Welcome to the new MagPi, now the official Raspberry Pi magazine. Before we go any further, massive thanks and congratulations need to go to the original MagPi team, and all the volunteers. You’ll be seeing many of them again, once they’ve had a well-earned break. Running a monthly magazine in a full-time capacity is challenging enough, let alone as a group of enthusiasts taking on the task during evenings and weekends. The success of The MagPi over the last few years is testament to the amazing Raspberry Pi community and a clear indication of what it can achieve. What else can it do? That’s what The MagPi is here to share!

The magazine might look a bit different, but it’s still made for and by the Raspberry Pi community and we’ll be sharing your amazing projects, tips and tricks every single issue. The MagPi is still very much committed to open source, too. You can download it free online and it still operates under the same Creative Commons licence. To download the magazine (in PDF format) and find out more about the licence, visit raspberrypi.org/magpi.

If you’ve made something with the Raspberry Pi, want to share your tips and tricks, need help with a technical issue, or simply want to let us know what you think of the new magazine, please get in touch via magpi@raspberrypi.org.

I’m really looking forward to hearing from you and sharing what you’re doing with the world’s favourite credit card-sized PC.

Russell Barnes

With thanks to this month’s contributors: David Crookes, Liam Fraser, David Hunt, Phil King, Simon Long, Simon Monk, Martin O’Hanlon, Les Pounder, Matt Richardson, Richard Saville, Simon M. Tracey and Robin Withers.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in this 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 2051-9990.
EVERYDAY ENGINEERING (PT 1) 34
Dr Simon Monk shows us how to build a parking sensor

MINECRAFT: PI CODING TIPS 38
Martin 'Minecraft' O'Hanlon shares five of his favourite creative coding tips and tricks

SHOOT SLOW-MOTION VIDEOS 42
"Drive through the boxes, it makes the car look good!"

MAKE A CANDLE LANTERN 44
Create romantic lighting using pulse-width modulation

WATER DROP PHOTOGRAPHY 46
Capture incredible DSLR photos with the help of Pi

COMMAND LINE Pi (PT 1) 48
Join Richard Smietal on the first part of his quest to help you love the command line as much as it loves you

GAMES WITH PYTHON (PT 1) 50
In part one of our new series, Sean M Tracey creates the building blocks of all games: shapes and paths

RASPBERRY Pi 2
It’s priceless and infinitely powerful, at least according to our account of the Raspberry Pi Foundation’s new credit-card-sized PC. Here’s why.

MEET THE MAKERS 30
Meet the technologists of tomorrow as we interview four young hackers and makers, and learn what they do with Pi

NEWS FEATURE 10
Find out how UK schoolchildren can win the chance to run experiments on the International Space Station

REGULARS

BOOK REVIEWS 62
The latest computer books reviewed and rated

COMMUNITY EVENTS 64
Find a community gathering near you in the coming weeks

THE FINAL WORD 68
Matt Richardson shares his earliest Raspberry Pi memories

RETIRO GAMING GROUP TEST 56
Les Pounder looks at three of the best emulation applications

SKYWRITER HAT 60
Should you control your Pi with a flick of the wrist?

ANONYMEBOX 61
Anonymise your web browsing with this new Pi-powered app

LEGO-LUTION OF PI 22
To celebrate the third birthday of the Raspberry Pi, the Hayler family did what they do best and built a massive LEGO diorama

Pi CLUSTER 24
We catch up with David Guitt to see what he’s been doing with his cluster powered by 40 Pis

iDATA TRUCK 26
The MagPi talks to Andy Proctor about his Raspberry Pi-powered truck

SNES Pi CASE 26
Mark Parrish shares details of his excellent Super Nintendo Pi hack

WHAT would you do with a Pi 2? Let us know for your chance to win!

YOUR PROJECTS

SAVE 45% with a Newsstand subscription

Coming soon

to iOS & Android devices
**FEATURE**

**TAKING UP ARMS**

When sales of the Raspberry Pi eventually exceed that of the ZX80, ZX81 and ZX Spectrum combined, the consensus is that the small-form computer will become Britain’s most successful of all time. But will it?

According to Stephen Furber, one of the designers of the BBC Micro, a lot depends on how the terms are defined. “More than 6 billion ARM processors have been shipped in total, and the rate is now over 12 billion a year in a small subset of which are, of course, going into the Raspberry Pi, so ARM is clearly the most successful British computer of all time, and indeed the most successful computer in the world,” he says. “But maybe that ARM doesn’t count as a computer because it is a microchip – or a part of a microchip – that needs other components to operate?”

---

**FIVE MILLION SALES**

**RASPBERRY PI SALES PASS FIVE MILLION**

But does that make it the bestselling UK computer of all time? The MagPi delves into the past to find out...

When Upton’s firm belief was that the microchip – or a part of a microchip – that needed other components to operate? is a microchip – or a part of a microchip – that needs other components to operate? is a microchip – or a part of a microchip – that needs other components to operate? is a microchip – or a part of a microchip – that needs other components to operate?.

**SHOULD THIS TREND CONTINUE, WE COULD, IN THEORY, SEE AN EXTRA 6 MILLION SALES IN A YEAR’S TIME**

The Raspberry Pi took 20 months to achieve sales of 1.75 million, yet just 18 months more to add 3.25 million. Should this trend continue, we could in theory see an extra 6 million sales in a year’s time, though Upton touted a million as a target for the year. Whatever the truth in that, from far waning, the Pi is becoming more popular. Even Sir Clive Sinclair, the brains behind the Spectrum, has been impressed by its impact.

“It’s very exciting,” he said. “I think it’s a dramatic and terribly clever.” He has praised both its low cost and its accessibility which, like his Spectrum, allows users to quickly start coding. “Suddenly people can again get their hands on computing power and play with it, manipulate it and really understand it.”

All of this is undoubtedly brilliant news for the Foundation, which created the computer as a tool to get children coding. Over the past three years, the Pi has become as ubiquitous in British schools as the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

After all, the BBC Micro was produced to complement the BBC Computer Literacy Project, which aimed to familiarise pupils with the ins and outs of these newfangled machines back in 1981. When the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

The Raspberry Pi took 20 months to achieve sales of 1.75 million, yet just 18 months more to add 3.25 million. Should this trend continue, we could in theory see an extra 6 million sales in a year’s time, though Upton touted a million as a target for the year. Whatever the truth in that, from far waning, the Pi is becoming more popular. Even Sir Clive Sinclair, the brains behind the Spectrum, has been impressed by its impact.

“It’s very exciting,” he said. “I think it’s a dramatic and terribly clever.” He has praised both its low cost and its accessibility which, like his Spectrum, allows users to quickly start coding. “Suddenly people can again get their hands on computing power and play with it, manipulate it and really understand it.”

All of this is undoubtedly brilliant news for the Foundation, which created the computer as a tool to get children coding. Over the past three years, the Pi has become as ubiquitous in British schools as the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

After all, the BBC Micro was produced to complement the BBC Computer Literacy Project, which aimed to familiarise pupils with the ins and outs of these newfangled machines back in 1981. When the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

The Raspberry Pi took 20 months to achieve sales of 1.75 million, yet just 18 months more to add 3.25 million. Should this trend continue, we could in theory see an extra 6 million sales in a year’s time, though Upton touted a million as a target for the year. Whatever the truth in that, from far waning, the Pi is becoming more popular. Even Sir Clive Sinclair, the brains behind the Spectrum, has been impressed by its impact.

“It’s very exciting,” he said. “I think it’s a dramatic and terribly clever.” He has praised both its low cost and its accessibility which, like his Spectrum, allows users to quickly start coding. “Suddenly people can again get their hands on computing power and play with it, manipulate it and really understand it.”

All of this is undoubtedly brilliant news for the Foundation, which created the computer as a tool to get children coding. Over the past three years, the Pi has become as ubiquitous in British schools as the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

After all, the BBC Micro was produced to complement the BBC Computer Literacy Project, which aimed to familiarise pupils with the ins and outs of these newfangled machines back in 1981. When the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

The Raspberry Pi took 20 months to achieve sales of 1.75 million, yet just 18 months more to add 3.25 million. Should this trend continue, we could in theory see an extra 6 million sales in a year’s time, though Upton touted a million as a target for the year. Whatever the truth in that, from far waning, the Pi is becoming more popular. Even Sir Clive Sinclair, the brains behind the Spectrum, has been impressed by its impact.

“It’s very exciting,” he said. “I think it’s a dramatic and terribly clever.” He has praised both its low cost and its accessibility which, like his Spectrum, allows users to quickly start coding. “Suddenly people can again get their hands on computing power and play with it, manipulate it and really understand it.”

All of this is undoubtedly brilliant news for the Foundation, which created the computer as a tool to get children coding. Over the past three years, the Pi has become as ubiquitous in British schools as the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

After all, the BBC Micro was produced to complement the BBC Computer Literacy Project, which aimed to familiarise pupils with the ins and outs of these newfangled machines back in 1981. When the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

The Raspberry Pi took 20 months to achieve sales of 1.75 million, yet just 18 months more to add 3.25 million. Should this trend continue, we could in theory see an extra 6 million sales in a year’s time, though Upton touted a million as a target for the year. Whatever the truth in that, from far waning, the Pi is becoming more popular. Even Sir Clive Sinclair, the brains behind the Spectrum, has been impressed by its impact.

“It’s very exciting,” he said. “I think it’s a dramatic and terribly clever.” He has praised both its low cost and its accessibility which, like his Spectrum, allows users to quickly start coding. “Suddenly people can again get their hands on computing power and play with it, manipulate it and really understand it.”

