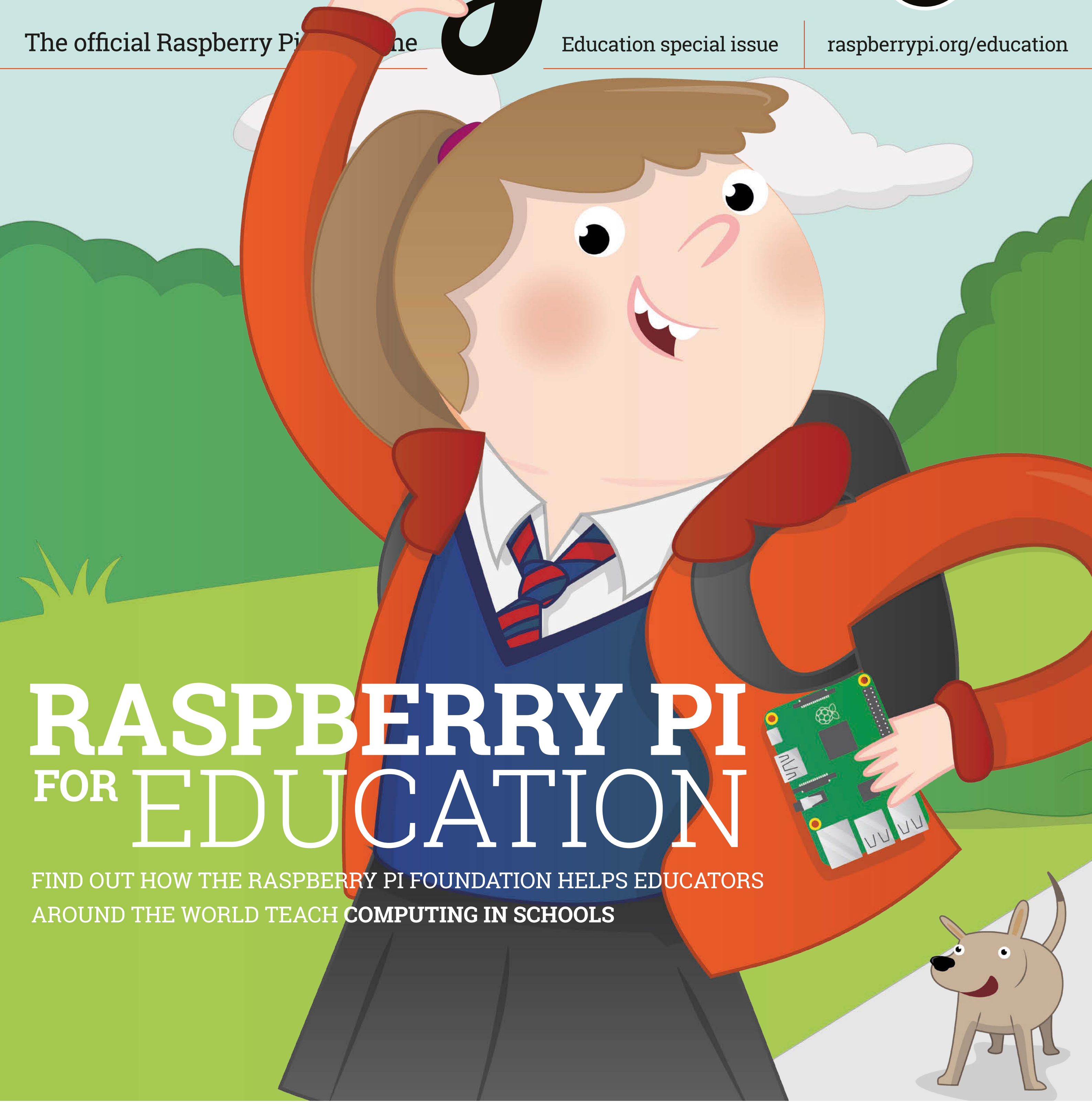
FREE RESOURCES FOR THE CLASSROOM

The MagPi

EDUCATOR'S EDITION

The official Raspberry Pi ne

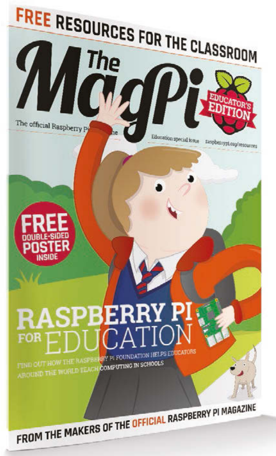
Education special issue | raspberrypi.org/education



RASPBERRY PI FOR EDUCATION

FIND OUT HOW THE RASPBERRY PI FOUNDATION HELPS EDUCATORS AROUND THE WORLD TEACH **COMPUTING IN SCHOOLS**

FROM THE MAKERS OF THE OFFICIAL RASPBERRY PI MAGAZINE


FREE RESOURCES FOR THE CLASSROOM
The MagPi
EDUCATOR'S EDITION
The official Raspberry Pi magazine
FREE DOUBLE-SIDED POSTER INSIDE
RASPBERRY PI FOR EDUCATION
FROM THE MAKERS OF THE OFFICIAL RASPBERRY PI MAGAZINE

Contents

Education special issue

raspberrypi.org/education

FREE TEACHER TRAINING

The Raspberry Pi Foundation offers free teacher training. Find out how to become a Raspberry Pi-certified educator today!

06

WELCOME TO THE COMMUNITY

CEO Philip Colligan introduces the UK tech charity, the Raspberry Pi Foundation

03

ADVANCING COMPUTER SCIENCE WITH RASPBERRY PI

12

The Raspberry Pi Foundation produces free resources for learners and educators all over the world. Learn more here...

ASTRO PI: YOUR CODE IN SPACE!

08

Find out how a collection of school coding projects from around the country found their way onto the ISS

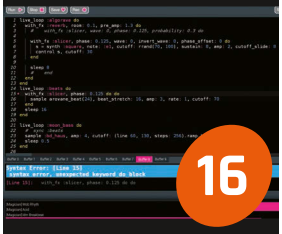
FREE COMPUTING RESOURCES

14

THE MAGIC 8 BALL

Help your students build their own soothsayer step-by-step on the Pi

MAKE MUSIC WITH CODE



```
 1 live_loop :algorave do
 2   with_fx :reverb, room: 0.1, pre_amp: 1.3 do
 3     #    with_fx :slicer, wave: 0, phase: 0.125, probability: 0.3 do
 4
 5     with_fx :slicer, phase: 0.125, wave: 0, invert_wave: 0, phase_offset: 0 do
 6       s = synth :square, note: :e1, cutoff: rrand(70, 100), sustain: 8, amp: 2, cutoff_slide: 8
 7       control s, cutoff: 30
 8     end
 9
10     sleep 8
11     #    end
12   end
13 end
14 live_loop :beats do
15   with_fx :slicer, phase: 0.125 do do
16     sample arovane_beat(24), beat_stretch: 16, amp: 3, rate: 1, cutoff: 70
17   end
18   sleep 16
19 end
20
21 live_loop :moon_bass do
22   # sync :beats
23   sample :bd_haus, amp: 4, cutoff: (line 60, 130, steps: 256).ramp
24   sleep 0.5
25 end
26
```

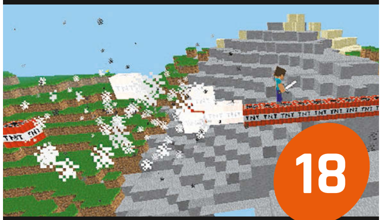
```
Syntax Error: [Line 15]
 syntax error, unexpected keyword_do_block
[Line 15]:   with_fx :slicer, phase: 0.125 do do
```

```
[Magician] Wob Rhyth
[Magician] Acid
[Magician] Idm Breakbeat
```

16

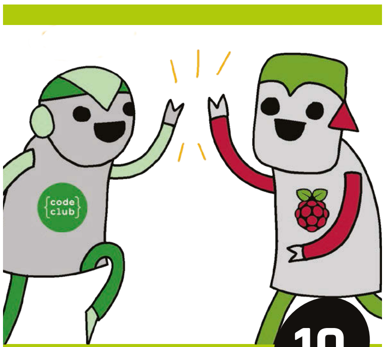
Learn how Sonic Pi crosses the line between music and code

HACK & CODE WITH MINECRAFT



18

Ten-year-old Jasper is a coder and writer. Here he demonstrates a simple but fun project for Minecraft on the Pi



10

CODE CLUB

Learn more about our nationwide network of volunteer-led after-school coding clubs for children aged between 9 and 11

WELCOME TO THE RASPBERRY PI COMMUNITY

Find out how the Raspberry Pi Foundation supports a global community of educators and how you can get involved.

