";
  const string HighLightString
= "I need to wear shades";
```

This part will help record the state of the light sensor:

```
  enum eState { unknown,
JustRight, TooBright, TooDark};
  eState CurrentState = eState.
unknown;
```

We now tell the code which ADC chip we're using:

```
  MCP3008 mcp3008 = new
MCP3008(ReferenceVoltage);
```

We're going to connect to the speech-to-text part of the Windows 10 Cognitive services, so we add:

```
  private SpeechSynthesizer
synthesizer;
```

And finally, we add a little bit so we can check the ADC at a set interval:

```
  public Timer timer;
```

With the start of the class filled out, we can now fill out some of the MainPage constructor (public MainPage()), right after this. InitializeComponent();. Here we add a line to start the speech synthesiser and another to initialise the ADC:

```
  synthesizer = new
SpeechSynthesizer();
  mcp3008.Initialize();
```

Now we can set up a timer in the OnNavigateTo method. After Debug.WriteLine("MainPage::OnNavigatedTo");, add the following:



"A Raspberry Pi that talks!"

```
  timer = new
Timer(timerCallback, this, 0,
1000);
```

Now we have the timer callback being called, let's fill it out. After the } following 'return;', add the following lines. Let's set the new light state and assume it's just right to start:

```
  eState newState = eState.
JustRight;
```

Next, we want to read the values of the dials from the ADC, so we use:

```
  int lowPotReadVal = mcp3008.
ReadADC(LowPotentiometerADCCha
nnel);
  int highPotReadVal = mcp3008.
ReadADC(HighPotentiometerADCCh
annel);
  int cdsReadVal = mcp3008.
ReadADC(CDSADCChannel);
```

Convert these readings into voltages to make them easier to use:

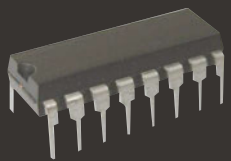
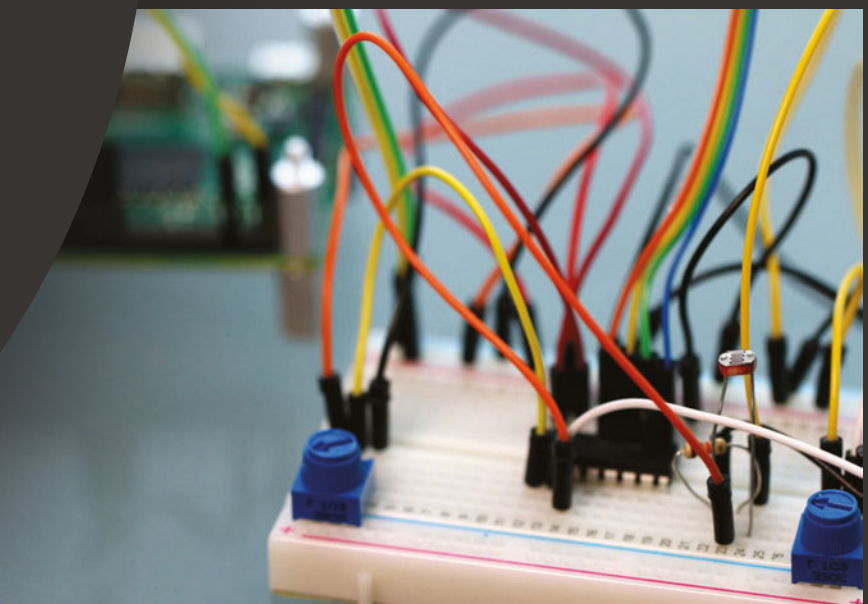


Fig 1 The notch on the chip denotes that this is where the 'front' of the chip is. In the Fritzing diagram, this notch is on the left side of the chip



The setup may end up looking a bit messy once you're done, but make use of colour-coding so you know what wire is related to what part of the system

PRACTICAL USES

How could such a project be used in the real world? Well, making sure some flowers are kept in proper sunlight may be useful for certain botanists. You could also place it near your TV to let you know when it's time to shut the curtains to avoid massive screen glare!



```
  float lowPotVoltage = mcp3008.
ADCToVoltage(lowPotReadVal);
  float highPotVoltage = mcp3008.
ADCToVoltage(highPotReadVal);
  float cdsVoltage = mcp3008.
ADCToVoltage(cdsReadVal);
```

Next, we'll make sure we can read the values when the code is running from the debug:

```
  Debug.WriteLine(String.
Format("Read values {0}, {1},
{2} ",
    lowPotReadVal,
highPotReadVal, cdsReadVal));
  Debug.WriteLine(String.
Format("Voltages {0}, {1}, {2}
",
    lowPotVoltage,
highPotVoltage, cdsVoltage));
```

We now check if the state of the light is below the values set by the variable resistors:

```
  if (cdsVoltage < lowPotVoltage)
  {
    newState = eState.TooDark;
  }
```

Next in the chain is to see if it was too high:

```
  if (cdsVoltage >
highPotVoltage)
  {
    newState = eState.TooBright;
  }
```

And then we let the code know to use a different method, to determine what to do with the state of the light:

```
  await
CheckForStateChange(newState);
```

The CheckForStateChange code has been mostly completed in this example, but you'll need to add the following under 'use another method to wrap the speech synthesis functionality' to do just that:

```
  await TextToSpeech(whatToSay);
```

Now we can add the part that lets the Pi talk to us! Under async, where it says 'insert code', insert the following code which first converts text input into a stream, and then plays the audio:

```
  SpeechSynthesisStream
synthesisStream;
  synthesisStream = await
synthesizer.SynthesizeTextToStr
eamAsync(textToSpeak);

  media.AutoPlay = true;
  media.
SetSource(synthesisStream,
synthesisStream.ContentType);
  media.Play();
```

Coding the ADC

Now we're done with the MainPage code, it's time to code up the ADC. Open up **MCP3008.cs** from the project file.

We need to start by creating an initialising method to set up communication with the SPI bus the ADC chip runs on. Just after the line 'Debug.WriteLin e("MCP3008::Initialize");' add the following, starting with the configuration of the SPI bus and telling it the ADC's clock speed:

```
  try
  {
    var settings = new
SpiConnectionSettings(SPI_CHIP_
SELECT_LINE);
    settings.ClockFrequency =
3600000;
    settings.Mode = SpiMode.
Mode0;
```

This line will return all SPI devices on the system:

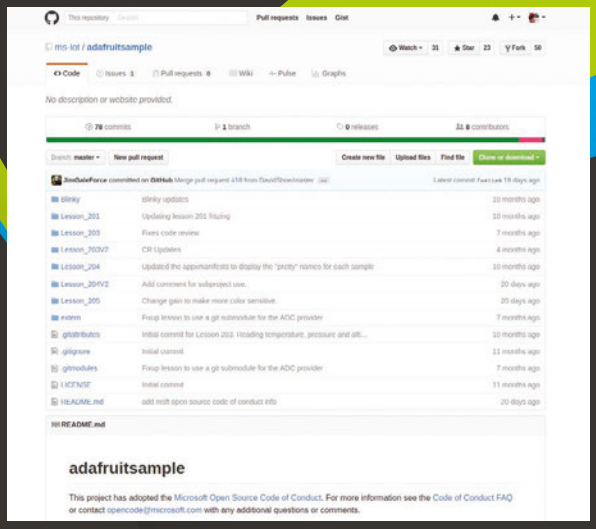
```
    string aqs = SpiDevice.
GetDeviceSelector();
```

This line finds the SPI bus controller from the previous line:

Language
>C#

FILENAME:
MCP3008.cs

DOWNLOAD:
magpi.cc/2agTxOR



There's a whole host of other tutorials on GitHub where this project lies; give them a look if you want to try out more!

```
    var dis = await
DeviceInformation.
FindAllAsync(aqs);
```

And the rest of the code creates an SPI device using all the information we just grabbed:

```
    mcp3008 = await SpiDevice.
FromIdAsync(dis[0].Id, settings);

    if (mcp3008 == null)
    {
      Debug.WriteLine(
     "SPI Controller {0} is
currently in use by another
application. Please ensure that
no other applications are using
SPI.",
      dis[0].Id);
      return;
    }

  }
  catch (Exception e)
  {
    Debug.WriteLine("Exception:
" + e.Message + "\n" +
e.StackTrace);
    throw;
  }
```

Now we have the information for the chip, we're able to add code so we can read it:

```
  public int ReadADC(byte
whichChannel)
  {
    byte command =
whichChannel;
    command |= MCP3008_
SingleEnded;
    command <<= 4;

    byte[] commandBuf = new
byte[] { 0x01, command, 0x00 };

    byte[] readBuf = new byte[]
{ 0x00, 0x00, 0x00 };

    mcp3008.
TransferFullDuplex(commandBuf,
readBuf);

    int sample = readBuf[2] +
((readBuf[1] & 0x03) << 8);
    int s2 = sample & 0x3FF;
    Debug.Assert(sample == s2);
    return sample;
  }
```

Remember when we were turning the input into a voltage so it was easier to read? We need to add a helper method here to work with that:

```
  public float ADCToVoltage(int
adc)
  {
    return (float)adc *
ReferenceVoltage / (float)Max;
  }
```

It's game time

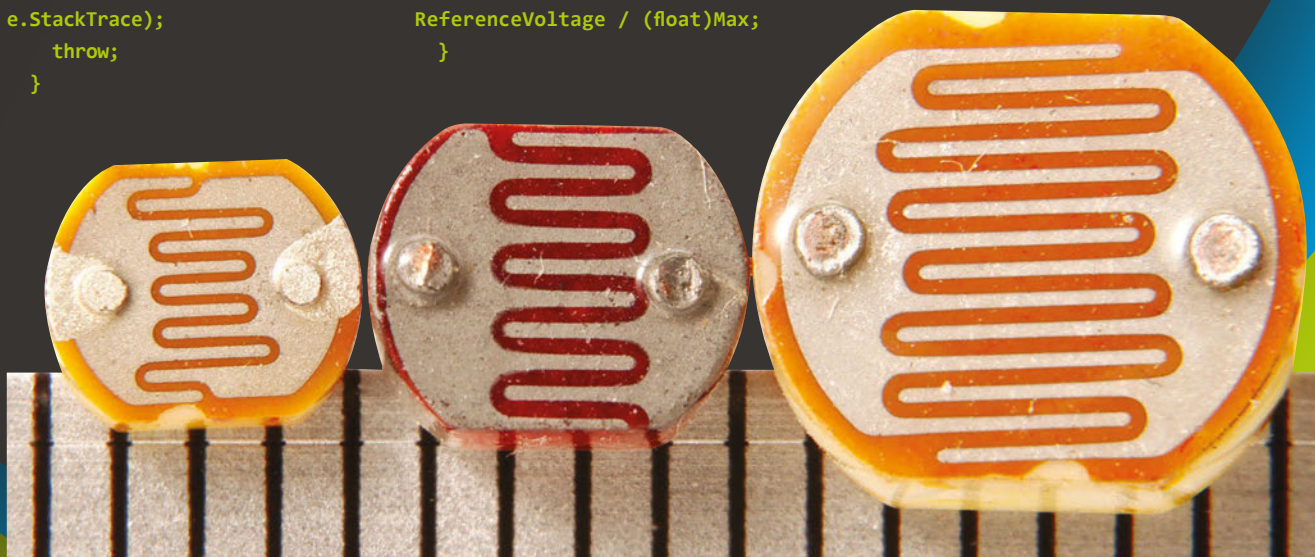
Your project is ready! Run the code with the setup in a normally lit area. The output window will show you the voltages read from the ADC: the lower-boundary resistor, the higher-boundary resistor, and the light sensor. It should look something like this:

```
  MainPage::timerCallback
  Read values 211, 324, 179
  Voltages 1.031281, 1.583578,
0.8748778
```

Turn the lower-boundary potentiometer, watching the value of the first number change. Adjust it so it's just a touch lower than the light sensor reading. Do the same with the other resistor, turning it to be just above the light sensor. This is now our 'just right' zone of light.

Now, when you put your hand over the light sensor, it will print in the output window 'I need a light'; it will also say it to you if you have speakers or a headphone connected. Experiment with uncovering and shining a light on it to see just how sensitive it is.

This tutorial has been adapted from Bright or Not? by the Windows IoT team consisting of David Shoemaker, Anthony Ngu, and Aparajita Dutta, from the Microsoft Projects site: magpi.cc/2abUvx0



The light sensor, or photoresistor, changes resistance depending on the amount of light hitting it. Like the variable resistor, this changes the voltage at the output

pi-top

LEARN
PLAY
CREATE

WWW.PI-TOP.COM



pi-topCEED

- Adjustable Viewing Angles
- 14" HD Screen
- Modular Components

$99.99
without Raspberry Pi
excluding VAT

pi-top

- 10 Hour Battery Life
- 13.3" HD Screen
- Modular Components

$264.99
without Raspberry Pi
excluding VAT



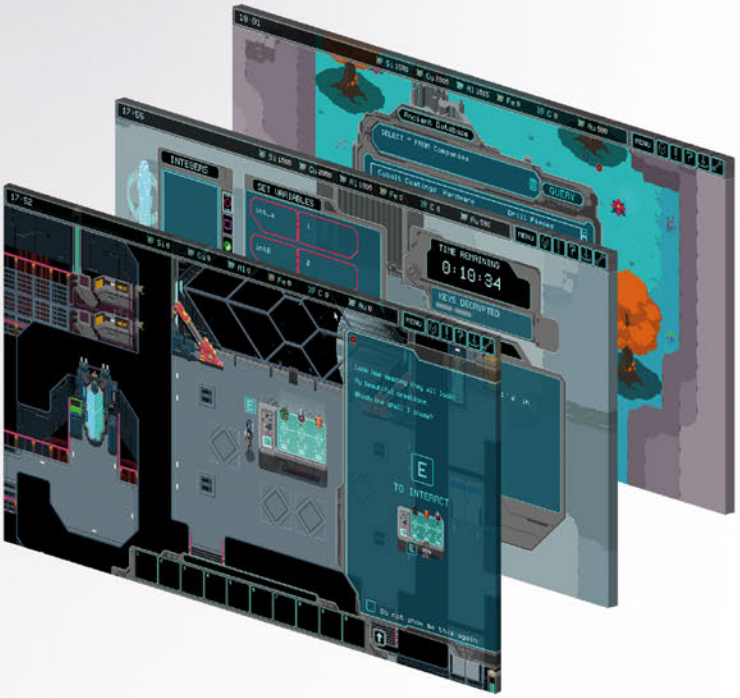
Available in green or grey colours

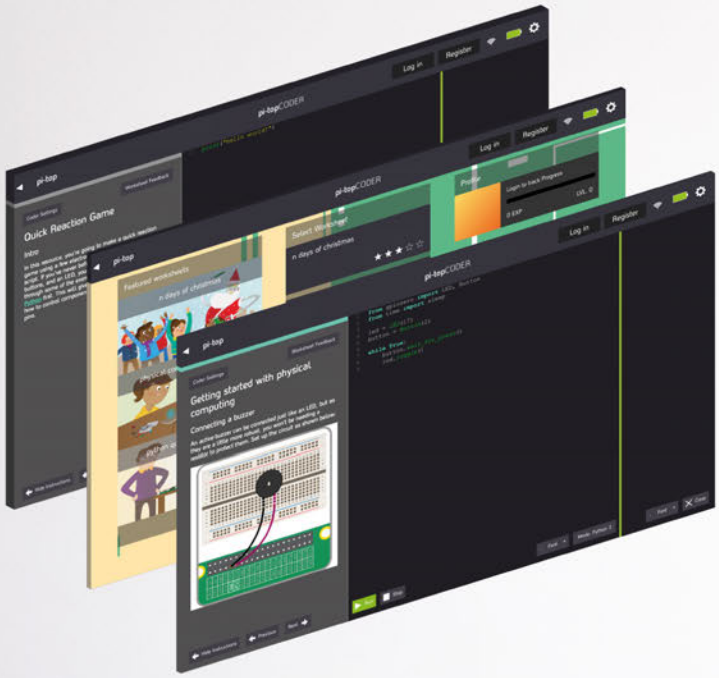
pi-top | Worldwide shipping available in green or grey at www.pi-top.com
Stay up to date with our latest news by following our social media

CEED**universe**

CEED**universe** is a world of fantasy grounded in computing reality! After crash landing on a strange new planet you will first encounter 'drag and drop' coding puzzles that improve your computational thinking skills.

Discover more about the planet you've landed on and the civilisation that used to inhabit it while learning harder and more complex concepts in fun and engaging ways! Before long you'll be writing your own conditional statements, loops and functions. Find out more about CEED**universe** at www.pi-top.com





pi-topCODER

Influenced by the workflow of makers and hackers, pi-top presents **pi-top**CODER - an integrated code editor which allows you to learn, write and test code all in one view. With intuitive syntax highlighting, dynamic views and customizable themes it makes for a versatile learning tool for your projects.

pi-topCODER also has every Raspberry Pi Foundation lesson plan created and will track and save your progress as you go through dozens of fun hardware and software projects.



pi-topPROTO

pi-topPROTO is a HAT compatible Add-on Board for your **pi-top** or **pi-top**CEED that allows you to prototype electronics. Create a Weather Station, HAM Radio, Heart Rate Monitor, or integrate any Arduino based maker kits into your own Raspberry Pi compatible prototyping board!



pi-topSPEAKER

Give a voice to your **pi-top** device with **pi-top**SPEAKER!

- Modular design, attach up to three in a row to give true stereo sound.
- 2W per module
- Left, Right and Mono mix selection
- High quality SPDIF digital audio from HDMI
- I²C controlled

Our Official Distributors
adafruit
HANDS-ON SCIENCE
KIWI electronics
KUBII
MODMYPI
MICRO CENTER computers & electronics
Pi-Shop.ch
RS
sparkfun ELECTRONICS



INGMAR STAPEL

Ingmar has been building Pi-powered robot cars since 2012. He's also working on a security robot for the home.
custom-build-robots.com

Quick Facts

- The Discoverer took nearly six months to build
- There's a 350m range for remote control
- It uses a Navilock NL-602U GPS receiver
- The original chassis was a smaller IKEA box
- The pan-and-tilt kit features two mini-servos

DISCOVERER
METAL-DETECTOR ROBOT

This smart Raspberry Pi robot is equipped with a metal detector, along with GPS tracking and a pan-and-tilt camera

We've seen all sorts of Pi-powered robots in our time, but the Discoverer is the first with a built-in metal detector. Mounted on a PVC pipe arm in front of the four-wheeled robot, the detector emits a beep whenever it passes over a metallic object. Prolific robot maker Ingmar Stapel, from Munich, came up with the idea after watching a TV show about people trying to find gold with a sophisticated metal detector. "I was immediately inspired to build my own affordable robot-car with a metal detector in order to discover some treasure in my garden," he tells us.

The basis for the new robot was his previous cardboard car, which required some major adaptations. "First, I had to look for a different chassis that is also suitable for outdoor usage. Second, I needed a metal detector that fits to the robot-car and the Raspberry Pi to ensure remote-controlled treasure hunting."

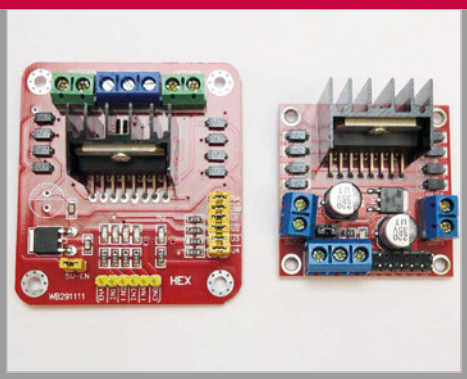
For the chassis, he used a plastic storage box to contain all the electronics, adding PVC piping around the exterior to hold a pan-and-tilt camera and the metal detector. At first, Ingmar tried using a cheap electric-cable detector from a DIY store, but its coil was too small to detect metal in the ground. He then came across the Gary's Pulse-AV metal detector (**magpi.cc/1XdKBeK**). "With support from Gary, I was able connect the metal detector to the Raspberry Pi and to get everything working." This involved using a step-down converter to change the detector's 12V output to 3.3V for a GPIO pin on the Pi.

As Ingmar wanted the Discoverer to stream live video to a laptop, it would need a camera. After mounting a Pi Camera Module on the front of the chassis, he found the angle of view was too limited. "I bought the pan-and-tilt kit

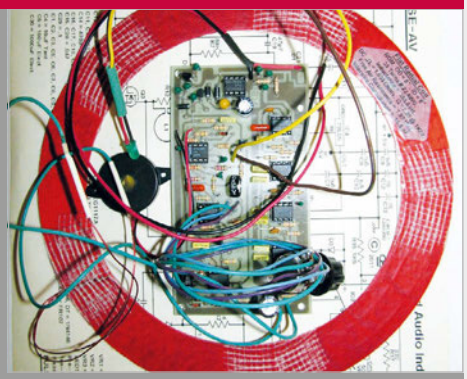
BUILDING A METAL-DETECTING ROBOT



>STEP-01
Chassis and wheels
The chassis is a standard plastic storage box bought from a DIY store. Four motors are attached to the bottom of the box using wall mounts for PVC pipes.



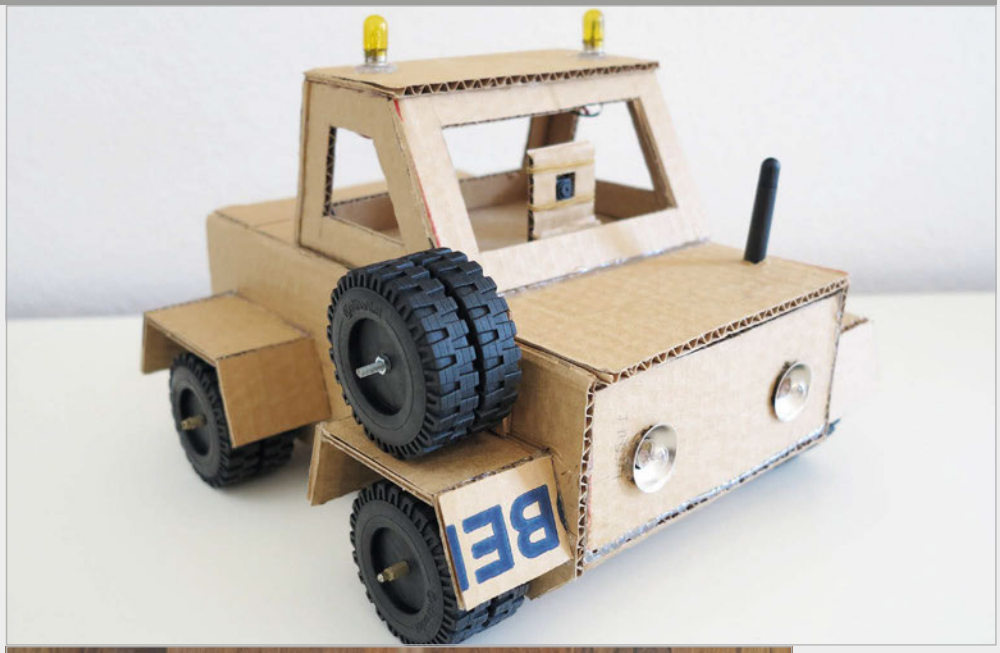
>STEP-02
Motor drivers
Standard L298N H-bridge motor controllers are used to trigger the DC motors. Ingmar plans to replace the latter with worm-gear motors and also use larger wheels.

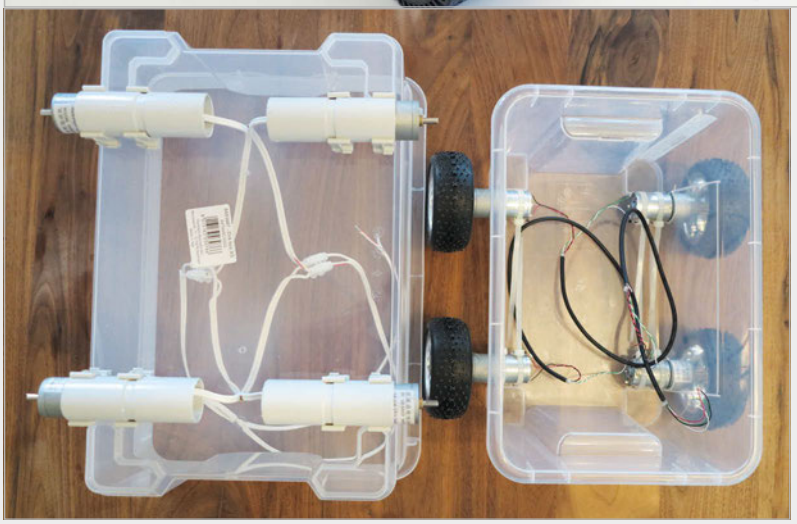


>STEP-03
Metal detector kit
A Gary's Pulse-AV PI detector kit (**magpi.cc/1XdKBeK**) is used for the metal detector part, which is mounted fairly low to the ground on a PVC pipe arm.

and mounted it in the middle of the robot and I got a much better overview." Some trial and error was involved in getting the live video streaming to work with very low latency, since even the tiniest delay would make remote control difficult. In the end, he used MJPG-streamer (**magpi.cc/24F96k6**) with some patches to get it to run on the Pi 3: "The video is very fast with a very low latency… it is even possible to make some robot-car races in our apartment."

While originally just remote-controlled, the Discoverer is now able to move autonomously using a GPS receiver, Sense HAT (for the compass), and a Python program Ingmar had already developed. "It is still in a beta version, but it is already able to import a KML file with GPS waypoints generated via Google Earth. Imported into my Python program, the robot-car is able to drive from one GPS waypoint to the next." When it's finalised, he plans to include a manual in his first book, due to be published (in German) this autumn. While the Discoverer is mainly for fun treasure hunting, Ingmar thinks it could also have some serious applications, such as detecting mines in war zones.





Above The Discoverer is loosely based on Ingmar's earlier cardboard car (**magpi.cc/1VQv8QJ**), which has live video streaming

Left Unable to fit all the components into the original IKEA plastic storage box, Ingmar replaced it with a larger one



TIM MAIER

Tim is currently studying for a Bachelor of Information Technology degree at Queensland University of Technology, with plans to major in Computer Science.
magpi.cc/29Burv1

Quick Facts

- The project cost $500 AUD (£280) to build
- It currently covers 2-3km per charge
- The board can currently reach up to 15km/h
- Tim plans a battery and ESC upgrade
- Build recipe is available on GitHub

MOTORISED SKATEBOARD

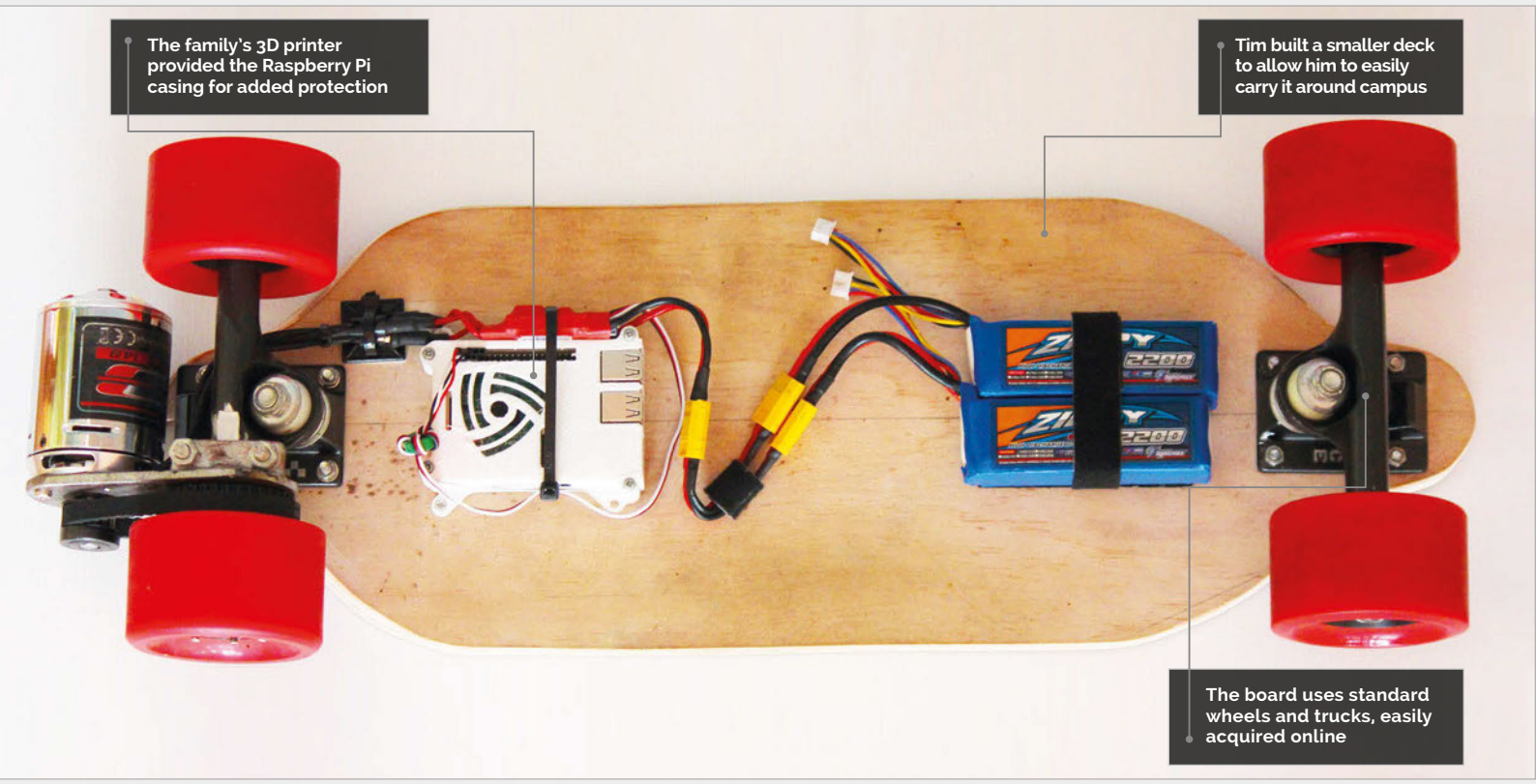
A university assignment allowed one student the chance to realise his dream of building the latest in whizzy commuter gadgets

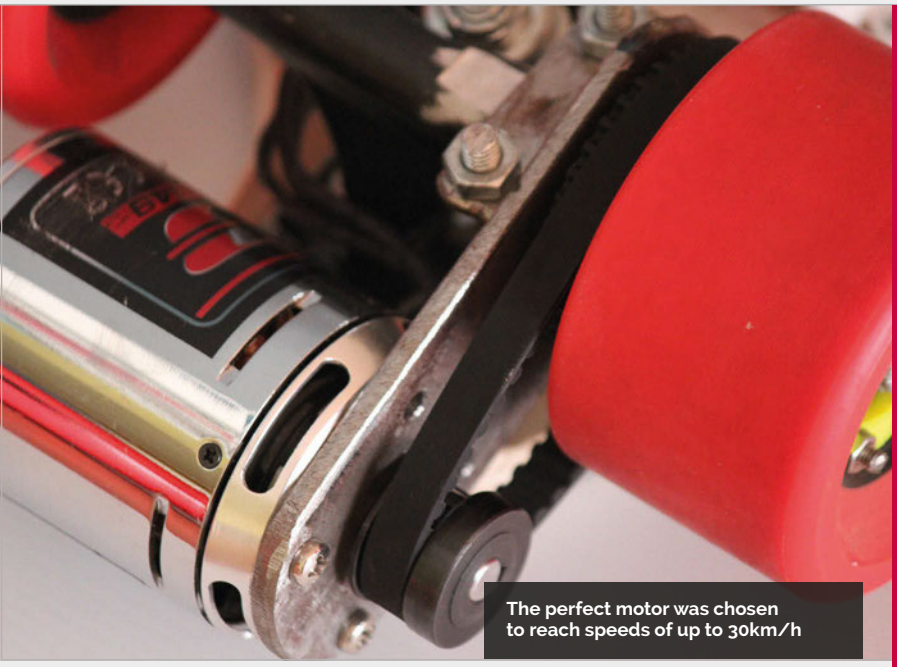
By now we're sure you're aware that e-boards have officially become a 'thing'. From knee-driven mini-Segways and two-wheeled 'hoverboards' to standard motorised decks, the streets are filled with wheeled commuters. And while Marty McFly may have failed to deliver on the true hoverboard of our dreams, search online for an electric skateboard and you'll find the next best thing, albeit with a hefty price tag.

So when Queensland University of Technology student Tim Maier was assigned with the task of 'building something with a Raspberry Pi', he already knew what he planned to create.

"Building an electric skateboard had been something on my mind for some time, as buying one was not a viable option," Tim explains, when we ask him if he had considered any other directions for his project. "So when we were told about the task, it all kind of linked up and I started to do my research on what to buy."

With a few requirements in mind, Tim started researching the perfect motor. He wanted to achieve an average speed of 30km/h to aid his commute, and knew the motor would easily





The perfect motor was chosen to reach speeds of up to 30km/h

be one of the most expensive components of the build. Finally, he decided upon a Turnigy Aerodrive SK3, matching it with two 2200mAh lithium polymer (LiPo) batteries and a basic electronic speed control (ESC).

Despite having to rely on YouTube and assorted literature to educate him on how to utilise Python, the biggest hurdle for Tim turned out to be the drive system. "Finding a way to attach the motor mount to the skateboard truck was a huge fiddle." He ended up creating a makeshift U-bolt system, though he plans to upgrade the mounting layout when attaching a new ESC.

> "The motor itself is controlled by a Pi and Wii Remote (a Wiimote to those in the know)"

The motor itself is controlled by a Pi and Wii Remote (a Wiimote to those in the know). Holding the '1' and '2' buttons will connect the Wiimote to the Pi. The 'B' button activates the motor, while 'up' and 'down' on the D-pad control speed.

Upon completing the build, Tim has been met with thousands of YouTube views and calls for how-to guides and board sales. Spurred by the positive response, he's provided the code and kit list on GitHub (**magpi.cc/29Burv1**), and plans to also create an instructional video of an upgraded design for anyone wanting to make their own. The Pi Skate 2.0 will house more batteries for longevity, a higher-quality ESC for greater speed and the ability to brake, and possibly LED lights because, well, why not? And as for the Wiimote? Tim hopes to move the board's control system to a mobile phone or smartwatch, thereby reducing the bulk of the console controller.

Of his future in the digital making industry, Tim suggests, "I will most probably venture into a career within the computing and engineering field, but haven't really thought too much about what area I will specialise in. I'm keeping my options open as it's such a broad field." In the meantime, his continued experimentation with Raspberry Pi will lead to further YouTube videos as and when he builds projects.

TAKE TO THE STREETS



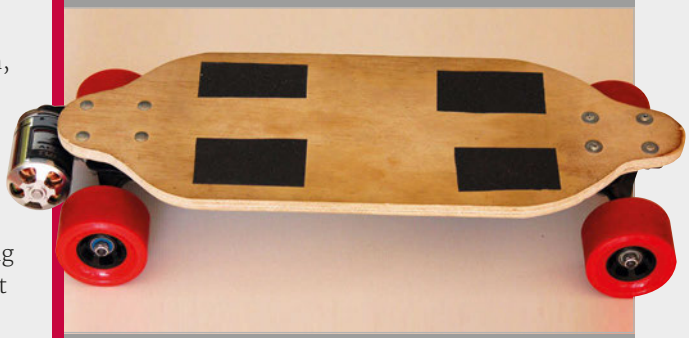
>STEP-01
The power

Two LiPo batteries are connected in parallel to the ESC, taking approximately an hour each to charge. They power both the motors and the Raspberry Pi.



>STEP-02
The control

The ESC runs 5V power to the GPIO pins, which powers the Raspberry Pi. This is where control from the Wiimote converts to a pulse-width modulation signal, sending it back to the ESC.



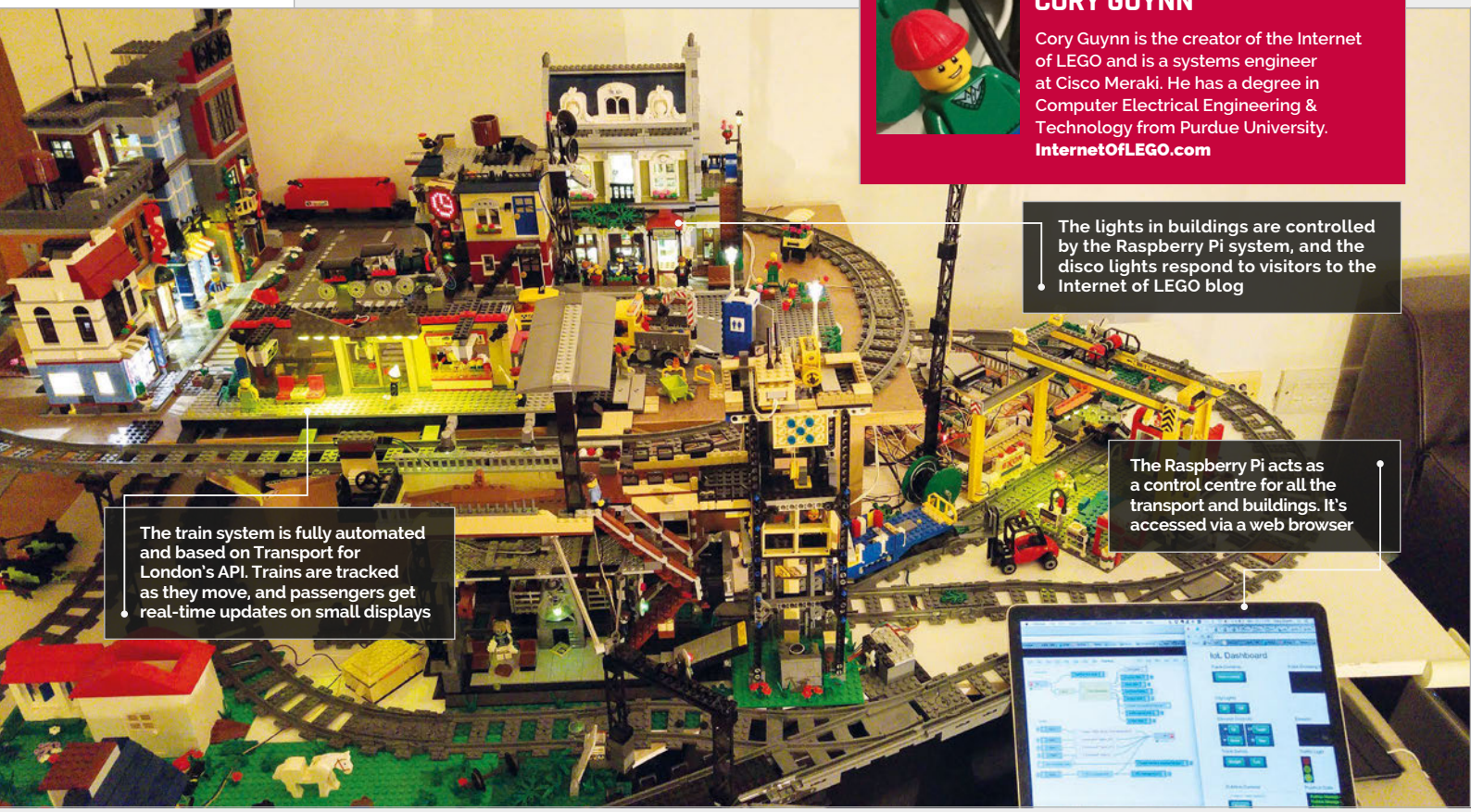
>STEP-03
The launch

The ESC then signals to the motor to go-go-go, and you're on your way. Make sure to keep balanced as you're zooming along!



CORY GUYNN

Cory Guynn is the creator of the Internet of LEGO and is a systems engineer at Cisco Meraki. He has a degree in Computer Electrical Engineering & Technology from Purdue University. **InternetOfLEGO.com**



Quick Facts

- Around 20,000 LEGO bricks were used
- It took one-and-a-half years to build
- Over 46,000 people have interacted with the blog
- 17 LEGO kits were used
- Most LEGO kits were from the Creator Expert series

THE INTERNET OF LEGO

One maker is taking a love of LEGO to a whole new level with Raspberry Pi. Take a tour of this incredible internet-connected cityscape being built brick by brick

We love LEGO and the internet, so what could be finer than this 'Internet of LEGO' project? Well, discovering the Raspberry Pi serving as its brain, and catching up with its maker to learn all about this amazing connected city.

"The Internet of LEGO is a living project where I set out to learn everything about the Internet of Things," says Cory Guynn. Packed with sensors, the city reports to a Raspberry Pi that acts as its brain. Cory has used this to create the Internet of LEGO blog (**internetoflego.com**).

"I grew up playing with LEGO bricks and model trains, which taught me about construction and electronics, and allowed me to be creative. The use of LEGO also allows me to represent a city or build prototype systems easily. Plus, it gives me an excuse to buy a bunch of LEGO bricks in my thirties," laughs Cory.

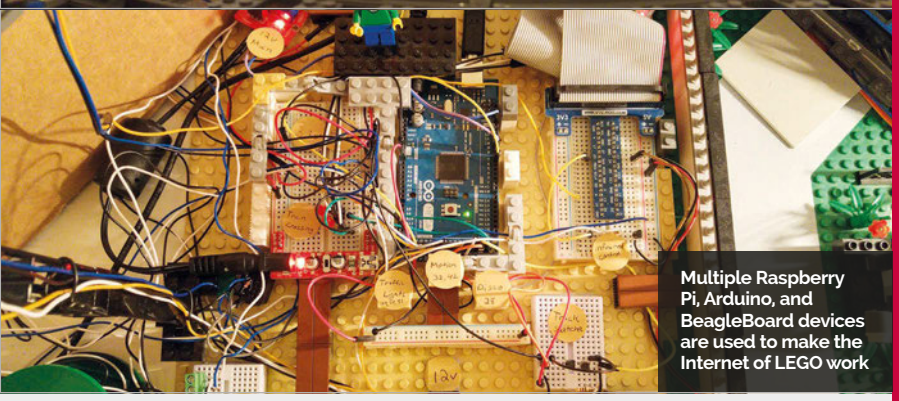
"A Raspberry Pi Model B+ is the heart of my city," he reveals, "and that was the starting point of the project." The Raspberry Pi is attached to Arduino boards that control most of the GPIO operations. Cory also uses Cactus Micro Rev2, BlueDuino, WeMos and NodeMCU boards, along with a Wio Link and BeagleBone Green.

The city itself is complex, with many structures and buildings hooked up to a huge range of sensors: RFID, ultrasonic proximity, infrared, and magnetic reed switches are used to keep track of the city environment.

The train system is Cory's favourite part of the city. "I love seeing things in motion," he reveals. "There are several things that I've been able to do that make for a dynamic environment." Cory has built a train scheduling



We love the detail of the Internet of LEGO city, with citizens going about their daily lives



Multiple Raspberry Pi, Arduino, and BeagleBoard devices are used to make the Internet of LEGO work

system using the Transport for London API. This system displays the schedule on an OLED screen and switches to the train track to match the destination.

The trains are controlled by WiFi and an infrared transmitter attached to a tower. Infrared sensors are used to detect incoming trains and trigger a crossing signal (with a servo controlling the arm, and LEDs for lights).

"Everything is connected to an Arduino Mega, which is then USB-tethered to the Raspberry Pi," explains Cory.

He has developed a huge amount of software to control all the structures in the Internet of LEGO. "The Johnny-five.io robotics framework made programming sensors and servos easy. I wrote a few projects, including a train crossing, train track switch, elevator, motion detector, and city lights."

Most of the orchestration of the city and its many sensors is handled by Node-RED. "This system allows me to add inputs and outputs from any of my projects that can interact with one another," says Cory. "For example, when somebody visits my blog, a REST call to Node-RED triggers an MQTT message to my disco lights, which turns on strobing LEDs in a LEGO club and also triggers the Palace Cinema marquee lights for five seconds."

> "Cory has developed a huge amount of software to control all the structures in the Internet of LEGO"

Cory tells us he is indebted to the open-source community, which has allowed him to learn so much about programming and hardware. "I hope to give back to the community through my projects and experience along the way."

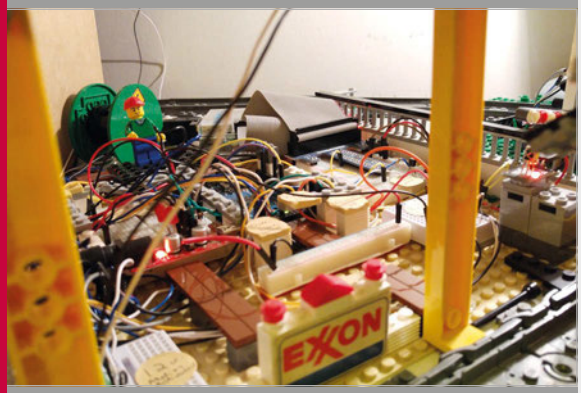
CITY PLANNING



>STEP-01

Regular LEGO

The Internet of LEGO is a Raspberry Pi-powered city built from regular LEGO bricks and kits. These are then given Internet of Things-style smarts, thanks to various electronic components.



>STEP-02

Raspberry and Arduino

A Raspberry Pi controls two Arduino boards using an MQTT broker. The Arduinos are assigned to GPIO operations, while a second Pi acts as an MQTT cluster member and runs Node.js automation scripts.



>STEP-03

A working city

The Internet of LEGO is fully automated, with reed sensors used to detect train presence. Timetable information is based on London API data. When people visit the blog, the Palace Cinema lights up and the disco starts working.



DJORDJE UNGAR

Djordje Ungar is a programmer by day and a digital alchemist by night. He is a hobby artist, animator, musician, game maker, hacker, and tinkerer.
djordjeungar.com

HAL 9000

"Open the pod bay doors, HAL" are chilling words to anybody who has watched *2001: A Space Odyssey*, apart from one maker who decided it would be a good idea to build HAL 9000 for real

Quick Facts

- HAL stands for 'Heuristically programmed ALgorithmic computer'
- HAL is rumoured to be IBM with each letter shifted one backward
- The total cost for the build was $99 (including the Raspberry Pi)
- All the parts are off-the-shelf computer components
- The case is laser-cut acrylic covered in black paint



The case is made from 3mm black acrylic laser-cut into the right shape for HAL. It is spray-painted for a professional finish

A stripped-down web camera is fitted with a camera lens. This completes the HAL 9000 look, and the web camera also provides the device with a microphone

A USB speaker is fitted inside the device near the grille at the bottom. Jasper software provides HAL with a voice

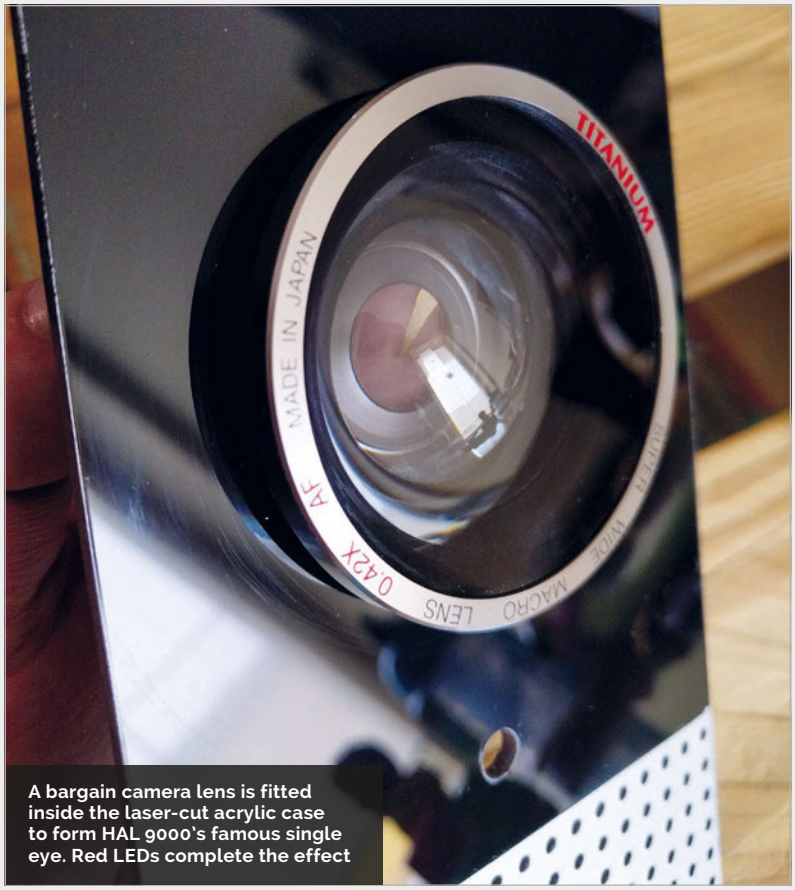
"I had this Raspberry PI Model B waiting to become something great," says Djordje Ungar, "and what greater thing can a computer hope for than to become the iconic computer from Stanley Kubrick's *2001: A Space Odyssey*? I mean, come on!

"The first time I heard synthesized speech, I thought to myself: 'Wow, how cool would it be if that was a voice of HAL 9000?' It was when I stumbled upon Jasper, this amazing open-source project that allows you to control a computer with your voice, that I knew I had to make HAL."

Aside from the Raspberry Pi, all the other parts are off-the-shelf computer components that you can buy online. "I used a 3mm thick black acrylic, and I've painted some parts to look metallic," Djordje reveals. "The box is 300 × 96 × 62mm, which is a bit larger than a tall carton of milk.

"I examined a dozen movie stills from the film," he continues. "I was only able to guess the actual size of the original HAL, so I based it off the lens. Some things like the number of holes on the speaker panel and the logo are spot on."

To allow HAL 9000 to see, Djordje added a webcam and camera lens. "I wanted to find a super convex lens like the one from the movie, but lenses of that calibre are anything but cheap. Even the used ones were way too expensive. So I had to settle for



A bargain camera lens is fitted inside the laser-cut acrylic case to form HAL 9000's famous single eye. Red LEDs complete the effect

a cheaper, non-convex lens." Luckily, he found a used one that he liked, via local classified advertisements: a Titanium Super Wide Lens 0.42x AF.

"The lens is mostly for style, to give the whole build a more polished look, but it does also expand the field of view of the webcam a little.

"The truth is: I didn't need the webcam. I only needed the microphone. But since they come bundled together, having a webcam was a plus. I removed the casing and left the circuit board exposed, to make it as compact as possible. Then I glued it onto the ring that goes on the back of the lens."

For HAL 9000's voice and brain, Djordje fitted a speaker and a Raspberry Pi inside the case. "My only criterion for the speaker was that it should be small and fit inside the case," he tells us. The brain is the Raspberry Pi running the Jasper project (**magpi.cc/29iKyxS**).

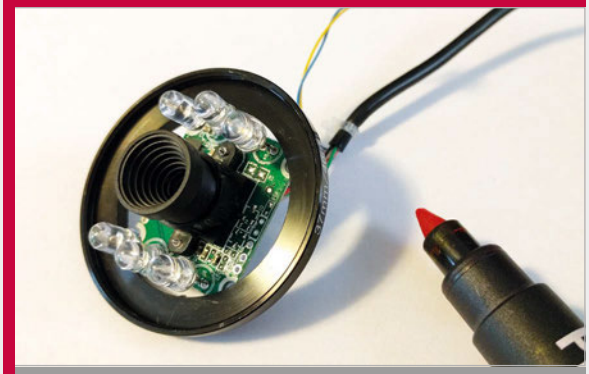
"The Jasper project offers a couple of TTS (Text-To-Speech) engines," explains Djordje. "I had a couple of options, but I chose eSpeak (**magpi.cc/29iKtdt**) since I'm most familiar with it."

> "It works as a voice assistant, but it can play any WAV files, including the classic lines from the movie"

It works as a voice assistant, but it can play any WAV files, including the classic lines from the movie. "I wrote a simple Jasper module that will play random WAV files."

Djordje's friends are impressed by the project: "In a world where you can talk to your phone's digital assistants, or ask Google a question with your voice, no one is blown away by the fact that HAL responds to voice commands, but everyone loved the case and wanted one."

BUILDING A HAL 9000



>STEP-01
Stripping the webcam

A recycled Insten USB digital six-LED webcam is stripped down. A red marker is used to paint the LEDs red, giving the device's eye the same ominous glow HAL has in the movie.



>STEP-02
Stuffing the case

The case is laser-cut from an acrylic sheet and glued together. The lens is fitted through a round hole in the front face and the webcam fitted inside. This provides HAL 9000 with a camera and, more importantly, a microphone.



>STEP-03
Assembled HAL

All the components are fitted inside, and the Raspberry Pi provides the software for human interaction. A speaker is fitted to the bottom of the device to provide the voice.

The MagPi ESSENTIALS

LEARN | CODE | MAKE

AVAILABLE NOW:

- > CONQUER THE COMMAND LINE
- > EXPERIMENT WITH SENSE HAT
- > MAKE GAMES WITH PYTHON
- > CODE MUSIC WITH SONIC PI



The MagPi ESSENTIALS

From the makers of the official Raspberry Pi magazine

OUT NOW IN PRINT

ONLY **£3.99**

from

raspberrypi.org/magpi



GET THEM DIGITALLY:

Available on the App Store

GET IT ON Google Play



THE HAYLER-GOODALLS

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

RGB LED TWEET-O-METER

You'll Need

- RGB LED
- Breadboard
- Jumper wires
- 3× 100 ohm resistors
- Twitter developer account
- TextBlob Python library
- Twython Python library

Use the GPIO Zero Python library to control an RGB LED and see how well your tweets are doing

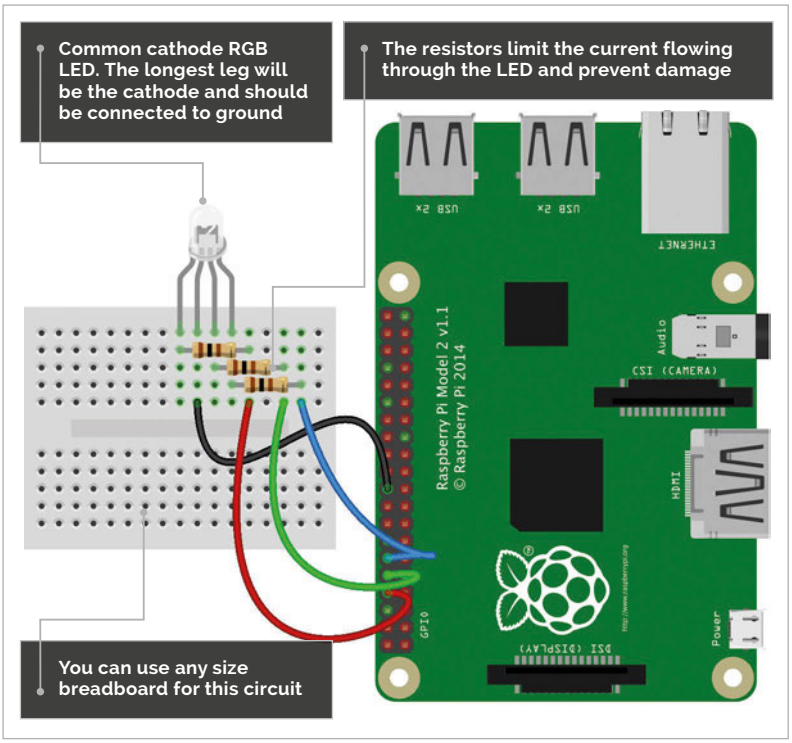
Keeping up to date with Twitter can be very time-consuming, especially if there are lots of tweets. What if you could see at a glance what the Twittersphere thinks about a certain topic? In this tutorial we're going to build a simple RGB LED circuit, and program it to change colour to indicate whether the tweets that include a given hashtag or keyword are using positive, negative or generally neutral language.

>STEP-01

Install Python libraries

Update your Pi to the latest version of Raspbian and download and install the additional software you'll need.

```
sudo pip3 install twython textblob
```



There are two libraries that make our project really easy. Twython allows you to contact Twitter using Python and collect tweets (you'll need to register for a Python developer account – see step 5). Then, to read the tweets in the code, we're going to use TextBlob; there are other libraries available, but this is one of the simplest.

>STEP-02

Do you like sausages?

Let's take a look at a simple example. Open a Python 3 interpreter (either use the command line or IDLE) and type:

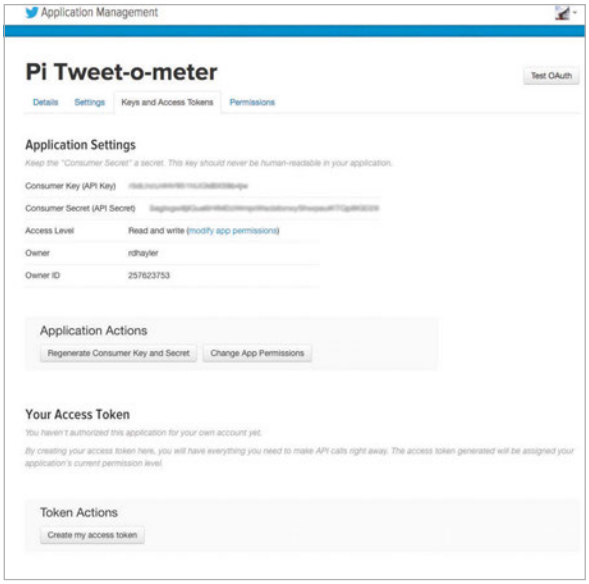
```
>>> from textblob import TextBlob
>>> sentence = TextBlob('I really like
sausages, they are great')
>>> sentence.sentiment.polarity
0.5
```

Any value for polarity greater than 1 indicates a positive sentiment (like); a value less than 1 suggests negative sentiment (dislike). Try changing the sentence and see how a different phrase will give a different result. Results will be more accurate if you have more text, although a 140-character tweet is normally good enough.

>STEP-03

Select your RGB LED

Light-emitting diodes (LEDs) are cool. Literally. Unlike a normal incandescent bulb which has a hot filament, LEDs produce light solely by the movement of electrons in a semiconductor material. An RGB LED has three single-colour LEDs combined in one package. By varying the brightness of each component, you can produce a range of colours, just like mixing paint. There are two main types of RGB LEDs: common anode and common cathode. We're going to use common cathode.



Above Once you've created your Twitter app, generate the required access tokens; copy these into your code

>STEP-04

Connect up the RGB LED

LEDs need to be connected the correct way round. For a common cathode RGB LED, you have a single ground wire and three anodes, one for each colour. To drive these from a Raspberry Pi, connect each anode to a GPIO pin via a current-limiting resistor. When one or more of these pins is set to HIGH (3.3V), the LED will light up the corresponding colour. Connect everything as shown in the diagram.

>STEP-05

Register as a Twitter API developer

Anyone with a Twitter account can register as a developer, although you might be asked to provide a mobile phone number or other identification details. Once you've registered, you need to create a new application at **apps.twitter.com**. Click the button to create a new app and then fill in the required fields. Once that's done, select the 'Keys and Access Tokens' tab and click on the 'Create my access token' button.

>STEP-06

Process some tweets

Download or type up the code from the **tweetometer.py** listing. Add the Twitter API keys and tokens generated in step 5 at the appropriate places. Now pick a hashtag or keyword for testing. As this is US presidential election year, we found that using the name of one of the candidates generated more than enough data! Run the code: you should see a running count of the analysed tweets on the console, and the LED should flash with each new matching tweet. Between new tweets, the LED will remain the colour of the sentiment with the biggest count.

tweetometer.py

Language >PYTHON 3

DOWNLOAD: magpi.cc/1WBerda

```
#Tweet-o-meter: Add your own Twitter API developer
keys (lines 9-12)
# and choose your own keyword/hashtag (line 56)
import time, sys
from textblob import TextBlob
from gpiozero import RGBLED
from twython import TwythonStreamer

# Add Python Developer App tokens and secret keys
APP_KEY ='ENTER APP KEY HERE' # <- CHANGE
APP_SECRET = 'ENTER APP SECRET HERE' # <- CHANGE
OAUTH_TOKEN = 'ENTER OAUTH_TOKEN HERE' # <- CHANGE
OAUTH_TOKEN_SECRET = 'ENTER OAUTH_TOKEN_SECRET HERE' # <- CHANGE

# Set our RGB LED pins
status_led = RGBLED(14,15,18, active_high=True)
# Set active_high to False for common anode RGB LED
status_led.off()
totals = {'pos':0,'neg':0,'neu':0}
colours = {'pos':(0,1,0),'neg':(1,0,0),'neu':(0,0,1)}

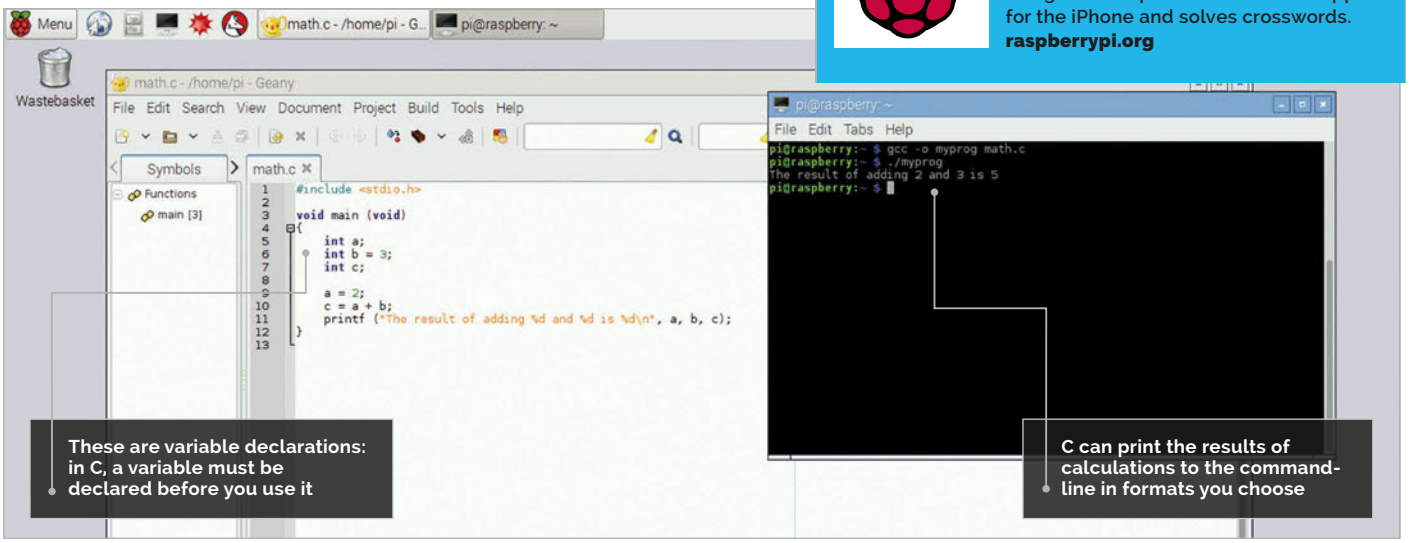
class MyStreamer(TwythonStreamer):
    def on_success(self,data): # When we get valid data
        if 'text' in data: # If the tweet has a text field
            tweet = data['text'].encode('utf-8')
            #print(tweet) # Uncomment to display  each tweet
            tweet_pro = TextBlob(data['text']) # Calculate sentiment
            # Adjust value below to tune sentiment sensitivity
            if tweet_pro.sentiment.polarity > 0.1: # Positive
                print('Positive')
                status_led.blink(on_time=0.4, off_time=0.2, on_color=(0, 1,
0), n=1, background=False)
                totals['pos']+=1
            # Adjust value below to tune sentiment sensitivity
            elif tweet_pro.sentiment.polarity < -0.1: # Negative
                print('Negative')
                status_led.blink(on_time=0.4, off_time=0.2, on_color=(1, 0,
0), n=1, background=False)
                totals['neg']+=1
            else:
                print('Neutral') # Neutral
                status_led.blink(on_time=0.4, off_time=0.2, on_color=(0, 0,
1), n=1, background=False)
                totals['neu']+=1
        overall_sentiment = max(totals.keys(),key=(lambda k: totals[k]))
        status_led.color = colours[overall_sentiment]
        print(totals)
        print('winning: ' + overall_sentiment)
        time.sleep(0.5) # Throttling

    def on_error(self, status_code, data): # Catch and display Twython errors
        print( "Error: " )
        print( status_code)
        status_led.blink(on_time=0.5,off_time=0.5, on_color=(1,1,0),n=3)

# Start processing the stream
stream2 = MyStreamer(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
while True:  # Endless loop: personalise to suit your own purposes
    try:
        stream2.statuses.filter(track='magpi') # <- CHANGE THIS KEYWORD!
    except KeyboardInterrupt: # Exit on Ctrl-C
        sys.exit()
    except: # Ignore other errors and keep going
        continue
```

SIMON LONG

Works for Raspberry Pi as a software engineer, specialising in user interface design. In his spare time he writes apps for the iPhone and solves crosswords.
raspberrypi.org



AN INTRODUCTION TO C

PART 02

VARIABLES & ARITHMETIC

Doing some real work in C: creating variables and doing mathematical operations on them

In some languages, you can create variables as you go along and put whatever data you want into them. C isn't like that: to use a variable in C, you need to have created it first, and at the time you create it, you have to set what type of value it is going to store. By doing this, a block of memory of the correct size can be allocated by the compiler to hold the variable. This process of creating a variable is known as declaration.

Integers

There are several fundamental data types in C, but we'll start by looking at one of the most commonly used: the **int** type, used to store an integer value.