All of this is undoubtedly brilliant news for the Foundation, which created the computer as a tool to get children coding. Over the past three years, the Pi has become as ubiquitous in British schools as the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

After all, the BBC Micro was produced to complement the BBC Computer Literacy Project, which aimed to familiarise pupils with the ins and outs of these newfangled machines back in 1981. When the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

The Raspberry Pi took 20 months to achieve sales of 1.75 million, yet just 18 months more to add 3.25 million. Should this trend continue, we could in theory see an extra 6 million sales in a year’s time, though Upton touted a million as a target for the year. Whatever the truth in that, from far waning, the Pi is becoming more popular. Even Sir Clive Sinclair, the brains behind the Spectrum, has been impressed by its impact.

“It’s very exciting,” he said. “I think it’s a dramatic and terribly clever.” He has praised both its low cost and its accessibility which, like his Spectrum, allows users to quickly start coding. “Suddenly people can again get their hands on computing power and play with it, manipulate it and really understand it.”

All of this is undoubtedly brilliant news for the Foundation, which created the computer as a tool to get children coding. Over the past three years, the Pi has become as ubiquitous in British schools as the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.

After all, the BBC Micro was produced to complement the BBC Computer Literacy Project, which aimed to familiarise pupils with the ins and outs of these newfangled machines back in 1981. When the BBC Micro was in the 1980s, and the irony of this situation is rather delicious.
GET A SLICE OF THE ACTION

The crowdfunded media player powered by the Raspberry Pi Compute Module is shipping soon, with online pre-orders coming immediately after...

Slice, the Raspberry Pi-powered media player, was successfully funded on Kickstarter in September last year, shattering its £90,000 goal on its way to raising over £227,000 (£345,000). While it was originally due for launch in December 2014, the team behind the HD media box, known as FiveNinjas, have now started the delivery process.

“The backers who just wanted a Slice remote control have already got them,” explains Paul Beech, member of FiveNinjas, a co-founder of Pimoroni.com and the designer of the Raspberry Pi logo. While delivery of actual Slice hardware still hasn’t taken place, it is coming soon. “Everything is ordered and in production, and we’ve already got things like cables, Wi-Fi, remotes, and power supplies in hand.

“We’ve had delays when getting the test plan perfect on the PCB [printed circuit board]. It’s a complex beast, so getting that right has taken more time and communication, which has been through intermediaries, slowing things down. A bad product is bad forever, so we’re being super-careful about the process. Even though we’re late on our deadline, this is still the quickest I’ve seen a complex hardware Kickstarter delivered,” he said.

It seems Slice buyers will be getting much more than they originally bargained for, with Beech revealing that a large amount of future-proofing has taken place: “It’s a box you’ll still be using years from now. The Compute Module is a good part of that. When the Raspberry Pi Foundation update the Compute Module, we expect it to be a slot-in replacement to upgrade your Slice.”

We also asked Beech what the FiveNinjas’ plans were, post-Kickstarter delivery. Will Slice be available to those that missed out on the Kickstarter? “Backers come first,” said Beech. “Once we’re happy that delivery to them is going smoothly, we’ll spin up a shop on FiveNinjas.com so everyone can get a Slice.”
LAUNCH YOUR CODE INTO SPACE WITH ASTRO Pi

UK astronaut Major Tim Peake offers UK schoolchildren the chance to launch their code into space with a competition that sees two Pis travel to the International Space Station...

E ve dreamed of getting into space? It takes many years of training before those with the right stuff are selected for missions beyond planet Earth, but thanks to the Raspberry Pi, UK schoolchildren have the chance this year to send their code up to the International Space Station (ISS), when Major Tim Peake takes off in November for a six-month mission. Each ISS-bound Raspberry Pi will carry a new Astro Pi board, loaded with sensors and gadgets well-suited to getting real science done in space. The gadget list of the Astro Pi is impressive and too long to detail in full here, but includes a gyroscope, accelerometer, barometric pressure sensor, and magnetometer, among other things. Add in the regular camera module or infrared camera (both versions will be travelling into space) and, like us, you may already be thinking of many possible experiments for more Earth-bound Pi boards as well. Speaking at the UK Space Agency during the launch of the Astro Pi in December, Business Secretary Vince Cable spoke of the government’s industrial strategy to create a new generation of engineers with world-class skills: “So much technology relies on big data, but not enough people are being trained in this field. This challenge helps the next generation to have fun whilst learning the skills that industry need.” Major Tim, also attending the launch event, spoke about his excitement at the project’s cooperation between UK industries and institutions, and the potential of the Astro Pi on board the ISS. “There is huge scope for fun science and useful data gathering, using the Astro Pi sensors on board the International Space Station. This competition offers a unique chance for young people to learn core computing skills that will be extremely useful in their future,” he commented. “It’s going to be a lot of fun!”

The major couldn’t be at the competition opening at BETT (education technology show), where the Pi Foundation was out in force – being an astronaut involves a lot of training and not much getting out to events – but he was there virtually, in cartoon form. Take a look at the competition launch video at vimeo.com/117274487, where Major Tim explains that “brilliant as the tech is, it’s nothing without you. We need your ideas for space experiments using the kit, and your code, to make it work.” The possibilities seem endless, but to help focus creative thinking about the uses of the Astro Pi, there are five themes to the competition: Spacecraft Sensors, Satellite Imaging, Space Measurements, Data Fusion and Space Radiation. The entries will be judged on creativity, originality, practicality and usefulness.

Ground control We got to speak about the Astro Pi with Libby Jackson – the UK Space Agency’s astronaut flight education programme manager, who is supporting Tim Peake’s flight to the International Space Station in 2015/16. Before joining the UK Space Agency, she was a Columbus flight director, working at the Columbus Control Centre in Munich, Germany – the European control centre for the ISS – and she is passionately enthusiastic about every aspect of the space programme: “I’ve been working on manned space flight for most of my career, and to be back in the UK working on the first government-supported manned British mission is fantastic.” We ask her about the possibilities of real and useful measurements with the Astro Pi on board the ISS. What is the UK Space Agency expecting? “I know we’ll be surprised,” Jackson tells us. “Every time we run competitions, children in the UK – and children generally – always have amazing ideas that we...
havens’t even thought of. That’s the exciting part of the competition. By putting all of these sensors together and letting everybody have a think, we know that we’ll get some ideas we never ever thought of,” Jackson mentions putting the camera in the cupola – the dome-shaped window on the ISS, looking down on Earth – and taking pictures, adding “maybe using the magnetometer to come up with where you are,” and speaks of the other sensors’ possibilities. She is enthusiastic about the Raspberry Pi (“fantastic British innovation, part of a coding renaissance”) and the part it can play in STEM (science, technology, engineering and mathematics) education.

Open source space

This isn’t the Raspberry Pi board’s first brush with space. Many readers will have followed with interest the helium-fuelled Pi missions of Dave Akerman (see below), or may have experimented with astrography, attaching the Pi camera board to an amateur telescope. Beyond the Pi, free and open source software has been democratising space research elsewhere. NASA’s SpaceGAMBIT is a US government-funded open source space programme, reaching out to makerspaces and hackspaces across the world to collaborate, last year focusing on projects for NASA’s Asteroid Grand Challenge.

More hands-on was the ISEE-3 Reboot project, which decommissioned the hardware necessary to communicate with the craft in 1993, but provided the archive material necessary for a group of amateurs to crowdfund provision of the software and hardware to communicate with ISEE-3 when it passed close to the Earth again last summer. Sadly, power issues on the craft meant the new data-collecting mission didn’t get far, but the project both showed the scope of collaborative amateur efforts, and underlined the importance of making the most of hard space. The chance to control the Astro Pi’s sensors while aboard the ISS is a phenomenal opportunity. The ISS itself made headlines following a number of what Eugene Honess excitedly tweets a link to the Astro Pi capsule, at the Gagarin Cosmonaut Training Centre in Russia

Below The Raspberry Pi Foundation’s David Honess excitedly tweets a link to the Astro Pi competition launch video.

The most rewarding thing was the opportunity to control the Astro Pi’s sensors while aboard the ISS is a phenomenal opportunity. The ISS itself made headlines following a number of what Eugene Honess excitedly tweets a link to the Astro Pi capsule, at the Gagarin Cosmonaut Training Centre in Russia.

Kaspersky called “virus epidemics” among the Windows laptops on board. Other space efforts have gone further towards open source – at the silicon level – with nano satellites based on the OpenSource system-on-chip, a MIPS design maintained and modified by academics, amateurs and several specialist chip-design companies.

Careering ahead

Meanwhile, back on Earth, the shortage of engineers and scientists – and not just in the space industry – has led to initiatives from the grassroots (like Code Club), as well as government and industry, to inspire more people to consider STEM careers. The Astro Pi competition is a result of UK Space companies, the UK Space Agency, and the Raspberry Pi Foundation working together: “UK Space – the industry consortium group – responding to Innovation Growth Strategy, the government’s call for ideas in space.” Space and science organisations involved include SSTL, Airbus DS, CIG, the Space KN, National Nuclear Laboratory, and National Physical Laboratory.

Libby Jackson tells us, “Studying science and maths at GCSE and A-Level doesn’t close your options down, but does the opposite and opens you up to a whole range of careers,” citing her own discovery of chances to work in the space industry while studying physics. The UK Space Agency’s educational aims encompass both Space Education and Education for Space: “We need to encourage people in,” Jackson explains, “but we also know space is a fantastic inspiration for learning in general. We’d like everyone to move one step up the ladder. So we’d like a lot of people who’ve never considered science before to start looking at it, the people who are interested in it maybe having a career in it, the people who are thinking about a career in it – maybe bring them into the space industry.”

Supported by £2 million of government outreach money, the UK Space Education Office and Raspberry Pi are developing teaching resources to link to the curriculum and assist teachers of STEM subjects in engaging their students in the competition, as well as explaining how to use and write code for the Astro Pi and its sensors.