The chances are that you've heard about Raspberry Pi; the low-cost, credit card-sized computer that was developed to encourage kids to learn how to code. We launched the world's first $25 computer in 2012. By the end of 2015, we'd sold over seven million and helped kick-start a global movement to create the next generation of digital makers.

What's perhaps less well known is that The Raspberry Pi Foundation is much more than a computer company. We are a UK-based educational charity with a mission to put the power of digital making into the hands of people all over the world. One of the ways that we pursue that mission is by providing low-cost, high-powered computers, but it isn't all we do.

A global community of educators

At the heart of Raspberry Pi is a global community of educators who are working inside and outside the classroom to inspire kids to get creative with technology. Our job is to provide that community with the support they need. One of the ways we do that is by developing high quality teaching resources and projects, many of which don't require a Raspberry Pi computer, all of which have been designed by educators, and which are all available for free.

> “We're proud to be part of a movement which aims to empower people to shape their world”

More than a computer company

Because we're a charity, we're able to use any and all profits that we generate from our commercial activity to invest in educational programmes and outreach, resources, training and support for educators, and building a global community that shares our mission.

Through our network of Code Clubs (see page 10), we're making sure that opportunities to get involved in digital making are as widely available as possible, mobilising a huge community of volunteers and educators in the process.

Through programmes like Astro Pi (see page 8), we're helping to make computing more relevant to young people who might not have thought that digital making was for them, but who are excited by human space exploration. We're doing the same with music, nature, and the arts, taking a deliberately cross-curricular approach that engages young people with very different interests.

Over the past three years we've also trained hundreds of Raspberry Pi Certified Educators through our Picademy programme of free professional development. It has been amazing for us to see so many of those Certified Educators go on to support other educators to develop their practice, whether as CAS master teachers, by organising meet-ups, or by creating and sharing their own resources. This is the most exciting part of our work: seeing the community of educators grow and support each other. We're constantly inspired by what they do.

We're proud to be part of a movement which aims to empower people to shape their world through digital technologies. If you're not already, then I hope you'll get involved.

Philip Colligan,
CEO, Raspberry Pi Foundation

DIGITAL MAKING REVOLUTION IN EDUCATION

The digital maker movement, a mix of traditional artisan arts and crafts combined with computer programming and electronics, has been taking the world by storm. Its ethos of tinkering and inventing is being used in the classroom to inspire a whole new generation of makers.

HACKATHONS

Hackathons are events at which groups of individuals will build a digital product from scratch, often over a single twenty-four hour period. Fuelled by pizza and cola, as well as by their own enthusiasm, students can work together to build anything from internet-connected Christmas trees to the next great social networking app. The events are competitive but very supportive, and always lots of fun.

Hackathons are a great way to encourage creativity, problem-solving and teamwork within the sphere of computing and digital making. There are plenty of student hackathons organised all over the country, and **mlh.io** is a good place to start if you're looking for an event near you.

Making lessons more engaging and relevant for learners

Raspberry Pi computers, and other similar devices, are unlike the traditional computers you would find in classrooms up and down the country. Rather than a hermetically sealed box, designed specifically to prevent a student from poking and prodding around with the internal components, you are presented with a single board computer with all the parts exposed. This design decision is not an accident: We want to demystify computers, to allow children to see that there is nothing to be afraid of, to show them exactly where the operating system can be found, and to let them experiment with controlling electronics using the General-Purpose Input/Output (GPIO) pins.

The design also poses questions for teachers who wish to deploy Raspberry Pi computers in their classrooms. Do you have them as fixed pieces of equipment, permanently attached to power supplies, keyboards and mice? Do you use cases, and, if so, which one? Do you give students their own SD card, or should they share? How do you keep the software up-to-date and ensure that students can always access their work?

These are all natural concerns, especially if you come from a traditional teaching background, but maybe the first question you should be asking yourself is: How do I make computing more engaging and relevant for my students?

Invent to learn

A recent Nesta report found huge positivity towards digital making: 82% of young people say they are interested in digital making and that parents are overwhelmingly supportive. 89% of parents think it is a worthwhile activity for their children and 73% encourage their children to make things with technology.

So how can you tap into this wealth of enthusiasm in your classroom? The first step is not to worry too much about the practicalities of using hardware such as the Raspberry Pi, Arduino or a Microbit. Things are going to get messy, no matter which platform you use, but that's part of the learning process for you and your students. In recent years, much has been made of the long-forgotten art of tinkering. Children have been encouraged to not break anything, but this is essential to discovering what works and what doesn't. The word 'fail' in digital making is used as an acronym for 'first attempt in learning'.

By bringing physical computing devices into the classroom, students gain from learning how to set kit up themselves, including monitors, keyboards, and mice. Playing with breadboards, buttons, and other electronic components teaches students to not be afraid of technology.

Tackling the digital divide

Computing skills can provide opportunities for social mobility. A recent Naace report on Computing in the National curriculum discusses this, noting that: "those who excel...are in high demand across large parts of the economy". The report advises that it is important to ensure that all pupils have the opportunity to study outside the classroom. Free and/or open-source software and low-cost hardware massively lower the barriers to participation in computing. Raspberry Pi Certified Educator James Robinson explains:

"We asked our GCSE class to purchase Raspberry Pi computers to use both at home and at school and, where appropriate, we were able to use pupil premium funds to support students. This meant that our entire cohort had their own general-purpose computer they could use for classwork, but more importantly their own projects.

"Once all the students had access to identical hardware, we were able to set much more challenging, open-ended and engaging tasks...We used some lessons to cover the basics of programming a Minecraft world while students worked on project-based homework. The results were fantastic: each student interpreted the brief differently and used varying techniques to solve the problem."

RASPBERRY JAMS

Raspberry Jams are community-organised events with a focus on digital making and the Raspberry Pi computer. Jams are filled with talks, show and tell sessions, and workshops, all showcasing the wonderful, useful and often wacky projects that can be created with a Raspberry Pi, a little skill, and a lot of creativity.

If you're interested in attending a Jam near you, then check out **raspberrypi.org/jam** for a list of up-and-coming events.

How the foundation supports makers

Bringing the maker movement into the classroom is one of the charitable aims of the Raspberry Pi Foundation. Our team of experienced educators write fun, engaging, and flexible learning resources, all of which are linked to curriculum objectives. They are published under a Creative Commons license that allows you the flexibility to adapt them to suit your needs (see pages 12 & 13). We provide free professional development for teachers worldwide through our Picademy programme. You can learn more about our training initiatives on page 6.

We have sent two Raspberry Pis and Sense HATs to the International Space Station as part of British ESA Astronaut Tim Peake's Principia mission. In 2015 we also gave away 1000 Raspberry Pi weather stations worldwide for students to build, program, and collect data. We also work closely with developers and academics to build education-tailored applications like Sonic Pi, a cross-platform programming and music making tool.

Fired up to find out more and get involved? Visit our education web page for access to news, events, free resources, our educator community, maker project articles, and much more! **www.rpf.io/edu**

> "82% of young people say they are interested in digital making and their parents are overwhelmingly supportive"

PICADEMY: TEACHING THE TEACHERS

Picademy is the Raspberry Pi Foundation's FREE teacher training initiative, and aims to give educators the skills they need to begin teaching computing with confidence.

Why Picademy?

Getting started with the computing curriculum can be daunting. There's a host of hardware and programming languages, and a baffling array of technical terminology. However, the barrier to getting started rarely has much to do with technical proficiency or programming knowledge. Confidence is typically the main issue, and early setbacks can have a huge impact on this.

Picademy aims to give teachers the skills and knowledge they need to get creative with computing, no matter what their level of experience. It's a two-day course that allows educators to experience some of what can be achieved with a little help and a lot of imagination. We are regularly blown away by what educators create.

What's Involved?