```
#include <stdio.h>

void main (void)
{
        int a;
        int b = 3;
        int c;

        a = 2;
        c = a + b;
        printf ("The result of adding %d and
%d is %d\n", a, b, c);
}
```

ARITHMETIC SHORTHAND

C allows shortcuts for some common operations; for example, instead of typing `a = a + 1`, you can just enter `a++`. Or for `a = a * 3`, you can enter `a *= 3`

The top three lines inside the main function here are declarations. They tell the compiler that we would like to use variables called **a**, **b** and **c** respectively, and that each one is of type **int**, i.e. an integer.

In the second line, we see an example of an initialisation at the same time as a declaration: this stores an initial value of 3 in the variable **b**. Note that the values of **a** and **c** at this point are undefined; you might assume that a variable which hasn't had a value stored in it is always 0, but that isn't the case in C. Before reading the value from a variable, or using it in a calculation, you must store a value in it; reading a variable before initialising it is a common error in C.

The next two lines do some actual work with the variables we have declared.

```
a = 2;
```

This stores a value of 2 in the variable **a**, which will now have this value until it is changed. (The reason **a** is called a variable is that it can vary: you can change its value as often as you like, but only to another integer. The value of a variable can change, but its type is fixed when it is declared.)

```
c = a + b;
```

This line adds **a** to **b**, and stores the result in **c**.

DECIMAL PLACES

You can set the number of decimal places to display for a floating point type-specifier in `printf` by putting a decimal point and the number of places between the % and the f – so `%.3f` will show a float value with three digits after the decimal point.

```
printf ("The result of adding %d and %d is %d\n", a, b, c);
```

This is another use of the formatted print function we saw in the previous instalment. Note the three **%d** symbols inside the string: these are called format specifiers, and they are how you output numbers in C. When the **printf** function is executed, each **%d** is replaced by a decimal representation (d for decimal integer) of the variable in the corresponding position in the list after the string. So the first **%d** will be replaced by the value of **a**, the second with the value of **b** and the third with the value of **c**.

Compile the program above and then run it. You should see this:

```
The result of adding 2 and 3 is 5
```

...in the terminal.

Floating-point numbers

So we can add two integers together; what else can we do? One thing we might want to do is to use floating-point numbers. These are numbers with a decimal point. These have a different type, called **float**. Try changing the code above so instead of:

```
int a;
```

...you have:

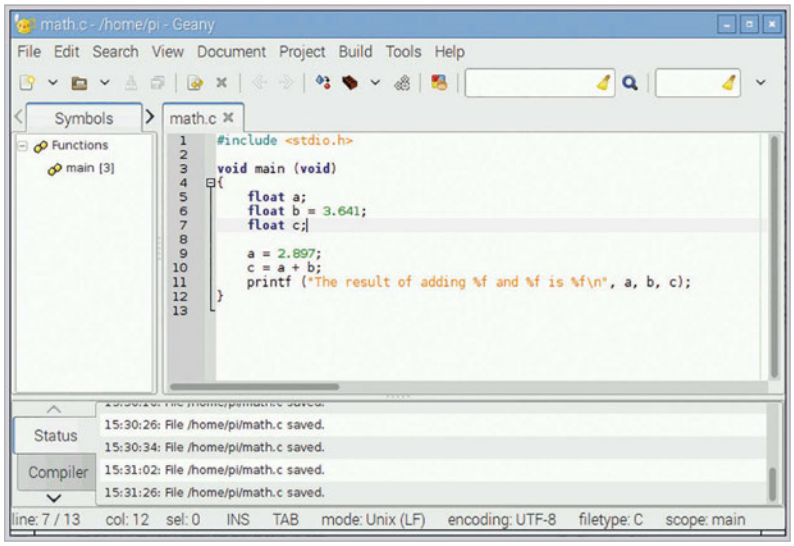
```
float a;
```

This tells the compiler that **a** is now a floating-point value, rather than an integer. Compile and run your program. What happens?

Oops! That doesn't look right, does it? What has happened is that, while the maths is still all correct, the **printf** statement is now wrong, because you are telling it to print **a**, which is a floating-point value, as a decimal integer. To fix that, change the first **%d** in the **printf** function to **%f**, which is the format specifier for a floating-point number, like this:

```
printf ("The result of adding %f and %d is %d\n", a, b, c);
```

That should produce something a lot more sensible when you run it. This is an important lesson about C: it will do exactly what you tell it to, even if it makes no sense. You told it to show you a floating-point number as if it were a decimal integer, and the compiler assumed that was what you wanted, even though the result was nonsense.



Above Don't forget to use %f instead of %d as the print specifier when changing the int values to float values in the example

When you are working with variables, always keep track of what values you are putting in what types, as it is easy to introduce errors by assuming a variable is of one type when it is actually another. One common error is to put the results of a calculation on floating-point values into an integer.

Try this: make **b** a float as well (not forgetting to change its format specifier in the **printf**), but leave **c** as an int, and set the two floats to values with decimal points, like this:

```
float a;
float b = 3.641;
int c;

a = 2.897;
c = a + b;
printf ("The result of adding %f and %f is %d\n", a, b, c);
```

You'll see a result like:

```
The result of adding 2.897000 to 3.641000 is 6
```

6? That's not right! But it is exactly what you have asked for. What the compiler did was to add the two floating-point values together, and got the answer 6.538, but you then told the compiler to put that into **c**, an integer variable. So the compiler just threw away everything after the decimal point! If you change **c** to a float, and change the final **%d** to **%f**, you'll find it gives the correct answer.

That gives you some idea about how C handles numbers, and how you can use it for arithmetic; in the next instalment we'll look at how to use the results of calculations to make decisions.

MULTIPLE DECLARATIONS

You can declare multiple variables of the same type in one line, separated by commas. For the example here, instead of three separate int declarations, you could type `int a, b = 3, c;` on one line.

REMEMBER PRECEDENCE

C obeys the common rules for operator precedence – so `a = a + 2 * 3` evaluates the multiply first and then adds the result, 6, to a. You can use round brackets to change precedence – `a = (a + 2) * 3` gives 3a + 6.

MIKE'S PI BAKERY



MIKE COOK

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

You'll Need

- GP2Y0A2YK0F IR distance sensor
- LM339 comparator
- 26-way dual line pin header socket
- 3× LEDs, coloured red, green & blue
- 21×14-hole stripboard
- 50mm length of angled aluminium 15×15mm
- Assorted resistors and capacitors

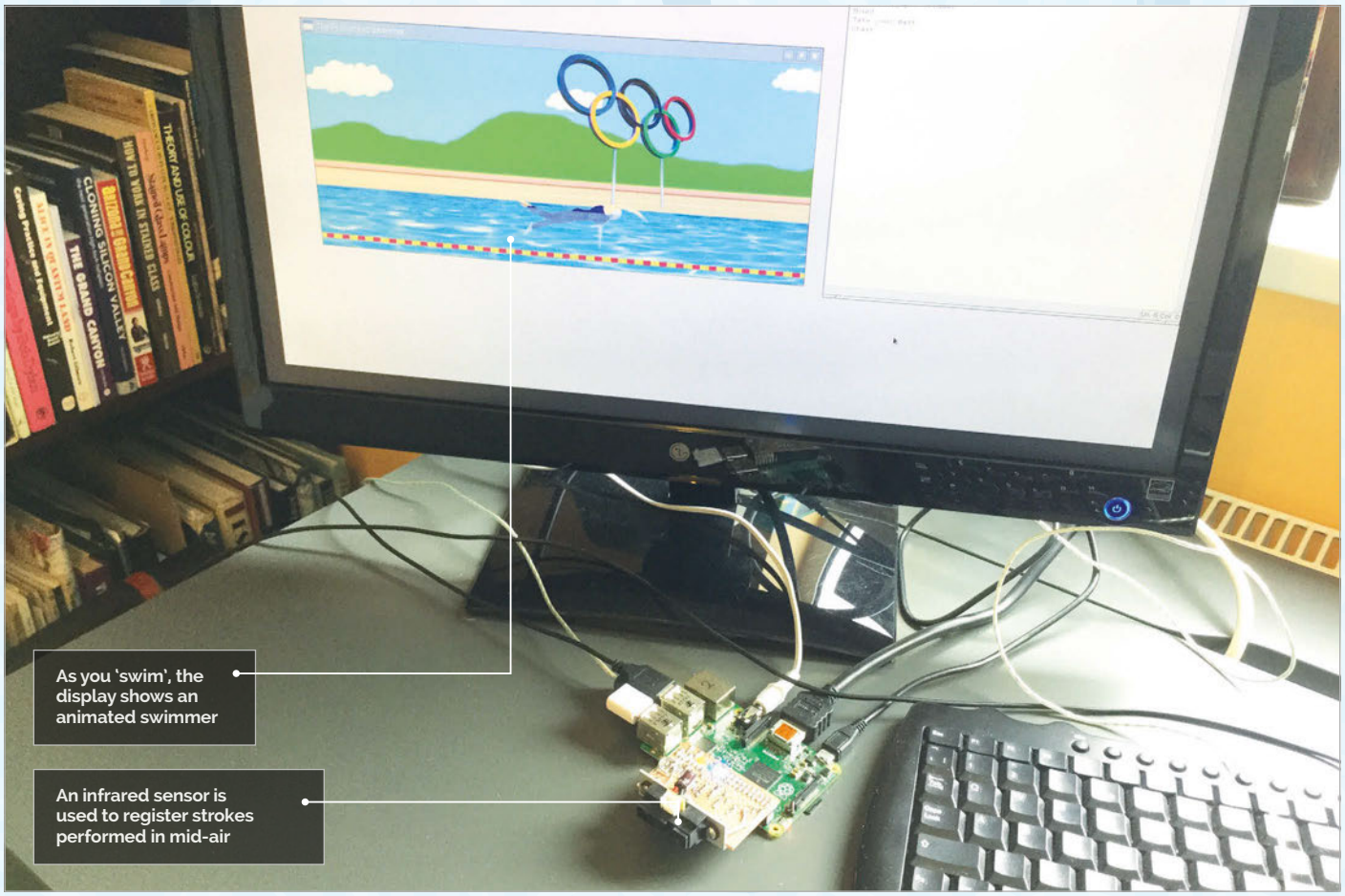
OLYMPIC SWIMMING

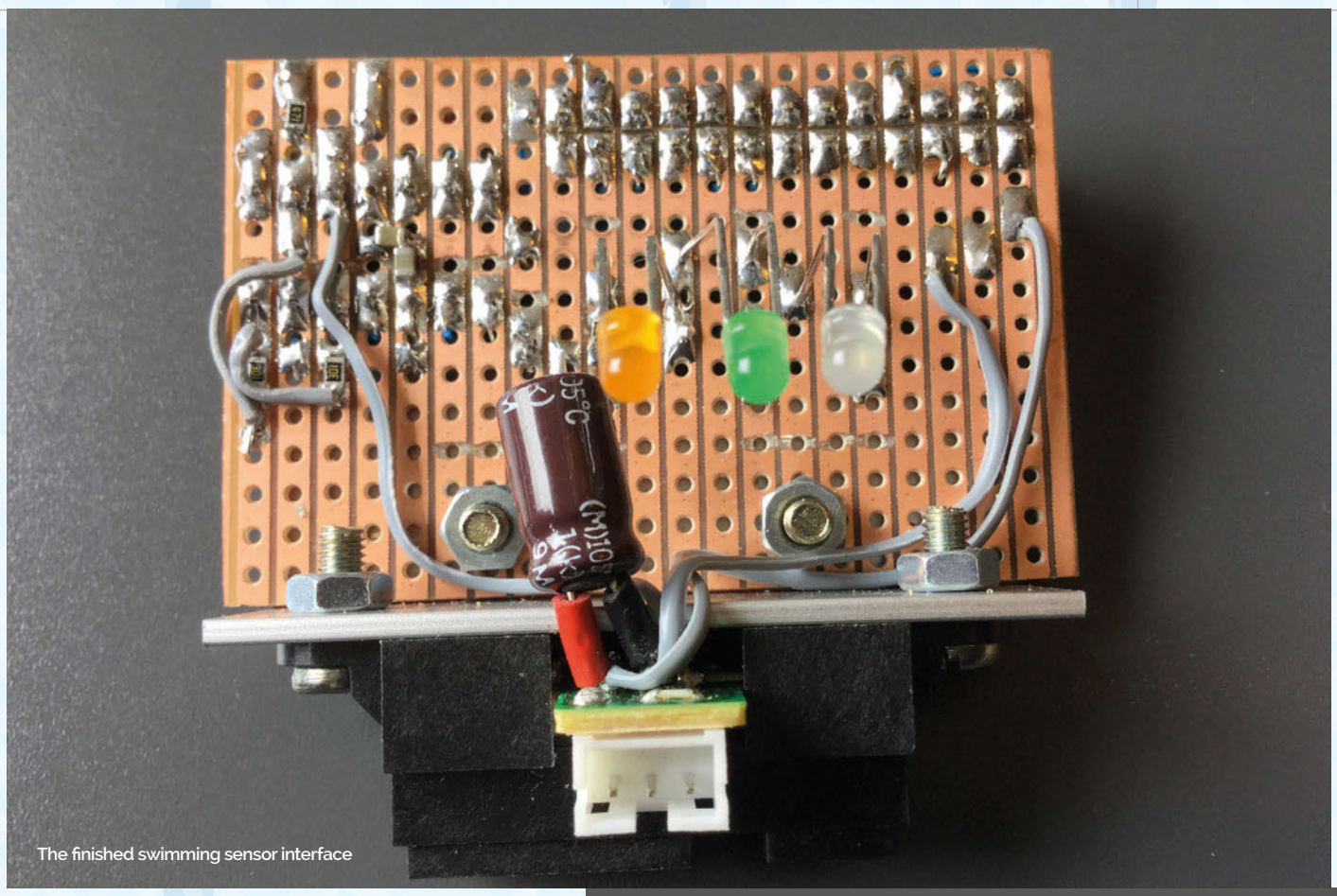
A UNIQUE SWIMMING RACE SIMULATOR

Try to beat the Olympic 50-metre freestyle record

Last year in *The MagPi* #35 (July 2015), we showed you how to make a Pi Sprinting simulator, where you ran on the spot, tapping foot sensors. Now, with Olympic fever about to grip the nation, we bring you a swimming simulator. We are going to use some of the elements of the Pi Sprinter software, but introduce a new way of interacting with the Pi.

The idea is that an IR distance sensor is used to register imaginary swimming strokes, although the optimum stroke to use is not a graceful front crawl. In order to complete a stroke, you need to register three readings of increasing distance; the best way to do this is to do a sort of doggy paddle in mid-air in front of the sensor. The setup has the added





The finished swimming sensor interface

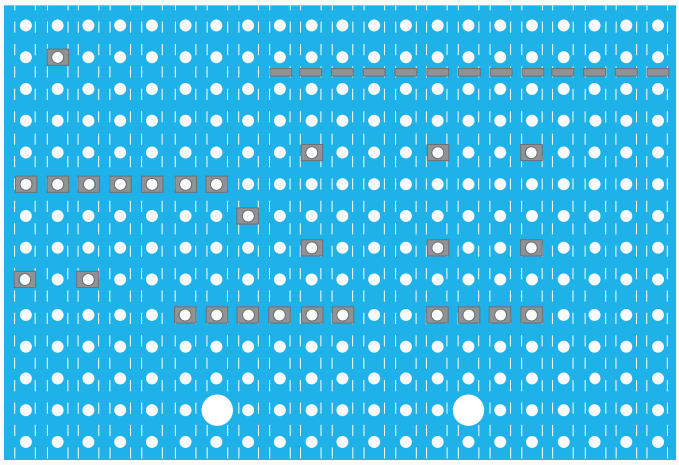
advantage of being totally immune to the ravages of overenthusiastic kids, because there is nothing that they have to touch.

The hardware

The IR distance sensor we used is the GP2Y0A2YK0F, which is the same type we used in *The MagPi* #44 infinity mirror project. The method of converting the output into a digital reading is the same as well, only this time the resistor values have been altered to trigger the distances we need. There are also three coloured LEDs that help you know when a stroke is being registered, so you don't drift out of the line of the sensor. We built the sensor onto a piece of stripboard that plugs directly into the Pi's GPIO header. We used a mixture of surface-mount and through-hole techniques, although you can make it all using through-hole parts if you prefer. The full schematic of this is shown in **Fig 1** (overleaf), and the finished interface can be seen above. The construction details are in the step-by-step guide.

> "Totally immune to the ravages of enthusiastic kids"

BUILDING THE SWIM SENSOR



>STEP-01

Prepare the board

Cut out a piece of 21×14-hole stripboard and make the cuts in the tracks shown above. Note that the cuts between the GPIO header are not done at a hole, but between the holes. To do this, make a cut across the board with a scalpel and then make another one in parallel with the first, as close as you can. This second cut should be angled in a bit so you remove a sliver of copper. Drill two 3mm holes for the sensor mounting bracket.

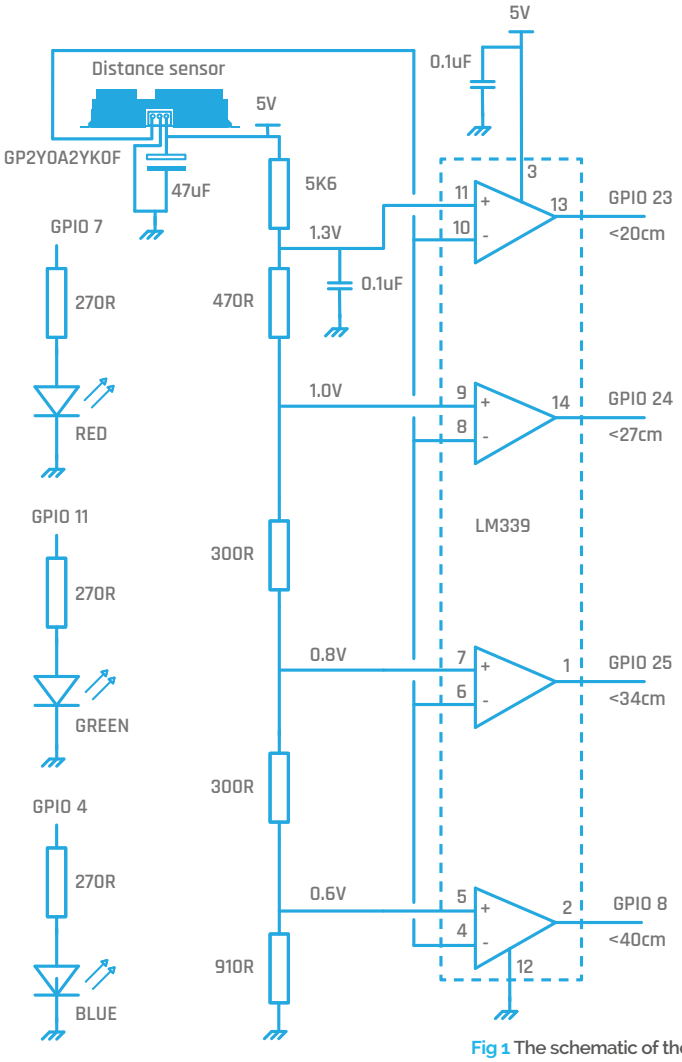


Fig 1 The schematic of the project

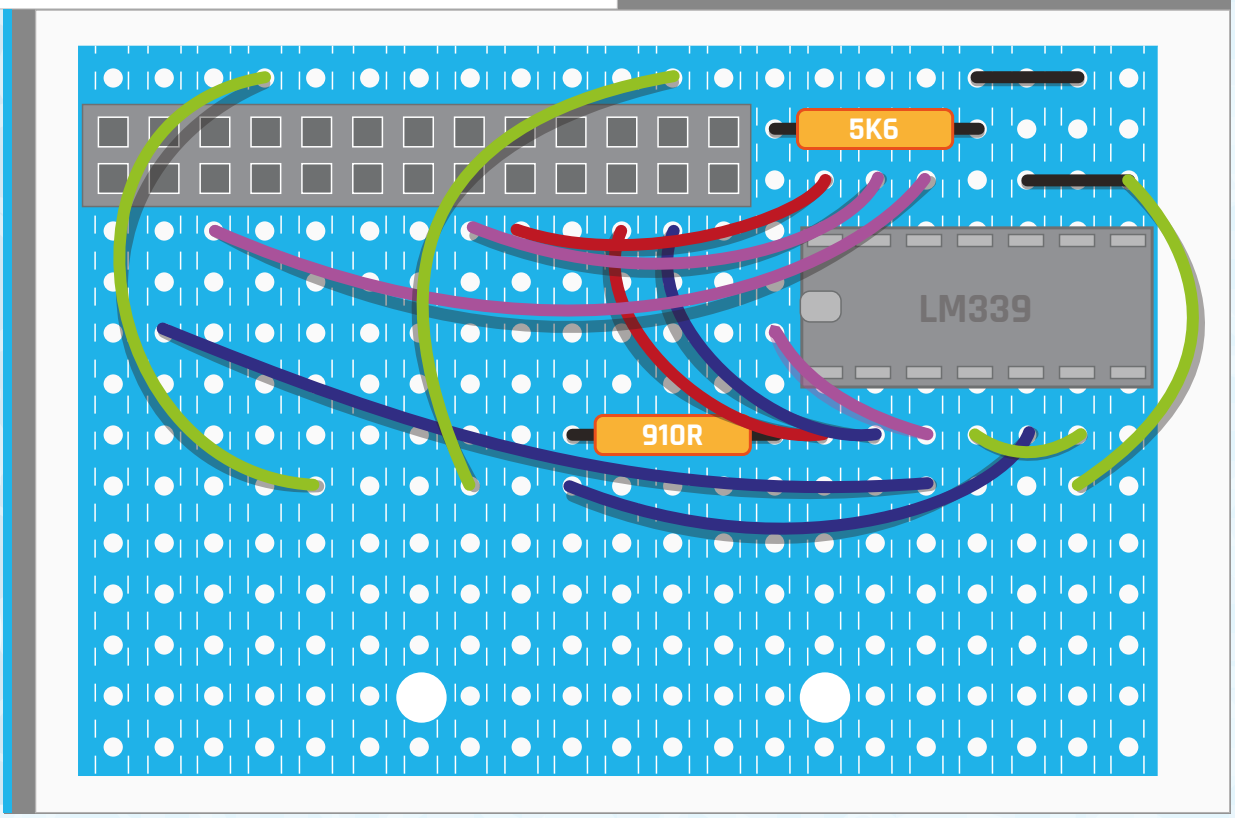
Graphics work

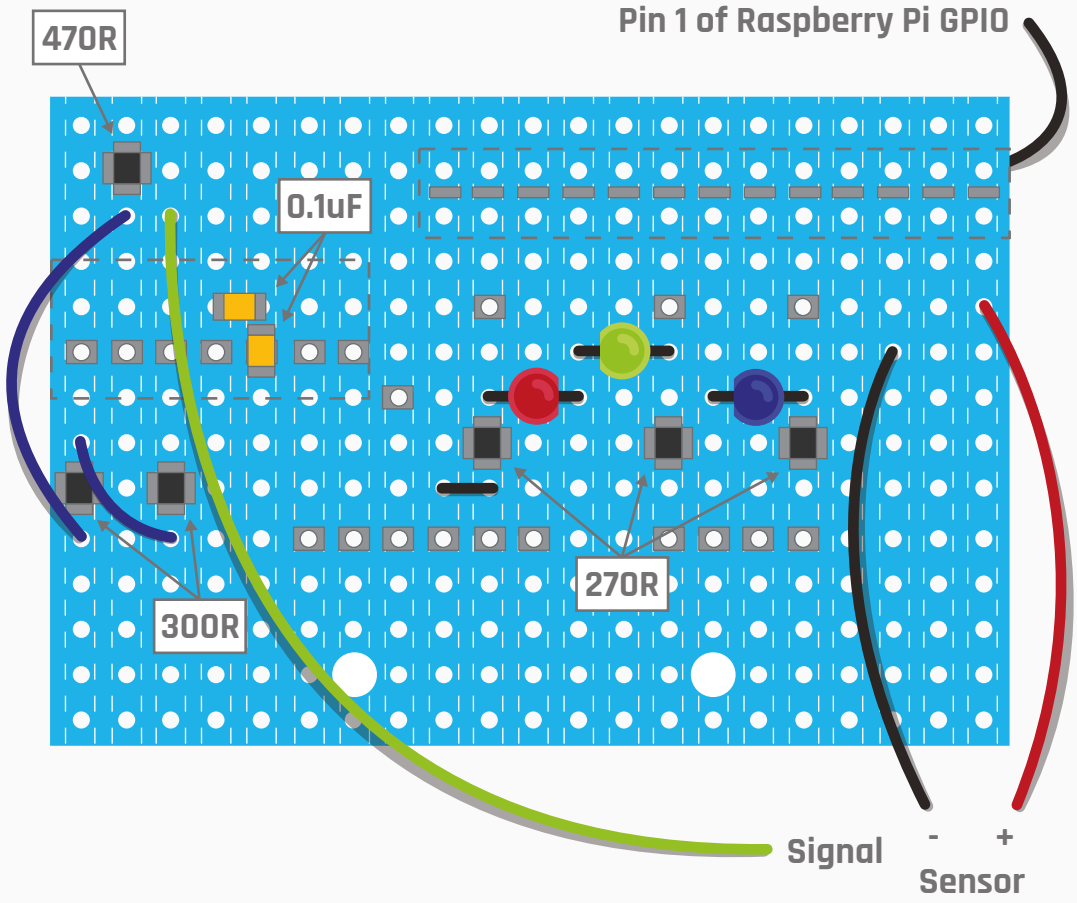
The big difference between this project and the Pi Sprinter one is the graphical display, which is split into two parts: the swimmer and the background. We used the Pi sprinter as the starting point for the background and added both a textured water surface in place of the track, and some Olympic sculptures on the side of the outdoor pool. The problem with the water surface was that it didn't wrap round well: there was a discontinuity at each end. This was solved by splitting the water surface graphic into two, left and right, and then swapping the sides over. This had the effect of putting the discontinuity in the middle of the frame, where it could be made to blend together using the clone and blur tools found in any photo-processing package. The swapping of left and right halves is easier using Photoshop: use the filter in Filters>other>over. The GIMP package can be used for an all-Pi graphics solution. A repeating pattern of red and yellow rectangles make up a lane divider for the bottom of the screen.

>STEP-02

Mounting the components

Attach the GPIO plug and IC socket as shown below. Add the solid wire links, shown in black, and solder the resistors in place. Complete the wiring by using flexible wires where indicated. Prepare the sensor bracket by drilling 3mm holes on one side of the angle aluminium to match those in the stripboard, and in the other to match the sensor.



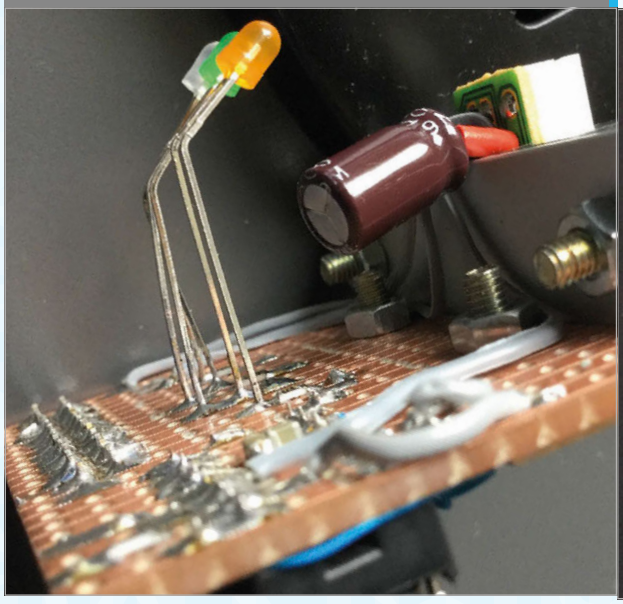


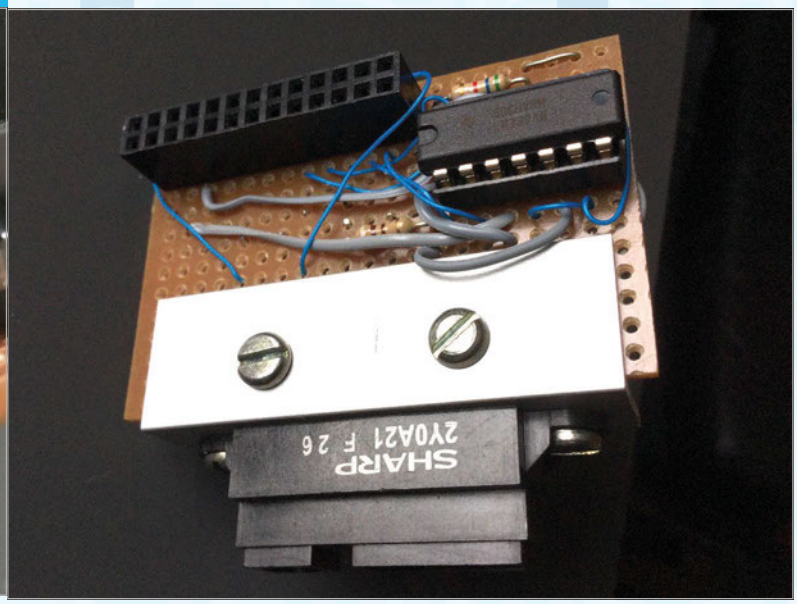
>STEP-03

Wire up the track side

Flip the board over and add the surface-mount resistors and capacitors; through-hole parts can be substituted if you prefer. Use flexible wires to make the other connections. Mount the three LEDs and angle them by about 45° towards the sensor; note here that the flat spot on the LED denotes the cathode, which is usually the shortest leg. Fix the sensor to a bracket, attach the bracket to the board, and wire it up to the sensor.

The swimming figure was isolated frame by frame from a swimming animation, and the complete sequence for a full stroke is 24 frames long. The figure had a water level line where the colours changed slightly, as if seen underwater. This was achieved by adding a blue tint to the colours under this line. However, this still gave the impression that the swimmer was swimming on top of the water. To get round this, the part of the figure under water was given a 66% opacity





value. This was done by creating two layers in the photo-editing package with the same contents. Then the out-of-water half was removed from one layer and the underwater half from the other. The underwater layer was then set to 66% opacity and the picture saved as a PNG file. This merges or flattens the image before storage. The upshot is that the lower half of the figure is translucent and you can see some of the water surface ripple through the lower part of the swimmer. However, in practice, thanks to the way the brain interprets images, this looks like you are seeing the swimmer through the water ripple. **Fig 2** shows this effect halfway through a race.

The software

The software is shown in the **swimmer.py** listing. The background scrolls repeatedly as the swimmer makes their way from the left to the right of the screen. The action looks smooth as long as you keep paddling with your hands. The LEDs will light in the sequence red, green, blue as you make one stroke. The start sequence is like a swimming race,

> "The sounds were pulled from a real race video"

with a whistle summoning the swimmers, then the instruction to 'take your mark', and finally a buzzer starting the race. The sounds were pulled from a real race video; we found the best races to use were the long ones, as they did not suffer from voice-over by commentators like the sprints did. All the sound and graphics files, along with the code, are on our GitHub repository (**magpi.cc/1NqJjmV**).

Taking it further

You could duplicate the distance sensor to make a side-by-side real-time race. Alternatively, using two sensors, you could detect a butterfly stroke.

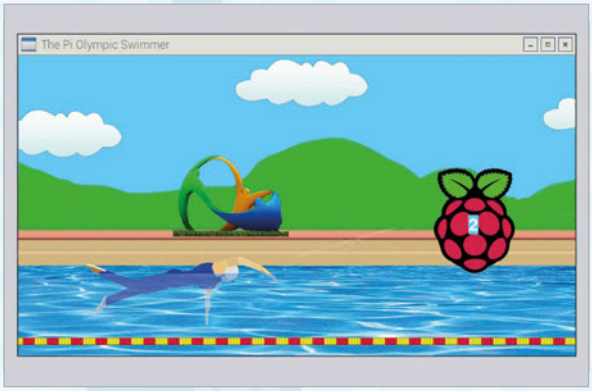


Fig 2 The screen halfway through a race

swimmer.py