Having kept the project secret for most of 2014, David Honess – education resource engineer at the Raspberry Pi Foundation – is delighted to talk about it at last: “We think this competition has the power to motivate a whole generation of coders for the UK science, technology and space industries to employ in the future… We want every school in the UK to enter!”

We know from some of the great Pi projects that we’ve seen that whatever makes it to the ISS is bound to be something special. From astrophotography, through space-bound balloons, to the Astro Pi, the Raspberry Pi is making giant leaps for a little computer. So, in the words of Major Tim at the competition launch, “What are you waiting for, Earthlings? Get those intergalactic hats on and let’s get coding. See you in space!”

Kaspersky called “virus epidemics” among the Windows laptops on board. Other space efforts have gone further towards open source – at the silicon level – with nano satellites based on the OpenSource system-on-chip, a MIPS design maintained and modified by academics, amateurs and several specialist chip-design companies.

Careering ahead

Meanwhile, back on Earth, the shortage of engineers and scientists – and not just in the space industry – has led to initiatives from the grassroots (like Code Club), as well as government and industry, to inspire more people to consider STEM careers. The Astro Pi competition is a result of UK Space companies, the UK Space Agency, and the Raspberry Pi Foundation working together: “UK Space – the industry consortium group – responding to Innovation Growth Strategy, the government’s call for ideas in space.” Space and science organisations involved include SSTL, Airbus DS, CIG, the Space KN, National Nuclear Laboratory, and National Physical Laboratory.

Libby Jackson tells us, “Studying science and maths at GCSE and A-Level doesn’t close your options down, but does the opposite and opens you up to a whole range of careers,” citing her own discovery of chances to work in the space industry while studying physics. The UK Space Agency’s educational aims encompass both Space Education and Education for Space: “We need to encourage people in,” Jackson explains, “but we also know space is a fantastic inspiration for learning in general. We’d like everyone to move one step up the ladder. So we’d like a lot of people who’ve never considered science before to start looking at it, the people who are interested in it maybe having a career in it, the people who are thinking about a career in it – maybe bring them into the space industry.”

Supported by £2 million of government outreach money, the UK Space Education Office and Raspberry Pi are developing teaching resources to link to the curriculum and assist teachers of STEM subjects in engaging their students in the competition, as well as explaining how to use and write code for the Astro Pi and its sensors.

Having kept the project secret for most of 2014, David Honess – education resource engineer at the Raspberry Pi Foundation – is delighted to talk about it at last: “We think this competition has the power to motivate a whole generation of coders for the UK science, technology and space industries to employ in the future… We want every school in the UK to enter!”

We know from some of the great Pi projects that we’ve seen that whatever makes it to the ISS is bound to be something special. From astrophotography, through space-bound balloons, to the Astro Pi, the Raspberry Pi is making giant leaps for a little computer. So, in the words of Major Tim at the competition launch, “What are you waiting for, Earthlings? Get those intergalactic hats on and let’s get coding. See you in space!”

Kaspersky called “virus epidemics” among the Windows laptops on board. Other space efforts have gone further towards open source – at the silicon level – with nano satellites based on the OpenSource system-on-chip, a MIPS design maintained and modified by academics, amateurs and several specialist chip-design companies.

Careering ahead

Meanwhile, back on Earth, the shortage of engineers and scientists – and not just in the space industry – has led to initiatives from the grassroots (like Code Club), as well as government and industry, to inspire more people to consider STEM careers. The Astro Pi competition is a result of UK Space companies, the UK Space Agency, and the Raspberry Pi Foundation working together: “UK Space – the industry consortium group – responding to Innovation Growth Strategy, the government’s call for ideas in space.” Space and science organisations involved include SSTL, Airbus DS, CIG, the Space KN, National Nuclear Laboratory, and National Physical Laboratory.

Libby Jackson tells us, “Studying science and maths at GCSE and A-Level doesn’t close your options down, but does the opposite and opens you up to a whole range of careers,” citing her own discovery of chances to work in the space industry while studying physics. The UK Space Agency’s educational aims encompass both Space Education and Education for Space: “We need to encourage people in,” Jackson explains, “but we also know space is a fantastic inspiration for learning in general. We’d like everyone to move one step up the ladder. So we’d like a lot of people who’ve never considered science before to start looking at it, the people who are interested in it maybe having a career in it, the people who are thinking about a career in it – maybe bring them into the space industry.”

Supported by £2 million of government outreach money, the UK Space Education Office and Raspberry Pi are developing teaching resources to link to the curriculum and assist teachers of STEM subjects in engaging their students in the competition, as well as explaining how to use and write code for the Astro Pi and its sensors.

Having kept the project secret for most of 2014, David Honess – education resource engineer at the Raspberry Pi Foundation – is delighted to talk about it at last: “We think this competition has the power to motivate a whole generation of coders for the UK science, technology and space industries to employ in the future… We want every school in the UK to enter!”

We know from some of the great Pi projects that we’ve seen that whatever makes it to the ISS is bound to be something special. From astrophotography, through space-bound balloons, to the Astro Pi, the Raspberry Pi is making giant leaps for a little computer. So, in the words of Major Tim at the competition launch, “What are you waiting for, Earthlings? Get those intergalactic hats on and let’s get coding. See you in space!”
RASPBERRY PI

Better, faster, stronger

he Raspberry Pi is a really tiny, really cheap computer. So cheap, in fact, you could sacrifice your Starbucks coffee for a week to afford one. Because it’s so small and affordable, it’s an excellent tool to teach computer education in schools. As the Raspberry Pi Foundation learned, however, the “bigger” kids among us quite enjoy playing with the Pi too.

Regardless of how old you are, you can hack, make, watch movies, and even play games with it. With the Raspberry Pi, though, it’s just as easy to start making your own game or movie as it is to passively consume one. This is one of the key things that sets the Raspberry Pi apart from its growing competition.

With the Raspberry Pi 2, those goal posts have shifted. With its sixfold increase in power, whether you’re making or playing games, movies or music, the experience is all the better for it.

On paper, the upgrade itself is pretty mundane stuff. The Pi 2 is essentially identical to a Model B+ in almost every respect. Other than its four 900MHz ARM Cortex-A7 cores and 1GB of RAM (as opposed to one 700MHz ARM Cortex-A6 core and 512MB of RAM), you could easily confuse the two.

The fact that there’s no killer application exclusive to the Raspberry Pi 2 is, conversely, its biggest asset and a super-weapon primed to stave off even the stiffest competition in the burgeoning ‘maker’ marketplace.

Best-of-British design and engineering is one thing, but a real commitment to powerful and flexible open source software, that offers near-total cross-compatibility between models, ensures this tiny, cheap computer is one of the most powerful in the world today, regardless of the model you’re using…
"We released the original Raspberry Pi on the 29th February 2012," says its creator Eben Upton. "It's been successful beyond our wildest dreams. Three years in, we've sold five million, and we think somewhere between one and two million Raspberry Pis are in the hands of children." While the launch of the Raspberry Pi 2 on the 35th floor of the Shard, one of London's most impressive new landmarks, was quite grand, the humble aim of the Raspberry Pi Foundation was never far from view. The goal has always been to get more kids into computing; to give children of today the same kind of experience people growing up during the home computing boom of the '80s and early '90s had – people of Eben Upton's generation.

"It's the idea of having a computer in the bedroom that's hackable and fun," says Upton. "In the first few months we were concerned they were only going in the pocket of people like me, but over time it's become clear there is interest from children in learning computing with the Raspberry Pi. As much as anything, there's interest from children in learning something their parents don't understand." Of course, even the Raspberry Pi Model B+ wasn't perfect. You can't build a $35 computer without making compromises. "The Raspberry Pi has a level of computing power of a PC from the turn of the century. Even when we doubled the RAM six months in, it still only had half a gigabyte." The Raspberry Pi 2 has been released to address these deficiencies, and more besides.

"The Raspberry Pi 2 takes us to a level of performance that makes it a genuine PC. That's the money shot: the realisation that actually you can have your cake and eat it. It's also the moment you realise that the single-board computer revolution just got interesting. The Raspberry Pi was the hacking and making board with brains, but the Raspberry Pi 2 takes the formula much further: real-time physics calculations, complex computer vision projects with the Camera Module, and anything else – up to and including complex weather simulation – with minimal investment in hardware. The Raspberry Pi 2 represents the backbone of the perfect university computer cluster.

"The Raspberry Pi 2 takes us to a level of performance that makes it a genuine PC. That's the money shot: the realisation that actually you can have your cake and eat it. It's also the moment you realise that the single-board computer revolution just got interesting. The Raspberry Pi was the hacking and making board with brains, but the Raspberry Pi 2 takes the formula much further: real-time physics calculations, complex computer vision projects with the Camera Module, and anything else – up to and including complex weather simulation – with minimal investment in hardware. The Raspberry Pi 2 represents the backbone of the perfect university computer cluster."
RASPBERRY PI 2

SIX THINGS THAT ARE BETTER WITH RASPBERRY PI 2

**Home office**
You can install the Raspberry Pi as a productivity machine. Now packages like LibreOffice.org’s open source alternative to Microsoft Office are responsive and usable.

**Web browsing**
Enjoy loading times four times faster than the old Model B – the Pi 2 hardware is much more adept at running modern websites. It’s still not perfect, but it’s nowhere near as frustrating as it used to be.

**Sonic Pi 2**
Despite a cruel bug that initially meant the live coding music application couldn’t use the full amount of RAM afforded to the Pi, the ability to create better beats and more complex compositions has drastically increased.

**Computer vision**
With the addition of the affordable Raspberry Pi Camera Module, the Raspberry Pi’s four faster cores make the evaluation and processing of images and video streams much easier.

**Retro gaming**
The Pi’s already a very popular solution for playing retro games, but the extra power from the Pi 2 opens up a whole world of new possibilities, including the emulation of fifth-generation consoles like the Nintendo 64.