Day one involves running through a series of workshops introducing our teachers to physical computing using the General-Purpose Input/Output (GPIO) pins on the Raspberry Pi computer to control electronic components like LEDs and buttons. All programming is done either in Scratch or Python. There are also sessions based around using add-on boards like the camera module or Sense HAT, coding music with Sonic Pi, or terraforming the world of Minecraft.

On day two, the teachers have free rein to come up with their own project ideas. Previous projects have included killer robots, Christmas jumpers with twinkling LEDs, bespoke games in Minecraft, and many more. They reinforce the skills they learned on day one, and also practice decomposing problems, testing, debugging, and building resilience. Our attendees leave as Raspberry Pi Certified Educators, ready to share their newfound enthusiasm with other educators and children.

Thanks to the generosity and support of our friends at Google, teachers can now apply for Picademy hosted at Google's new venue, the Digital Garage in Manchester in February and March before we hit the road and set up in another city in the UK. To find out when and where our next Picademy event is running, visit **rpf.io/train**.

MANY USEFUL IDEAS

"Picademy's Continuing Professional Development was excellent and it gave me many useful ideas to go back to school with, particularly the use of Minecraft Pi. It was a great experience to spend time with like-minded people who could see how the Pi can be of benefit in supporting the computing curriculum. I use the Pi in my school, and I teach workshops on Minecraft Pi and Python programming." Sarah Zaman, primary computing teacher and Raspberry Pi Certified Educator

Below Teachers graduate as Raspberry Pi Certified Educators

SKYCADEMY: MISSION ACCOMPLISHED

24 teachers took to the outdoors to launch some high-altitude balloons in a brand-new Picademy

Hitching a ride on a Soyuz rocket is not the only way to get a Raspberry Pi into the stratosphere. High Altitude Ballooning involves sending a small electronic device or 'tracker' into the upper atmosphere using a meteorological balloon. That's what our recent Skycademy course was all about: a Picademy designed to show educators how to deliver their own near space project.

James Robinson, who leads the project, explains: "Having run a [HAB] launch myself in school, I'd seen how engaged and inspired my students were, seeing that reaching near space was within their grasp. We wanted to share that experience with other educators and, while the project involves several disciplines, getting to grips with each is not too difficult." The initial goal was simple: give some enthusiastic educators the hands-on experience and confidence to do their own supervised launch, and then hope they return to their students fired up to run their own projects.

Teacher Training

After juggling the places and managing a waiting list, on 24 August 2015, four teams of six teachers from around the UK met to begin their training. Over three days the teams learned how to build, configure, and track their High Altitude payload. On the second day the teams launched their own payloads following a demo launch by seasoned professional Dave Akerman. Our educators then had the task of tracking their flight by car in the hope of recovering their payload as it landed. The payloads reached heights of around 35km, capturing images throughout their flight. At the apex people could view the curvature of the Earth and the black of space!

Skycademy Continues

Skycademy wasn't just those three days though, the attendees will report back over the next several months about their own launches. James explains:

"The event marks the beginning of a year-long project where we've funded each of the newly-qualified educators to run a launch themselves with their students. It's really important to us that it's not just a bit of fun for a few days, it's a great opportunity to explore the power of project-based learning using a range of disciplines including science, technology, engineering, and computer science."

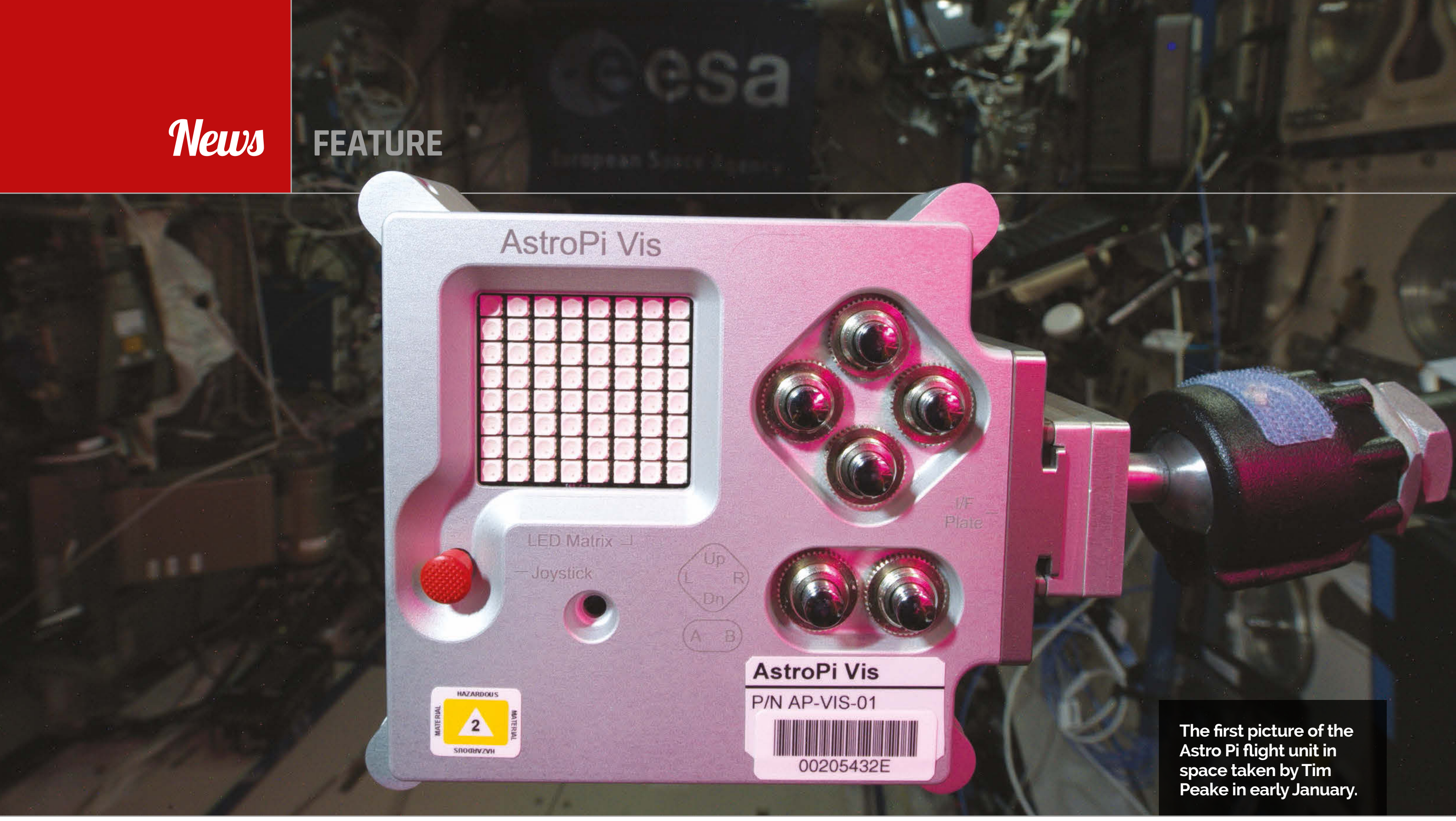
We've since seen several launches by our educators, including a flight from PyCon UK in which teachers sent trinkets like Lego figures up with the payload. Bourne Grammar School ran our first entirely student-led flight, with pupils managing everything including the chase and recovery. Glebe House School captured amazing images of the Norfolk Coast during their flight.

We're looking forward to seeing what the rest of our educators do over the coming year. To find out more about Skycademy, visit **rpf.io/skycademy**.

ANATOMY OF A FLIGHT

- A large helium balloon lifts a small (<500g) payload up to 35km, where the balloon bursts.
- After the burst, the payload falls using a small parachute to slow its descent.
- The payload contains a flight computer (Raspberry Pi) connected to a GPS antenna, a radio transmitter and camera.
- The payload continually collects GPS data and images and transits them via its radio.
- On the ground listeners can receive the GPS data and use it to predict the landing site.

Left High above the Earth, our intrepid weather balloons take shots of the world below



The first picture of the Astro Pi flight unit in space taken by Tim Peake in early January.

ASTRO PI: YOUR CODE IN SPACE

Find out how your students can code software for British ESA Astronaut Tim Peake to use on the International Space Station

What is Astro Pi?

If you're looking for a way to inspire your students to learn about computer programming you won't struggle to get them excited about this! Simply say: "Do you want your code in space?" and watch as their mouths hang open while they take in the scope of what you just said.