```
01.  import pygame, time, os, random
02.  import wiringpi2 as io
03.
04.  pygame.init()       # initialise graphics interface
05.  pygame.mixer.quit()
06.  pygame.mixer.init(frequency=22050, size=-16, channels=2,
     buffer=512)
07.
08.  os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
09.  pygame.display.set_caption("The Pi Olympic Swimmer")
10.  pygame.event.set_allowed(None)
11.  pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])
12.  screen = pygame.display.set_mode([1000,415],0,32)
13.  textHeight = 36
14.  font = pygame.font.Font(None, textHeight)
15.  random.seed()
16.
17.  compPins = [ 8,25,24,23]
18.  ledPins = [ 7, 11, 4 ]
19.  restart = True ; strokeState = 0
20.
21.  soundEffects = ["whistle","mark","go","end"]
22.  swimingFrames = [ pygame.image.load("images/
     S"+str(frame)+".png").convert_alpha()
23.                  for frame in range(0,24)]
24.  background =  pygame.image.load("images/BackgroundPi.png").
     convert_alpha()
25.  gameSound = [ pygame.mixer.Sound("sounds/"+soundEffects[sou
     nd]+".ogg")
26.                  for sound in range(0,4)]
27.
28.  def main():
29.     global restart, strokeState
30.     initGPIO()
31.     print"The Pi Olympic Swimmer"
32.     while True:
33.       if restart:
34.          frame = 0 ; distance = 0 ; manDistance = -85
35.          posts = 3 ; ledOff() ; strokeState = 0
36.          restart = False ; showPicture(frame,distance,-
     400,posts)
37.          time.sleep(2.0) ; gameSound[0].play()
38.          print"Mount" ; time.sleep(4.0)
39.          showPicture(frame,distance,manDistance,posts)
40.          print"Take your mark" ; gameSound[1].play()
41.          time.sleep(random.randint(2,5))
42.          print"Start" ; startTime = time.time()
43.          gameSound[2].play()
44.       strokeDetect()
45.       showPicture(frame,distance,manDistance,posts)
46.       manDistance += 4
47.       distance += 40
48.       if distance > 3000:
49.          distance -= 2000
```

Language
>PYTHON 2.7

DOWNLOAD:
magpi.cc/1NqJjmV

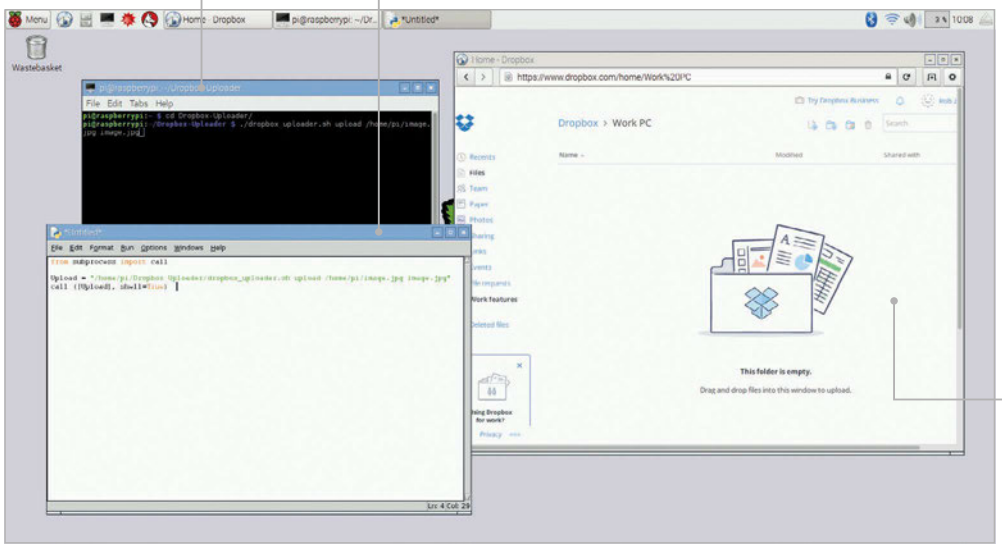
PROJECT VIDEOS
Check out Mike's Bakery videos at:
magpi.cc/1NqJnTz

```
50.         posts -=1
51.      frame = frame + 1
52.      if frame > 23:
53.         frame = 0
54.      if posts == 0 and distance >=100 :
55.         raceTime = int(100*(time.time() - startTime))
56.         gameSound[3].play()
57.         drawWords("Finished "+str(raceTime / 100.0)+"
    Seconds",400,258)
58.         pygame.display.update()
59.         print"Finished - type return for another race"
60.         while not restart:
61.            checkForEvent()
62.
63. def initGPIO():
64.    try :
65.       io.wiringPiSetupGpio()
66.    except :
67.       print"start IDLE with 'gksudo idle' from command
    line"
68.       os._exit(1)
69.    for pin in range (0,4):
70.       io.pinMode(compPins[pin],0) # mux pin to output
71.       io.pullUpDnControl(compPins[pin],2) # input enable
    pull up
72.    for pin in range (0,3):
73.       io.pinMode(ledPins[pin],1) # LED pin to output
74.       io.digitalWrite(ledPins[pin],0)
75.
76. def showPicture(frame,distance,manDistance,post):
77.    screen.blit(background,[-distance,0])
78.    if distance > 1000 :
79.       screen.blit(background,[1999-distance,0])
80.    drawWords(str(post),-distance+1932,220)
81.    screen.blit(swimingFrames[frame],[60+manDistance,230])
82.    pygame.display.update()
83.
84. def drawWords(words,x,y) :
85.         textSurface = pygame.Surface((14,textHeight))
86.         textRect = textSurface.get_rect()
87.         textRect.left = x
88.         textRect.top = y
89.         pygame.draw.rect(screen,(102,204,255),
    (x,y,14,textHeight-10), 0)
90.         textSurface = font.render(words, True,
    (255,255,255), (102,204,255))
91.         screen.blit(textSurface, textRect)
92.
93. def strokeDetect():
94.    global strokeState
95.    if strokeState == 0:
96.       while getSensor() != 1 :
97.          checkForEvent()
98.       io.digitalWrite(ledPins[2],0)
99.       io.digitalWrite
    (ledPins[0],1)
100.      strokeState = 1
101.      return
102.   if strokeState == 1:
103.      while getSensor() != 3 :
104.         checkForEvent()
105.      io.digitalWrite(ledPi
    ns[0],0)
106.      io.digitalWrite(ledPins[1],1)
107.      strokeState = 2
108.      return
109.   if strokeState == 2:
110.      while getSensor() != 7 :
111.         checkForEvent()
112.      io.digitalWrite(ledPins[1],0)
113.      io.digitalWrite(ledPins[2],1)
114.      strokeState = 0
115.      return
116.
117. def getSensor():
118.    sensor = 0
119.    for i in range(0,4) :
120.         sensor = (sensor << 1) |
    io.digitalRead(compPins[i])
121.    return sensor
122.
123. def ledOff():
124.    for pin in range (0,3):
125.       io.digitalWrite(ledPins[pin],0) # LED off
126.
127. def terminate(): # close down the program
128.    print "Closing down please wait"
129.    for pin in range (0,3):
130.       io.pinMode(ledPins[pin],0) # LED pin to input
131.    pygame.mixer.quit()
132.    pygame.quit() # close pygame
133.    os._exit(1)
134.
135. def checkForEvent(): # see if we need to quit
136.    global restart
137.    event = pygame.event.poll()
138.    if event.type == pygame.QUIT :
139.         terminate()
140.    if event.type == pygame.KEYDOWN :
141.       if event.key == pygame.K_ESCAPE :
142.          terminate()
143.       if event.key == pygame.K_RETURN :
144.          restart = True
145.          print"New Race"
146.
147. # Main program logic:
148. if __name__ == '__main__':
149.    main()
```



ROB ZWETSLOOT

Tinkerer, sometime maker, other-times cosplayer, and all-the-time features editor of *The MagPi*.
magpi.cc / @TheMagPi



GET DROPBOX ON RASPBERRY PI

You'll Need

- Dropbox account **dropbox.com**
- Dropbox Uploader **magpi.cc/2aaHoJN**
- Your Dropbox API key

Connect to the most ubiquitous cloud service on your Raspberry Pi, perfect for uploading pictures and video in a project!

Dropbox's relationship with Linux has always been slightly weird, and as Raspbian is a version of Linux, that too means it's not so straightforward to get the file-syncing behaviour of Dropbox to work. There are definitely ways around this, though, and with a little bit of hacking and tweaking, we can get automatic uploads (and downloads!) of items to Dropbox. This method was created by Alex Eames of **RasPi.TV** and is perfect for many types of Raspberry Pi project, especially those where you're taking pictures and want to view them remotely or free up some space on the Raspberry Pi after they've been taken.

>STEP-01

Get a Dropbox account

If you don't already have one, sign up for a Dropbox account at **dropbox.com**. It offers a couple of GB for free, but you can pay a small amount a month for a whopping 1TB of space. There are some other cloud services around, such as Google Drive, but they have even less Linux support than Dropbox. As with most cloud storage services, you can view, download, and upload files from the browser. So if you want to download anything to the Raspberry Pi, it can be quick and easy to go through there.

>STEP-02

Get Dropbox uploader

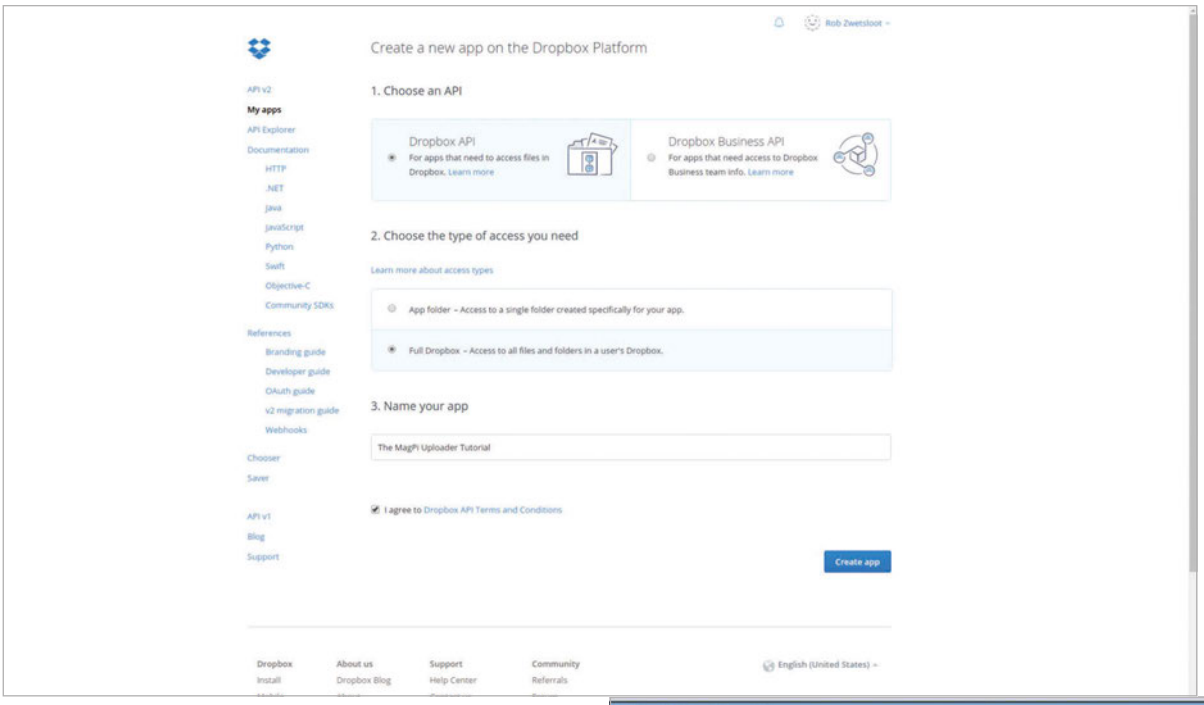
Now we need to grab Dropbox Uploader on the Raspberry Pi. Boot into Raspbian if you're not already using it, and either open a Terminal or SSH into the Raspberry Pi if you prefer. From there, you'll need to download the install files with:

```
github.com/andreafabrizi/Dropbox-Uploader.git
```

Once that's downloaded, you'll need to move to the folder (`cd Dropbox-Uploader`) to begin installing. You can start this off with:

```
./dropbox_uploader.sh
```

It will ask for your API key, which is our cue to move onto to the next step.



Left Creating an app on Dropbox is easy; just make sure it has a unique name so you can get it working

FREEING UP SPACE

In a Bash or Python script using the function, you can always set it to delete the upload once it's sent, to free up space on the Pi.

>STEP-03

Find your API key

You need to head to the developers' section of Dropbox (**magpi.cc/2aaQnKQ**) so you can create a new app and get a unique API key to use on the Raspberry Pi. Click on **Create App** to start.

As we're working towards a personal use application, the first option we'll chose is Dropbox API rather than business. The next two options don't really matter: if you want to access full Dropbox, you can, but it may be better for privacy and security reasons if you're just able to use a specific folder on your Dropbox. Finally, name it whatever you want and click **Create App**.

>STEP-04

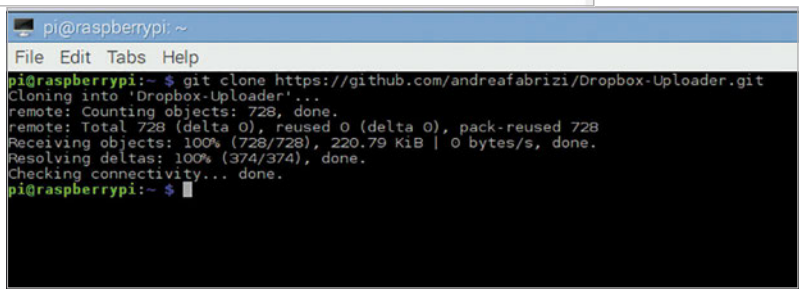
Enter your API key

On the settings page for the app you created, there will be an 'App key' field. Note it down or simply copy and paste it in into the Terminal if you're still on your Raspberry Pi. It will then ask for the 'App secret', which is right below the key in your settings page. Click on 'show' and then enter that. It will then ask you to confirm what type of permission you gave it (full or just a folder) and then it will drop a link to put in the browser to confirm everything. Press **ENTER** to finish the setup and if everything has gone correctly, it will flash up a message to let you know!

>STEP-05

Add a progress bar

Without a progress bar, you won't always know if everything is working. Luckily, you can add one to this project: open up the installed file we just used (**`nano dropbox_uploader.sh`**) and look for the line that says **`SHOW_PROGRESSBAR`** under Default values.



Above From the Terminal, all you need to do is download the project to begin with: it's just a simple git request

If this line ends with **`=0`**, then change the **`0`** to a **`1`** and save the file. There are also some other options under Default values (such as the ability to skip existing files), so have a quick look and see if there's anything else you feel confident to change.

>STEP-06

Start uploading!

Now everything should be working and you can start uploading. Everything revolves around the **dropbox_uploader** file, so stay in the folder or make sure to have your code point towards the folder in the future. The code to upload is something like:

```
./dropbox_uploader.sh upload path/to/file
dropbox_filename
```

You can use this code in Python 3 by creating an OS call, using something like:

```
from subprocess import call
Upload = "home/pi/Dropbox_Uploader/
dropbox_uploader.sh upload path/to/file
dropbox_filename"
call ([Upload], shell=True)
```

Time to get uploading and experimenting!

OTHER COMMANDS

As well as uploading, you can use it to download files. A full list of commands is at: **magpi.cc/2aaHoJN**

Sonic Pi PART 12



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

EXPLORE THE POWER OF AMPLITUDE MODULATION

Learn how to master Sonic Pi's powerful slicing capabilities with its creator, **Sam Aaron**....

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`

This month, we take a deep dive into one of Sonic Pi's most powerful and flexible audio FX – the **:slicer**. First, listen to the deep growl of this code, which triggers the **:prophet** synth:

```
synth :prophet, note: :e1, release: 8, cutoff: 70
synth :prophet, note: :e1 + 4, release: 8, cutoff: 80
```

Now, let's pipe it through the **:slicer** FX:

```
with_fx :slicer do
  synth :prophet, note: :e1, release: 8, cutoff: 70
  synth :prophet, note: :e1 + 4, release: 8, cutoff: 80
end
```

Hear how the **:slicer** acts like it's muting and unmuting the audio with a regular beat. It also affects all the audio generated between the **do**/**end** blocks. You can control the speed which it turns the audio on and off with the **phase:** (phase duration) opt. Its default value is **0.25**, which means 4 times a second at the default BPM of 60. Let's make it faster:

```
with_fx :slicer, phase: 0.125 do
  synth :prophet, note: :e1, release: 8, cutoff: 70
  synth :prophet, note: :e1 + 4, release: 8, cutoff: 80
end
```

Now, have a play with different **phase:** durations. Good values to try are **0.125**, **0.25**, **0.5**, and **1**. **Fig 1** shows how different **phase:** values alter the number of amplitude changes per beat.

By default, the **:slicer** FX uses a square wave to manipulate the amplitude through time. Other control waves supported by **:slicer** are saw, triangle, and (co)sine – see **Fig 2**. The following code uses (co)sine as the control wave. Hear how the sound doesn't turn on/off abruptly, but instead smoothly fades in/out:

```
with_fx :slicer, phase: 0.5, wave: 3 do
  synth :dsaw, note: :e3, release: 8, cutoff: 120
  synth :dsaw, note: :e2, release: 8, cutoff: 100
end
```

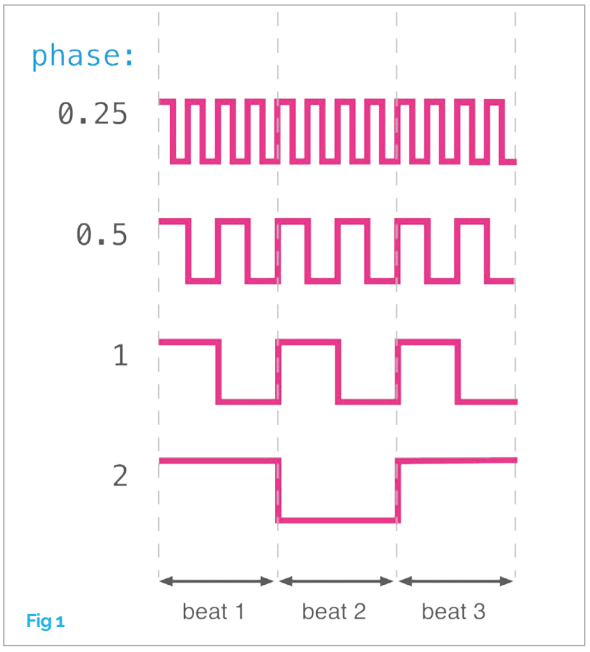


Fig 1

Play around with the different wave forms by changing the **wave:** opt to **0** for saw, **1** for square, **2** for triangle, and **3** for sine. See how different waves sound with different **phase:** opts, too. Each wave can be inverted vertically using the **invert_wave:** opt. The control wave can also be started at different points with the **phase_offset:** opt – a value between **0** and **1**. By playing with **phase:**, **wave:**, **invert_wave:**, and **phase_offset** opts, you can dramatically change how the amplitude is modified through time.

By default, **:slicer** switches between amplitude values **1** (fully loud) and **0** (silent). This can be altered with the **amp_min:** and **amp_max:** opts. Use this with the sine wave setting to create a simple tremolo effect:

```
with_fx :slicer, amp_min: 0.25, amp_max: 0.75,
wave: 3, phase: 0.25 do
  synth :saw, release: 8
end
```

This is like moving your hi-fi's volume knob up and down just a little so the sound 'wobbles' in and out.

One of **:slicer**'s powerful features is its ability to use probability to choose whether or not to turn the slicer on or off. Before the **:slicer** FX starts a new phase, it rolls a dice and, based on the result, either uses the selected control wave or keeps the amplitude off. Let's take a listen:

```
with_fx :slicer, phase: 0.125, probability: 0.6 do
  synth :tb303, note: :e1, cutoff_attack: 8, release: 8
  synth :tb303, note: :e2, cutoff_attack: 4, release: 8
  synth :tb303, note: :e3, cutoff_attack: 2, release: 8
end
```

We now have an interesting rhythm of pulses. Try changing the **probability:** opt to a different value between **0** and **1**. Values closer to **0** will have more space between each sound, due to the likelihood of the sound being triggered being much lower.

Another thing to notice is that the probability system in the FX is just like the randomisation system accessible via fns such as **rand** and **shuffle**. They are both completely deterministic. This means that each time you hit **Run**, you'll hear exactly the same rhythm of pulses for a given probability. If you would like to change things around, you can use the **seed:** opt to select a different starting seed. This works exactly the same as **use_random_seed**, but only affects that particular FX.

Finally, you can change the 'resting' position of the control wave when the probability test fails, from **0** to any other position, with the **prob_pos:** opt:

```
with_fx :slicer, phase: 0.125, probability: 0.6,
prob_pos: 1 do
  synth :tb303, note: :e1, cutoff_attack: 8, release: 8
  synth :tb303, note: :e2, cutoff_attack: 4, release: 8
  synth :tb303, note: :e3, cutoff_attack: 2, release: 8
end
```

One really fun thing to do is to use **:slicer** to chop a drum beat in and out:

```
with_fx :slicer, phase: 0.125 do
  sample :loop_mika
end
```

This allows us to take any sample and create new rhythmical possibilities, which is a lot of fun. However, one thing to be careful about is to make sure that the tempo of the sample matches the current BPM in Sonic Pi, otherwise the slicing will sound totally off. For example, try swapping **:loop_mika** with the **:loop_amen** sample to hear how bad this can sound when the tempos don't align.

As we have already seen, changing the default BPM with **use_bpm** will make all the sleep times and synth envelope durations grow or shrink to match the beat. The **:slicer** FX honours this too, as the **phase:** opt

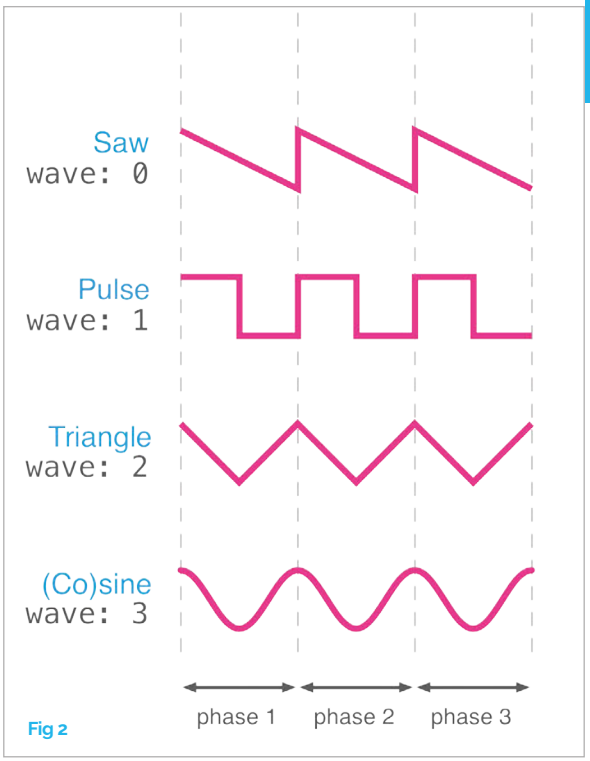


Fig 2

is actually measured in beats, not seconds. We can therefore fix the issue with **:loop_amen** above by changing the BPM to match the sample:

```
use_sample_bpm :loop_amen
with_fx :slicer, phase: 0.125 do
  sample :loop_amen
end
```

Now, let's apply all these ideas into a final example that only uses the **:slicer** FX to create an interesting combination. Go ahead, start changing it and make it into your own piece!

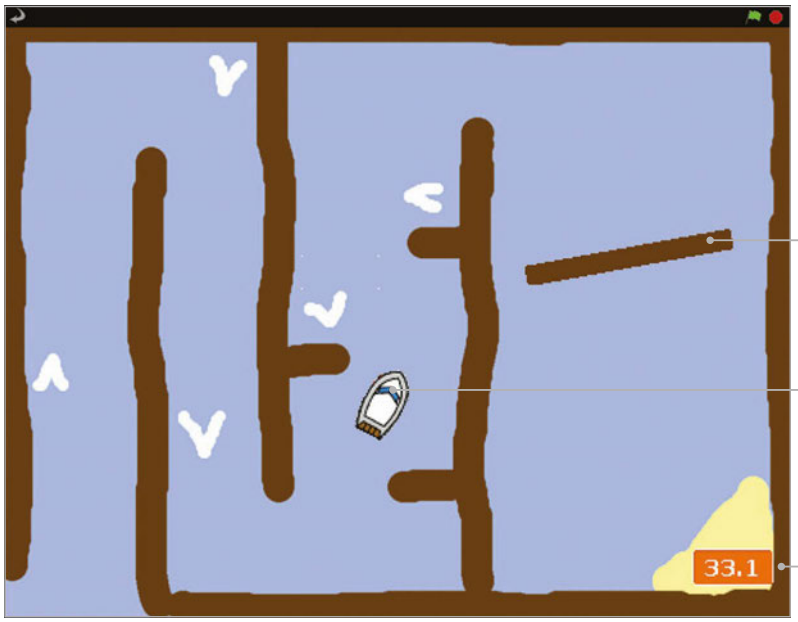
```
live_loop :dark_mist do
  co = (line 70, 130, steps: 8).tick
  with_fx :slicer, probability: 0.7, prob_pos:
1 do
    synth :prophet, note: :e1, release: 8,
cutoff: co
  end
  with_fx :slicer, phase: [0.125, 0.25].choose
do
    sample :guit_em9, rate: 0.5
  end
  sleep 8
end

live_loop :crashing_waves do
  with_fx :slicer, wave: 0, phase: 0.25 do
    sample :loop_mika, rate: 0.5
  end
  sleep 16
end
```



PHIL KING

When not sub-editing *The MagPi* and writing articles, Phil loves to work on Pi projects, and to help his six-year-old son learn Scratch coding.
@philking68



The boat crashes if it hits something brown, like this revolving gate

The boat sprite is programmed to move towards the mouse pointer

The timer is shown on screen, and stops when the boat reaches the yellow beach

You'll Need

- A mouse
- Art assets - **magpi.cc/scratch_art**
- Speaker (optional)

MAKE A BOAT RACE GAME IN SCRATCH

Create your own boat race game, complete with mouse control, collision detection, and on-screen timer

In this tutorial, you'll be making your own arcade game in which the player attempts to guide a boat safely around a maze-like course, including a revolving gate, and get to the finish in as fast a time as possible. You can even design your own custom course if you like. As well as moving a sprite towards the mouse pointer, this project involves collision detection, using the `touching color` Sensing block to determine whether the boat has hit something. Let's dive in and start coding...

code club

This tutorial was adapted from a Code Club project (**codeclubprojects.org**) and you can find more in *Learn to Code with Scratch*: **magpi.cc/Scratch-book**

>STEP-01

Prepare your artwork

First, delete the cat! You should then import the two sprites for the boat and gate. Since they're not in the Scratch 1.4 library, you can download them (**magpi.cc/scratch_art**). Just click the star/folder icon above the Sprite List (bottom-right), then navigate to the folder where you've stored the downloaded graphics for this project. Import the Boat and Gate sprites. If you aren't designing your own course, you can also download and import our Course backdrop: click Stage in the Sprite List, select the Backgrounds tab (top-middle), then click Import and navigate to the folder.

>STEP-02

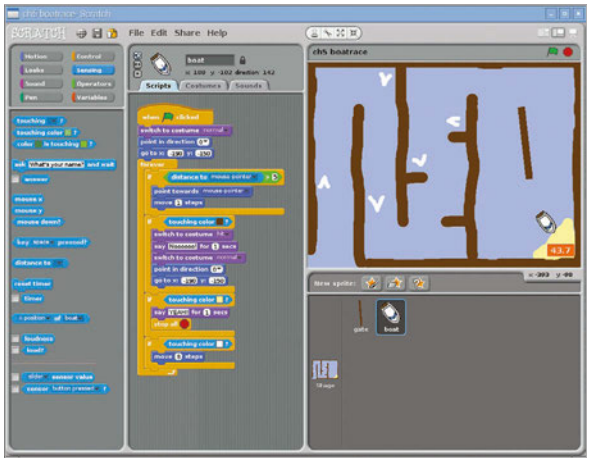
Design a course

You could just edit our course. Alternatively, to create a brand new one, click on the Stage in the Sprite List, then the Backgrounds tab, and Paint. Use the paint bucket tool to fill the canvas with a blue colour for the water. Then use a brown colour, which should be the same as in the Gate sprite, to draw the walls of the course. Use a yellow colour to draw some sand for the finish. Finally, add some white arrows which will act as speed boosters. Once this is done, let's make our Gate sprite rotate by adding the simple code in **Listing 1** to its Scripts area.

>STEP-03

Controlling the boat

In this game we'll be controlling the boat with a mouse, using the code in **Listing 2** in the Scripts tab of the Boat sprite. To do this, we simply point it towards 'mouse pointer' and move it one step at a time, within



Above: We used `touching color` Sensing blocks to detect when the boat has hit a hazard, booster, or the finish

a **`forever`** loop. To stop it from moving when near the pointer, we put the control code in an **`if`** block that only tells it to move if the distance to the pointer is greater than 5. Run the code and guide the boat: at the moment, it sails straight through barriers.

>STEP-04

Make it crash!

What we need is some collision detection to check whether the boat has hit a hazard. Within your **`forever`** block, add the code from **Listing 3** under your boat control code. Here, we use the **`touching color`** Sensing block to see if the boat has hit anything brown: click the colour square to get a dropper tool, then click on a brown part of the course. When it crashes, we switch the boat's costume, say 'Noooooo!', then place it back at the start point in its normal costume.

Let's add two more **`if touching color`** blocks, shown in **Listing 4**, to our **`forever`** loop. The first checks whether the boat has reached the yellow beach, which acts as the finish line, and stops the program. The second detects the white of our booster arrows and moves the boat three steps.

>STEP-05

Boosters and time

To make our game a bit more exciting, we need a timer. Click the Stage and add the **Listing 5** code to its Scripts area. This sets the time to zero at the start of the game, then gradually increases the **`time`** variable in line with real time; you'll need to create the latter in Variables, and make sure it's ticked so that it's shown on the stage.

>STEP-06

Taking it further

You could easily add a sound effect for when the boat crashes, using a Sound block. You could even add background music, composing it using Sound blocks with various drums, instruments, and notes. The best time(s) could also be stored in a variable or list.

.01

```
when clicked
forever
  turn 1 degrees
```

.02

```
when clicked
switch to costume normal
point in direction 0
go to x: -190 y: -150
forever
  if distance to mouse-pointer > 5
    point towards mouse-pointer
    move 1 steps
```

.03

```
if touching color ?
  switch to costume hit
  say Noooooo! for 1 secs
  switch to costume normal
  point in direction 0
  go to x: -190 y: -150
```

.04

```
if touching color ?
  say YEAH! for 1 secs
  stop all
if touching color ?
  move 3 steps
```

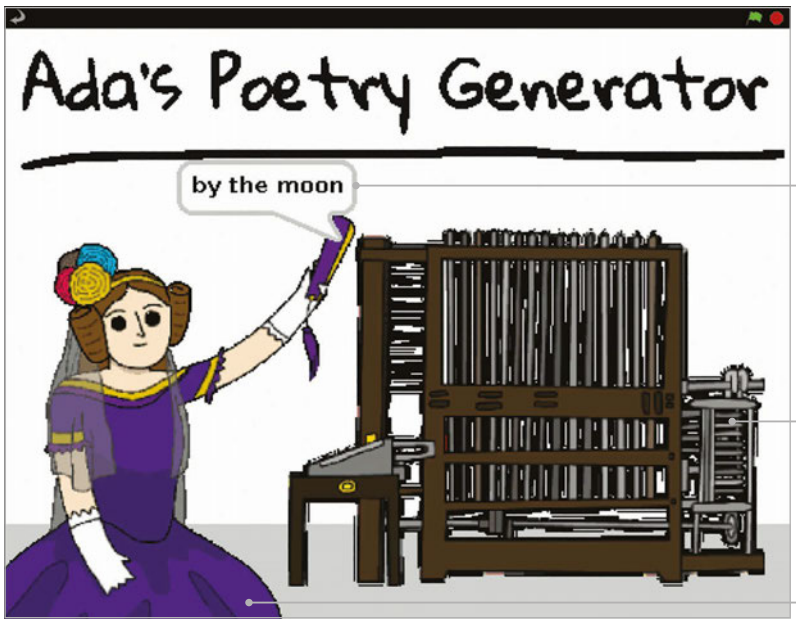
.05