**WaspPi 1**
```
Benedict Ryan Wednesday start a new project. The idea is simple: instead of doing science, we’re doing science.
```

**Future**
```
June 2015
Silver is better than gold
Eben Upton, creator of the Pi, receives the Royal Society of Public Understanding of Science award for outstanding service to UK engineering.

August 2015
A weekly slice of Pi
The official release of the Camera Module is a big step forward.

October 2015
Infrared in action!
This major update for the Camera Module in Raspberry Pi 2 provides near-infrared support.

November 2015
WaspPim is all about!
This is a major update for the WaspPi magazine, now available in PDF format.

December 2015
Adventures in Raspberry Pi
This month the Raspberry Pi Foundation reveals the Raspberry Pi 2, the latest computer of the Raspberry Pi family.

March 2016
The MagPi comes of age
The Raspberry Pi Foundation reveals the Raspberry Pi Magazine, now available in print.

April 2016
A new PC to count on!
A major update to the Compute Module 3 is now available, enabling huge improvements in performance.

May 2016
The Model B+ comes out
The Raspberry Pi Foundation announces the Raspberry Pi 3, with more power and more features.

June 2016
Ben goes to America
The Raspberry Pi Foundation is heading to America to showcase the latest in technology.

July 2016
An Internet of Things
The Raspberry Pi Foundation announces a new project to bring the Internet of Things to the Raspberry Pi.

August 2016
The Model A+
Announced, the Raspberry Pi Foundation announces the latest addition to the Raspberry Pi family.

September 2016
Astro Pi is announced
The Raspberry Pi Foundation announces the Astro Pi, a new project for space scientists.

October 2016
Raspberry Pi 2 is announced
The latest in Raspberry Pi technology is announced.
W hen Richard Hayler isn’t working for the Foreign Office he’s a Raspberry Pi enthusiast, CoderDojo mentor and Code Club volunteer. “Pretty much everything else revolves around my sons, who love getting involved with all things Pi,” he says. Besides test-driving his educational material, Richard’s sons Ozzy (aged 8) and Jasper (9) are often to be found hacking and making with the Pi. Their latest creation is this rather marvellous LEGO scene designed to celebrate the evolution of everyone’s favourite credit-card-sized PC.

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

“I asked the boys if they had any ideas of how to make the pictures more exciting than just a bunch of Pis on a desk, and they immediately suggested this.” After discussing a few ideas, Richard’s youngest, Ozzy, suggested creating a LEGO timeline showing the Pis being used in different ways. “This morphed into a scene which follows the Pi from the design phase, through manufacture in a Pi factory, to being loaded onto a lorry. Then we have some children using it in a school, and finally, a Pi being strapped to a rocket, ready to launch up to the ISS to celebrate Astro Pi.”

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

Ozzy, Jasper and Richard Hayler celebrate their collection of Raspberry Pis the only way they know how...
David Guill shows us what happens when he’s left in a room with 40 Raspberry Pis, two 24-port switches, 5TB of storage, and an ATX power supply.

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

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

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

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

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

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

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

www.dexterindustries.com
A Raspberry Pi-powered lorry? It’s not as strange as you think, as Andy Proctor shows us how he automates deliveries with Pi

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

Humble beginnings

“I started off with Tweepy and ‘#m12’, and I noticed it was being retweeted automatically. I contacted the guy who was doing it and within six weeks I had created the box with the four buttons that will display on his dashboard – in a blind spot of his lorry that I want to develop it further.”

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

“What’s next?

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

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

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

Quick Facts

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

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

You walk into a room and see a SNES. A classic, a legend, one of the greatest videogame consoles to ever be crafted by the hands of man. Beside it is a cartridge of the old F-Zero, perhaps not the best in the series but an excellent game nonetheless. You slam it in (gently though, they’re both 25 years old), flick the power switch and look for a controller. Suddenly the power light is fixed to a Raspberry Pi logo shows up. This isn’t a Super Nintendo. It’s a Raspberry Pi case that used to be an SNES he was working on. Most of the work on this project was the physical customisation part. "The software side is easy since there are numbers of solutions out there that have already been proven to be successful. The majority of the work I’ve done is with the physical part, and is easily 90-95% of the time invested."

F-Zero is one of Mark’s favourite SNES games. His own creative way is frankly incredible. Instead of just fitting the Raspberry Pi into an empty case, he connected it inside an actual game cartridge. I’ve noticed a few more failed boots than normally would be expected. Other than that, it works beautifully!""
To improve the Raspbian user experience and give the Pi its own identity, the Foundation has begun to work on customising and improving the desktop. Simon Long tells us more...

The term ‘user interface’ covers two aspects of software. First, its appearance – does it look good, does it draw you in and make you want to interact with it? Second, its behaviour – does it work well, is it intuitive and logical? I’ve been working on improving both of these for Raspbian’s default desktop environment, with the first changes appearing in the December 2014 release.

The Raspbian desktop is an X Window system called LXDE (Lightweight X Desktop Environment). There are many good things about it, the most important from the point of view of the Pi being its low usage of memory, disk space, and processor. It is also very customisable, even without changing a line of code. However, the default appearance of LXDE on Raspbian was a bit dated and unfriendly-looking, and I wanted to improve on that to provide a desktop that looks more like those people used to Mac OS X or Windows might expect.

Starting small
I started out with some minor tweaks to the appearance, I added a new font (Google’s Roboto, which is also used on the Pi website), cleaned up the colours and decluttered by removing seldom-used taskbar and desktop icons. I also chose a different icon theme, in which the icons are less complex graphically and therefore easier to understand. All these are aimed at making the interface nicer to look at and less intimidating when you first see it.

For me, the more interesting part of UI design is the second one I mentioned above: making it intuitive. This is where some basic psychology comes in. User interface design is mostly about applying consistency – you get used to the way something works and if something else works slightly differently, it jars. (As a designer, you can use that to your advantage sometimes, for drawing attention to something, but you don’t want it happening all the time!) Some of the changes are quite subtle. For example, when you move the mouse pointer over the menu bar at the top, everything now highlights in the same colour – previously, hovering the mouse over something on the menu bar had different effects on different items. This helps to add consistency, to make the desktop behave the way you expect.

SIMPLIFYING RASPBIAN
In coming releases, some of the more complex settings – for things such as connecting to Wi-Fi networks – are going to be simplified, all with the intention of making the Pi feel as good to use as a PC or a Mac. Watch this space...

PERSONALISE YOUR PI
This is just the start of the UI work planned for Raspbian. There’s much more in the pipeline, including a new configuration application to enable you to customise the colours, desktop picture and more, so you can make your Pi look the way you want it to. There are going to be other graphic tweaks too, including a new set of icons we are having custom-designed.

The intention is to make it easier for new users to get to grips with Raspbian - some of the names and tooltips, to make it a bit easier to understand what is actually in the menu. The intention with all the changes is to make it easier for new users to get to grips with Raspbian, and for them to feel comfortable with how it works as quickly as possible. Some of these changes may seem awkward to experienced users, if you don’t like them, it’s pretty straightforward to undo most of them.

OUT WITH THE OLD: The original appearance of LXDE on Raspbian prior to December 2014.

IN WITH THE NEW: A cleaner appearance, new colours and font, and a revised menu bar.

RASPBIAN DESKTOP THEN & NOW

UPDATING THE DESKTOP

The intention is to make it easier for new users to get to grips with Raspbian...
MAKERS: THE NEXT GENERATION

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

---

**Zach**

Name: Zachary Igieelman  
Age: 14  
Location: London  
Studying: GCSEs  
Twitter: @ZacharyIgieelman

After teaching himself to code in Visual Basic at just 12 years old, Zach moved on to Objective-C and released several apps on the iOS App Store. Since discovering the Pi, he’s learnt to code in Python and has built his own autonomous robots, enhancing them with a variety of sensors. He also helped 4tronix develop the PiXio. A regular Raspberry Jam attendee, he has learnt to develop the Pi2Go. A regular

A

fter teaching himself to code in Visual Basic at just 12 years old, Zach moved on to Objective-C and released several apps on the iOS App Store. Since discovering the Pi, he’s learnt to code in Python and has built his own autonomous robots, enhancing them with a variety of sensors. He also helped 4tronix develop the PiXio. A regular Raspberry Jam attendee, he has learnt to develop the Pi2Go. A regular

**Amy Mather**

Name: Amy Mather  
Age: 15  
Location: Manchester  
Studying: A Level Computing  
Twitter: @minigirlgeek

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

---

**BIG PROJECT: PIPIANO**

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

---

**BIG PROJECT: PI-LIFE**

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

**What’s so great about the Raspberry Pi?**

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

**Do you have any new Pi-based projects planned?**

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

**By Amy Mather**

What advice would you give to other young coders? Just get involved, find out where all the local events are, and get involved with the community, you’ll definitely learn a lot more from talking with other people who are interested in similar things, rather than just sitting alone at home and doing it by yourself.

---

Find out more at bit.ly/1wsBmci

---

Find out more at bit.ly/1wsBmci

---

Find out more at bit.ly/1wsBmci

---

Find out more at bit.ly/1wsBmci
INTERVIEW

What do you especially like about the Pi?
One thing that I really love is its versatility. I’ve used it in a variety of projects, and seen it in use even more! The Pi is so small that it can be used practically anywhere, which means it can be used in [so many different] projects.

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

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

MAKERS: THE NEXT GENERATION

Why did you decide to set up your own YouTube channel?
I am a very recent convert to the ways of Pi. As a result, I originally found computing fairly hard to get started with, and I built up a considerable amount of knowledge on the subject and thought it would be a great side project to teach people some of the stuff I’ve had so much fun learning. I turned to YouTube as a way of doing this because I found the most easy way to learn something is by watching someone go through something step by step… On 1 September 2012, the Raspberry Pi Guy was born and I have been publishing videos ever since; I am just about to hit the 2 million view barrier on YouTube, something I never imagined!