Computer programming is an incredibly important skill and one that is in high demand within the UK's space industry. So, back in 2014, Raspberry Pi joined forces with the UK Space Agency, the European Space Agency and the UK Space Trade Association to run a competition. This gave school-age students in the UK the chance to devise computer science experiments for British ESA Astronaut Tim Peake to run aboard the International Space Station (ISS).

Two augmented Raspberry Pi computers (called Astro Pis) have been flown to the ISS as the platform on which to run these experiments. They include a special sensor board, a Raspberry Pi peripheral called the Sense HAT. This can measure the environment inside the ISS, detecting how the station is moving through space, and measuring the Earth's magnetic field.

What would you do with a piece of equipment like the Sense HAT? Look for cosmic rays? Make a timelapse of the Earth through a hatch window? Maybe just give Tim games to play?

This was the premise of the 2015 Astro Pi student competition which produced seven winning teams. Their experiments are in the form of Python programs, written and tested by the school pupils using their own Sense HAT boards and Raspberry Pis. The winning programs, now part of the Astro Pi payload, blasted off from Cape Canaveral Air Force Base in Florida back in December and arrived at the ISS a few days before Tim. They range from fun reaction time games to serious science experiments looking at space radiation.

During his mission Tim Peake will deploy the Astro Pis inside the European Columbus module and run each of the student programs in sequence. Finally, the results will be downloaded back to Earth, returned to the students that took part, and made available online for everyone to see.

Want to learn more? Visit **astro-pi.org** for more information.

FOLLOW US ONLINE

The Astro Pi units will remain on the ISS until 2022 and we hope to run many more Astro Pi competitions, challenges, and interactive activities between now and then. If you follow us on social media you can keep up to date on everything that is happening!

Twitter
@astro_pi
@astro_pi_vis
@astro_pi_ir

Facebook
rpf.io/apfb

What age is this for?

Astro Pi is aimed at both primary and secondary level. The primary school competition was aimed at students aged 11 and under. They were required to submit an idea for how they would use the Astro Pi computers on the ISS. These were then judged based on their originality, creativity, practicality, and usefulness. Two winners were chosen who then worked with the Raspberry Pi Foundation to code, test and prepare their software.

The secondary competition was for students aged 11 to 18. Phase one was identical to the primary competition and ran in parallel; in phase two, students were required to code their experiments on their own Raspberry Pis with Sense HATs and to submit the completed software. These were judged again against a number of space themes set out by the UK Space Trade Association companies. You can find out about all the winners here: **rpf.io/apwinners**

Can I get involved?

Yes! The project has, for the first time, made it possible for students to have access to the exact same computer hardware being used by the crew of the ISS. Everything apart from the aluminium flight case is available to buy online. We also have a variety of free educational resources to introduce students to using each sensor, followed by examples of how the hardware can be used in space. Check them out here: **astro-pi.org/resources**

The code that won the Astro Pi competition can also be downloaded and run by your students as an example of what can be achieved with the hardware: **github.com/astro-pi**

Alongside the competition winners' programs, Tim will be using the Astro Pis to run a long-term ISS environmental monitoring program which will produce a CSV file full of time-stamped sensor readings which can be downloaded and analysed. Look for evidence of the crew being near the Astro Pi, the effect of the South Atlantic anomaly or changes in the accelerometer when the station's thrusters are fired and much more. The resource for this is online here: **rpf.io/apfda**

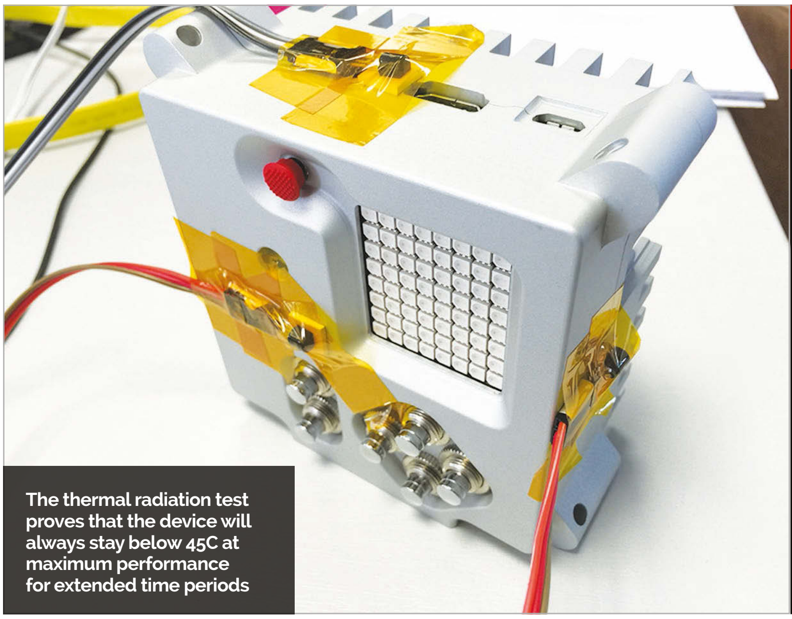
Tim is also planning to devote some of his free time on Saturday afternoons to educational outreach. Depending on his available time, we hope he will invite more students to write programs for him to run in space. So there could be yet more opportunities to get your code running on the ISS during Tim's mission!

> "Get your code running on the ISS during Tim's mission!"

There is even a friendly web forum that you can post on should you have any questions. This is frequented by lots of experienced Raspberry Pi users who are eager to help you.

Go to **raspberrypi.org/forums** and you'll find the **Astro Pi** forum listed under **Education**.

Above Hannah Belshaw's program records sensor data from the Astro Pi and represents it in a Minecraft world. She appeared alongside Tim Peake on The One Show broadcast on the BBC on the 6th of November 2015



The thermal radiation test proves that the device will always stay below 45C at maximum performance for extended time periods

FLIGHT SAFETY TESTING

To be allowed on a rocket, you need a flight safety certificate for your device, and these can only be obtained by presenting a whole host of measurements and test results to a panel of experts at the European Space Agency in Holland.

The expertise and equipment to carry out many of these tests was well outside the capabilities of the Raspberry Pi Foundation. Without the facilities and personnel available through our UK Space partners this would not have been possible. We had to use facilities and partners all over Europe to get the work done.

If you want to find out about the process we had to go through to get the Astro Pis certified for space flight you can read about it on the main Raspberry Pi blog here: **rpf.io/ap5**

START A CODE CLUB IN YOUR SCHOOL

Are you a teacher interested in setting up a Code Club in your school? Find out how simple it is to get started.

It's easier than you think to run a Code Club yourself: you don't need existing coding skills, just a can-do attitude to get stuck in learning alongside your students for an hour a week!

If you haven't yet heard about Code Club, they are a UK-based not-for-profit organisation offering free learning materials and support for teachers, volunteers and parents running after-school coding clubs for children aged 9-11.

Code Club's specially designed projects offer structured and fun content for the clubs. The projects are step by step guides for children to follow to create animations, games, websites, and much more. Children build up their programming skills as they move through the projects. There are also challenges to provide opportunities to apply what they've learnt.

Caroline Harding, a Year 4 teacher who helps to run a Code Club at her school in Croydon, told us about the benefits the club has brought the children. "Making Code Club available to the children in our school has helped tremendously with their confidence and engagement in coding and computing in general," Caroline said. "It taps into their problem solving skills and enables them to develop critical thinking skills."

"Programming and coding is an area of the curriculum that many staff can find intimidating. Knowing that the children have some experience of the program can help ease some anxieties and enables that 'have a go' attitude!"

By starting a club at your school you'll be joining a huge community of teachers who do the same thing - around 50% of Code Clubs are run by teachers.

If you're considering getting a Code Club started, we've come up with a few tips to help you.

Below Code Club allows children to experiment and invent, using different languages to create their own games, animations and websites

Register your club online

To access Code Club's project materials, you will need to register online. You can sign up as a Code Club Host by visiting codeclub.org.uk/bett, making sure to use your school email address so we can validate you as a teacher.

Once you've entered your details, you will be able to select the option to run the club yourself. Your club will then be automatically activated and you will have immediate access to all Code Club's online resources.