```
when clicked
set time to 0
forever
  wait 0.1 secs
  change time by 0.1
```



PHIL KING

When not sub-editing *The MagPi* and writing articles, Phil loves to work on Pi projects, and to help his six-year-old son learn Scratch coding.
@philking68



ADA POETRY GENERATOR

Ada Lovelace unveils the Analytical Engine in Scratch! This early computer looks a bit primitive, but can generate random poems

You'll Need

- Art assets - **magpi.cc/scratch_art**
- A list of words
- An eye for poetry

code club

This tutorial was adapted from a Code Club project (**codeclubprojects.org**) and you can find more in *Learn to Code with Scratch*: **magpi.cc/Scratch-book**

In this Scratch project, the user first chats to Ada, before clicking on her computer to generate a random poem. To achieve this, we'll be creating and using lists, found in the Variables block category, containing words of a certain type: verbs, nouns, adjectives, and adverbs. We'll then select randomly from these lists to create the poem, which should be different each time. They can be quite amusing.

>STEP-01

Prepare your artwork

After deleting the cat sprite as usual, you need to import the sprites and backdrop. Since they're not in the Scratch 1.4 library, you can download them (**magpi.cc/scratch_art**). As the Poetry backdrop is so simple – just a grey stripe at the bottom of a white canvas – you could paint it yourself, or just use ours by importing it from the folder where you've stored the downloaded graphics for this project. The same goes for the Banner sprite. Otherwise, import each sprite as usual, by clicking the star/folder icon above the Sprite List.

>STEP-02

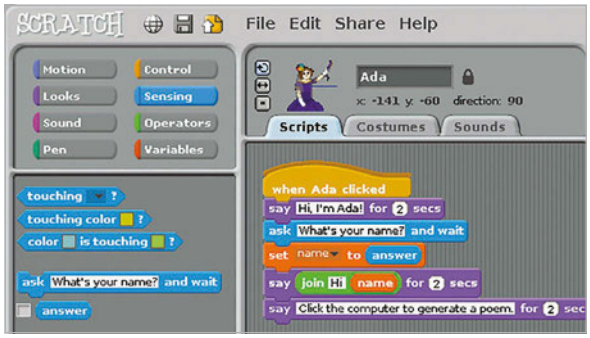
Ada says hello

First, we'll get our Ada sprite to interact with the user via speech bubbles and text input when clicked, using the **say** and **ask** commands. Open the Ada sprite's Scripts tab and then type in the code from **Listing 1**. Note that you'll first need to create a **name** variable: select the Variables block category from the top-left, then click 'Make a variable', 'For this sprite only', and enter 'name' in the text field. You should untick the **name** block to stop it showing on the stage. We can now set **name** to **answer** (the user's text input), and then add it into Ada's response by using the **join** Operator block. Make sure you put a space after 'Hi' to avoid it being joined together with the name. After this, we add a block to get Ada to tell the user to click the computer.

>STEP-03

Computer beeps

Click the Computer sprite and select its Scripts tab. This is where we'll add the workings of our poetry generator. To start with, type in the code from



Above: The script for Ada asks for the user's name, then says hi before telling them to click on the computer for a poem

Listing 2. After a block to say 'Here is your poem' and the user's name, we'll use a Sound block to make our computer beep. Our Computer sprite already has the sound for this, or you can record/import a new one in its Sounds tab. We also add a **repeat** loop with two **turn** blocks to make the computer shake.

>STEP-04

Create word lists

You can't make a poem without words. We'll store ours in four lists: **verbs**, **adverbs**, **nouns**, and **adjectives**. Create each of these in Variables by clicking the 'Make a list' button, then 'For this sprite only', and typing its name. It will then appear on the stage: to add words to it, click the '+' icon and type them in, one by one. When done, untick this list block to make it vanish from the stage. We used the following words for our lists:

Adjectives: happy, tired, hungry
Adverbs: loudly, silently, endlessly
Nouns: sea, moon, tree
Verbs: laugh, dance, burp

>STEP-05

Poetry in motion

Now we have our word lists, we can use them to generate a random poem each time the computer is clicked by the user. Join the code from **Listing 3** to the bottom of your existing script for the Computer sprite. It comprises four **say** blocks, each of which includes an **item of** Variables block; this should have 'any' selected from its drop-down menu, to make a random selection from the list. Test the project out a few times to check that it works properly and generates random poems.

>STEP-06

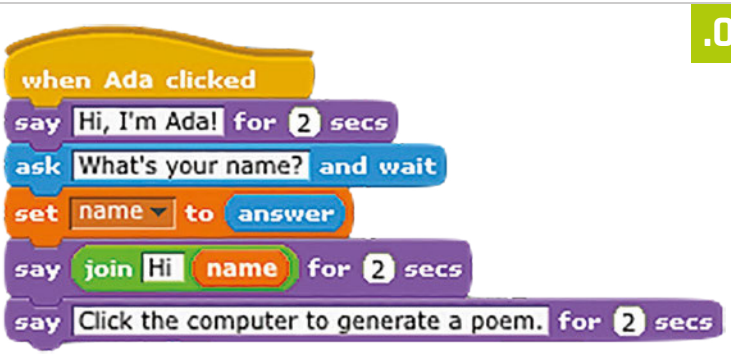
Taking it further

While we've only created short lists for this example, you could add lots more words to them for greater variation in the random poems created by the computer. More, and differently constructed, **say** blocks can also be added to make poems longer. If you're not keen on blank verse, why not create lists of rhyming words?

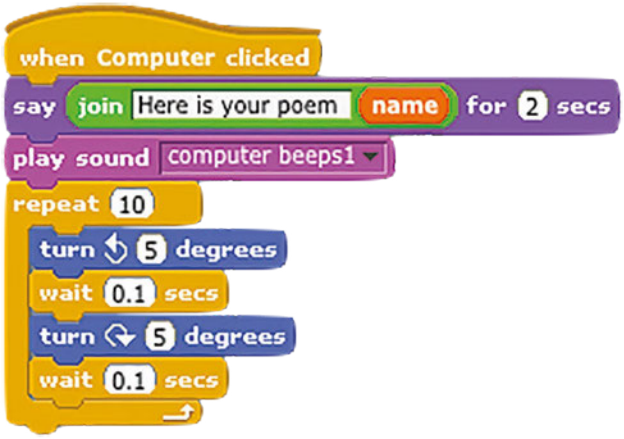


Above: To add words to each list, tick it to make it appear on the stage, then click its '+' icon

.01



.02



.03





WESLEY ARCHER
Self-taught Raspberry Pi enthusiast, founder of Raspberry Coulis, and guide writer for Pi Supply and Cyntech.
raspberrycoulis.co.uk
@RaspberryCoulis



You'll Need

- Picade PCB (**magpi.cc/29DpDCz**)
- Zippy ball-top arcade stick (**eBay**)
- 2.8mm & 4.8mm arcade daisy chain wires (**eBay**)
- Female A panel mount USB socket (**eBay**)
- 8× 30mm arcade buttons (**modmypi.com**)
- Various 12.7mm standoffs (**modmypi.com**)
- 4× M4 countersunk 16mm machine screws and bolts
- 3.5mm female-to-male stereo jack extension

BUILD YOUR OWN RASPCADE: CONTROLS



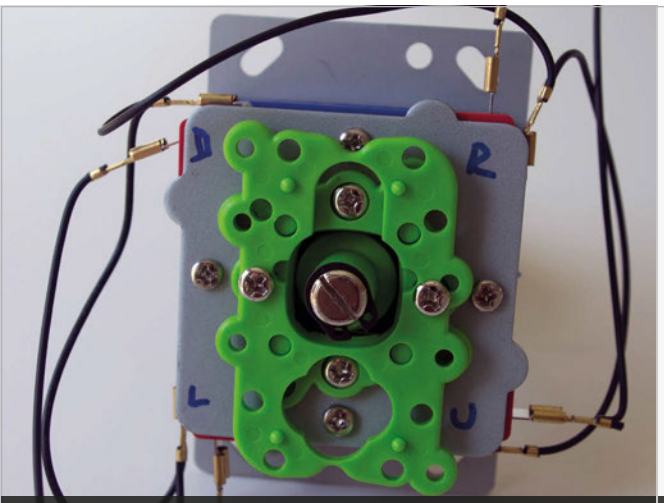
In this second part of the build, we'll be showing you how to assemble your controls as part of your RaspCade home-build!

Arcade controls can be mind-boggling, especially with all the wires involved. It is no surprise that people get confused, but there's no need to panic. The joystick and buttons are essentially switches with positive and negative terminals. Today we'll be going over wiring basics, and using Pimoroni's Picade PCB as a brilliant way of setting up your RaspCade controls quickly and easily. If you haven't already done so, buying dedicated arcade wiring looms or harnesses makes the whole process much easier because they are designed to be assembled quickly, without any soldering. Are you ready? Then let's get started!

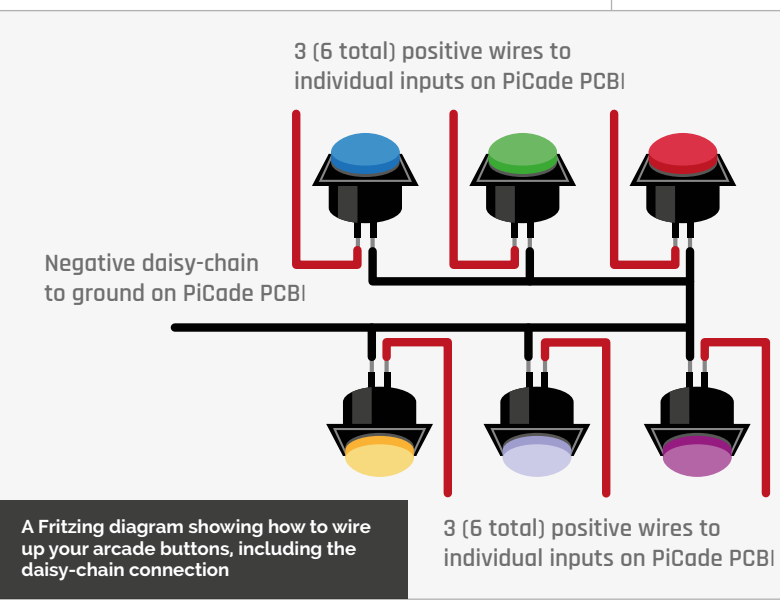
>STEP-01

Plan your wiring first

Before you put your controls in place, it is worth doing a test run first. Connect your wires and be sure you understand what goes where before placing them in your cabinet. The arcade wiring harnesses are perfect for this as the spade connectors simply slide on (and off) the buttons, so no soldering is required. Each button (including the joystick) has a positive and negative terminal. You can 'daisy-chain' the negative terminals (connect them together), which means you only have one ground wire instead of several. Once you're happy, carefully press the buttons into place: they are a tight fit!



It's worth labelling your joystick when wiring. The switch is in the opposite direction, as it is activated when the joystick moves in that direction



A Fritzing diagram showing how to wire up your arcade buttons, including the daisy-chain connection

>STEP-02

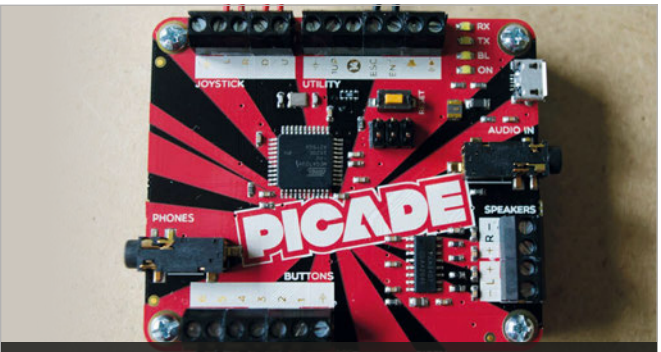
Bolt your joystick in place

If you're using our arcade cabinet design, the four holes should line up with the mounting plate on the joystick. Unscrew the ball-top and slide the plastic collar off, then bolt the joystick in place using four M4 screws and bolts. You can use a countersink drill bit to tidy this up if you like, but it isn't essential. Also, do not over-tighten the bolts as this could damage the cabinet. Once you have done this, replace the plastic collar and screw the ball-top back on. Now onto the wiring underneath...

>STEP-03

Better get yourself connected!

Connecting your controls is not as difficult as you might think. Grab your 4.8mm arcade harness and slot one connector onto each negative (black) terminal. Daisy-chain these so you have one wire connecting the four negative terminals. Next, use some wire cutters to cut and strip the end so you have a wire that can be connected to the Picade. You may need to extend this wire, but this can be done easily, even by a soldering newbie. Next, connect one wire to each of the positive (red) terminals. Each of these will be connected to the Picade, too.



The Picade PCB is well worth the money. Button inputs are clearly labelled, which makes setting up your controls a breeze!

>STEP-04

Push the button

Before wiring your buttons, carefully push them through the holes in the cabinet. They are a tight fit, so be gentle where you push to avoid snapping the cabinet. Once they are in, use the 2.8mm arcade harness and daisy-chain all the negative terminals (any can be used, but use the same one on each button) and strip the end again. Then connect one to each positive terminal so you end up with seven wires ready to connect to the Picade (six positive and one negative). We can now move onto the final two buttons on the front panel.

TIN THE WIRE ENDS

If the wires don't stay in the Picade PCB, tin their ends with a soldering iron to fatten them up.

>STEP-05

Assembling the front panel

The front panel consists of two arcade buttons, a USB port, and the headphone jack. The buttons pop into place the same way the others, and you also need to daisy-chain the negative terminals: you should have three wires (two positive and one negative). The USB port screws into place and the other end will be connected to our Pi. You can use this to connect a WiFi dongle, or a USB thumb drive to store all your games, making it simple to add more. The headphone jack screws into place, allowing you to play your RaspCade without disturbing everybody.

>STEP-06

Connecting to the Picade PCB

Connecting all the wires is really simple thanks to the Picade PCB. This has several screw terminals, all nicely labelled for our arcade controls. Simply find the relevant terminal and then screw the wire into place. You should have eight wires for the buttons and four for the joystick, as well as three ground (negative) wires. It isn't essential to put the wires in exactly the right place, so if your 'up' button is connected to the 'down' terminal, it can still be configured once in RetroPie. The Picade simply connects to the Pi via USB and that's it!

BE GENTLE WHEN ASSEMBLING!

The buttons are a tight fit and the cabinet is relatively thin. Be gentle when pushing the buttons into place.

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved...

NEED A PROBLEM SOLVED?

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

RASPBERRY PI CABLES

WHAT POWER CABLES DO I NEED FOR MY RASPBERRY PI?

Recommended power supplies

If you want to use the Raspberry Pi to its fullest ability, with add-ons like the Camera Module, Sense HAT and suchlike, the official Raspberry Pi power supply provides enough juice to get it all working properly: **magpi.cc/2a14pye**

Mobile phone charger

The Raspberry Pi is powered by a micro USB cable, the same one used by most mobile phones. The recommended minimum power for a Raspberry Pi is 2A (2.5A for a Pi 3), so check your charger to see if it will power it properly like the official power supply.

USB cable

For light operations, especially if you're running the Pi in command-line mode, you can power a Raspberry Pi from a computer's USB port. This isn't really recommended, though, so if you start having issues you'll need to upgrade to a proper power supply.



Right The official power supply for the Raspberry Pi provides very consistent power

WHAT AV CABLES DO I NEED?

HDMI

The HDMI port is your best bet for 1080p video and digital audio out from the Raspberry Pi. It's the easiest one to use and a lot of software is preconfigured to use that by default, such as Kodi, the media centre software.

Audio out

The 3.5mm jack can be used like the headphone port on music players, phones, and other PCs. However, you can also use it as an audio out to speakers that take auxiliary cables, or by using a 3.5mm-to-composite stereo output converter.

Composite video

On the Raspberry Pi A+, B+, 2, and 3, there isn't an obvious composite video out. However, the 3.5mm jack also allows for analogue video out if you have the correct RCA converter cable to use it. The original Pi models have a dedicated composite out, and the Zero needs a port soldered to it.

HOW DO I USE WIRED NETWORKING ON THE PI?

Ethernet port

On most versions of the Raspberry Pi, there's an Ethernet port next to the USB ports that you can use to connect to a wired network. In Raspbian, it will automatically connect to a network, although in rare cases you may need to specify an IP address for it.

USB Ethernet

The Raspberry Pi Model A, A+, and Zero don't include an Ethernet port. However, many USB-connected Ethernet ports can be plugged in and used in the same way as the standard Ethernet on other Pi models.

USB hubs with Ethernet

These are a bit more interesting: USB hubs that also act as an Ethernet port, allowing you to connect all your USB devices to something like a Pi Zero, as well as keeping it hooked up to the wired network.

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

What displays can I use?
Most Pi models have composite and HDMI outputs. So you can hook it up to an old analogue TV through the composite or a composite-to-SCART connector, or to a digital TV or to a DVI monitor (using a cheap, passive HDMI-to-DVI cable for DVI). The Pi Zero uses a mini HDMI port.

There's no VGA support, but active adapters are available. Passive HDMI-to-VGA cables won't work with the Raspberry Pi. When purchasing an active VGA adapter, make sure it comes with an external power supply. HDMI-to-VGA adapters without an external power supply often fail to work.

Why is there no VGA support?
The chip we use supports HDMI and composite outputs, but doesn't support VGA. VGA is considered to be an end-of-life technology, so supporting it doesn't fit with our plans at the moment. However, if you really want to use a VGA monitor with a Raspberry Pi, then it is possible using an HDMI-to-VGA adapter.

Does the HDMI support CEC?
Yes, the HDMI port on the Raspberry Pi supports the CEC standard. CEC may be called something else by your TV's manufacturer; check the Wikipedia entry on CEC for more information.

Can I add a touchscreen?
The Foundation provides a 7-inch capacitive touchscreen that utilises the Pi's DSI port; the screen is available through the usual distributors. Alternatively, several third-party retailers offer a range of touchscreens for the Raspberry Pi.

What codecs can it play?
The Raspberry Pi can encode (record) and decode (play) H.264 (MP4/MKV) out of the box. There are also two additional codecs you can purchase through the Swag Store: MPEG-2, a very popular and widely used format to encode DVDs, video camera recordings, TV, and many others; and VC-1, a Microsoft format found in Blu-ray discs, Windows Media, Slingbox, and HD-DVDs.

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.

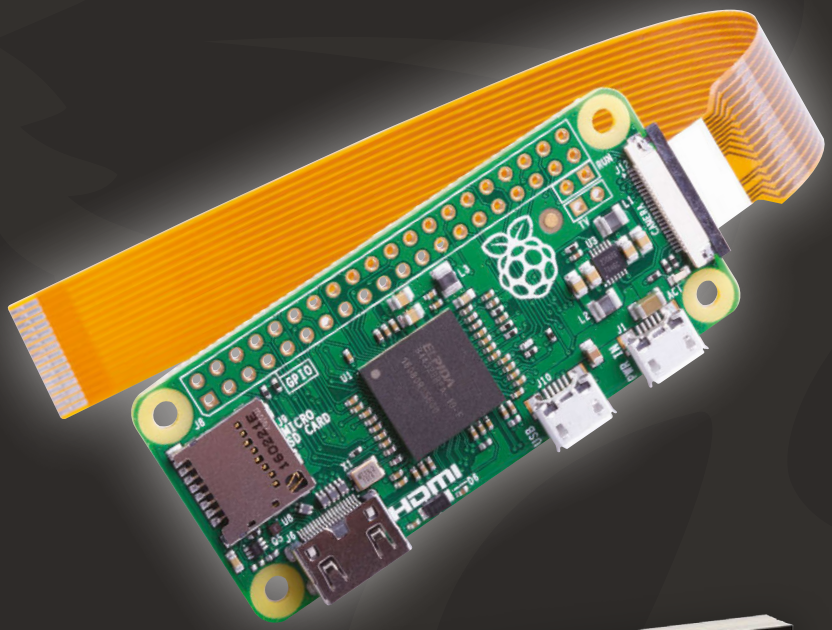


GET IT ON Google Play

Available on the App Store

FREE PI ZERO!

Subscribe in print for six or 12 months to receive this stunning free gift



Subscribe today & receive:

- A free Pi Zero v1.3 (the latest model)
- A free Camera Module connector
- A free USB & HDMI cable bundle

Delivered with your first issue!

Other benefits:

- Save up to 25% on the price
- Free delivery to your door
- Exclusive Pi offers & discounts
- Get every issue first (before stores)

SAVE UP TO 25%



Pricing

Get six issues:

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

FREE PI ZERO!

Subscribe for a year:

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

FREE PI ZERO!

Get three issues:

£12.99 (UK) (Direct Debit)
$37.50 (US) (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#48

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 ☐☐/☐☐/☐☐

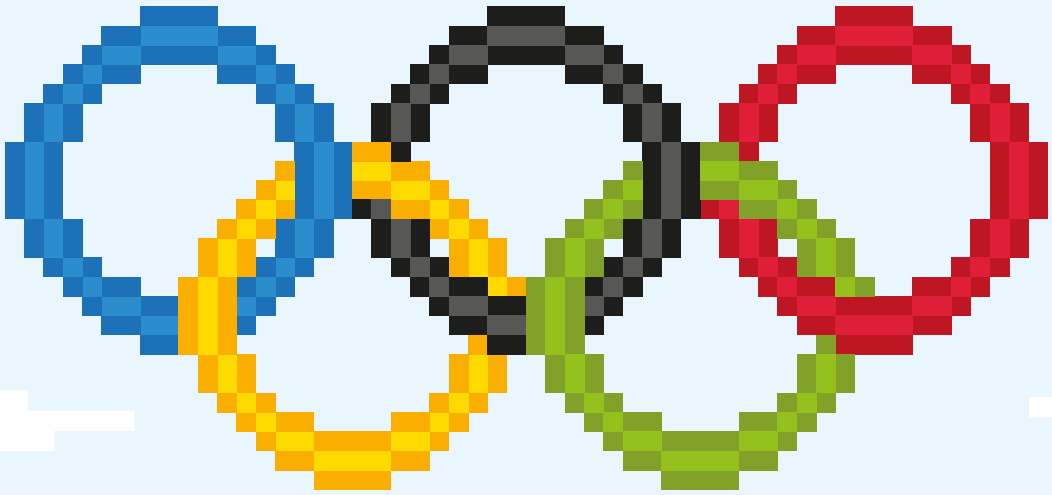
I would like my subscription to begin from issue ..... (month + year)

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

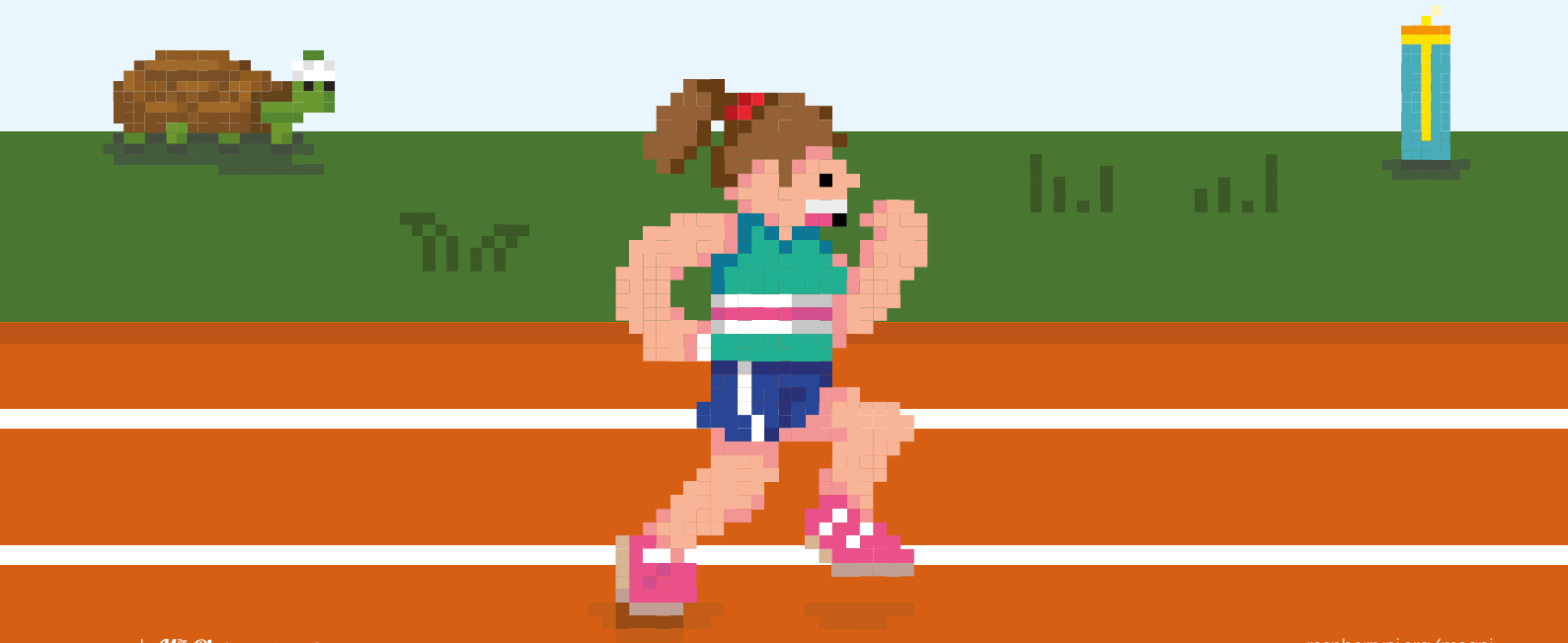
☐ Please tick this box if you DO NOT want to receive any other information from Select Publisher Services Ltd.

☐ 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.



SCRATCH OLYMPICS 2016



Want to check your work? Go to: **magpi.cc/ScratchOlympics**



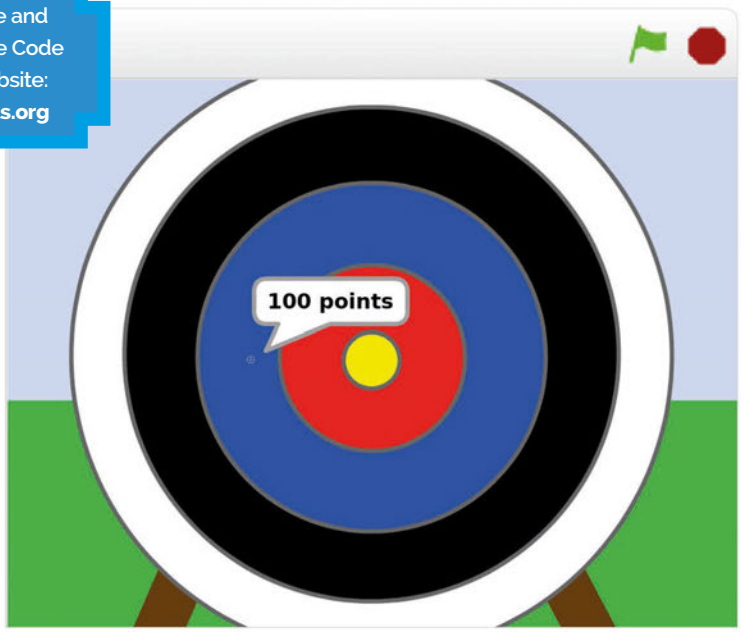
GO FOR GOLD WITH THESE PROJECTS THAT WILL TURN YOU INTO AN INTERNATIONAL CODING ATHLETE

The Olympics are upon us! That sporting spectacle of incredible skill, where the world's finest athletes come together to represent their country and show the world just how hard the human body can be pushed. If you're not an Olympic athlete, you can always watch it on TV, but if you want to take part in some way, we have a solution.

Using Scratch – the Pi version or online editor, depending on the project – we can create our own Olympic games in the comfort of our own homes and maybe, just maybe, it can prepare you for one day training up to become an Olympic athlete yourself.

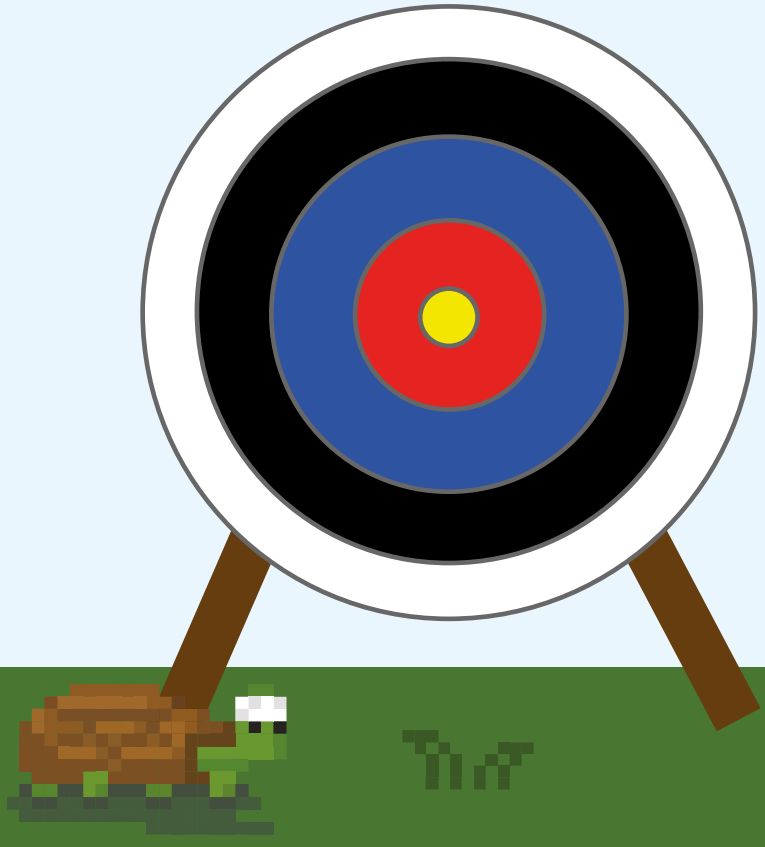
{code club}

Find this resource and more like it on the Code Club projects website: **codeclubprojects.org**



ARCHERY

Let loose a few virtual arrows and fulfil your fantasy of being Robin Hood



Let's start by creating an arrow that moves around the screen for this fun archery event. Note that since this project uses a block type not present in Scratch 1.4, you'll need to work on it in the online Scratch editor; to open the resources, go to **magpi.cc/29KO9S3** and follow this tutorial there.

When your game starts, broadcast a message to shoot a new arrow. Once this message has been received, set the arrow's position and size (**Fig 1**). Click the green flag to test your game; you should see your arrow get bigger and move to the bottom-left of the stage.

.01





Add the code circled in **Fig 2** to your arrow so that it glides randomly around the stage forever. Then test your game again, and you should see your arrow move randomly around the stage.

Shooting arrows

Let's code your arrow to shoot when you press the space bar (**Fig 3**). This will stop the other script, the one moving the arrow, when the key is pressed. Test your project again. This time, your arrow should stop moving when you press the space bar.

> When you press the space bar you should see your arrow get smaller, as if it's moving

Now we can animate your arrow by reducing its size, using a **change size** block inside a **repeat** loop. Test your game again. This time, when you press the space bar you should see your arrow get smaller, as if it's moving towards the target.



Once your arrow is at the target, you can tell the player how many points they've scored. For example, they could score 200 points for hitting the yellow. You can also play a sound if they hit that colour (**Fig 4**).

Finally, you need to broadcast the new arrow message again to get a new arrow.

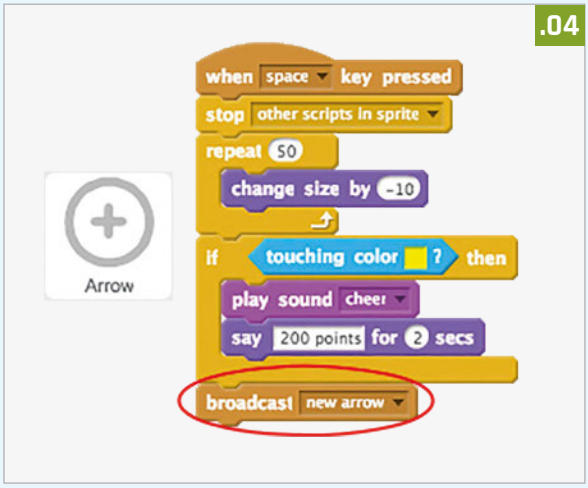
Event Info

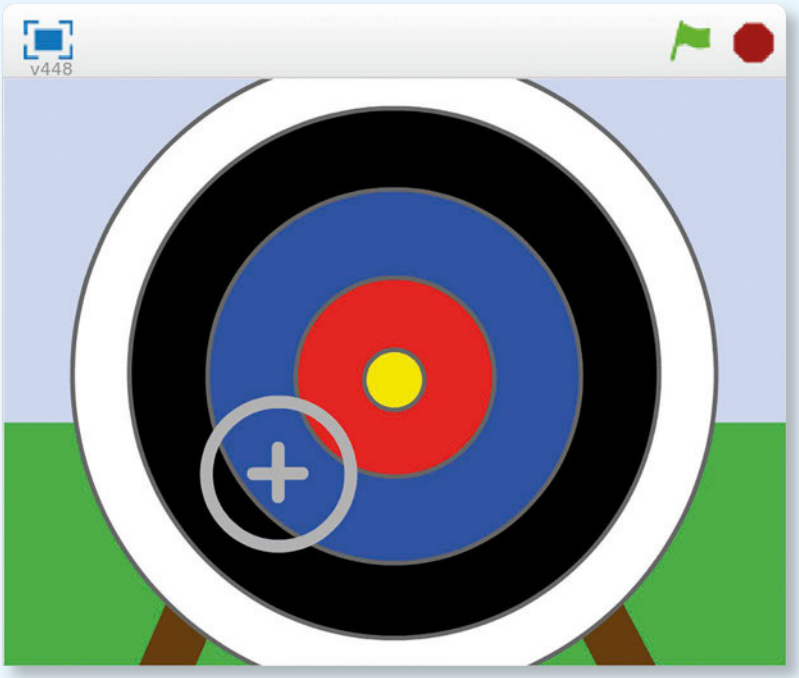
NAME Archery

VARIATIONS Individual, Team

INTRODUCTION 1900

DESCRIPTION
An ancient weapon-turned-sport of skill, the aim is to pierce the centre of a target for maximum points





WEIGHTLIFTING

Lift the heaviest thing you can to win big. Be careful with your keyboard!

In this activity, you will make a game in Scratch on the Pi that allows you to test your keyboard-mashing skills. By repeatedly hitting the keyboard, the character will lift their barbell high into the air. There will also be the opportunity to have a play with some physical computing, and introduce some arcade-style buttons into the game.

For this project, you're going to need a sprite and a background. You can download a zip file containing all the game assets by going here: **magpi.cc/29AVkj5**.

Once the file has been downloaded, you can unzip the archive by right-clicking on it and choosing unzip. You should see two directories: one containing the weightlifter's costumes and the other containing the background.

Event Info

NAME: Weightlifting

VARIATIONS: Snatch, Clean and Jerk

INTRODUCTION: 1896

DESCRIPTION:
The ultimate test in strength, divided by weight class to create a more equal playing field



Importing the assets into Scratch

Open Scratch by clicking **Menu > Programming > Scratch**. Now click on the **Stage** icon and then drag and drop the Olympics background into the **Backgrounds** tab. You can delete the original background. Next, click on **Sprite1** and then, one by one, drag and drop the Weightlifter costumes into the **Costumes** tab. You can delete the original costumes.

Testing the animation

You can check whether the animation will work using the simple script in **Fig 1**:



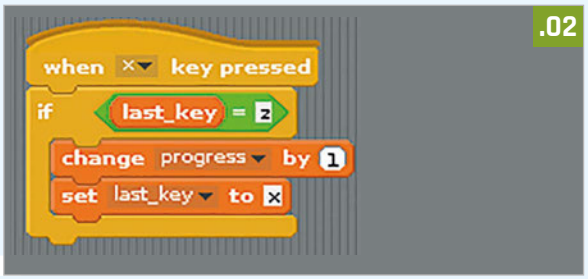
Click on the green flag and watch the weightlifter do his thing!

Capturing the speed of key presses