What’s so great about the Pi?
There is no other product out there that you can so easily use to impress your average person seen? How many single-board computers has the Raspberry Pi Guy ever been run for over two years now and has proven immensely popular, amassing over 32,000 subscribers. “Dedicated to teaching the masses how to make the most of their Raspberry Pi computer”, it provides a plethora of step-by-step video tutorials. These range from basic setups to attaching various add-ons and creating numerous projects – including, of course, robotics. The latest addition is the Raspberry Pi Robots series, which Matt hopes “will engage people in fuller science through the most exciting medium: world-conquering robots.”

Making!

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

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

BIG PROJECT: PI VIDEO WALL

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

Get making!

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

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

BIG PROJECT: THE RASPBERRY PI GUY

Matt’s YouTube channel (youtube.com/user/TheRaspberryPiGuy) has been running for over two years now and has proven immensely popular, amassing over 32,000 subscribers. “Dedicated to teaching the masses how to make the most of their Raspberry Pi computer”, it provides a plethora of step-by-step video tutorials. These range from basic setups to attaching various add-ons and creating numerous projects – including, of course, robotics. The latest addition is the Raspberry Pi Robots series, which Matt hopes “will engage people in fuller science through the most exciting medium: world-conquering robots.”

MAKERS: THE NEXT GENERATION

Why did you decide to set up your own YouTube channel?
I am a very recent convert to the ways of Pi. As a result, I originally found computing fairly hard to get started with, and I built up a considerable amount of knowledge on the subject and thought it would be a great side project to teach people some of the stuff I’d had so much fun learning. I turned to YouTube as a way of doing this because I found the most easy way to learn something is by watching someone go through something step by step… On 1 September 2012, The Raspberry Pi Guy was born and I have been publishing videos ever since; I am just about to hit the 2 million view barrier on YouTube, something I never imagined!

What’s so great about the Pi?
There is no other product out there that you can so easily use to impress your average person seen? How many single-board computers has the Raspberry Pi Guy ever been run for over two years now and has proven immensely popular, amassing over 32,000 subscribers. “Dedicated to teaching the masses how to make the most of their Raspberry Pi computer”, it provides a plethora of step-by-step video tutorials. These range from basic setups to attaching various add-ons and creating numerous projects – including, of course, robotics. The latest addition is the Raspberry Pi Robots series, which Matt hopes “will engage people in fuller science through the most exciting medium: world-conquering robots.”

Get making!

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

- There are lots of free online resources, such as Codecademy, to help you learn to code. Just search ‘Coding courses’.
- Look out for local events and get involved – it’s much more fun than trying to do it alone at home!
- Events such as Maker Faires and Raspberry Jams can give you ideas to try and a place to start exhibiting your projects and make contacts.
- Join a robotics team if you’re interested in making robots and entering them into competitions.
- Even if something seems hard to start with, keep trying. If you tell your mind to it, you’ll achieve it. And you’re never too young to start!
- Have fun! Whether you like something or don’t, don’t waste your time with it.
Solve real-world electronic and engineering problems with your Pi and the help of renowned technology hacker and author, Simon Monk

electronics permeates every aspect of modern life and it’s easy to take such technology for granted without ever stopping to think just how these things work. Small, low-cost computers like the Raspberry Pi make it possible for hobbyists to put their own take on commercially available products, and also invent new gadgets simply for the fun of it.

In this series, we will be exploring the use of the Raspberry Pi GPIO pins to make all kinds of everyday electronic devices, starting with an ultrasonic parking sensor.

Each of these projects will be constructed using a solderless breadboard and readily-available components, so even if you don’t want to develop and install these projects for real, you can prototype them to learn more about engineering and electronic invention.

Why use four resistors?
A Raspberry Pi’s GPIO pins operate at 3.3V, whereas the rangefinder module’s pins operate at 5V. This does not cause a problem when connecting the output of the Raspberry Pi to the input of the rangefinder (for example, a GPIO output on the Raspberry Pi to the Trigger input on the rangefinder) because even though the voltage at the input is a bit low, at 3.3V it will still be high enough to activate the trigger input.

The problem arises when you are going in the opposite direction and the 5V Echo-output of the rangefinder needs to connect to a GPIO input on the Raspberry Pi. Putting 5V on a GPIO pin only designed for a maximum of 3.3V could easily damage the pin. Therefore, we use an arrangement of two resistors to reduce this voltage from 5V to 3.3V, where it will be high enough to register as a high input, but still be well below the maximum of 3.3V.

More advanced readers may prefer to use different combinations of resistor to set the voltage a bit closer to 3.3V, but the advantage of just having the voltage is that all four resistors can be of the same value.

Building your parking sensor
Even if you plan to install the project for real, it’s a good idea to start with the rangefinders plugged directly into the breadboard, with the ultrasonic transducers pointing outwards. This will allow you to experiment with the project and make sure that everything is working as it should be, before you commit to some more permanent setup.

We need to do a voltage level conversion to use 5V rangefinders with 3.3V Raspberry Pi GPIO pins

The HC-SR04 rangefinder module measures distance from a few cm to several meters. Just search for HC-SR04 and remember to order two.

The HDMI display is only needed if you plan to install the project for real in your car or garage; otherwise, you can just use your usual Raspberry Pi monitor. Again, you will find mini HDMI displays at a very reasonable price on eBay. The model we used had a 7-inch display and separate controller board. Look for a display that will operate from 5V if you are going to connect it to your car.

The other parts are probably best bought as an electronics starter kit. The Monk Makes Electronic Starter Kit for Raspberry Pi includes the breadboard and all the parts and wires except the rangefinders. Most starter kits for the Raspberry Pi will include the breadboard, jumper wires and some resistors.

The HC-SR04 rangefinder module needs to connect to a GPIO input on the Raspberry Pi. Why use four resistors?

A Raspberry Pi’s GPIO pins operate at 3.3V, whereas the rangefinder module’s pins operate at 5V. This does not cause a problem when connecting the output of the Raspberry Pi to the input of the rangefinder (for example, a GPIO output on the Raspberry Pi to the Trigger input on the rangefinder) because even though the voltage at the input is a bit low, at 3.3V it will still be high enough to activate the trigger input.

The problem arises when you are going in the opposite direction and the 5V Echo-output of the rangefinder needs to connect to a GPIO input on the Raspberry Pi. Putting 5V on a GPIO pin only designed for a maximum of 3.3V could easily damage the pin. Therefore, we use an arrangement of two resistors to reduce this voltage from 5V to 3.3V, where it will be high enough to register as a high input, but still be well below the maximum of 3.3V.

More advanced readers may prefer to use different combinations of resistor to set the voltage a bit closer to 3.3V, but the advantage of just having the voltage is that all four resistors can be of the same value.

Building your parking sensor
Even if you plan to install the project for real, it’s a good idea to start with the rangefinders plugged directly into the breadboard, with the ultrasonic transducers pointing outwards. This will allow you to experiment with the project and make sure that everything is working as it should be, before you commit to some more permanent setup.
The Python code for this project is very well commented on GitHub

The Python code for this project is very well commented on GitHub (bit.ly/2DUtY2R), so you’ll probably find it handy to have the code up in an editor while we go through it.

The program starts by importing the Python libraries that will be used, and some constants for the GPIO pins. If you wanted to swap things around you could just change the GPIO pins. So, if you wanted to swap things around you could just change the GPIO pins. So, if you wanted to swap things around you could just change the GPIO pins. So, if you wanted to swap things around you could just change the GPIO pins.

The rest of the code is concerned with the user interface for the project. The function colour_for_distance returns a colour to use when drawing the rectangle for that sensor, depending on how large the distance detected is.

The next few lines initialise the Pygame window and define a font to use for the distance readout. While Pygame is designed primarily to make games, it’s excellent for any project that uses graphics, like this one. You will likely want to alter the width and height variables to match the resolution of your display.

The while statement starts the main loop. The program will keep looking around the instructions inside, from while until the end of the file, until the program window is closed – this kind of loop is known as an infinite loop.

Each time around the loop the Pygame events are checked, and if the Pygame window has been closed (by clicking the little cross in the corner of the window), the GPIO pins are set to be a safe input mode using the GPIO.cleanup function and the program exits.

Most of the time the window will not have been closed, so the remainder of the loop will measure the two distances from the rangefinders and then draw rectangles on the screen, using the distance readings to set the height of the rectangle. The height of each rectangle is the distance in cm multiplied by 5 pixels. Finally, there is a delay of half a second to stop the distance figures updating too fast to read clearly.

Using your parking sensor

Although you can extend the leads to the ultrasonic rangefinder by perhaps a metre or so, any longer than that and you are likely to have problems with the signal. So, if you are installing this project for real in a vehicle, it may be better to sit the Raspberry Pi fairly near the sensors and use a longer HDMI lead to connect the Raspberry Pi to the display.

If you are installing this project for real, then you will probably want to make the program start automatically. That way, you don’t even need to have a keyboard and mouse attached to the Raspberry Pi. You can find links on how to do this on the Raspberry Pi Forum (bit.ly/DaU7R).

The ultrasonic rangefinders are great little devices. You can take the range-finding part of the program for this project and use it in lots of other projects. You could, for instance, use it to just make a distance meter, perhaps displaying the distance in inches or cm. You could also use it to create a theremin-like musical instrument that changes the pitch of the note, depending on the distance of your hand from the rangefinder.