Code Club have projects in three different coding languages: Scratch, HTML/CSS and Python. Beginning with Scratch is recommended, as

Left There is no cost involved in starting a Code Club. It's free for schools, and the kids who attend

this visual block-based language provides a great introduction to key programming concepts. If your pupils are already experienced with Scratch, though, you may wish to get started with HTML/CSS or Python. There are twelve Code Club projects in each language to keep your club occupied for a full term.

Your first Code Club

It's worth preparing for your first Code Club session by working through the project in advance, so that you are aware of all the instructions and the places where pupils could possibly get stuck.

Code Club is fun and it offers children (and their teachers) the opportunity to get creative with coding. It's a chance to experiment and invent, helping children to learn an important skill for their future, while engaging with technology and creating things that they can get excited about.

The model can be adapted to suit different venues and educational needs. Most clubs run through one coding project per week, but some children like to spend longer perfecting their designs. Some clubs have pairs of students sharing computers, and many clubs also like to experiment with physical computing. You can customise your club to suit you and the children.

Code Club in practice

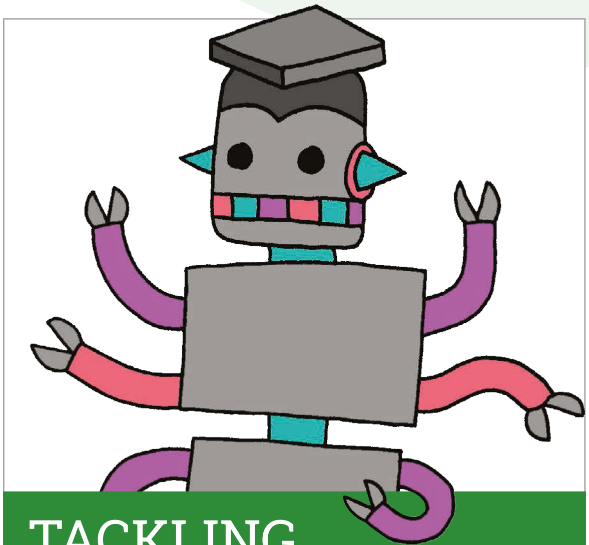
There are thousands of teachers running their own Code Clubs across the country, and around the world. We spoke to Matthew Cave, Assistant Headteacher at West Town Lane Academy in Bristol, who told us about his club.

Beginning with Year 5 and 6 students, Matthew and his team introduced Code Club's Scratch projects for all Key Stage 2 children. Now, they have a whole-school approach, with Scratch Jr introduced for Key Stage 1, and they have invested in new technology including Lego WeDo and My Romo.

Matthew said; "We've been running our Code Club for over a year now, with 40 children attending. The club is in high demand."

Code Club's fun approach has provided other benefits: "It's amazing to see the sense of achievement the children get when they finish their projects. We can really see them starting to persevere with the tasks in Code Club, using analytical thinking to troubleshoot."

What advice does Matthew have for teachers who are thinking of starting a club? "It's dead easy, so take the plunge! The children will run with it, so don't worry about not being an expert."



TACKLING THE COMPUTING CURRICULUM

Running an after-school Code Club can help you develop more confidence to teach the computer curriculum and to integrate computing into your everyday lessons.

If you and your colleagues are keen to get some additional, more formal training, you may be interested in Code Club's Teacher Training courses. There are three modules on offer, focussing on 'Computational Thinking', 'Programming & Networks' and 'The Internet'.

Many of the sessions are now free for teachers, so if you are interested, you can make an enquiry with the Code Club team by emailing **hello@codeclubpro.org.**

FIND THIS & OTHER RESOURCES: rpf.io/learn

ADVANCING COMPUTER SCIENCE WITH RASPBERRY PI

The Raspberry Pi Foundation produces free resources for learners and educators all over the world. Head of Curriculum Development, **Marc Scott**, explains why and how...

One of the core goals of the Raspberry Pi Foundation is to advance Computer Science education, and one of the key ways in which we attempt to achieve this is in the publication of free educational resources. Our aim is to be the first port of call for educational materials, whether you are a learner who has just unboxed your first Raspberry Pi computer, or an educator teaching Computer Science to eager students of any age and level.

SIMULATING WEIGHTLESSNESS

In this resource you can learn how to simulate the effects of weightlessness in space using Scratch. The resource teaches about some fundamental computer science concepts such as sequencing, looping, variables, and conditionals. It's an ideal activity for those just starting out with a Raspberry Pi and wanting to learn a little bit more about learning to program in Scratch. See it at **rpf.io/weightlessness**.

Teach, Learn and Make

Head on over to **raspberrypi.org/resources** and you'll find three categories of resources from which to cherry-pick: *Teach*, *Learn* and *Make*.

Our rapidly growing set of *Teach* resources are designed with educators in mind. There you can find complete sets of lesson plans, student worksheets and resource files ready for you to print off, project, or just to direct your students to. Some schemes require Raspberry Pis, such as the Networking lessons. Some are cross-platform, such as the Sonic Pi lessons on making music. Some tackle tricky concepts in Computer Science, such as the Sorting Algorithm lessons. There are even some cross-curricular schemes, such as *Sensing Space*.

Our *Learn* and *Make* resources are designed to be accessed by learners without necessarily requiring the support of an educator. They could also easily be used in lessons and are especially suitable for after-school or lunchtime clubs. If you've ever wanted to build a device to tell you how many astronauts are currently in space, predict the future with a digital Magic 8-Ball, or learn what it's like to float around in zero-g, this is the place to go. Our *Learn* resources focus on developing new skills and knowledge through project based learning, while the *Make* resources engage learners with fun and often hands-on software and hardware projects.

We add new resources to the site almost every week, so it's worthwhile checking back every now and then to see what new and exciting projects we've been dreaming up lately.

FLAPPY ASTRONAUT

This resource is for the more advanced learner. It uses the new Sense HAT hardware along with some Python code to produce a clone of Flappy Bird, in which the player navigates between pipes by vigorously shaking the Raspberry Pi to overcome gravity. It's a tricky game to master, but a lot of fun to play. Test it yourself at **rpf.io/flappy**.

Free as in Beer and Speech

A key difference between the Raspberry Pi Foundation resources, and those produced by many other organisations, is that ours are free. That's free in both senses of the word!

Both time and money are finite, and we recognise that educators can spare little of either. That's why we've created these resources for you to use in any way you wish, and why you'll never be charged for downloading, printing or sharing them. In fact, we would encourage you to do all those things.

We're big believers in open source software at the Foundation, and our attitude to educational resources mirrors this belief. The license under which we publish our work is very permissive. We use a Creative Commons Attribution and Share-Alike license for all our educational materials, and encourage others to do the same. This means that if you want to make changes to our resources, you are welcome to go ahead and do so. If you want to alter our work so that it is more appropriate to your learners, better suited to your teaching style, or just plain better than the original, then that is not a problem. Copy the text, change it in whatever way you like, and then publish it yourself. All we ask is that you attribute us as the original authors and that any derivative works you produce are shared under the same license.

> "Both time and money are finite, and we recognise that educators can spare little of either."

Get involved

We're a very small team here at the Raspberry Pi Foundation, and as such we rely a huge amount on our wonderful community of enthusiasts, tinkerers and educators. Not only do they use our resources, they contribute as well, doing anything from correcting a few typos right up to producing full schemes of work for us to publish on our site. You could get involved too, and help to improve or create educational materials. For the technically minded, just head on over to our GitHub page at **github.com/raspberrypilearning** and you can fork our resource repositories, make any improvements you like, and then submit a pull request for us to look over. If the idea of using Git is a little bit daunting though, then you can just use GitHub reporting feature to raise an issue, explain what the problem is and we'll get the message and can make the changes for you.

If you want to have a go at making a resource yourself then go ahead. Share it with us and if we like it you could well find your work being proudly displayed on our website.