The progress the weightlifter makes is going to be controlled by the speed at which the player can hit the Z and X keys, so you need to create some scripts that will capture this data.

You're going to need two **variables** in this game. The first, called **progress**, will be used to record how far into the lift the weightlifter has managed to get. The second, called **last_key**, will be used to store the last key press the player made.

Create these two variables by clicking in Variables and then clicking on **Make a variable**. Start your script by setting **progress** to be **1** and **last_key** to be **x** when the game starts. The player must switch between hitting the X key and the Z key for the **progress** to increase. So when the X key is pressed, your script needs to check that the last key press was Z. If it was, then **progress** can be increased and the **last_key** can be switched to **x**. This is called **conditional selection**. The action only occurs if a variable is at the correct value (**Fig 2**):



Of course, you want the same sort of thing to happen when you press the Z key. You can duplicate the script by right-clicking on it and selecting **duplicate** from the context menu, then change the duplicated script to switch **x** for **z** and vice versa.

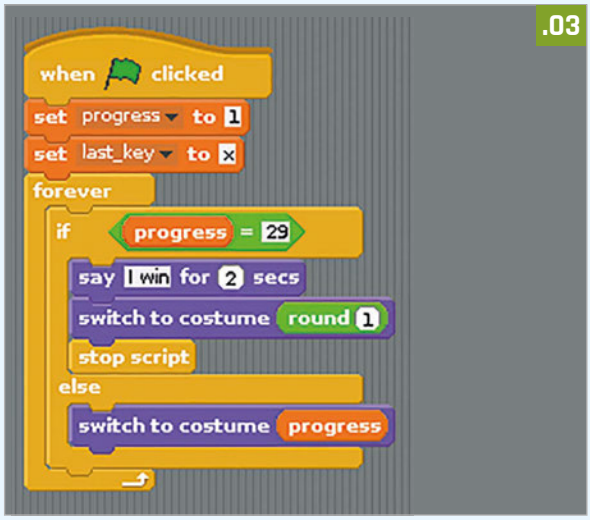
Test that your game works by clicking on the green flag, and then repeatedly hitting the X and Z keys on the keyboard. You should see the variable **progress** increasing. The faster you hit the keys, the faster it will increase.

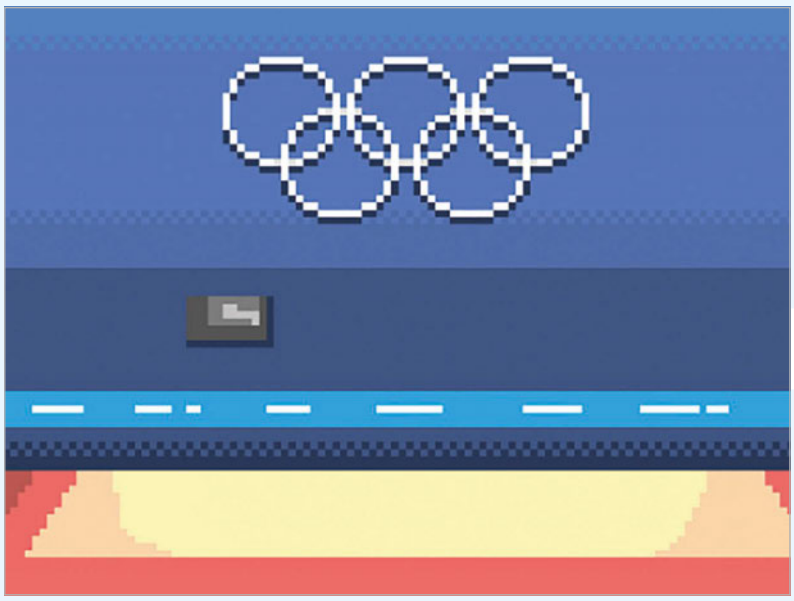
Making the character lift

There are a total of 29 costumes in the game. The sprite's costume can be continually set so that it's the same as the **progress** variable. That way, as **progress** increases, the costume will change. When **progress** reaches **29**, the game can end.

You will need a **forever** loop for the main logic of the game. Find a **forever** loop in the Control section and add it to the bottom of the first/main script. Now place another conditional block within the **forever** loop; this time, you can use an **if else** block. If **progress** reaches **29** then the sprite will **say "I win"**, the costume is set back to costume 1, and the script is stopped. If **29** has not yet been reached, then the costume can be set to the same value as **progress**.

You can't directly set a costume to a specific number in Scratch, but you can use the **round** Operator block to set the costume to its number (**Fig 3**):





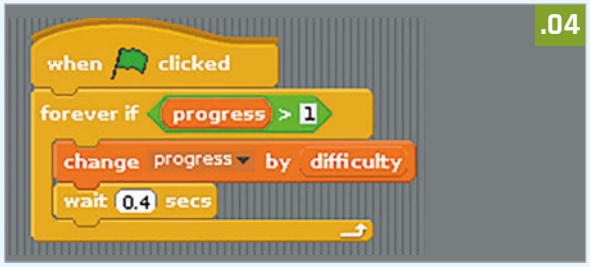
Have a go at testing your script. Click the green flag and then start hitting the X and Z keys alternately to watch the weightlifter go. It's a good idea to reset the costume back to number 1 each time the script starts:

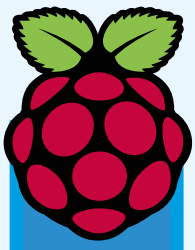
Making it a little trickier

If you stop hitting the X and Z keys, then the weightlifter just stops lifting. It would be good if he started to put the weight back down if the player's speed on the keyboard decreases. This can be done by decreasing the value of **progress** every once in a while. To start with, create a new variable and call it **difficulty**. This can be set to **-1** in the main script.

Grab a new **when green flag clicked** block and place it into the Scripts area. You can now use a **forever if** loop, which will run as long as a variable is at a certain value. You want it to run as long as **progress** is greater than **1** (**Fig 4**).

Inside the **forever if** loop, you'll need to keep changing **progress** by the value of **difficulty**. As **difficulty** is a negative number, this will keep reducing **progress** until it reaches **1**. You'll need to use a little **wait** command as well; since computers are so fast, there's no way a player could keep up with the computer otherwise. Waiting for **0.4** seconds will do to start with.





Find this resource and more like it on the Raspberry Pi resources website: **magpi.cc/1qEg9Nh**

Test your game to see the weightlifter pick up the weight as you hit the keys, but lower it again if you stop pressing them.

Looping the animation

Let's make the game a little more interesting now. You can start off by using a little loop effect at the start of the game. Pay attention to this next part, as you'll be using the same techniques later on.

The first four costumes can be looped to make the sprite look like he's getting ready to start. You'll need to get a **when I receive** block. Click on the

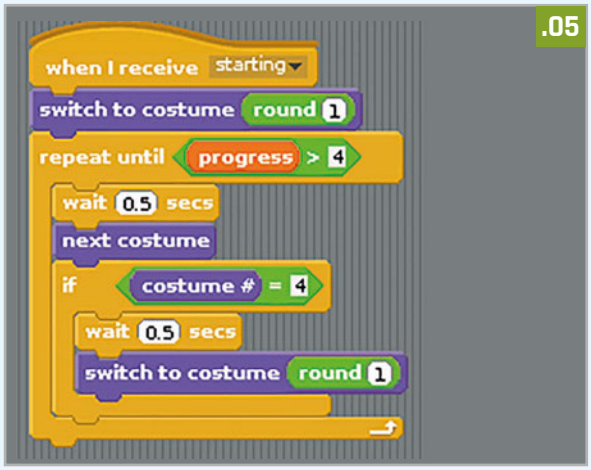
> See the weightlifter pick up the weight as you hit the keys, but lower it again if you stop

little black arrow and create a new broadcast called **starting**. The first thing that happens in **starting** is a switch to the first costume. This is followed by a **repeat until** block; you can make the code inside it run until the game begins. You'll know that the game has begun because **progress** will increase above **1**. Use a **>** operator from Operators to help you build the script (**Fig 5**).

You can change the costume inside this loop. If the costume reaches number 4, then it needs to be reset back to costume 1 again. You'll need a couple of **wait** blocks as well, so you can actually see the costume changing.

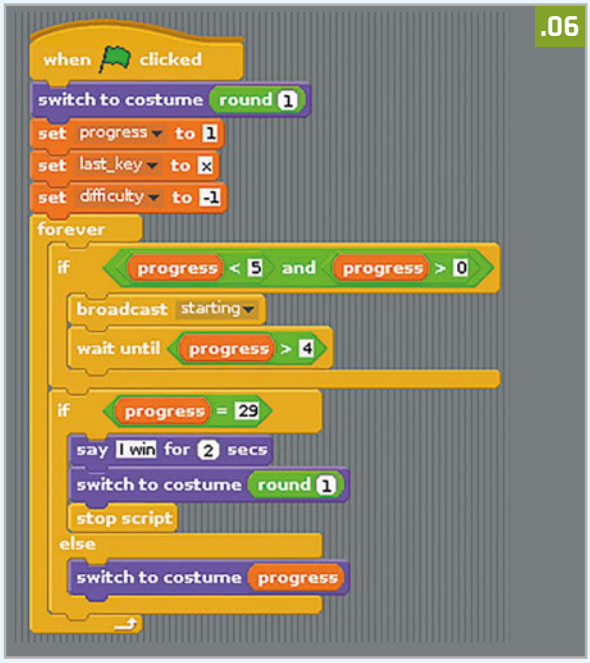


.05



If you click on this block, it should have a halo of white around it, and the sprite should start looping. To finish this section off, the **starting** block needs to be triggered from within the main script (**Fig 6**) whenever **progress** is **1**, **2**, **3** or **4**, or in other words, **<5** and **>0**. If it is, then **starting** can be triggered and the script can wait until **progress** is greater than **4**. This all goes into the main script.

.06



Run your script and the weightlifter should glance left and right. When you start hitting the X and Z keys, he should start to lift. If you stop hitting the keys, he'll return the weight to the floor and then start glancing left and right again.

Adding in a strain stage

You can add a second difficulty level by playing around with the timings and how they affect the costume; go to **magpi.cc/29MuJPv** to find out how to complete it!

HURDLES

What's more exciting and challenging than running on a track? Adding some obstacles to jump over!

In this activity you will make a hurdles game using Scratch, where the speed of the runner is controlled by how fast you can hit the X and Z keys, and perfect timing is required to jump over the hurdles at exactly the right time. Download the zip file to your home directory and unzip its contents: **magpi.cc/29NUFYk**.

Open Scratch by clicking on **Menu > Programming > Scratch**. Now, click on the **background** icon and import the new background from the **assets** directory. You can then delete the old background.

Click on the icon to import a new sprite and then choose the **run-1** image. Next, import **run-2**, **run-3**, and **run-4** as additional costumes. You can then delete the old cat sprite.

Event Info

NAME Hurdles

VARIATIONS 100 metres, 110 metres, 400 metres

INTRODUCTION 1896

DESCRIPTION
Get to the end of the course as fast as you can, but remember to jump over the hurdles in your way

Capturing the key mashing

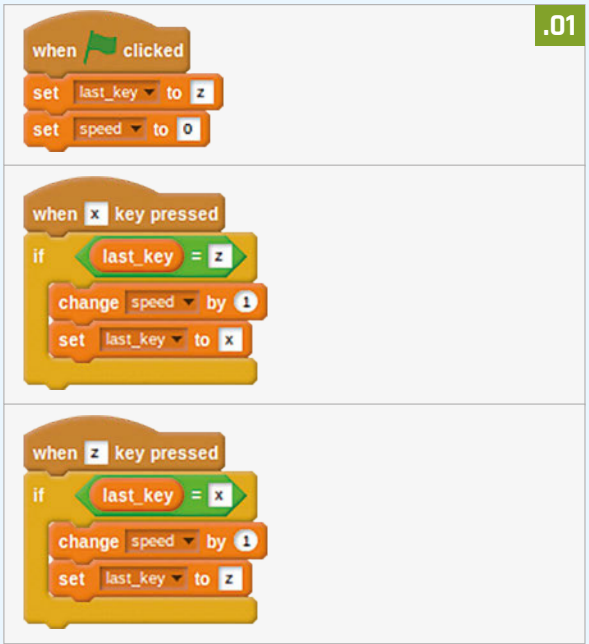
The first step is to capture the X and Z key presses, and use the speed at which the player is pushing the keys to control the size of a variable. To do this you'll need a variable that stores the last known key press. Create a variable called **last_key** and set it to **z** when the green flag is clicked.

For the next script you'll need a new variable called **speed**, so go ahead and create it now. It can be set to **0** when the game begins (**Fig 1**).

When the X key is pressed, if the **last_key** is equal to **z**, then the **speed** variable can be increased and the **last_key** can be set to **x**. This will ensure that the player can't cheat and keep hitting the X key to make the speed increase. The same can be done for the Z key. In combination, these two scripts force the player to hit the keys alternately in order to increase the **speed** variable.

Now test your script. Click the green flag, then repeatedly press the X and Z keys and watch the **speed** variable increase.

.01



Animating the hurdler

At the moment, the hurdler has four costumes as part of what's called a walk cycle (or run cycle in this case). When these costumes are switched, the character appears to run on the spot. The time delay between costume switches should depend on the **speed** variable. The higher the **speed**, the quicker the costume change should be and therefore the smaller the delay. You can get this effect by dividing **1** by the **speed** variable to calculate a delay.

If you run this script as it is, you'll get an error, because **speed** starts off with a value of **0**. This means

the computer is trying to calculate 1 / 0, which it can't do. It's a very common error that programmers make in their code. To fix this, you can use a conditional to make sure that the calculation only occurs when **speed** is larger than **0** (**Fig 2**).

.02

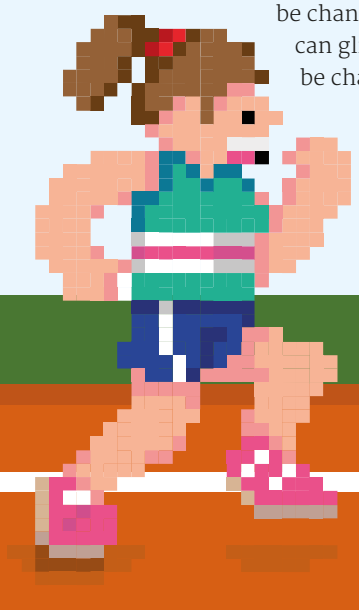


Now you should be able to test your script and watch the hurdler running on the spot as you press the X and Z keys.

Get jumping

Hurdlers need to jump. You'll need a few more costumes for this part, so look in the **runner** directory and import the **jump-1** and **jump-2** costumes for your hurdler. You'll need a new variable for this part called **jumping**. This is because other scripts will need to know when the character is jumping. Create the new variable on the **green flag** block and set it to **False**.

The character should jump when the space bar is pressed. The first thing that happens is the **jumping** variable should be set to **True**, then the costume can be changed to **jump-1** and the character can glide upwards. Next, the costume can be changed to **jump-2** and the character can glide back down again. Finally, the **jumping** variable can be returned to **False** to indicate that the jumping animation has finished (**Fig 3**).

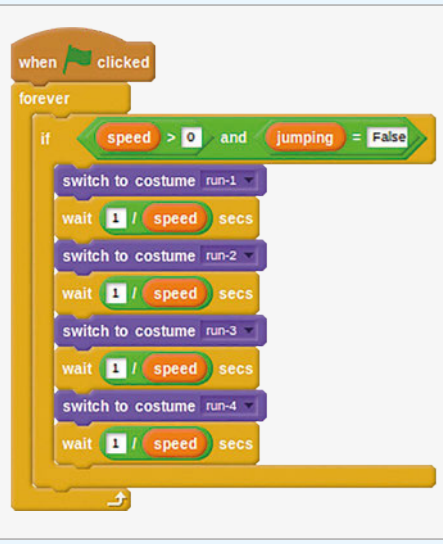


.03



Test your script; it might surprise you to see that the character's costume doesn't change. This is because the walk cycle you set up previously is still working. You'll need to stop this walk cycle when the character is jumping. To do this, you can use an **and** conditional operator to check that both **speed > 0** and **jumping = False** for the walk cycle to work (**Fig 4**).

.04



Now have a go and you should find your character jumps when the space bar is pressed.

Slowing down

At the moment, the more you press the X and Z keys, the faster the character runs. There needs to be a way of slowing the hurdler down, so she doesn't win too easily. This can be done on your initial script that sets the starting variables. You just need to add an infinite loop that will check if the **speed** is greater than **1**, and then lower it every few 100ths of a second (**Fig 5**).

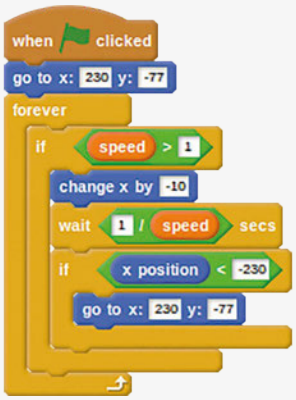
.05



Adding in hurdles

For the final part of this tutorial, you can add in hurdles that the character will have to jump over. Import the **hurdle.png** sprite from the **assets/items** directory. This sprite needs to begin at the far right of the screen, then it should continually move left across the screen at a pace that's proportional to the speed of the character. When it hits the far left of the screen, it should instantly appear on the right again (**Fig 6**).

.06

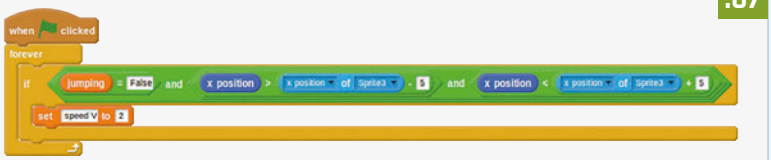


Making the hurdles an obstacle

At the moment, the runner can just plough straight through the hurdles. She needs to be slowed down if she doesn't jump.

Back on the hurdler sprite, add in a new **when green flag clicked** block. This next part is a little complicated. The runner should be slowed down if she isn't jumping, has an x position just before the hurdle, or has an x position just after the hurdle (**Fig 7**).

.07



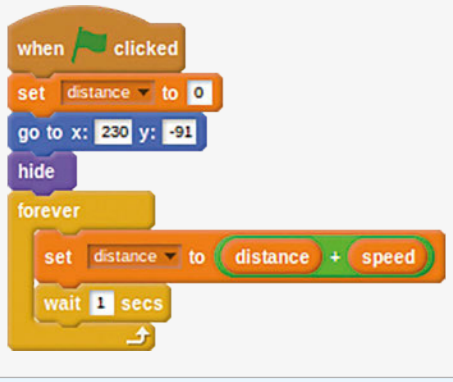
This can be achieved using two **and** logical operators, checking if **jumping = False**, that **x position > x position of hurdle – 5**, and **x position < x position of hurdle + 5**. If all those conditions are met, then she must have hit the hurdle and her speed can be dropped.

Making an end to the game

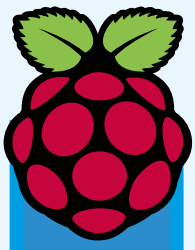
To finish off a completed game, you need to add in a finishing line. You can find one in the **assets/items** directory. Import this as a new sprite into your Scratch game, and approximately position it into the runner's lane.

To start off, you need to use a variable to control how far the hurdler has to run. Create a new variable and call it **distance**. The first script to be added to the finish line will set **distance** to **0** when the game begins, position the finish line to the far right of the screen, and hide it. Next, **distance** has to be increased by the **speed** of the runner every second (**Fig 8**).

.08



Now that the finish line is ready to go, you can make it appear when the value of distance hits whatever value you desire (200 in this example). It can then begin to move across the screen towards the hurdler. When she touches the finish line, detected by **touching color**, all the game scripts should end – see **magpi.cc/29WZV9f**.



Find this resource and more like it on the Raspberry Pi resources website: **magpi.cc/1qEg9Nh**



SYNCHRONISED SWIMMING

Swim to the beat and wow the judges with your stylish water moves

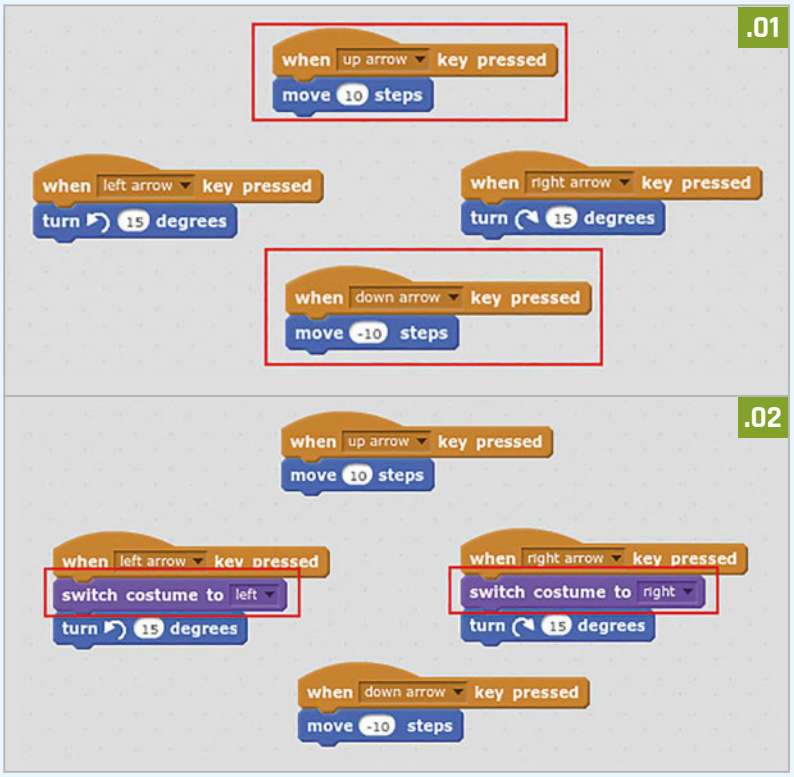
Event Info

NAME: Synchronised swimming

VARIATIONS: Team, Duet

INTRODUCTION: 1984 Summer Olympics, Los Angeles, USA

DESCRIPTION:
Originally known as water ballet, it involves team of swimmers performing a coordinated routine to music



Let's start by getting one cat swimming. Note that since this project uses a block type not present in Scratch 1.4, you'll need to work on it in the online Scratch editor. First, let's turn the stage blue, like a swimming pool. Click on **Stage**, then the **Backdrops** tab. Choose a blue colour from the palette; click the 'Fill with color' tool, then click the backdrop.

You're going to use a different cat sprite, so right-click on the walking cat to delete it. Now click 'Choose sprite from library', select **Animals**, choose **Cat1 Flying** and click **OK**.

Now let's get the cat swimming. Click on the **Cat1 flying** sprite, then **Scripts**, and add code to make the cat rotate left and right when you press the left and right arrow keys (**Fig 1**). Test your code by pressing the left and right arrow keys on the keyboard, and add forward and backward movement. Test your code by swimming around the stage using the arrow keys.

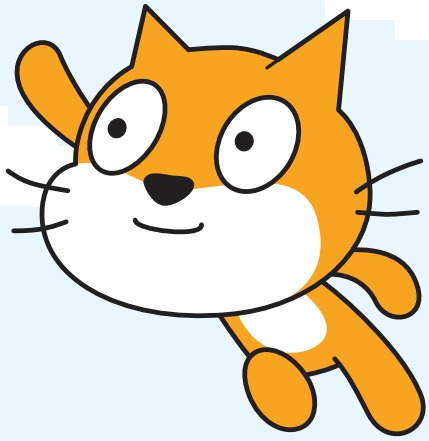
Changing costume

This would look better if the cat sprite changed direction when it turns left. Click on **Costumes** and delete the **cat1 flying-a** costume. Rename the remaining costume from **cat1 flying-b** to **right**. Right-click on the costume and choose **duplicate** to create a copy, then click the 'Flip left-right' icon to reverse the copy and name it **left**.

Click **Scripts** to return to your code and add blocks to change the costume when the direction is changed (**Fig 2**). Test your code by swimming around the stage using the arrow keys.

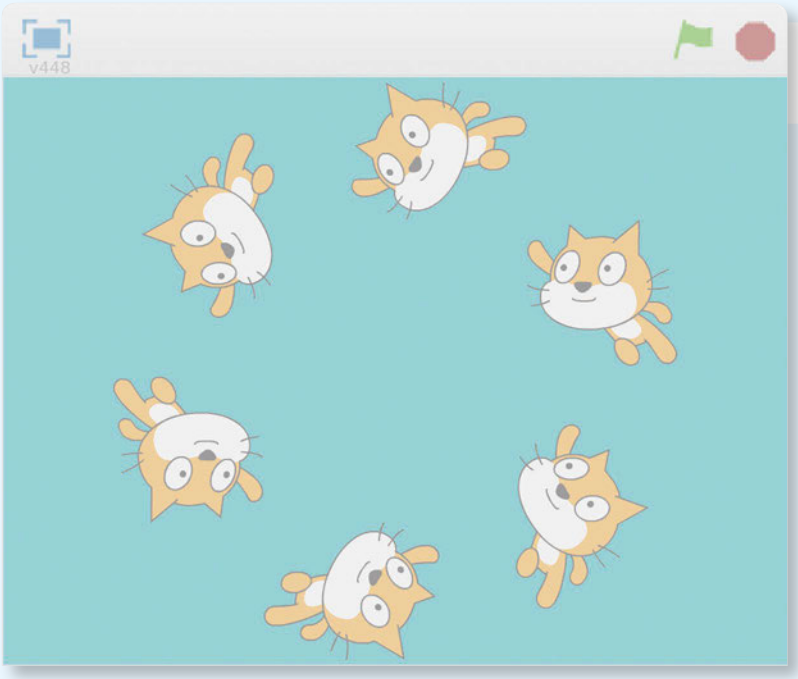
Create the team

Synchronised swimming needs more than one cat! We can use `create clone of` to make copies of the cat sprite that behave in the same way. First, let's add code to make sure the cat always starts in the same position when you click the green flag (**Fig 3**). Test your code by pressing some arrow keys and then clicking the green flag to return to the start position.

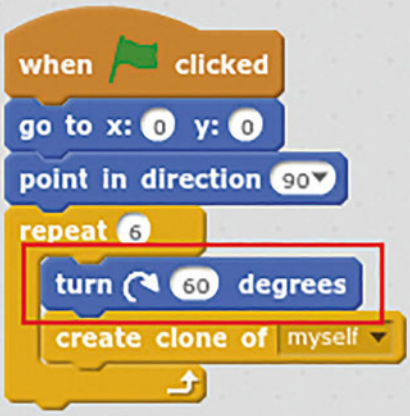


code club

Find this resource and more like it on the Code Club projects website: **magpi.cc/29KATgh**



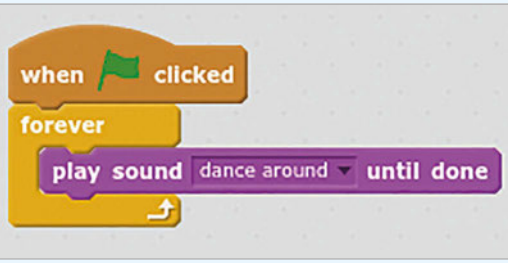
.03



Now we can use a **repeat** loop to create six clones (copies) of the cat. Loops are used to do the same thing multiple times. You don't want all the cats to be in the same position, though, so add code to rotate 60 degrees before creating each clone.

Test your code by using the arrow keys. You should be able to create some amazing synchronised swimming patterns!

.04



Music!

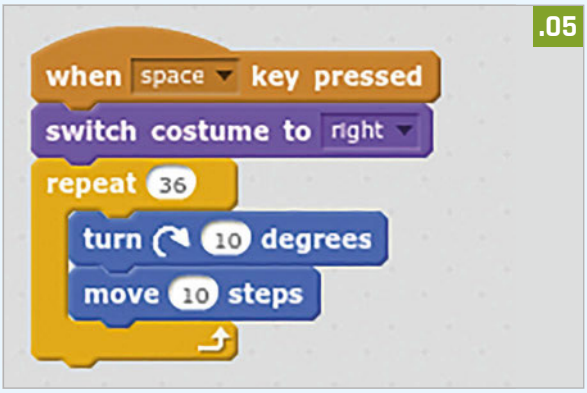
A synchronised swimming routine needs music. (But if you can't play sound then you can skip this step.) Click on the **Sounds** tab and then click 'Choose new sound from library'. Select **Music Loops**, choose some music, and click **OK**. Then go back to **Scripts** and add the blocks to play your music (**Fig 4**).

Putting the **play sound** block inside a **forever** loop means the music will keep repeating. You can now test your project again; remember, you can click on the red stop button to stop the music playing!

Programmed routines

Would you like to be able to perfect a routine and easily repeat it? Let's add some moves to be performed when the space bar is pressed (**Fig 5**). Run your project and press the space bar to test the new routine. Try using the arrow keys to move to a different position before pressing space.

.05





WILDLIFE CAM KIT

Capture wildlife photos with this weatherproof, Pi-powered camera trap

Related

CAMDEN BOSS ENCLOSURE

This mini, wall-mountable acrylic case will protect just the Camera Module, not the Pi, and isn't weatherproof.



£9 / $12

magpi.cc/29Hsnh0

Ever wondered what kind of critters visit your garden whenever you're not around to scare them off? With the Wildlife Cam Kit, you can find out. Its PIR sensor will sense any movement in the vicinity and trigger its Pi Camera Module to take a stealthy snap of whatever's passing by.

You may recall that we've followed the progress of this Kickstarter-funded project in previous issues of *The MagPi*, but now it's finally out in the wild. Designed by Naturebytes, a trio of digital makers and wildlife enthusiasts, its aim is to give users a fascinating insight into the natural world while also enhancing their digital making skills. To this end, it comes in kit form, although no soldering is required. It takes an hour or so to put together, following the detailed online PDF instructions (**magpi.cc/29HamiP**). The latter are well-illustrated with plenty of photos, even if a couple were slightly misleading.

Constructing your camera

A laser-cut plastic insert is provided to suit whichever Raspberry Pi model you're using; the standard kit comes with an A+ due to its lower power usage, but it could even be used with a new Pi Zero v1.3. Screws and plastic spacers are supplied to fit most of the components – including PIR sensor, Camera Module, and Pi – to the insert, threading jumper wires through its strategically placed holes. It's clear that much thought has gone into its design; you even get bendy ties to push through pairs of holes to secure the wires tidily. Still, it wasn't so easy to fit the jumper wires to the Pi's GPIO pins through the large cut-out, requiring a small screwdriver to push them into place. A real-time clock module is also fitted to five of the pins (since the Pi doesn't have one built-in), to enable accurate date/time-stamping of photos without an internet connection.

Finally, an Adafruit Powerboost is fitted to the insert and connected to a LiPo battery. This is to boost the latter's 3.7V output to the 5.2V needed to power the Pi and other components. A power switch has been added to the Powerboost to make it easier to switch the camera on and off when outdoors. Before you can use it, however, you'll need to charge up the battery, which takes around 17.5 hours; it might be advisable to start doing this before assembling the rest of the kit. Naturebytes claims it will provide around 30 hours of power; in our tests it lasted up to three days between charges.

magpi.cc/2902evf

From £140/$185



Maker Says

"A camera that anyone can build to take stealthy high-definition images of wildlife"

Naturebytes

Testing it out

Before positioning the camera trap outdoors, it's a good idea to hook it up to a TV to check everything's working properly. The supplied SD card features a customised version of the Raspbian operating system, featuring a Naturebytes logo and desktop shortcuts for various tools, including PIR and camera tests. The main Python script for taking photos can also be tweaked. By default, it adds graphical overlays – a date/time stamp and Naturebytes logo – to each photo taken. However, we found that this was causing a lengthy delay (of up to two minutes) when writing to the supplied USB stick, possibly caused by a lack of RAM in the 256MB Pi A+ to handle the I/O process. Naturally, this is undesirable when capturing wildlife photos, as the critter may have vanished in the meantime. Fortunately, commenting out the code lines for the overlays solved the problem. Writing to an SD card is even quicker, although it's far more convenient to simply remove the USB stick and plug it into another PC in the house to view the photos, without having to bring the whole device back inside.