```python
# You could use the affordable (H3H4) screen in your garage

import RPi.GPIO as GPIO
import time, sys, pygame

GPIO.setmode(GPIO.BCM)
GPIO.setup(trigger_pin_left, GPIO.OUT)
GPIO.setup(echo_pin_left, GPIO.IN)
GPIO.setup(trigger_pin_right, GPIO.OUT)
GPIO.setup(echo_pin_right, GPIO.IN)

def send_trigger_pulse():
    GPIO.output(trigger_pin, True)
    time.sleep(0.00001)
    GPIO.output(trigger_pin, False)

def wait_for_echo(echo_pin, value, timeout):
    count = timeout
    while GPIO.input(echo_pin) != value and count > 0:
        count -= 1

def get_distance(trigger_pin, echo_pin):
    send_trigger_pulse()
    wait_for_echo(echo_pin, GPIO.HIGH, 10000)
    start = time.time()
    wait_for_echo(echo_pin, False, 10000)
    finish = time.time()
    pulse_len = finish - start
    distance_cm = pulse_len / 0.000058
    return int(distance_cm)

def colour_for_distance(distance):
    if distance < 30:
        return red
    if distance < 150:
        return orange
    return green

pygame.init() size = width, height = 800, 600 # the variables alter window size offset = width / 8
screen = pygame.display.set_mode(size) myfont = pygame.font.SysFont("monospace", 15)
while True: # the main loop starts here
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            GPIO.cleanup() # set GPIO pins to be inputs sys.exit() # quit the program entirely

    left_distance = get_distance(trigger_pin_left, echo_pin_left)
    right_distance = get_distance(trigger_pin_right, echo_pin_right)

    pygame.draw.circle(screen, colour_for_distance(left_distance), (width / 2 + offset, 4, width / 4, right_distance))

    left_label = myfont.render(str(left_distance) + ' cm', 1, black)
    screen.blit(left_label, (offset + 10, height / 2))

    right_label = myfont.render(str(right_distance) + ' cm', 1, black)
    screen.blit(right_label, (offset - width / 2, height / 2))

    pygame.display.update()
    time.sleep(0.1)
```
Minecraft: Pi Coding Tips

If you've completed the Minecraft Pi learning resources at raspberrypi.org, check out these pro tips and mini programs to learn more about the coding in Minecraft…

**BUILD A HOUSE**

The quickest way to make a house in Minecraft: Pi Edition is to use code and the API. By programming a house rather than building it by hand, it can be any size you want – 10 blocks across or 100!

Create a simple program which will use the `setBlocks()` function, once to create a cube 10×10×10 of wood (17) and then again to create a cube of air (0) 9×9×9 inside the wooden cube.

```python
from mcpi import minecraft
mc = minecraft.Minecraft.create()
pos = mc.player.getTilePos()
mc.setBlocks(pos.x + 0, pos.y + 0, pos.z + 0, pos.x + 10, pos.y + 10, pos.z + 10, 17)
mc.setBlocks(pos.x + 1, pos.y + 1, pos.z + 1, pos.x + 9, pos.y + 9, pos.z + 9, 0)
```

You can then use `setBlocks()` again to create an entrance by building another block of air (0).

```python
mc.setBlocks(pos.x + 4, pos.y, pos.z, pos.x + 6, pos.y + 3, pos.z, 0)
```

The limits of coding a house are endless – why not add a stone roof, a wool floor, some torches to the outside?

**USE GRAVITY-FFECTED BLOCKS**

Sand and gravel block types in Minecraft are affected by gravity and will fall down if the block below is air.

The same gravity effect occurs if a block is placed in the world using the API. So if you were to create a block of gravel (13) 25 blocks above the player, it would fall on the player's head. In a new program, type:

```python
from mcpi import minecraft
c = mc.player.getTilePos()
mc.setBlock(c.x, c.y + 25, c.z, 13)
```

Using the gravity effect of blocks is a great way to add something new to your Minecraft programs.
TIPS AND TRICKS

CHANGE THE CAMERA

Bored of always following Steve around? You can alter the position of the ‘camera’ in Minecraft to change how you see the world.

You can change the camera to follow Steve while looking directly down at him, or to look down on the world from any coordinate in Minecraft. The `camera.setFollow()` function will change your view so you are looking down at Steve. In a new script:

```python
from mcpi import minecraft
mc = minecraft.Minecraft.create()
mc.camera.setFollow()
```

To change the camera to look down on any position, you use the `camera.setFixed()` function before using `camera.setPos()` to change the position of the camera. If you wanted to set the camera 25 blocks above the spawn position, you would use:

```python
mc.camera.setFixed()
mc.camera.setPos(0,25,0)
```

To set the camera back to normal, you would use the `camera.setNormal()` function.

```python
mc.camera.setNormal()
```

Using the camera functions, you could hide a diamond block (57) in the world, then tease the player by changing the camera to show them where it is before challenging them to find it. Try this in a new program:

```python
from mcpi import minecraft
from time import sleep
mc = minecraft.Minecraft.create()
mc.postToChat("Here is the diamond block I have hidden.")
mc.setBlock(100,25,100,57)
mc.camera.setFixed()
mc.camera.setPos(100,30,100)
sleep(10)
mc.postToChat("Go find it!")
m.postToChat(y)
m.camera.setNormal()
```

LEARN THE HEIGHT OF THE WORLD

If you want to code structures to always be ‘on top’ of the land, you need to know how high the world is — or, put another way, how far the air comes down!

In Minecraft, the height is the Y coordinate, while X and Z are the horizontal dimensions — if you pass X and Z coordinates to the API function `getHeight()`, it will return the Y coordinate. In a new program:

```python
from mcpi import minecraft
mc = minecraft.Minecraft.create()
y = mc.getHeight(0,0)
m.postToChat("Height of the world at spawn is")
m.postToChat(y)
```

If you know the height of the world, you can cover the top layer of the world in a different type of block by looping through the X and Z coordinates. What about covering the world in snow?

You can do this in a new script by looping through the coordinates around your player, finding the height for that position and setting the block to snow (78).

```python
from mcpi import minecraft
mc = minecraft.Minecraft.create()
pos = mc.player.getTilePos()
for x in range(pos.x, pos.x + 10):
    for z in range(pos.z, pos.z + 10):
        y = mc.getHeight(x,z)
        mc.setBlock(x,y,z,78)
```

What other types of block could you cover the world in? Lava perhaps?!
**SHOOT IN SLOW-MOTION WITH THE CAMERA MODULE**

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

---

**STEP-01**

Connect the camera module

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

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

---

**STEP-02**

Configure the camera module

You need to make sure the camera module is enabled, so connect your Pi to a screen and keyboard, turn it on and log in, then type `raspi-config`. Open a terminal window and type `sudo raspi-config` to enter the configuration menu. Use the arrow keys, scroll down the list that appears and select ‘Enable Camera’ using the right arrow key. In the next menu, select ‘Enable’ with the right arrow key to turn on the camera module, then hit Return. You should be prompted to reboot after this, otherwise type `sudo reboot` to restart your Pi.

---

**STEP-03**

Install a video converter

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

---

**STEP-04**

Test the camera

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

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

---

**STEP-05**

Create a Python script

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

---

**STEP-06**

Run the Script

To run the script, simply open a terminal window, type `cd` and hit Return to ensure you’re in the Home folder, then type `sudo python slowmotion.py`. You will see the status of the script printed in your terminal window as it carries out its commands, and the camera module’s LED will light up while it’s recording.

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

---

**RECORDING TIPS**

- **Prepare**
  - Have good lighting and a steady mount for your camera module. Also, charge your portable power options if you’re shooting outside. Finally, don’t forget to use an SD card with enough storage space.

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

---

**Code Language > PYTHON**

```python
import os
import time

print("Starting program")

time.sleep(2)

# Look for the video files
os.system("ls")

print("Finding video files")

time.sleep(2)

# Add the video files to the list
os.system("ls -a")

print("Adding video files")

time.sleep(2)

# Convert the raw video files to playable mp4 files
os.system("for f in *.h264; do ffmpeg -i "$f" -c:v libx264 -c:a aac -b:a 128k -movflags +faststart "$f.mp4"; done")

print("Converting to mp4")

time.sleep(2)

print("Closing program")