FIND THIS & OTHER RESOURCES: rpf.io/learn

MAKE A DIGITAL MAGIC 8 BALL

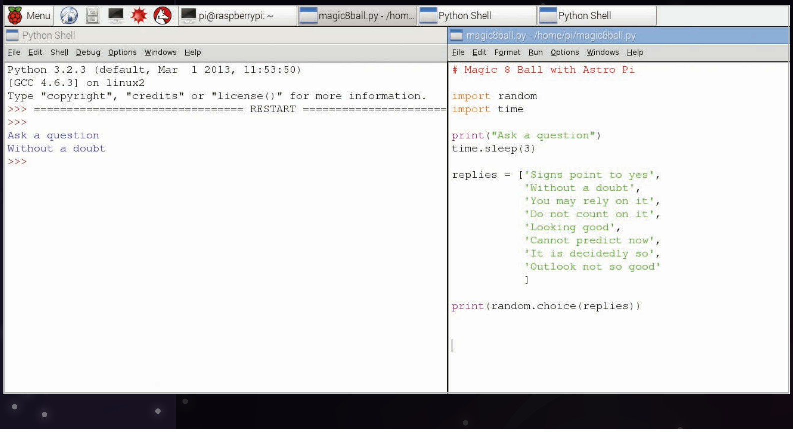
Bring the time-honoured tradition of shaken-not-stirred fortunes to the Sense HAT, making use of those motion sensors

You'll Need

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

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

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



Menu
pi@raspberrypi: ~
magic8ball.py - /hom...
Python Shell
Python Shell
Python Shell
File Edit Shell Debug Options Windows Help
magic8ball.py - /home/pi/magic8ball.py
File Edit Format Run Options Windows Help

```
Python 3.2.3 (default, Mar  1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ======================
>>>
Ask a question
Without a doubt
>>>
```

```
# Magic 8 Ball with Astro Pi

import random
import time

print("Ask a question")
time.sleep(3)

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

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

Fig 1 Test the program first in a Python shell window

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

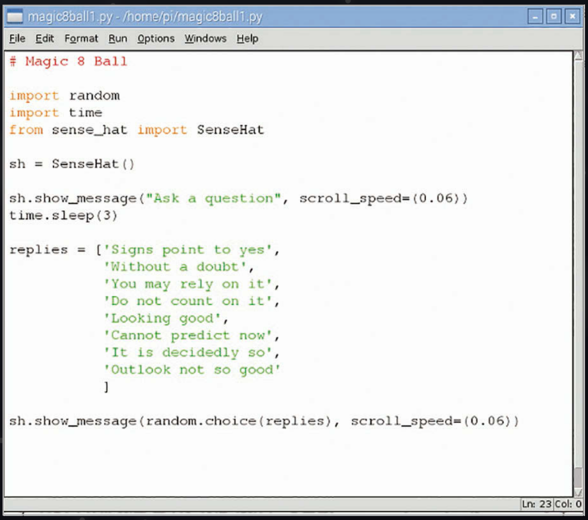
Printing replies to the screen

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

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

```
import random
import time
```

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


magic8ball1.py - /home/pi/magic8ball1.py
File Edit Format Run Options Windows Help
Ln: 23 Col: 0

```
# Magic 8 Ball

import random
import time
from sense_hat import SenseHat

sh = SenseHat()

sh.show_message("Ask a question", scroll_speed=(0.06))
time.sleep(3)

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

sh.show_message(random.choice(replies), scroll_speed=(0.06))
```

Fig 2 You can alter the speed of the scrolling text

Type:

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

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

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

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

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

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

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

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

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

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

Display text on the Sense HAT

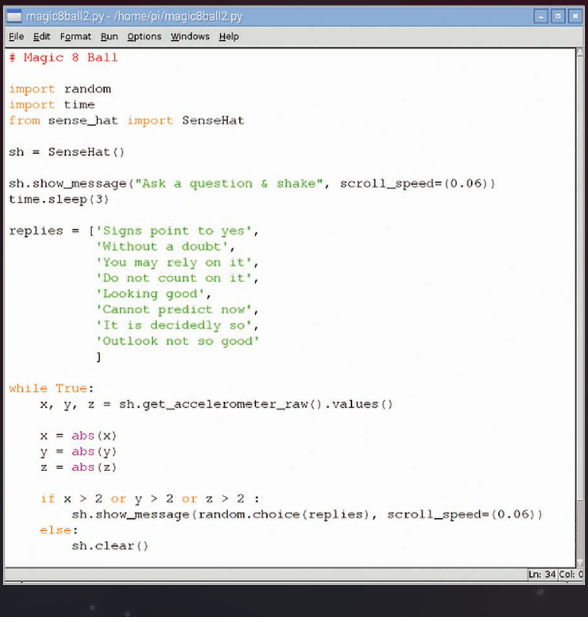
Now that you have text outputting to the Python 3 shell window on your screen, let's change the code so that the text scrolls across the LED matrix on your Sense HAT. To do this, you will need to use the **SenseHat** library and replace the 'print' functions with a 'show message' function. Underneath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

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

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

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

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


magic8ball2.py - /home/pi/magic8ball2.py
File Edit Format Run Options Windows Help
Ln: 34 Col: 0

```
# Magic 8 Ball

import random
import time
from sense_hat import SenseHat

sh = SenseHat()

sh.show_message("Ask a question & shake", scroll_speed=(0.06))
time.sleep(3)

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

while True:
    x, y, z = sh.get_accelerometer_raw().values()

    x = abs(x)
    y = abs(y)
    z = abs(z)

    if x > 2 or y > 2 or z > 2 :
        sh.show_message(random.choice(replies), scroll_speed=(0.06))
    else:
        sh.clear()
```

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

SAM AARON

Sam is the creator of Sonic Pi. By day he's a Research Associate at the University of Cambridge and by night he writes code for people to dance to.

sonic-pi.net

π))) Sonic Pi

LIVE CODING

Digital musician and Cambridge Computer Lab researcher **Sam Aaron** introduces Sonic Pi and the art of live coding

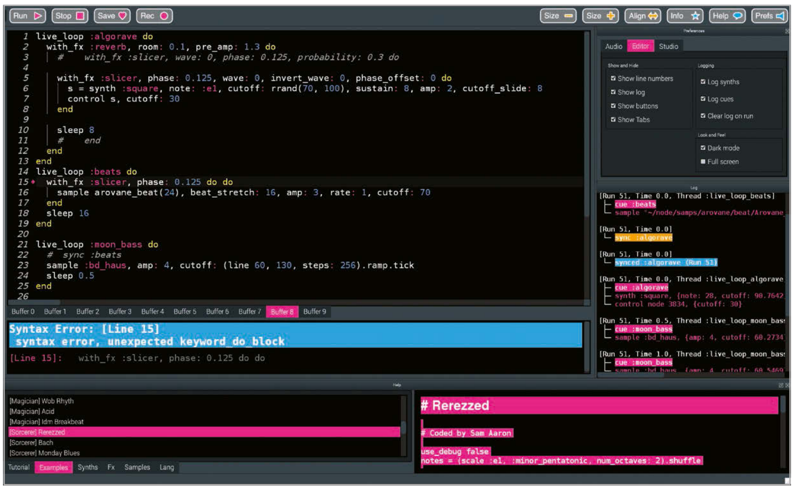
You'll Need

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

The laser beams sliced through the wafts of smoke as the subwoofer pumped bass deep into the bodies of the crowd. The atmosphere was rife with a heady mix of synths and dancing. However, something wasn't quite right in this nightclub. Projected in bright colours above the DJ booth was futuristic text – moving, dancing, flashing. This wasn't fancy visuals; it was merely a projection of Sonic Pi running on a Raspberry Pi. The occupant of the DJ booth wasn't spinning discs; she was writing, editing, and evaluating code. Live. This is live coding.

This may sound like a far-fetched story from the future, but coding music like this is a growing trend and is often described as live coding (**toplap.org**). One of the recent directions this approach to music-making has taken is the Algorave (**algorave.com**); these are events where artists code live music for people to dance to. However, you don't need to be in a nightclub to live-code; with Sonic Pi v2.6+, you can do it anywhere you can take your Raspberry Pi and a pair of headphones or some speakers. Once you reach the end of this article, you'll be programming your own beats and modifying them live. Where you go afterwards will only be constrained by your imagination.

Below The new Dark theme for v2.6 is lovely!