Venturing outdoors, a Velcro strap is used to fasten the Wildlife Cam to a tree or fence post. The green acrylic case itself seems very tough and durable and snaps tightly shut, secured by two plastic clamps. Naturebytes has tested it in all weather conditions, and we experienced no leakage whatsoever following a downpour. On the front, a circle of clear plastic protects the Camera Module, while another translucent cover guards the PIR sensor.

Start snapping

Once in position, toggling the Powerboost switch turns on the camera and the picture-taking Python script will start running automatically. Our first attempt yielded a shot of a wood pigeon in flight, plus numerous photos of nothing: false positives triggered by the changing amount of sunlight. This effect can be reduced by repositioning the camera to avoid morning or afternoon glare, or by adjusting the sensitivity of the PIR. You can lure wildlife in front of the camera trap by placing it next to a bird table – we improvised by pouring some seeds onto an old garden table and managed to obtain a great shot of a magpie! There's also an optional wooden arm to hang a bird-feeder in front of the camera.

Whatever tactic you use, viewing the photos at the end of the day is an exciting prospect, just to see what has been captured. While not ready at the time of writing, Naturebytes is also working on an online community hub where users can share their wildlife photos, which should add an extra dimension to the project. Ideal for educational use, the Cam Kit is also quite versatile and could be used for time-lapse photography, night-time shots (with a Pi NoIR camera and IR LED lighting), or even a live video feed.

Last word

While more expensive than originally envisioned, the Wildlife Cam Kit is easy enough to assemble and fun to use. Once we'd sorted out the photo-taking delay issue, it worked well outdoors and the anticipation of seeing what it had captured was exciting. That lengthy battery recharge time is a little annoying, but it does power the unit for quite a while. Most importantly, the weatherproof case is very robust and may soon also be available as a standalone unit.

★★★★☆

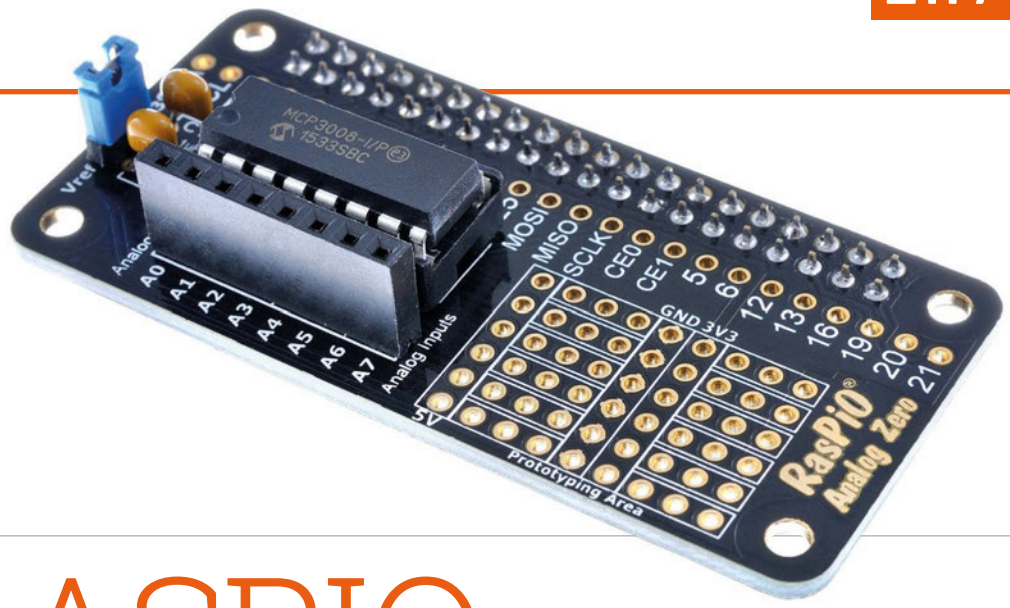
rasp.io/analogzero

£11 / $15

Maker Says

"Read up to eight analogue inputs at once"

RasPiO



RASPIO ANALOG ZERO

It makes the reading of analogue sensors as easy as Pi

While its mini form factor makes the Analog Zero a perfect partner for the Pi Zero, it's a great way to add easy-to-use analogue inputs to any Raspberry Pi model. Supplied as a kit, it's based around the MCP3008 analogue-to-digital converter (ADC) chip, but avoids all the intricate wiring usually required when using an ADC. The great thing about using this particular chip is that it's already supported by the GPIO Zero Python library with its own class, so it's a doddle to start writing programs to read and compare up to eight analogue inputs at once. Just use jumper wires to hook up your analogue sensors – temperature probes, light-dependent resistors, humidity sensors, gas detectors, potentiometers etc. – to any of the eight inputs in a female header, then write a few lines of code to get instant readings. Voltages up to 3.3V can be read directly; if the input is higher, you'll need to use a voltage divider made from resistors. Potential projects include a digital thermometer, voltmeter, and weather station, and kits for all of these were offered as part of the Analog Zero's successful seven-day Kickstarter campaign.

If you require greater accuracy than the MCP3008's 10 bits, you always have the option of swapping it out for your own 12-bit MCP3208 ADC chip for extra precision, since it fits the same socket and is also supported by GPIO Zero. Even so, the MCP3008's 1,024 steps should be enough for most projects.

Although the Analog Zero makes things easier once assembled, note that you do have to do a bit of soldering beforehand, but everything's well marked out on the board. As well as the chip socket, you'll need to solder on the small female header for the analogue inputs, a couple of capacitors, a jumper switch, plus a 40-way female header to connect to the Pi's GPIO port. Handily, the board features through-holes for 25 GPIO pins, along with a mini 54-point prototyping area. There's also the option to create a sleeker version by soldering the chip directly to the board and using surface-mount capacitors on the rear.

Last word

We'd have appreciated a pre-assembled option, but once you've soldered the kit components onto the board, the Analog Zero really does make it much easier to use multiple analogue inputs for projects, particularly when using GPIO Zero.

★★★★☆

Related

AD/DA EXPANSION BOARD

Based on the PCF8591T 8-bit ADC chip, it has four analogue inputs, WiringPi support, and can also do DAC conversion.

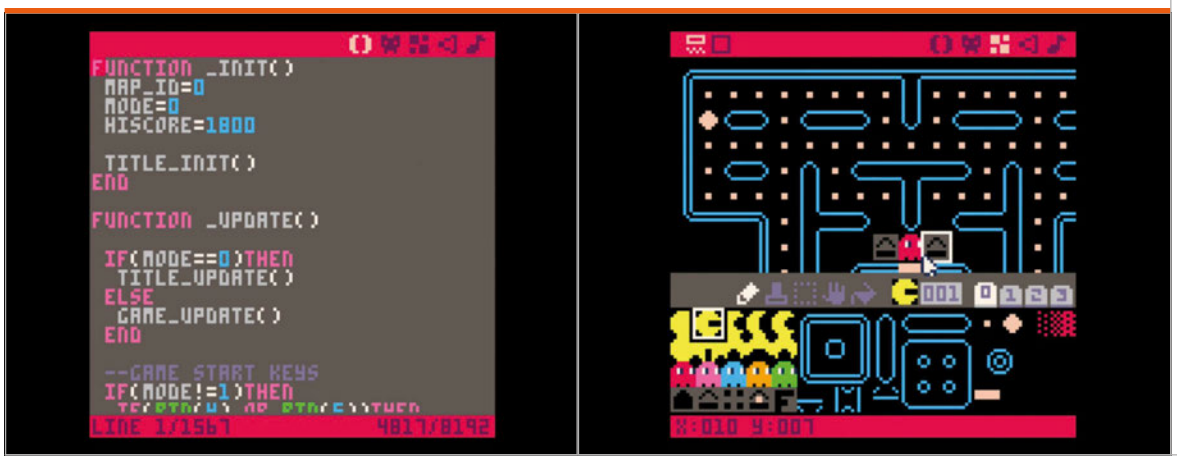


£9 / $12

magpi.cc/1syGYDs

lexaloffle.com

£12 / $15



Maker Says

"A fantasy console for making, sharing, and playing tiny games and other computer programs

Lexaloffle

PICO-8

Build, play, and share 8-bit games with this imaginary console with built-in code-, sprite-, and music-editing tools

Imagine the best console from the 1980s that you never owned. As well as being able to play free games, you would be able to stop them at any time and dive backstage.

You could tweak the code to make it easier, harder, or different. The built-in sprite editor and sound effects tools would let you customise the graphics. And there'd be a level editor for building new stages or changing existing ones.

That's PICO-8. It's a virtual games console, but instead of being an actual 8-bit console, it recreates a virtual console with built-in editing tools.

PICO-8 was developed by Joseph White from his base at Pico Pico Cafe in Kichijouji, Tokyo. It's been around for Windows, Mac OS X, and Linux machines, but feels more naturally at home on the Raspberry Pi, where it can become the console it was meant to be.

PICO-8 runs just fine on the Raspberry Pi hardware. We'd be surprised if it didn't, because a key aspect is a deliberately limited feature set. The display is just 128×128 blocks with 16 colours. Sprites are 8×8 pixels, and you can only have 128 on screen at once.

The virtual cartridges (saved as PNG files) are limited to 32kB, and it has a four-channel sound chip.

Imagine it sitting alongside a NES or PC Engine in terms of technical prowess. These deliberate limitations make PICO-8 more engaging. The games have a retro-cool aesthetic and they're easier to build. The low-spec nature of the console helps newcomers get started.

Code is written in Lua. It's a relatively simple language to learn and mostly used for scripting Adobe programs. While it's no Python, Lua is worth learning. Games can be shared in a web browser, so you can see what sort of thing is possible on the Lexaloffle website.

Community service

A vibrant community has sprung up around PICO-8, with one of the most active fan bases we've seen. The forums are packed with interaction between developers, with everybody chipping in and offering to help.

If there's one downside, it's that PICO-8 is a paid-for program. But we think it's great value given the amount of fun we had, along with the active community.

Last word

We had a huge amount of fun with PICO-8, and it's a natural fit for the Raspberry Pi.

★★★★★

Related

PYGAME

Pygame is a more powerful set of gaming modules that uses Python. It's not got the same holistic environment as PICO-8, though, but you can build some impressive games with it.



Free

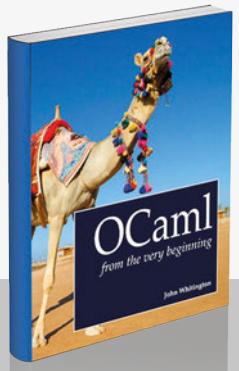
pygame.org

RASPBERRY PI BESTSELLERS

OCAML

Combines a static type system and functional first design with a type-inferred object system.

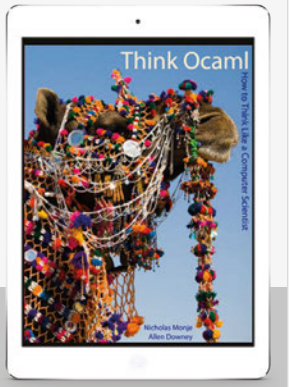
OCAML FROM THE VERY BEGINNING



Author: John Whitington
Publisher: Coherent Press
Price: £24.99
ISBN: 978-0957671102
ocaml-book.com

John Whitington's introduction to OCaml as a modern general-purpose programming language will get any newbie up to speed, and its exercises will deepen your understanding.

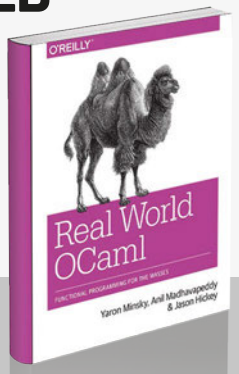
THINK OCAML



Authors: Allen Downey and Nicholas Monje
Publisher: Green Tea Press
Price: Free download
greenteapress.com

OCaml as a first programming language? Yes, with this rewrite of *Think Python* (see last month), for functional programming. A free download that's well worth the time you'll need to invest.

REAL WORLD OCAML



Authors: Anil Madhavapeddy, Yaron Minsky and Jason Hickey
Publisher: O'Reilly
Price: £26.50
ISBN: 978-1449323912
oreil.ly/29UoLGY

For confident beginners, or those with a little experience in OCaml or any other language, this is a deep dive into the possibilities of this functional programming language.

MAKE: FUN!



Author: Bob Knetzger
Publisher: Maker Media
Price: £16.50
ISBN: 978-1457194122
oreil.ly/29UpdVL

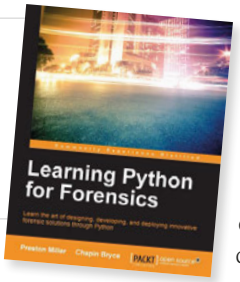
As the summer holidays stretch ahead, with the inevitable promise of rainy days and bored kids indoors, it's time to turn to some quick and simple toy making. Independent toy inventor Bob Knetzger passes on his passion for making and inventing through a series of small projects to show just what's possible when a little imagination is applied to a few everyday objects.

Knetzger's projects get you learning by making, giving the reader the skills and tools needed to finish a Raspberry Pi project's casing and for physical interactions. While some of the parts listed will take a little research for European readers to translate from American trade names and store references, almost everything is readily available, and no project here is beyond a keen beginner on a budget.

Starting with kit-bashing and non-destructive modifications to existing toys, Knetzger ranges widely, taking in the mathematics of gnome hats and the myriad possibilities of polystyrene packaging foam. When a foundry is needed, or a strip heater to bend acrylic, *Make: Fun!* even shows you how to make these tools. From talking toys and 3D hacks, to edible lenses and a homemade yakitori grill, there's something to inspire everyone's imagination here.

Score ★★★★★

LEARNING PYTHON FOR FORENSICS



Author: Preston Miller and Chapin Bryce
Publisher: Packt
Price: £38.99
ISBN: 978-1783285235
magpi.cc/29UoPX6

Continuing the theme of learning by doing, here's an immersive coding experience for intermediate and not-quite-beginner Python programmers that teaches both forensic analysis and real-world Python scripting in tandem. Right from the opening Python intro, best practices for forensics are integrated into the learning.

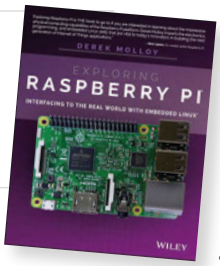
Python 2.7 is used, which is no longer satisfactory, and with the rise of Python 3 this is hopefully one of the last books to be written on Python 2. Nevertheless, even for relatively new learners, a familiarity with Python 2 is still useful for working on, understanding, and re-factoring old code. You'll still want to supplement this book with one of the many Python 3 books we've reviewed, and move your forensic scripts on where possible.

After chapters on working with serialised data, and with databases for dealing with large quantities of data, a framework is created for parsing embedded metadata – useful programming skills for other fields – and this is returned to for building a full forensic framework in the final chapter. Rounded off by a useful appendix on troubleshooting exceptions, this look at an unfortunately all too necessary subject is a good way of developing Python skills, albeit Python 2.

Score ★★★★☆

EXPLORING RASPBERRY PI

Author: Derek Molloy
Publisher: Wiley
Price: £23.99
ISBN: 978-1119188681
exploringrpi.com



As the author notes in the introduction, the Pi's integration of high-level Linux software with low-level electronics circuits represents a paradigm shift in embedded systems development. Molloy complements this statement with an introduction to embedded development for the Pi that takes in low-level hardware interfaces and high-level libraries, embracing every bus and interfacing option.

The first section features hardware, software, and electronics tutorials to prime you with the basics you'll need for the rest of the book, from using GitHub, through scripting language choice, to the role of pull-up resistors. Along the way you'll read some great little intros to everything from how computers perform binary arithmetic, to crimping together custom cables for the GPIO, and plenty of useful tips to make your work easier.

Weighing in at nearly 700 pages, the bulk of the text is in the second and third parts, detailing interfacing the Pi to the physical environment and the internet, rounded off with a short intro to kernel programming. From adding extra UARTs, to writing C and C++ code for the Pi's communication buses, Molloy's succinct and useful examples ensure that this reference will spend as much time on your workshop desk as your bookshelf.

Score ★★★★★

EXPERT F# 4.0 FOUTH EDITION

Authors: Don Syme, Adam Granicz and Antonio Cisternino
Publisher: Apress
Price: £41.50
ISBN: 978-1484207413
magpi.cc/29UpGXK



F# 4.0 is a "mature, open-source, cross-platform, functional first programming language"; if you still think of Microsoft as the 'embrace and extend' company driving everyone onto the Windows platform, it's time to make a fresh assessment. Not only is F# all the things quoted and available on Raspbian (`sudo apt-get install fsharp`), it's also a decent language, and well worth investigating if you've any work to get done on the Mono/.NET platform.

Expert F# 4.0 gives you the wisdom of the language's creator, Don Syme, aided by two maintainers of important F# projects, in a comprehensive introduction that embraces functional and imperative programming, working with data, and using the language's strengths in diverse problem areas.

Although F# is a fairly compact and expressive language, the authors' emphasis on examples using popular libraries means the book's 550 pages should be worked through with less skipping than occurs with some tutorials; the introduction emphasises where programmers from different backgrounds should put in the most effort, such as getting the hang of static types in early chapters if you're a Python or Ruby coder. A little dry at times, but the best intro for .NET newbies.

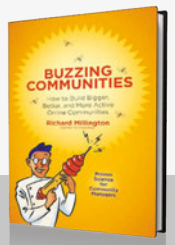
Score ★★★★☆

ESSENTIAL READING: COMMUNITY ESSENTIALS

Running a hackspace, Raspberry Jam, or user group demands soft skills and hard organisation.

Buzzing Communities

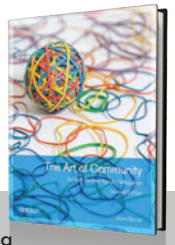
Author: Richard Millington
Publisher: FeverBee
Price: £15.00
ISBN: 978-0988359901
feverbee.com



Helps you create thriving online communities, with real-life examples, practical tips, and trusted community-building methods.

The Art of Community

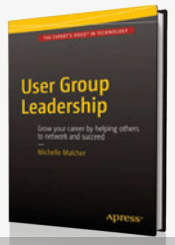
Author: Jono Bacon
Publisher: O'Reilly
Price: £26.50
ISBN: 978-1449312060
oreil.ly/2a0buA4



Comprehensive, open-source-focused guide to all aspects of community building from the Ubuntu community manager.

User Group Leadership

Author: Michelle Malcher
Publisher: Apress
Price: £14.50
ISBN: 978-1484211168
magpi.cc/2a0bZue



The essentials for those who have community leadership thrust upon them, or need to build a professional user group.

Social by Social

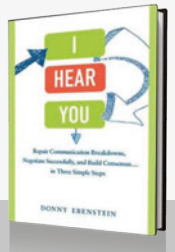
Author: Andy Gibson et al.
Publisher: Nesta Lab
Price: Free Download
ISBN: 978-1906496418
magpi.cc/2a0bro9 (PDF)



Freely available guide to putting social technologies to use in community engagement, whose wisdom hasn't dated.

I Hear You

Author: Donny Ebenstein
Publisher: Amacom
Price: £18.99
ISBN: 978-0814432198
magpi.cc/2a0bZKF



If it all goes wrong, skilled mediator and conflict negotiator Donny Ebenstein will get you back on track.

THE MONTH IN RASPBERRY PI

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

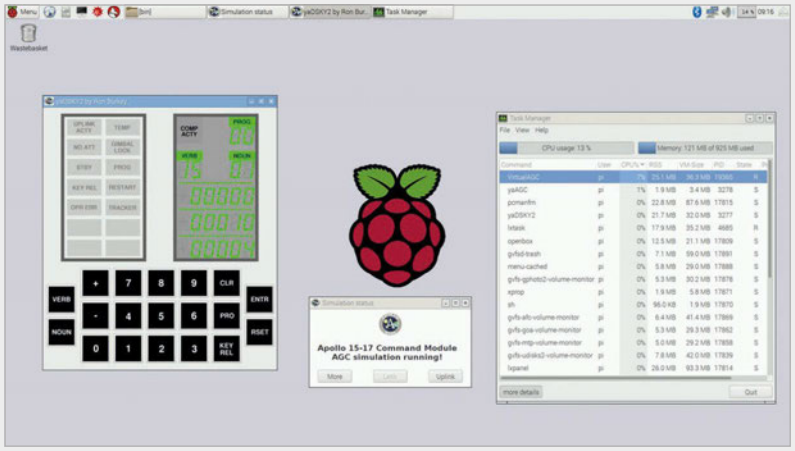
APOLLO MISSIONS ON THE RASPBERRY PI



Above The code for the Apollo missions all on paper, next to the director of software engineering at MIT, Margaret Hamilton. MIT was tasked with developing the software for Apollo

You may have seen in the news recently that the full source code for the Apollo space missions has been published to GitHub (**magpi.cc/2abpPcb**). We understand that the code has been available in some form or another to the public for a few years now (NASA has a mandate to be open in everything it does, including making its code open source where possible); however, at the time of writing, this its first official presence on GitHub.

What is the code actually for? Well, it's the Apollo Guidance Computer (AGC) source code for the Command Module and Lunar Module of Apollo 11; for those not experts on the details of the Apollo missions, the Lunar Module was the moon lander, which would detach from the Command Module when the craft got to the moon. The Command Module would then orbit the moon until the Lunar Module returned and redocked, then return to Earth.



Right The Virtual AGC running on the Raspberry Pi: it barely takes up any processing power! It's designed to look like one of the original AGC control pads

Building on Pi

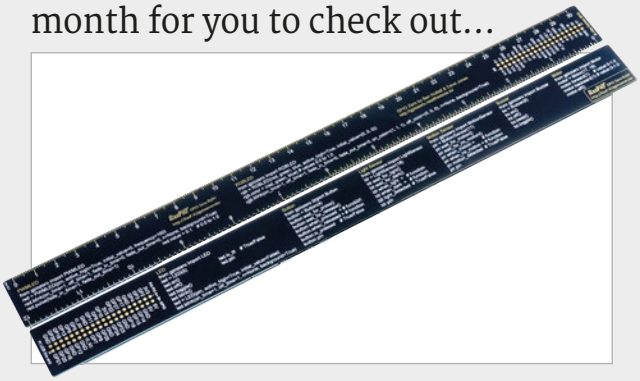
You may be wondering what this has to do with the Raspberry Pi? As you may be aware, the computers on the Apollo missions were not very sophisticated by today's standards (they were similar to pocket calculators), so the Raspberry Pi should be more than capable of running the code. While still high on the success of the Astro Pi mission, Dave Honess of the Raspberry Pi Foundation found that Ron Burkey (**github.com/rburkey2005**) had already ported the code to Linux after its initial 2009 release and it was only a small step to get it going on the Raspberry Pi.

"It takes about three minutes to build," Dave told us as he proudly showed off the screenshot of the virtual AGC. This version not only runs Apollo 11's code, but also Apollo 9, 13, 15, 16, and 17. Number 17 was the last Apollo mission.

We're hoping to bring you some articles about running the code on the Raspberry Pi in the next issue, giving you the chance to see how space technology worked in the Sixties and Seventies.

CROWDFUND THIS!

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



RASPIO GPIO ZERO RULER

kck.st/2acCLB3

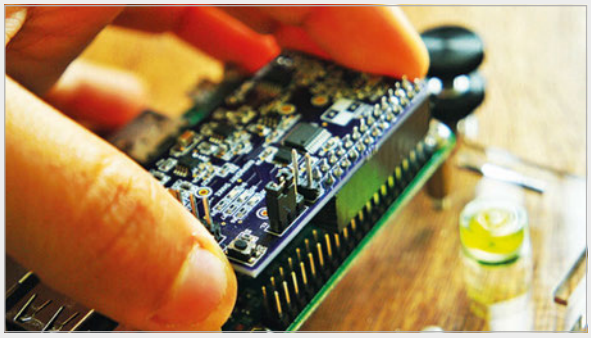
The RasPiO GPIO rulers have been a great little addition to the collection of Raspberry Pi accessories for a while now. As well as being a genuine 30cm ruler, the original had a layout of the GPIO pins on a Raspberry Pi and some quick bits of information on Rpi.GPIO, the original way to program GPIO pins on the Pi. Now there's a new version you can Kickstart that will have GPIO Zero functions on there instead. This is the very easy-to-use module for Python that allows you to simply connect components to the Pi, and now you can own a quick reference guide that also explains the GPIO pins to you. It's already hit its target, so get in there and bag yourself one!



ADVENTURE VENDING MACHINE

magpi.cc/2aaY7PS

This is the first Pi-powered art project we've come across on a crowdfunding site while writing this section, but we're happy it exists. When funded, this vending machine will be set up at Burning Man 2016, and will dispense a quest for you to go on. Once you complete the quest, you'll get a coin which can be used to open a door and claim a prize. Apparently, it's inspired by Leonardo da Vinci, and the creator's aim is for it to inspire people to connect in meaningful ways.



RASPBERRY SHAKE - SEISMOGRAPH

kck.st/2a1bCPf

We like this one: a seismograph add-on for your Raspberry Pi which allows you to detect seismic events (in other words: earthquakes). It's sensitive enough to detect earthquakes of as little as magnitude 2 within a radius of 50 miles; if it's a magnitude of 4 or higher, you can sense it from up to 300 miles away. Its maker says it will even be able to detect earthquakes further away, but you won't get as much useful data. With the Raspberry Pi being used for more and more science experiments, we like the idea of a few geologists having one of these at home.

BEST OF THE REST

Here are some other great things we saw this month

creator.matrix.one



MATRIX CREATOR

We're going to look at this more closely in a future issue, but this is an interesting development kit for the Raspberry Pi. It has a lot of different sensors and optimisations that make it great for studying places and taking good photos and video of them.

farmbot.io



FARMBOT