time.sleep(2)
```

---

**Average Man vs Raspberry Pi**

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

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

>STEP-01
Pick a resistor for your LED

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

\[ R = \frac{V_{LED} \times V_{power}}{0.02} \]

We’ll be using a 220 ohm resistor, but you may want to spruce up your project before you use it on a loved one. Pretty lanterns are available very cheaply from most department stores, just make sure you select one that obscures the view of the interior. If the candle isn’t big enough, though, it might not reach the garage level of the Pi and breadboard, solder the resistor to the LED and hide the Pi behind it.

>STEP-02
Setup the breadboard

Unplug your Pi and follow the breadboard illustration setup. Make sure you use the same GPi pin we have, as only a couple are capable of pulse-width modulation (on the B+). We’re using GPi number 18 for PWM, which is described as P8M in an infinite loop within the function, which handles the all-important cleanup of the GPIO library when CTRL+C is pressed by the user.

>STEP-04
Test your creation

Exit your editor and run the code by typing sudo python2 candle.py into your terminal (you need root privileges to access the GPi pins). Now that you’ve tested it, you can exit with CTRL+C and we’ll make it run at boot. This way, the Pi can run headless and not need any user interaction.

The candlelight project is surprisingly effective, but presentation is key in matters of the heart, so you may want to spruce up your project before you use it on a loved one. Pretty lanterns are available very cheaply from most department stores, just make sure you select one that obscures the view of the interior. If the candle isn’t big enough, though, it might not reach the garage level of the Pi and breadboard, solder the resistor to the LED and hide the Pi behind it.

For the amperage the script will go to the background and let the boot process continue. Notice how sudo isn’t required because rc.local is executed as root. Reboot the Pi with sudo reboot to verify that it works.

>STEP-05
Packaging it up

Now that the script is started when the Pi boots, you could package it up into a nice container using a portable phone charger as a power supply. Arts and crafts are out of the scope of this tutorial, but there are plenty of candle holders that can be fashioned out of paper if you search the internet. Paper is ideal, especially with lots of holes in, since the LED probably isn’t throwing out much light.

>STEP-06
Presentation, presentation, presentation

The candlelight project is surprisingly effective, but presentation is key in matters of the heart, so you may want to spruce up your project before you use it on a loved one. Pretty lanterns are available very cheaply from most department stores, just make sure you select one that obscures the view of the interior. If the candle isn’t big enough, though, it might not reach the garage level of the Pi and breadboard, solder the resistor to the LED and hide the Pi behind it.

Candle.py

import GPi, GPIO as GPi
import time
import random

# Set the PWM output we are using for the LED
LED = 18

def setup():
    # GPIO setup
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(LED, GPIO.OUT)
    # Set the LED pin as an output
    GPIO.setup(LED, GPIO.OUT)
    # Start PWM on the LED pin at 200Hz with a
    # 100% duty cycle. At lower frequencies the LED
    # would flicker even when we wanted it to solidly
    # pulsate (PWM: LED, 200Hz)
    pwm = GPIO.PWM(LED, 200)
    pwm.start(0)

    # Set the brightness of 100%
    pwm.start(100)

    def set_brightness(new_brightness):
        # Sets brightness of the LED by changing duty cycle
        pwm.ChangeDutyCycle(new_brightness)

    def setup():
        # Set the hardware
        setup()

    def loop():
        try:
            while True:
                # If the flicker function is called, it
                # starts the flickering
                if flicker():
                    # The wrapper around the flicker function makes sure the
                    # GPIO hardware is cleaned up when the user presses CTRL-C
                    GPIO.cleanup()

                    # The loop is now finished
                    pass

                finally:
                    GPIO.cleanup()

        # Setup the pi
        setup()

    # Start the flickering
    loop()

    # Start the flickering
    loop()
WATER DROPLET PHOTOGRAPHY

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

DAVID HUNT
David has been making projects for the Raspberry Pi since the early days. These include a Camera Controller, Time-Lapse Rail, Focus Stacker, and even a Battery Activated Doggy Door Opener. Oh, and let’s not forget the PiPhone! DavidHunt.ie

You’ll Need
- Solenoid Valve
- x1 Match's diode
- x1 NPN Transistor
- x2 Resistor
- x1 Power supply
- x1 Shutter release cable
- x1 Wiring Pi

STEP BY STEP

<table>
<thead>
<tr>
<th>STEP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>The solenoid driver</td>
</tr>
<tr>
<td>02</td>
<td>The camera shutter driver</td>
</tr>
<tr>
<td>03</td>
<td>Setting up the solenoid</td>
</tr>
<tr>
<td>04</td>
<td>Trigger the camera</td>
</tr>
<tr>
<td>05</td>
<td>Get the lighting right</td>
</tr>
<tr>
<td>06</td>
<td>Adjust the camera settings</td>
</tr>
</tbody>
</table>

STEP-01 The solenoid driver

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

STEP-02 The camera shutter driver

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

STEP-03 Setting up the solenoid

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

STEP-04 Trigger the camera

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

STEP-05 Get the lighting right

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

STEP-06 Adjust the camera settings

You should be shooting on manual setting, with a shutter speed as high as your camera will allow for flash. For Canons this is about 1/1000th of a second, and maybe 1/250th of a second for Nikons. Use ISO 100-400 and then adjust your aperture till you get a decent exposure. You can then tweak the flash power down to get shorter flash durations, which will tend to freeze the motion of the liquid more. Open up the aperture more if you need to, but be aware that your depth-of-field will be reduced.

Code Language

**>PYTHON**

```python
from time import sleep
import wiringpi2
import wiringpi

solenoidpin = 18
shutterpin = 17

# Set up the GPIO pins
gpio = wiringpi2.GPIO(wiringpi2.GPIO.WPI_MODE_GPIO)
shutterpin = 17
solenoidpin = 18
gpio.pinMode(shutterpin, 1)
gpio.pinMode(solenoidpin, 1)

gpio.digitalWrite(solenoidpin, gpio.LOW)
sleep(0.06)
gpio.digitalWrite(solenoidpin, gpio.HIGH)
```

Above An example of the type of Image that can be achieved
Richard Smedley presents a cut-out-and-keep guide to using the command line on the Raspberry Pi. In part one, we take a look around and discover things aren’t as strange as they might appear...

**File path**

You can list files and folders anywhere in your system (or other connected systems) by providing the path as an argument to your command. The path is the folder hierarchy – on a Windows computer, in a graphical file browser, it starts with “My Computer”; on your Pi it starts at ~, pronounced ‘root’ when used on its own as the root of your filesystem. Try it: `ls ~` – again we get terseness; names like bin, which is short for binary, and is the directory where many system programs are kept. If you’re into binary, you’ll know that’s not really a place for ordinary files. We’re already home – to check your location, you’ll see that ‘pi?’ is you; you’re logged in as user pi. If you’ve changed your login name, or created extra users, they’ll all be listed there too – every user gets her own directory; yours is the /home/pi folder we found ourselves in earlier. Before, with python_games, we used the relative path (the absolute path would be /home/pi/python_games); because we’re already home – to check your location, type `pwd` (present working directory).

**Time for change**

Now enough looking, let’s start moving:

**Look around**

If you’re used to looking at files and folders in a file manager, try to clear your mind of the icons and concentrate on the names. Type `ls` and press Return (see the ‘Press Return’ boxout on the next page). On a fresh Raspbian install you’ll just see two directories: python_games and Desktop. Type `ls python_games` (also see the Lazy Completion boxout) and you should see a listing like Fig 1 on the next page.

Commands like `ls` are not cryptic (at least not intentionally) but they are terse, dating back to a time when the connection to the computer was over a 110 baud serial line, from an ASR 33 teletype terminal. And if you think it’s strange to be defined by 50-year-old technology, just remember that your QWERTY keyboard layout was both to stop mechanical typewriter keys jamming, and to enable salespeople to quickly type ‘typewriter’ using the top row!

**Try the command line**

You can list the hidden files without the dot (.) at the beginning of the name, by chaining switches together: `ls ~ . ~` will show you all the files and directories you have.

Commands are not cryptic (at least not intentionally), but they are terse...
In this new ten-part series, **Sean M Tracey** teaches us how to make a game on our Raspberry Pi from the ground up. In part one we learn the basics...

Hello, and welcome to this first part of ten tutorials in which we’ll learn to make games on the Raspberry Pi with Pygame. Over the course of this series, we’ll look at drawing, animation, keyboard and mouse controls, sound, physics, and maybe even the installation of a kitchen sink. Each part will add to our knowledge of Raspberry Pi game development, allowing us to understand the games we play and to create almost anything our imaginations can come up with.

This series isn’t for absolute programming beginners, but it’s not far from it – we’re going to guess that you’ve written some simple Python (or similar) programs in the past, and are able to do things like creating files and get around your Pi’s filesystem without too much difficulty. If you’re not set up your Pi and are a little lost on how to go about it, there are lots of easy-to-follow guides on the web which will help bring you up to speed. You could point your web browser to raspberrypi.org/resources to get started.

In this first part of the series, we’re going to look at drawing and colouring various shapes in a window. This isn’t quite Grand Theft Auto V, granted, but drawing shapes is the first step in building just about anything.

To start off, open your favourite text editor, create a new file, insert the following code into it and save it as hello.py:

```python
import pygame

pygame.init()

window = pygame.display.set_mode((500, 400))

while True:
    pygame.display.update()
```

Let’s run that code and see what it does. In your terminal window, enter `python hello.py`. If all has gone well, a new window will have opened showing you a red square on a black background in the top-left corner of the window. We’ve just created our first Pygame program – let’s walk through it.

### Understanding hello.py

The first two lines of our first program are very simple – all we’ve done is tell Python that we want to use Pygame. `import pygame` loads all of the Pygame code into our script, so we don’t have to write all of that code ourselves. Pygame is designed to make the creation of games and interactive software easy. `pygame.init()` tells Pygame that we’re ready to start using it.

Let’s look at the third line of code:

```python
window = pygame.display.set_mode((500, 400))
```

`window` is the parameter we’re going to use to tell our Pygame program about how it should look when it runs: each parameter affects the application window’s shape and size – note that width always comes before height. `window` is also the parameter that we’ll use to tell other lines of code the surface on which they should draw shapes and set colours. With `window`, we’re calling the `set_mode` function of Pygame’s display module – the latter is responsible for how the game window and surface (an informal term for the pixels we’ll be manipulating) behaves. We’re passing a tuple (which we can think of as a special list of things – in this case numbers) to `set_mode()` to tell it how big we want our game window to be.

In this case, the application window is 500 pixels wide by 400 pixels tall. If you pass numbers that are bigger, the game window will be bigger; if we pass numbers that are smaller, the game window will be smaller.

The next few lines are where we make our program draw shapes on that window. When programs run, they execute their code, and when they’re finished, they close themselves. That’s fine unless, of course, you want your program to be interactive or draw/animate shapes over time (exactly what we need from a game). So, in order to keep our program from exiting, we make a `while` loop and put all our code inside. The `while` loop will never finish because `True` is always `True`, so we can keep running our program and drawing our shapes for as long as we like.

The first thing we do in our `while` loop is draw a rectangle. A rectangle is the simplest shape that we can draw in Pygame:

```python
pygame.draw.rect(window, (255, 0, 0), (300, 200, 50, 50))
```

This isn’t quite Grand Theft Auto V, granted, but drawing shapes is the first step