```
 1 live_loop :algorave do
 2   with_fx :reverb, room: 0.1, pre_amp: 1.3 do
 3   #    with_fx :slicer, wave: 0, phase: 0.125, probability: 0.3 do
 4
 5     with_fx :slicer, phase: 0.125, wave: 0, invert_wave: 0, phase_offset: 0 do
 6       s = synth :square, note: :e1, cutoff: rrand(70, 100), sustain: 8, amp: 2, cutoff_slide: 8
 7       control s, cutoff: 30
 8     end
 9
10     sleep 8
11   #    end
12   end
13 end
14 live_loop :beats do
15   with_fx :slicer, phase: 0.125 do do
16     sample arovane_beat(24), beat_stretch: 16, amp: 3, rate: 1, cutoff: 70
17   end
18   sleep 16
19 end
20
21 live_loop :moon_bass do
22   #  sync :beats
23   sample :bd_haus, amp: 4, cutoff: (line 60, 130, steps: 256).ramp.tick
24   sleep 0.5
25 end
26
```

```
Syntax Error: [Line 15]
 syntax error, unexpected keyword_do_block
[Line 15]:   with_fx :slicer, phase: 0.125 do do
```

```
[Run 51, Time 0.0, Thread :live_loop_beats]
 ├─ cue :beats
 └─ sample "~/node/samps/arovane/beat/Arovane_

[Run 51, Time 0.0]
 └─ sync :algorave

[Run 51, Time 0.0]
 └─ synced :algorave (Run 51)

[Run 51, Time 0.0, Thread :live_loop_algorave
 ├─ cue :algorave
 ├─ synth :square, {note: 28, cutoff: 90.7642
 └─ control node 3834, {cutoff: 30}

[Run 51, Time 0.5, Thread :live_loop_moon_bas
 ├─ cue :moon_bass
 └─ sample :bd_haus, {amp: 4, cutoff: 60.2734

[Run 51, Time 1.0, Thread :live_loop_moon_bas
 ├─ cue :moon_bass
 └─ sample :bd_haus, {amp: 4, cutoff: 60.5469
```

```
# Rerezzed

# Coded by Sam Aaron

use_debug false
notes = (scale :e1, :minor_pentatonic, num_octaves: 2).shuffle
```

Preferences: Audio | Editor | Studio
Show and Hide: ☑ Show line numbers ☑ Show log ☑ Show buttons ☑ Show Tabs
Logging: ☑ Log synths ☑ Log cues ☑ Clear log on run
Look and Feel: ☑ Dark mode ■ Full screen

Live loop

The key to live coding with Sonic Pi is mastering the **live_loop**. Let's look at one:

```
live_loop :beats do
  sample :bd_haus
  sleep 0.5
end
```

There are four core ingredients to a **live_loop**. The first is its name. Our **live_loop** above is called **:beats**. You're free to call yours anything you want. Go crazy. Be creative. I often use names that communicate something about the music they're making to the audience. The second ingredient is the **do** word, which marks where the **live_loop** starts. The third is the **end** word, which marks where the **live_loop** finishes. Finally, there is the body of the **live_loop**, which describes what the loop is going to repeat – that's the bit between the **do** and **end**. In this case, we're repeatedly playing a bass drum sample and waiting for half a beat. This produces a nice regular bass beat. Go ahead: copy it into an empty Sonic Pi buffer and hit Run. Boom, boom, boom!

Redefining on-the-fly

OK, so what's so special about the **live_loop**? So far it just seems like a glorified 'loop'! Well, the beauty of **live_loops** is that you can redefine them on-the-fly. This means that while they're still running, you can change what they do. This is the secret to live coding with Sonic Pi. Let's have a play:

```
live_loop :choral_drone do
  sample :ambi_choir, rate: 0.4
  sleep 1
end
```

Now hit the Run button or press **ALT+R**. You're now listening to some gorgeous choir sounds. Now, while

Language
>RUBY

it's still playing, change the rate from **0.4** to **0.38**. Hit Run again. Whoa! Did you hear the choir change note? Change it back up to **0.4** to return to how it was. Now, drop it to **0.2**, down to **0.19**, and then back up to **0.4**. See how changing just one parameter on-the-fly can give you real control of the music? Now play around with the rate yourself, choosing your own values. Try negative numbers, really small numbers, and large numbers. Have fun!

Sleeping is important

One of the most important lessons about **live_loop**s is that they need rest. Consider the following **live_loop**:

```
 live_loop :infinite_impossibilities do
  sample :ambi_choir
 end
```

If you try running this code, you'll immediately see Sonic Pi complaining that the **live_loop** did not sleep. This is a safety system kicking in! Take a moment to think about what this code is asking the computer to do. That's right, it's asking the computer to play an infinite amount of choir samples in zero time. Without the safety system, the poor computer will try to do this and crash and burn in the process. So remember- your **live_loop**s must contain a **sleep**.

Combining sounds

Music is full of things happening at the same time. Drums at the same time as bass at the same time as vocals at the same time as guitars... In computing we call this concurrency, and Sonic Pi provides us with an amazingly simple way of playing things at the same time. Simply use more than one **live_loop**!

```
  live_loop :beats do
   sample :bd_tek
   with_fx :echo, phase: 0.125, mix: 0.4 do
    sample  :drum_cymbal_soft, sustain:
0, release: 0.1
    sleep 0.5
   end
  end

  live_loop :bass do
   use_synth :tb303
   synth :tb303, note: :e1, release: 4,
cutoff: 120, cutoff_attack: 1
   sleep 4
  end
```

Here, we have two **live_loop**s: one looping quickly, making beats; another looping slowly, making a crazy bass sound.

One of the interesting things about using multiple **live_loop**s is that they each manage their own time. This means it's really easy to create interesting polyrhythmical structures and even play with phasing, Steve Reich-style. Check this out:

Steve Reich's piano phase

```
  notes = (ring :E4, :Fs4, :B4, :Cs5,
:D5, :Fs4, :E4, :Cs5, :B4, :Fs4, :D5, :Cs5)

  live_loop :slow do
   play notes.tick, release: 0.1
   sleep 0.3
  end

  live_loop :faster do
   play notes.tick, release: 0.1
   sleep 0.295
  end
```

Bringing it all together

We are going to round this tutorial up with a new piece of music which draws from all of the ideas we have introduced so far. Read this code and see if you can imagine what it's doing. Then, copy it into a fresh Sonic Pi buffer, hit Run and hear what it actually sounds like. Finally, change one of the numbers, or comment and uncomment things some of the lines out. See if you can use this as a starting point for a new performance. Above all, have fun!

```
with_fx :reverb, room: 1 do
  live_loop :time do
   synth :prophet, release: 8, note: :e1, cutoff: 90, amp: 3
   sleep 8
  end
 end

 live_loop :machine do
  sample :loop_garzul, rate: 0.5, finish: 0.25
  sample :loop_industrial, beat_stretch: 4, amp: 1
  sleep 4
 end