A Pi-powered robot which makes creating your own little vegetable farm a doddle thanks to superb components and code. It's still in development and a little expensive, but it's a really cool idea that may help out small communities.

WIMBLEDON RASPBERRY JAM: AN INTERVIEW WITH **CAT LAMIN**



CAT LAMIN

Cat is a former primary school teacher, computing coordinator, maths teacher, and real-life geek girl. She's enthusiastic about getting teachers and children interested in coding and computing, any way she can.

Practical tips for setting up a Raspberry Jam from educator and community cheerleader **Cat Lamin**

The MagPi caught up with Cat Lamin after her recent success at the Wimbledon Raspberry Jam to find out why she loves the Raspberry Pi and Raspberry Jams.

How did you get involved with the Raspberry Pi community?

"As a teacher, various people mentioned Raspberry Pi to me, including one of the technicians in school, but I wasn't brave enough to get one. Eventually, the school bought me four to use with my Code Club and I sat them on a shelf admiring them, but didn't really know where to start until I went to Picademy and had some training in how to use one! My biggest problem was understanding how to set up the Pi and what to do if something went wrong. I'm now an expert in debugging basic user errors; you'd be amazed at how many parents tell me that their Raspberry Pi isn't working until I ask them to check the SD card is fully pushed in.

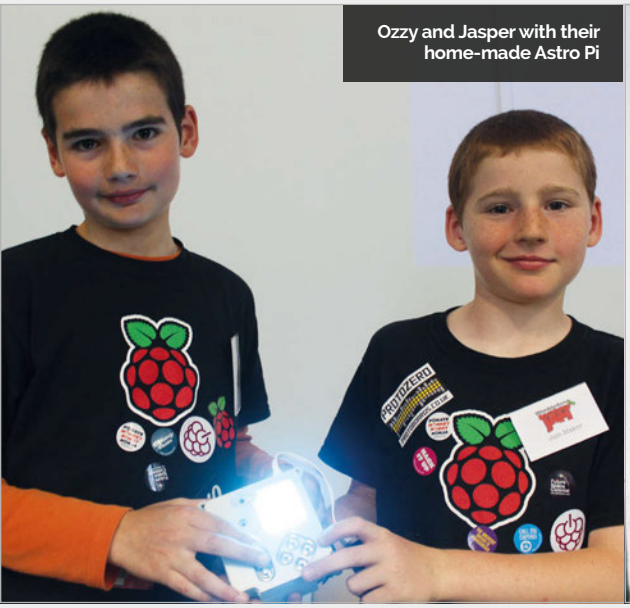
"I went to Picademy in July 2014. While there, people were adding me on Twitter and Google+ and were chatting to me about projects and ideas. I got really excited about the teaching possibilities of Pi, but was very aware of my own limitations. So I started a blog to keep track of my many errors, and to let other teachers know what they could do in their classroom. I then came up with the idea of Coding Evenings as a way to further allow sharing between teachers and community members, which meant that lots of amazing community people got involved."

What was your first Raspberry Jam?

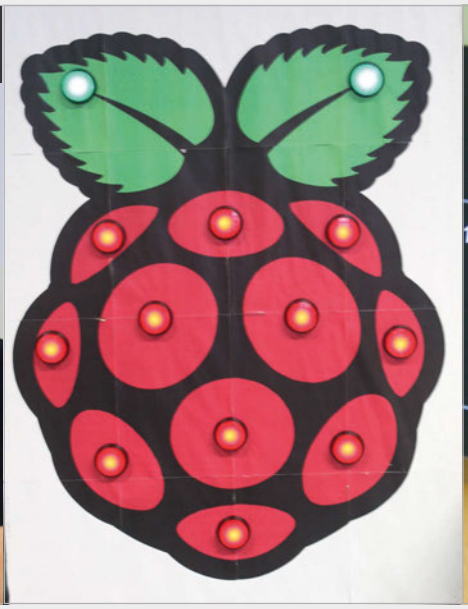
"I think the first actual Jam I attended was the third birthday party in Cambridge, when I accidentally volunteered to help out as a Jam maker, with no idea what I was letting myself in for. I went to Egham Jam when Albert Hickey asked Carrie Anne Philbin to put out a plea for a teacher to head over and do a talk about Picademy and using Pi in school. I've been to CamJam, Egham Jam, Peterborough Jam, and the 4th birthday party, before organising my own Jam in Wimbledon with help from Albert Hickey."

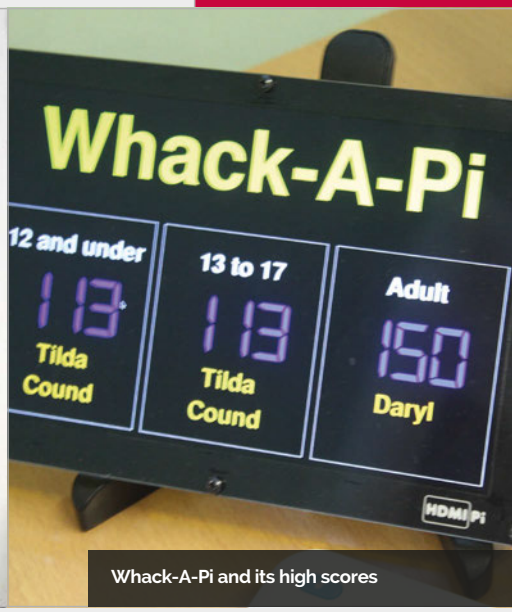
How did organising that first Jam come about?

"Albert, who organises the Egham Jam, approached me to see if I was interested in helping him run the Jam in Wimbledon; he had been offered a venue and wanted me to be involved from the start. Wimbledon is close to the school



Ozzy and Jasper with their home-made Astro Pi





Whack-A-Pi and its high scores

I teach in, and I knew this would be an excellent opportunity to give some of the children from school the opportunity to help develop their passions outside of school. What I really enjoyed about the Jam was seeing all of the families there, and several parents asked if we could let their children's school know about the next one because they were keen to bring more families down!

"I was really lucky with Wimbledon Jam as so many people offered their help almost straight away, and it was great having Ben [Nuttall] along as a representative from the Raspberry Pi Foundation: it added a sort of official stamp of approval to the day."

> "I was really lucky with Wimbledon Jam as so many people offered their help"

What's your favourite thing about a Raspberry Jam?

"I really like having workshops, talks, and show-and-tells going on and we were really lucky that loads of people were interested in doing everything. Our talks ranged from getting girls to code, to teaching, to using RPi as a print server, and people seemed to enjoy the content. Workshops included Crumble Bot, CEEDuniverse and micro:bit, as well as *Minecraft* and Sense HAT. The workshops had varying numbers of signups, but all of them ended up being full because all of the children who attended were really keen to get involved and try things out. For show-and-tell we had Brian Corteil's Micro PiNoon, Carl Monk's Whack-A-Pi, and then Albert's Pi-controlled crane. The children (and adults) had so much fun playing with the games, especially as Redfern and 4Tronix kindly donated prizes for some of them! One of my highlights from the day was watching the mums creep over to Whack-A-Pi and sneak a go while their children were taking part in workshops: it was very funny!

"We had 80 people sign up to the Jam and around 60 turned up, including some 'walk ins'. Twelve of the children attending the Jam were from my own school and there were probably another twelve who came with their parents."

SETTING UP YOUR OWN JAM: CAT'S TOP TIPS

01. ATTEND ANOTHER JAM

Attend a couple of Jams somewhere else to get a feel for them – Jams are all slightly different and you need to decide what you're interested in doing.

02. PEOPLE

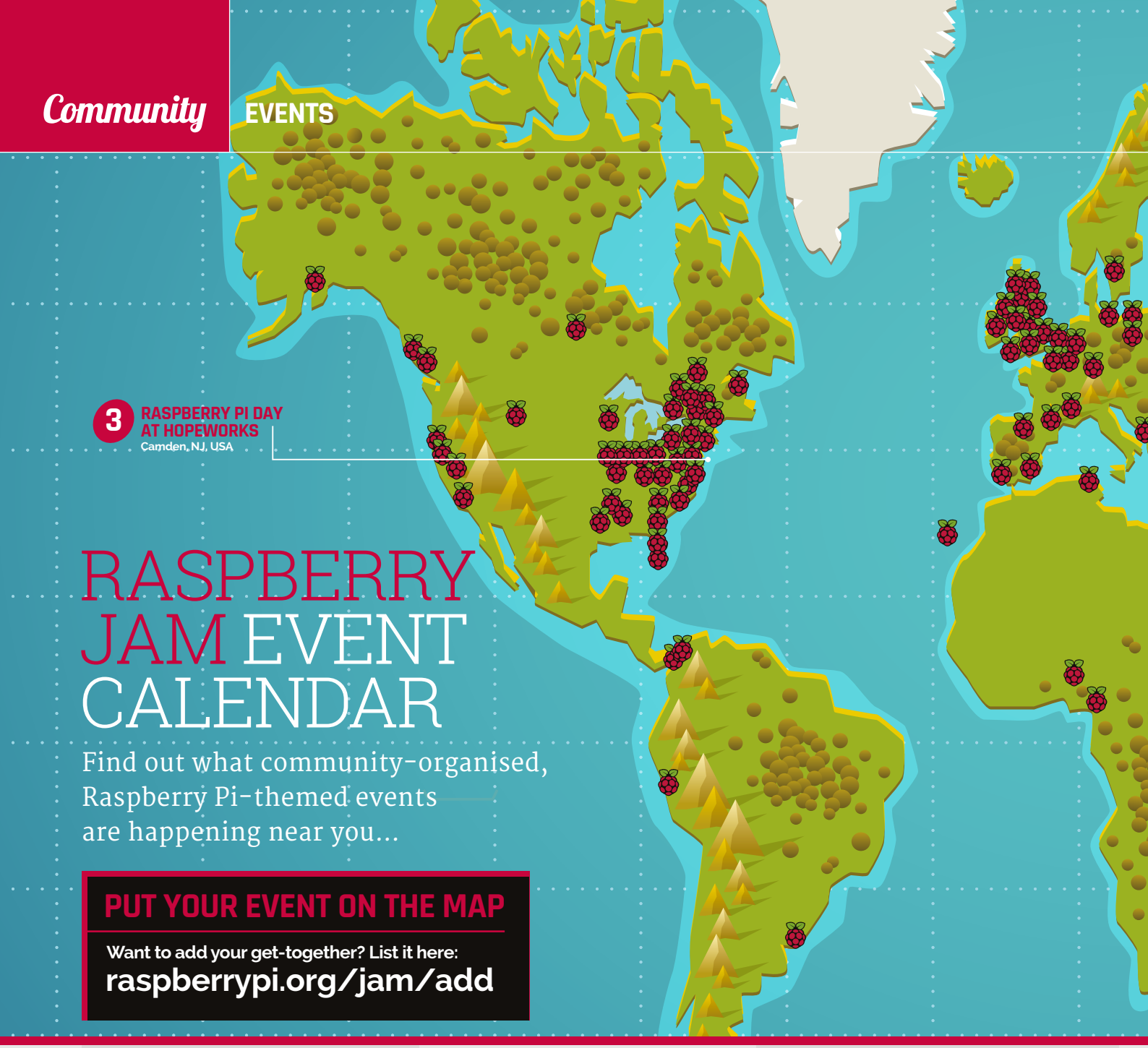
It's really important that you have a reliable core of people to help out, so I was really lucky that volunteers from my Coding Evenings agreed to come along to help out.

03. ASK FOR SUPPORT

Don't be afraid to ask for support from other community members and even businesses. In the past, I've found that a lot of the companies involved in selling Pi-based products are really keen to support events, and very generous with their donations of stickers and prizes.

04. CHIN UP

Don't be disheartened if things don't come together easily. I've been organising my second Jam in Truro in Cornwall and compared to Wimbledon, it's felt like an uphill struggle. I've had support from the Foundation and from the Code Club coordinator in the South West, and I'm feeling much more happier about it now.



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 THE FIRST TAUNTON RASPBERRY JAM

When: Saturday 6 August
Where: Taunton Library, Taunton, UK
magpi.cc/2a8P41N
Beginners and experts welcome at the first Taunton Jam, covering computing, robots, and more.

2 EAGLE LABS RASPBERRY JAM

When: Saturday 13 August
Where: Innovation Birmingham Campus, Birmingham, UK
magpi.cc/2a8OPUo
Raspberry Pi enthusiasts of all ages are welcome to explore show-and-tell projects and talks about the Pi.

3 RASPBERRY PI DAY AT HOPEWORKS

When: Saturday 13 August
Where: Camden Colab, Camden, NJ, USA
magpi.cc/2a6XvaS
Learn how to set up and use the Raspberry Pi thanks to mentors from around the region.

4 SOUTHEND RASPBERRY JAM 10

When: Saturday 20 August
Where: Hive Enterprise Centre, Southend-on-Sea, UK
magpi.cc/2a8Q3PD
A Jam for those who are interested in Raspberry Pi, Astro Pi, Sonic Pi, Code Club, and much more!

5 MINECRAFT PIBRARY JAM

When: Thursday 25 August
Where: Central Library, Coventry, UK
magpi.cc/2a6XWlS
A Minecraft Library Jam where you'll learn to code with Python and Java with *Minecraft Pi*.

6 RASPBERRY JAM PRESTON

When: Monday 5 September
Where: Media Innovation Studio, Preston, UK
magpi.cc/2a6XXG7
Learn, create, and share the potential of the Raspberry Pi at a family-friendly event.



7 RASPBERRY JAM LEEDS

When: Wednesday 7 September
Where: Swallow Hill Community College, Leeds, UK
magpi.cc/2a6Yhoz

Everyone is invited for a couple of hours of computing fun, talks, demonstrations, and hands-on workshops.

8 CAMJAM - CAMBRIDGE RASPBERRY JAM

When: Saturday 17 September
Where: Institute of Astronomy, Cambridge, UK
magpi.cc/2a6Vr2W

The famous CamJam is back this September, with plenty of people showing off projects and giving talks.

DON'T MISS: CAMJAM – CAMBRIDGE RASPBERRY JAM

When: Saturday 17 September **Where:** Institute of Astronomy, Cambridge, UK

Tickets are now on sale for the insanely popular CamJam, the Raspberry Jam from the Raspberry Pi's home town of Cambridge. The CamJam is back at the Institute of Astronomy and will have the usual selection of dealers' tables, people with projects to show, wonderful workshops, and plenty of talks from excellent people within the Raspberry Pi community. Get your tickets fast before they sell out! You can find out more information on the event's page here: **magpi.cc/2a6Vr2W**



YOUR LETTERS



Above A lot of people like the mission patches that came with issue 47 – send us your pictures of you wearing them!

Astro Pi patch

I loved issue 47 of *The MagPi* as I've been following along with Astro Pi very closely since the beginning. To see some of the results (especially the pictures from the hatch window!) was very exciting, and I'm glad to hear there will be more Astro Pi in some capacity in the future.

I was also delighted to see the poster and patch with the issue. I'm trying to find a space to put the poster, which I can do, but I must confess I don't actually know how to sew the patch on. Do you have any tips for a sewing beginner to attach it to my bag?
Jake

Thank you for your kind words about the feature, Jake. We're really looking forward to covering more Astro Pi missions in the future. As for the patch, it's actually quite simple! You can do it with just a sewing needle and some thread.

We like to pin the patch in place so it stays in position; you can easily use a safety pin to do this. We also thread our needle by getting a length of thread and folding it in half; it then goes through the eye a little better, and you can tie it around the needle by just looping it over the point. No knots required! A few stitches around the edge is all you really need to do before tying it in place. There's plenty of videos on YouTube (like this one: **youtu.be/WzL1AEXAd7Y**) that can help you out.

Being social

I'm a recent convert to the Raspberry Pi and I buy the magazine from the shop every month. I've never really had the chance to get into computing, but I'm having a lot of fun learning! Who said you couldn't teach an old dog new tricks? I've also recently started to get into social media; you don't find many people in their seventies on Twitter, I bet! I was wondering where I can follow you online?
Barry Harper

We're on several social channels around the web! On Twitter you can find us with the user name TheMagP1 (**twitter.com/TheMagP1**), which is where we post a lot of updates about what's going on with the mag and the Raspberry Pi community. We're also on Facebook (**facebook.com/MagPiMagazine**) and Google+ (**plus.google.com/+TheMagPi**), if you prefer to use other forms of social media. We hope you enjoy the mag and get to learn much more!

International Pi

Hi there! I know you're a UK-based company, but I keep hearing about the great events that are put on over there and I was wondering if they'd ever come to the States? We have a budding Raspberry Pi community here and it would be great if we could attend some Jams, Picademies, and such in America. Maybe even an Astro Pi contest here as well – we do have NASA, after all!
James Alex

We do have a small but very dedicated American team working on promoting the Raspberry Pi in America, James. You can see them at Maker Faires, and recently Matt from the American team visited a big education conference called ISTE. At the time of writing, he's just sorting out a Picademy in America and it's not the first one he's done either; keep an eye on the blogs and Twitter feeds in the future to find out when new ones are organised.

As for Jams, these are all community-run! There are plenty that are held in America and if you take a look at our Events page in this issue, there may be some you can attend. Failing that, a full list of Jams can be found online here: magpi.cc/28Nxeff

Raspberry Pi is definitely in America!

FROM THE FORUM:

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community – join in via **raspberrypi.org/forums**

I C THE PROBLEM

On the article An Introduction to C, Simon Long presents a Hello World program in C. Unfortunately, Simon presents the main function with a return type of void, which as far as I know has never been the correct signature for main in any of the C standards. While a number of compilers will accept this code and produce the expected result, I think it would be better to have an introduction to C article that presents the signature of main that conforms to the standard.

AndyD

We asked Simon Long about it and he had this to say:

"In the example in the first episode, I have shown a simplified prototype for main. For the record, in 25 years of writing C professionally, I have yet to come across a compiler which considers main returning void to be an error, and for a lot of real-world C programming (embedded systems, standalone PC applications etc.), nothing is ever done with the returned value anyway.

"Had I decided to show the strictly correct prototype for main in the first episode of an introduction to C, I would have had to explain what an int is, which would mean I would have had to explain variable types. I'd also have had to explain what an array of string pointers is (argv) and by the time I'd done that, I would a) have used up all the allotted words, and b) scared off every beginner who was reading it."

Apparently, there will be more on this in the future C tutorials in the magazine. We hope this clears up any problems!

One of Zero

Since November of last year, I've picked up a few Raspberry Pi Zeros when I've noticed them on sale and needed more. I know they're still quite popular, but I was wondering when the one Zero per person rule might start to be lifted? I have a project coming up that could definitely use a few Raspberry Pi Zeros if they were around!

Regards,

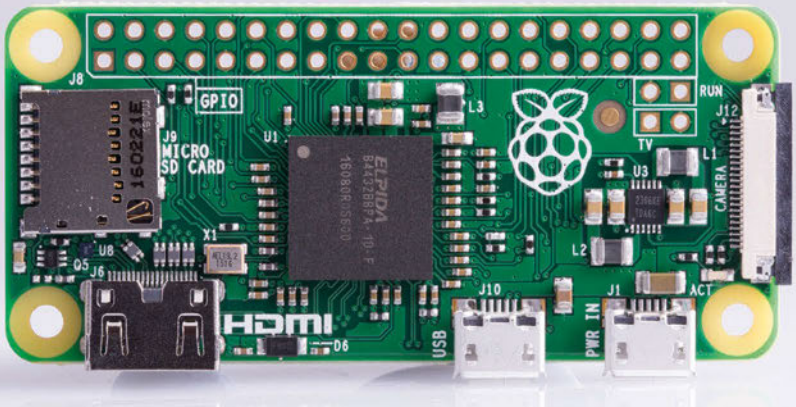
Terry P

We've seen a few stores that have actually begun to lift their one Zero per customer rule, but it seems it's not everywhere yet. The Raspberry Pi Zero is still selling out basically as fast as they can be made, so it's a tricky thing to predict – it's always been more up to the markets than anything else! As more stores lift the restriction, though, others will follow suit and by then the supply will probably be enough to meet demand.

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 US ANYWHERE



FREE: DOWNLOAD ALL 30 ORIGINAL ISSUES

The MagPi Magazine

Available now

for smartphones & tablets

Available on the App Store

GET IT ON Google Play

Subscribe from

£2.29 rolling subscription or **£26.99** full year subscription

Download it today – it's free!

- Get all 30 legacy issues free
- Instant downloads every month
- Fast rendering performance
- Live links & interactivity

TO CELEBRATE THEIR 4TH BIRTHDAY

Pimoroni are giving away a raft of their best boards to one lucky MagPi reader

In association with

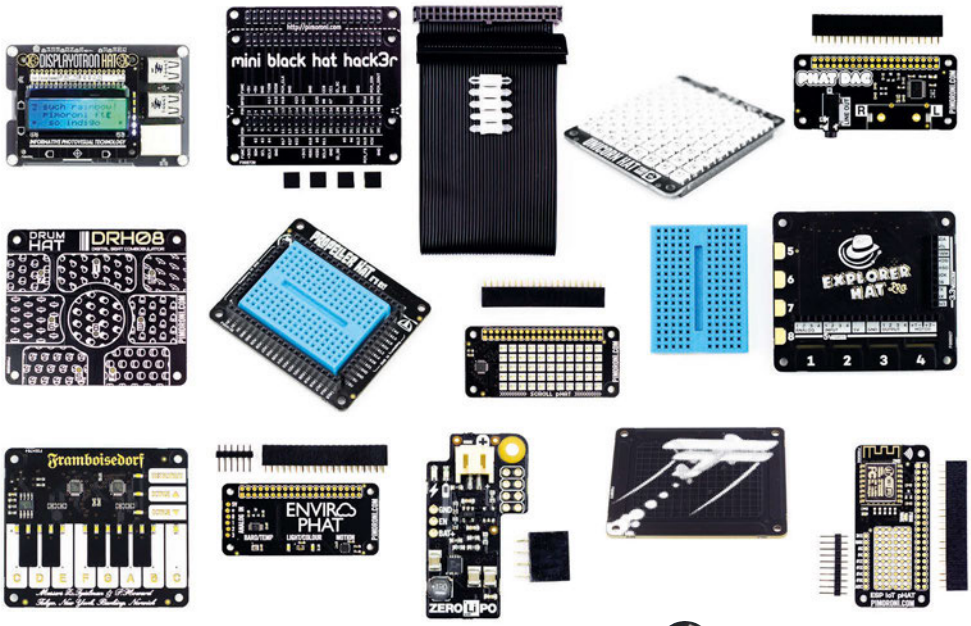
PIMORONI

SHOP.PIMORONI.COM

QUESTION:
WHO WAS CHINA'S MOST SUCCESSFUL PIRATE?

First prize contains:

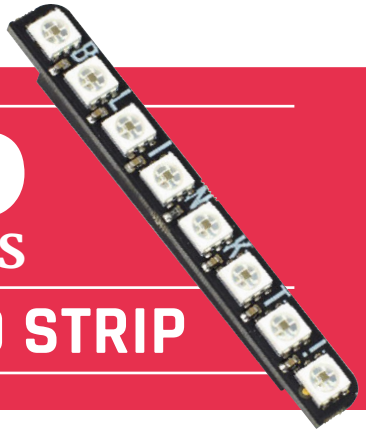
- 4th Birthday Pibow
- Display-O-Tron HAT
- Drum HAT
- Explorer HAT Pro
- Piano HAT
- Propeller HAT
- Skywriter HAT
- Unicorn HAT
- ESP8266 IoT pHAT
- Enviro pHAT
- Scroll pHAT
- pHAT DAC
- Zero LiPO
- Mini Black HAT Hack3r



Answer by 25 August for your chance to win!

Simply email **competition@raspberrypi.org** with your name, address, and answer!

PLUS 10 RUNNER-UP PRIZES OF A BLINKT LED STRIP



Terms & Conditions

Competition closes 25 August 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.



THE UNSUNG HERO

Matt Richardson shares how software support is critical to Raspberry Pi's success

As Raspberry Pi enthusiasts, we tend to focus a lot on hardware. When a new or updated board is released, it garners a lot of attention and excitement. On one hand, that's sensible because Raspberry Pi is a leader in pushing the boundaries of affordable hardware. On the other hand, it tends to overshadow the fact that strong software support makes an enormous contribution to Raspberry Pi's success in education, hobby, and industrial markets.

Because of that, I want to take the opportunity this month to highlight how important software is for Raspberry Pi. Whether you're using our computer as a desktop replacement, a project platform, or a learning tool, you depend on an enormous amount of software built on top of the hardware. From the foundation of the Linux kernel, all the way up to the graphical user interface of the application you're using, you rely on the work of many people who have spent countless hours designing, developing, and testing software.

The look and feel of the desktop environment in Raspbian serves as a good signal of the progress being made to the software made specifically for Raspberry Pi. I encourage you to compare the early versions of Raspbian's desktop environment to what you get when you download Raspbian today. Many little tweaks are made with each release, and they've really built up to make a huge difference in the user experience.

Skin deep

And keep in mind that's only considering the desktop interface of Raspbian. The improvements to the operating system under the hood go well beyond what you might notice on screen. For Raspberry Pi, there's been updates for firmware, more functionality, and improved hardware drivers. All of this is in addition to the ongoing improvements to the Linux kernel for all supported platforms.

For those of us who are hobbyists, we have access to so many code libraries contributed by developers, so that we can create things easily with Raspberry Pi in a ton of different programming languages. As you probably know, the power of Raspberry Pi lies in its GPIO pins which make it perfect for physical computing projects, much like the ones you find in the pages of *The MagPi*. New Python libraries like GPIO Zero make it even easier than ever to explore physical computing. What used to take four lines of code is boiled down to just `LED.blink()`, for example.

Not all software that helps us was made to run on Raspberry Pi directly. Take, for instance, Etcher, a wonderful program from the team at Resin.io. Etcher (**etcher.io**) is the easiest SD card flasher I have ever used, and takes a lot of guesswork out of flashing SD cards with Raspbian or any other operating system. Those of us who write tutorials are especially happy about this; since Etcher is cross-platform, you don't need to have a separate set of instructions for people running Windows, Mac, and Linux. In addition, its well-designed graphical interface is a sight for sore eyes, especially for those of us who have been using command line tools for SD card flashing.

The list of amazing software that supports Raspberry Pi could go on for pages, but I only have limited space here. So I'll leave you with my favourite point about Raspberry Pi's strong software support. When you get a Raspberry Pi today and download Raspbian, you can rest assured that, because of the rapidly improving software support, it will only get better with age. You certainly can't say that about everything you buy.





Raspberry Pi gets a home extension

Endless expansion possibilities with Raspberry Pi

The new RPI-BC enclosure is designed specifically to accommodate Raspberry Pi computers. The DIN rail or wall mountable enclosure provides access to all ports, GPIO pins and SD card without removing the lid. RPI-BC is part of the BC portfolio of electronics enclosures from Phoenix Contact.

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

PHOENIX CONTACT
INSPIRING INNOVATIONS

XXX00-00.000.X0
© PHOENIX CONTACT 2016

LEARN TO CODE WITH SCRATCH

Get started today for **just £2.99 / $3.99**

MAKE GAMES AND APPLICATIONS WITH YOUR *Raspberry Pi*

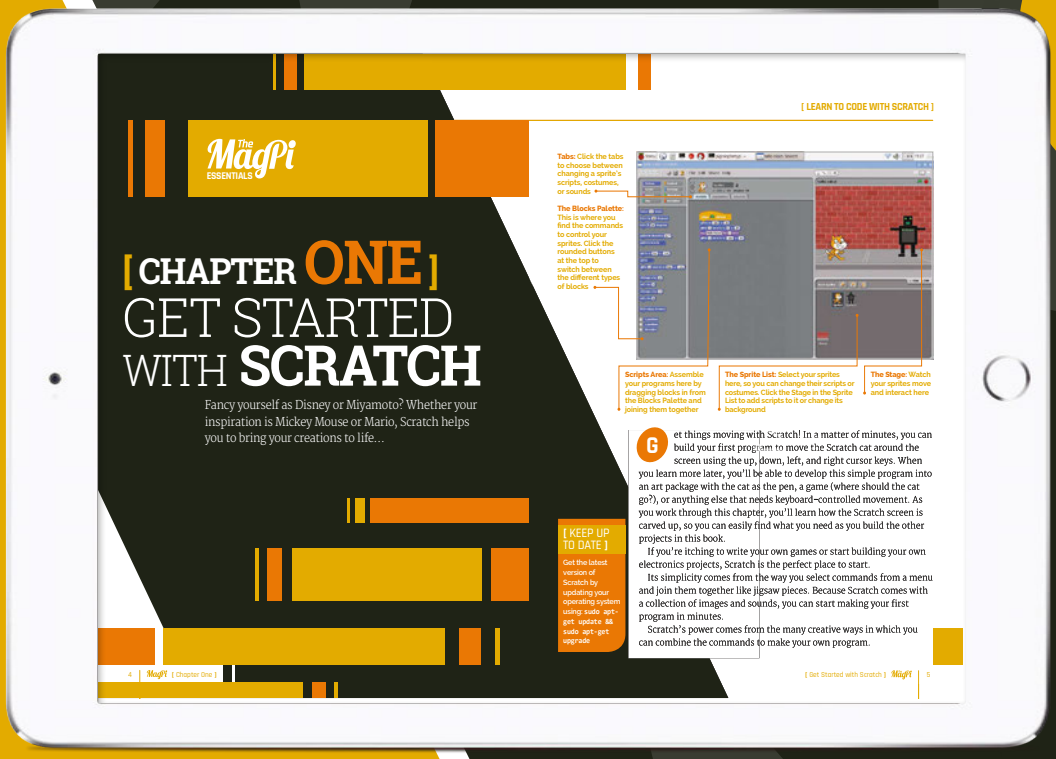
The MagPi ESSENTIALS

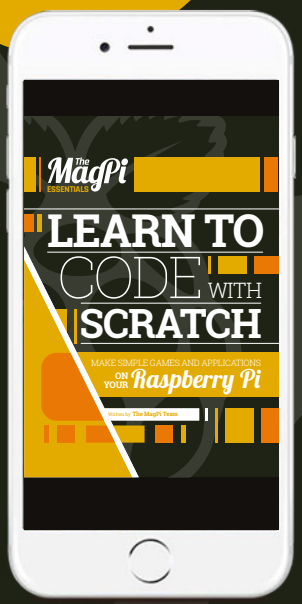
From the makers of the official Raspberry Pi magazine

Available on the App Store

GET IT ON Google Play

Find it on The MagPi digital app





magpi.cc/Scratch-book

Expand your Pi

Stackable expansion boards for the Raspberry Pi

Serial Pi Plus

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

Breakout Pi Plus

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

ADC Differential Pi

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

IO Pi Plus

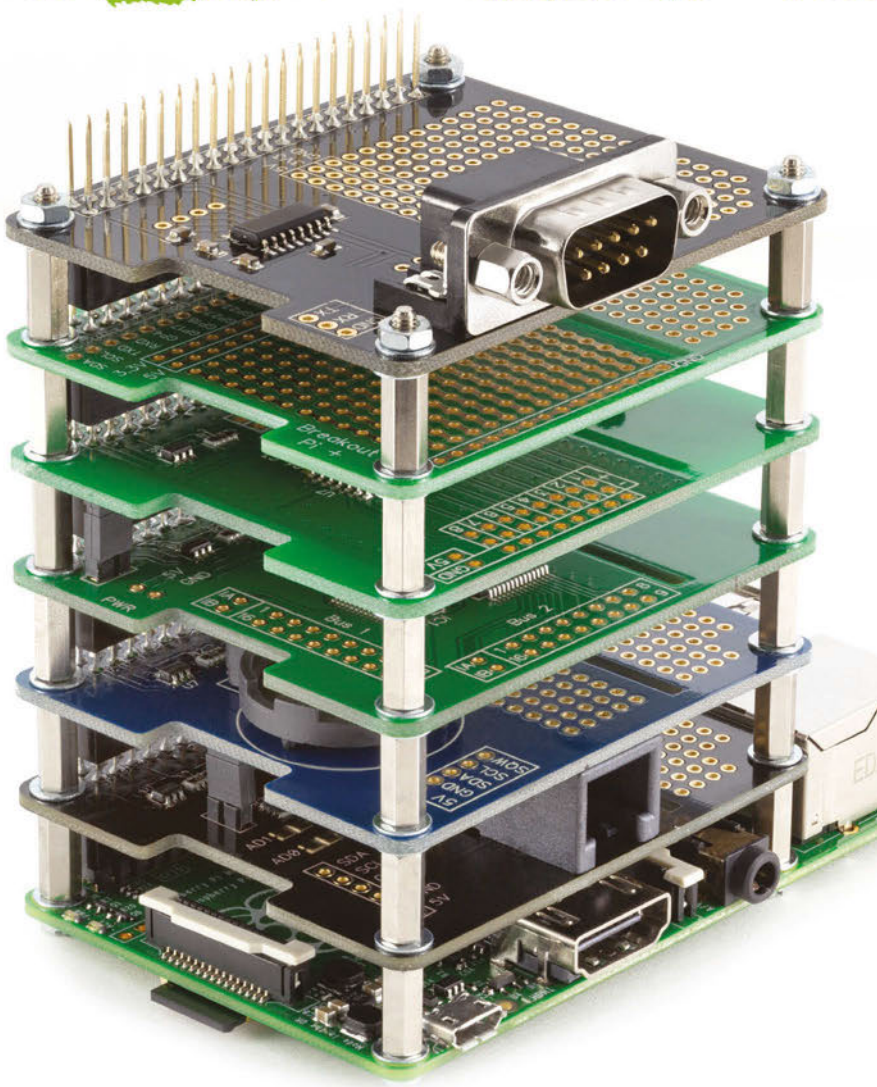
32 digital 5V inputs or outputs. I²C address selection allows you to stack up to 4 IO Pi Plus boards on your Raspberry Pi giving you 128 digital inputs or outputs.

RTC Pi Plus

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

1 Wire Pi Plus

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



Now Available for the **Pi Zero**



ABelectronics UK

www.abelectronics.co.uk

DEXTER INDUSTRIES

SAVE 15% "MagPi15" discount code



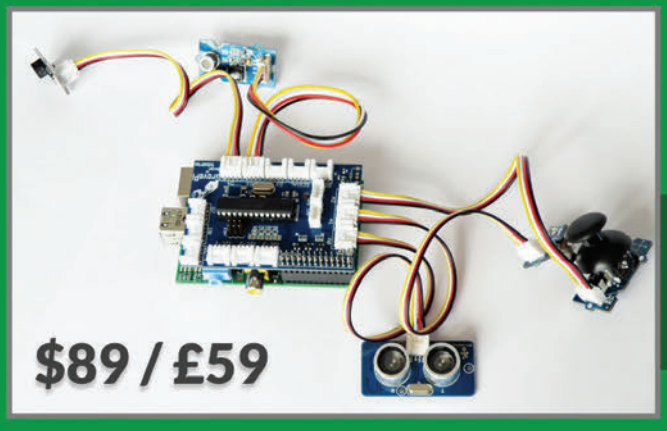
BrickPi

Build a LEGO robot with your Raspberry Pi!

GoPiGo

Everything you need to build a Raspberry Pi robot!





GrovePi

Connect hundreds of sensors to your Raspberry Pi!

www.dexterindustries.com