 live_loop :kik do
  sample :bd_haus, amp: 2
  sleep 0.5
 end

 with_fx :echo do
  live_loop :vortex do
   # use_random_seed 800
   notes = (scale :e3, :minor_pentatonic, num_octaves: 3)
   16.times do
    play notes.choose, release: 0.1, amp: 1.5
    sleep 0.125
   end
  end
 end
```

JASPER HAYLER-GOODALL

The brains behind one of the winning Astro Pi entries, and a regular helper at CoderDojo Ham and Code Club. He's ten.
coderdojoham.org

TNT RUN!

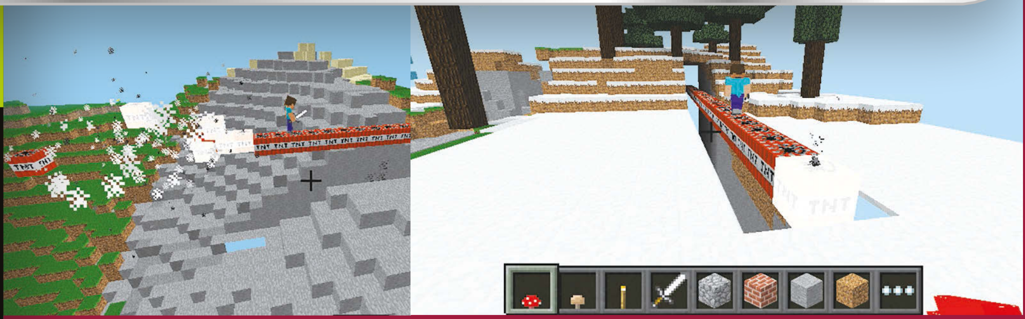
Can you outrun an explosion?
Test your speed with this marvellous *Minecraft* mini-game!

INFO CARD

Difficulty
Medium

Type
Programming

Requirements
Raspberry Pi computer
Raspbian from
raspberrypi.org/downloads



Bash the first block to trigger the chain reaction along the TNT

One of the many amazing things about Raspberry Pi computers is that they come with their own edition of *Minecraft* included for free. And even better, you can code it in Python using the Minecraft API! In the next 30 minutes you will create a game called TNT RUN, in which you start at one end of a long line of TNT and have to try to make it to the safe area without the TNT exploding in your face.

The Minecraft API gives us complete control over many elements of the game. This includes teleporting players around the world and displaying helpful messages to the screen. We are also able to place blocks automatically – not just one at a time, but as a three-dimensional collection of blocks.

This game will also include a block which is unique to Pi and Pocket editions: invisible bedrock. We use this block to keep the TNT from falling to the ground when lit, and as an invisible path leading to the safe area. You may find something strange about this block when you place it next to a non-invisible block, so try experimenting: you will find that you need to look directly into the invisible bedrock.

TNT also behaves differently in the Pi edition. Whereas in other editions you set TNT off with flint and steel, a fire charge or a flaming arrow, in the Pi edition you simply need to hit it a couple of times with anything. However, you can't just do this with any old TNT block: first, we need to set its block data value to an odd number (1, 3, 5, 7, or 9). Most blocks have a block data value; by changing this value, you will alter the block's behaviour. For example, when you set a nether reactor core's block data value to 1, it appears in a red colour; if you set it to 2 then it will come out as a dark blue colour. We use these cool-looking blocks to

TNTRUN.py

Language
>PYTHON 2.7

DOWNLOAD:
github.com/snake48/TNTRUN

```
# import all the necessary modules
from mcpi.minecraft import Minecraft
from mcpi import block
import time

# connect with the Minecraft world
mc=Minecraft.create()

# get the player’s position
pos=mc.player.getTilePos()

# check if the end of the world will engulf your
#creation & then move you if you’re too close
if pos.z<-40:
    mc.postToChat(
    'teleporting to safer distance in progress!')
    mc.player.setPos(pos.x,pos.y,-40)
    pos=mc.player.getTilePos()

# mark where the teleport is
zpos=pos.z-40

# create the valley by hollowing it out with air
mc.setBlocks(pos.x-1,pos.y+3,pos.z,pos.x+1,pos.y-
7,pos.z-88,block.AIR.id)

#build the invisible bedrock support
mc.setBlocks(pos.x,pos.y-1,pos.z,pos.x,
pos.y-7,pos.z,block.BEDROCK_INVISIBLE.id)
mc.setBlocks(pos.x-1,pos.y-1,pos.z,pos.x,
pos.y-7,pos.z,block.BEDROCK_INVISIBLE.id)
mc.setBlocks(pos.x+1,pos.y-1,pos.z,pos.x,
pos.y-7,pos.z,block.BEDROCK_INVISIBLE.id)
mc.setBlocks(pos.x,pos.y-1,pos.z-88,pos.x-1,pos.y-
7,pos.z-88,block.BEDROCK_INVISIBLE.id)
mc.setBlocks(pos.x-1,pos.y-1,
pos.z-88,pos.x,pos.y-7,pos.z-88,
block.BEDROCK_INVISIBLE.id)
mc.setBlocks(pos.x+1,
pos.y-1,pos.z-88,pos.x,
pos.y-7,pos.z-88,block.BEDROCK_INVISIBLE.id)
mc.setBlocks(pos.x,pos.y,pos.z,pos.x,
pos.y-7,pos.z-92,block.BEDROCK_INVISIBLE.id)

# build the bomb
mc.setBlocks(pos.x,pos.y,pos.z,pos.x,pos.y,
pos.z-88,block.TNT.id,1)

# build the end podium
mc.setBlocks(pos.x-2,pos.y,pos.z-93,
pos.x+2,pos.y,pos.z-97,block.GLOWING_OBSIDIAN.id)
mc.setBlocks(pos.x-1,pos.y+1,pos.z-94,pos.x+1,
pos.y+1,pos.z-96,block.NETHER_REACTOR_CORE.id,1)
mc.setBlock(pos.x,pos.y+2,pos.z-95,
block.REDSTONE_ORE.id)

# set how many teleports you have
teleport=1

# build the display teleport signal block
mc.setBlock(pos.x+1,pos.y+1,pos.z-44,
block.NETHER_REACTOR_CORE.id,2)
mc.setBlock(pos.x-1,pos.y+1,pos.z-44,
block.NETHER_REACTOR_CORE.id,2)


# teleport player when at a certain position
while teleport ==1:
    pos=mc.player.getTilePos()
    if pos.z==zpos:
        mc.player.setPos(pos.x,pos.y,pos.z-24)
        teleport=0
```

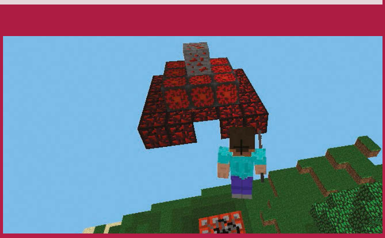
HOW TO PLAY:

- The start of the TNT line will appear directly underneath you.
- The TNT won’t just light itself; you need to detonate the first block of TNT by hitting it a couple of times.
- Once you’ve lit it (you’ll know when it starts flashing white), turn around and then run for your life!
- Try to get to the teleporter, as it will give you an 18-block boost.
- If the TNT catches up with you, you will be flung up into the air and will lose.

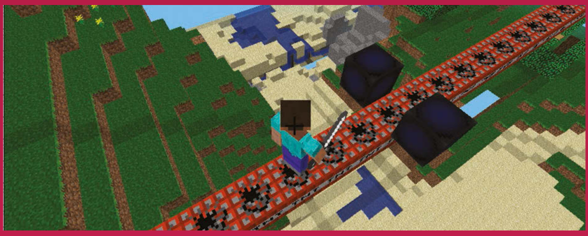
mark the position of the teleporter and as a part of the end podium. The teleporter function in the code is designed so that if you manage to get to a certain point along the line of TNT, it gives you a boost and teleports you forward.

When you create your *Minecraft* world, fly around it and find a cool location to create your TNT course (i.e. not when you are near a cliff or the end of the world). Your code takes the player’s position and constructs the TNT course, using this location as the starting point. When you run the code, make sure *Minecraft* is running, otherwise your code will give you a connection error.

Once you have finished the project, you can customise your own version of the TNT Run game by adding changes like making the row of TNT longer, or creating a fancier safe area. Also, when you’re connected to a network with other Raspberry Pis, you can join someone else’s world, create two lines of TNT and race to the safe areas simultaneously!



This is the safe area you need to run to. If you make it there, you’ve won



The dark blue nether reactor core marks the location of the teleporter

SUBSCRIBE **TODAY**

& SAVE UP TO 40%

Subscribe to the official Raspberry Pi mag at our exclusive educator's rate

Subscription benefits

- Get it first (before stores)
- Free delivery to your door
- Never miss a single issue

Pricing

Quarterly Direct Debit:
£10.50

Six issues:
£23

12 issues:
£45

NEW TEACHER OFFER!

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Issue 41 January 2016
raspberrypi.org/magpi
MINECRAFT MASHUPS
Amazing Pi projects that bring Minecraft's virtual world to life
Make music · Control electronics · Simulate natural disasters · and much more
SENSE HAT SPECIAL
10 pages of awesome games & guides
PI IN SPACE!
It's a giant leap for Raspberry Pi kind...
#PIZERO PROJECTS
More amazing creations with the $5 Raspberry Pi!
MAKE MUSIC WITH SONIC PI
...with its creator, Sam Aaron
NEW YEAR, NEW YOU!
New Year's resolutions made possible with your Raspberry Pi
Also inside:
MORE OF YOUR AMAZING PI PROJECTS
ALL YOUR #PIZERO QUESTIONS ANSWERED
BUILD A RETRO CONSOLE IN A CONTROLLER
GET STARTED WITH THE CAMJAM ROBOT KIT
THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY
Issue 41 • Jan 2016 • £5.99

How to subscribe:

- Call on **01202 586848** and ask us about our educator rates

Find us on your digital device by searching for 'The MagPi'

Available on the App Store

Get it on Google play