```python
pygame.draw.line(window, (255, 255, 255), (25, 75), (50, 50), 1)
pygame.draw.line(window, (255, 255, 255), (75, 75), (25, 75), 1)
pygame.draw.line(window, (255, 255, 255), (50, 50), (75, 75), 1)
```

### Pygame

Pygame is installed on Raspbian by default. For documentation detailing all its features, visit [Pygame.org](http://pygame.org/docs).

Below here we can see how each variable in a window affects the application window’s shape and size. Width always comes before height.
WALKTHROUGH

of colour should be in that shape. We told our rectangle that it should be the colour (255, 0, 0), which is pure red. If we had told it to be (255, 0, 255), it would have been a bright purple, because it’s being drawn with the maximum amount of red and the maximum amount of blue. If we had told our rectangle to be coloured (0, 0, 255), it would be a dark grey, because all of the colours would be equal.

After we’ve passed through a colour for our rectangle to be, we next tell it where it should go and how big it should be. We do this by passing a tuple of four numbers. The first number is an X coordinate – that is, how far from the left side of the window the left edge of our shape should be. The second number is a Y coordinate, this tells our shape how far from the top of our window the top edge of our shape should be. The third number is the width our rectangle should be, and the fourth number is how tall our rectangle should be. If we wanted our rectangle to be 50 pixels from the left of the window, 100 pixels from the top of our window, 20 pixels wide and 80 pixels tall, we’d pass (50, 100, 20, 80) to pygame.draw.rect().

The order never changes. If you tell Pygame how big you want the rectangle to be when it’s expecting a colour or vice versa, the program may crash, so take your time.

Our last line in hello.py is nice and simple: it tells Pygame that we’re done drawing shapes for the moment and that it can now refresh the window. This saves our Pi having to draw and redraw the screen for every shape that we’ve created, instead, it can get them all drawn in one go.

Adding more shapes

We’ve successfully drawn one shape, so let’s draw a few more. We’ll draw some squares around the screen and mess around with their properties a little bit. There’s no need to create a new file, so we’ll stick with hello.py for now. Edit the while loop so it’s the same as the following:

```python
while True:
    pygame.display.update()
```

Now we should have three squares: red, blue, and green - nice and simple, but those squares are placed right next to each other. What would happen if they were to overlap? Let’s find out. Change your code once more to the following:

```python
while True:
    pygame.display.update()

    # Not filled
    pygame.draw.rect(window, (255,0,0), (200, 100, 50, 50))
    pygame.draw.rect(window, (255,0,0), (40, 0, 50, 50))
    pygame.draw.rect(window, (0,0,255), (80, 0, 50, 50))
    pygame.display.update()
```

This time we get two rectangles and a square, but that’s not what we asked for! So what’s gone wrong? When we execute our code, it works through line-by-line what it has to do and where it has to put it. If one item is drawn and then another is over or on top of part of it, then we can no longer see what’s beneath that second shape. The pixels of the shape drawn first are lost when we overlap it with another shape. If we change the order of our code, we can see this effect in action. Cut and paste the code for the second square so that it becomes the third square drawn, like so:

```python
while True:
    pygame.display.update()

    # Not filled
    pygame.draw.rect(window, (255,0,0), (200, 100, 50, 50))
    pygame.draw.rect(window, (40, 0, 50, 50))
    pygame.draw.rect(window, (0,0,255), (80, 0, 50, 50))

    pygame.display.update()
```

Drawing circles

Drawing a circle is much like drawing a square except instead of passing a width and a height, we pass a radius and a point, around which we draw our circle. If we wanted to draw a yellow circle in the middle of our window with a diameter of 40 pixels, we would use the following code to replace the while loop in hello.py:

```python
pygame.draw.circle(window, (255,255,0), (200, 200), 20, 0)
```

What about ellipses? They are a slightly strange cross between drawing rectangles and circles. It’s the same as drawing a rectangle: we pass an X coordinate, a Y coordinate, a width, and a height, but we end up with a circle(s) shape. Let’s draw an ellipse or two...
A new path

So that’s rectangles, squares and circles, but what if we want to draw a triangle, a pentagon, a hexagon or an octagon? Are there functions for every single kind of shape? Well, no, but what we do have are paths. Paths let us draw irregular shapes by defining points in space, then joining them up with lines and filling in the space we’ve created.

This is a little more complex, so it’s time to move on from our old friend hello.py. Create a new file, call it paths.py, and save it with the following inside:

```python
import pygame

pygame.init()

# Rectangles, squares and circles
while True:
    pygame.draw.ellipse(window, (255, 255, 0), (250, 200, 50, 50))
    pygame.display.update()

# Circles
while True:
    pygame.draw.circle(window, (255, 255, 0), (250, 200), 20, 1)
    pygame.display.update()

# Lines
while True:
    pygame.draw.ellipse(window, (0, 0, 255), (100, 150, 80, 40))
    pygame.draw.ellipse(window, (0, 0, 255), (100, 190, 60, 30))
    pygame.display.update()
```

Each path is made of joined-together lines, but before we start joining things up, let’s draw a couple of standalone lines to familiarise ourselves with them. We can do this with `pygame.draw.line()`. Edit paths.py so your while loop is like the following:

```python
while True:
    pygame.draw.line(window, (255, 255, 255), (0, 0), (500, 400), 1)
    pygame.display.update()
```

Just as before, run your code. You should now see three ellipses—one red, one green, and one blue, each a different size. If you wanted to visualise how these shapes were generated, you could draw rectangles using the same coordinates as you used to draw an ellipse and it would fit perfectly inside that box. Guess what? That means you can also make circles by using `pygame.draw.ellipse` if the width and height parameters are the same.

This is our bare-bones Pygame app again, in fact. If you want to make a copy of this for experimentation without breaking anything, now would be a good time to do so.

There! We have a friendly, white triangle with a 1px edge. But when we look at that code, it looks like a lot, doesn’t it? So many things, like the colour or the width, are being written again and again just for the sake of it.

There must be a better way, and indeed there is! All we need is `pygame.draw.lines()`.

```python
while pygame.draw.
    lines() lets us draw a line between two points, `pygame.draw.lines()` enables us to draw a sequence of lines between numerous points. Each XY-coordinate point will be joined up to the next XY-coordinate point, which will be joined up to the next XY-coordinate point, and so on.
```

After running the code you’ll see that it’s exactly the same, except we did it in one line of code instead of three. You might have noticed that we didn’t actually close the triangle – Pygame did it for us. Just before we pass the points for our shape to be drawn from, we can pass either a `True` or a `False` value that will let Pygame know that we want it to close our shapes for us. Change it to `False` and we get the first two lines of our shape, but not the third.

What if we want a more complex shape? We simply add more points like so:

```python
while True:
    pygame.draw.lines((WHERE TO DRAW, COLOUR, CLOSE THE SHAPE FOR US?, THE POINTS TO DRAW, LINE WIDTH))
    pygame.display.update()
```

```
There you have it: a pentagon. If you want a hexagon or even a triacontagon, just add more points. Give it a go.
```

So that’s how you draw shapes, lines and paths in Pygame. Already we know enough to make programs that could be used to display pixel art to our friends and family.

### Tutorial

**WALKTHROUGH**

**TUPLE**

A tuple is like a list, but unlike a standard list, a tuple’s contents can’t be changed. It’s immutable.

**RASPBERRY PI**

This is where the spin, which is where the games begin.

**MAKE GAMES WITH PYTHON**

In part two, things will get a little more complicated—we’ll be animating all of the shapes we’ve learnt to draw so far. They’ll bounce, stretch and spin, which is where the games begin...

Next Month

March 2015

Rasberry Pi 204 | Made with <3 by Sarah Todd

Pygame.org/docs
emulation is a wonderful thing. Much like an old police box or a 1982 DeLorean, it can take us back in time to when we were driving in a Grand Prix, saving the world from the brink of destruction or simply creating lines out of falling blocks while waiting for the bus. We have a rich heritage of computer gaming, and with the help of the Raspberry Pi 1 or 2, it’s beautifully preserved through the emulation software we’re looking at today. It’s not all rose-tinted glasses, though. Emulation provides a virtual museum of computing in your home, and will most likely be used to keep the knowledge of these times alive, long after the hardware has gone to silicon heaven.

Legality aside, emulation provides a wonderful thing. Much like an old police box or a 1982 DeLorean, it can take us back in time to when we were driving in a Grand Prix, saving the world from the brink of destruction or simply creating lines out of falling blocks while waiting for the bus. We have a rich heritage of computer gaming, and with the help of the Raspberry Pi 1 or 2, it’s beautifully preserved through the emulation software we’re looking at today.

It’s not all rose-tinted glasses, though. Emulation does have its issues, and chief among them is the legal grey area surrounding ROMs. A ROM is a dump of the game code contained in a file so ROMs represent your favourite games from yesteryear when loaded into your chosen emulator.

Of course, that game code is copyrighted software, created by companies with the specific goal of making money. So how can you play retro game ROMs legally? This is the grey area we face. Obtaining retro-gaming ROMs is your own responsibility, so we’ll leave you to make your own investigations online.

One of the few issues we encountered otherwise was related to audio output, which forced itself via the HDMI port no matter how often we changed the settings or per emulator, enabling you to create your own investigations online.

With the ROMs uploaded, the relevant emulators will be activated and loading it will trigger PiPlay to scrape the web for thumbnail images for your collection, creating a slick-looking library. Configuring your joystick can be done as a default setting or per emulator, enabling you to create your perfect configuration. We used a USB SNES pad, which worked flawlessly. We also tried the Xbox 360 controller for Mac OS X.

Final word
PiPlay offers a good collection of emulators that are easily configured to work with many games and controllers. However, there are definite issues when running some games, which are down to PiPlay.
The RetroPie project is another emulation distribution that is based on Raspbian, and it provides a plethora of emulators. Raspbian may be the operating system, but RetroPie represents the glue that binds the OS to a beautifully simple interface called Emulation Station, a third-party themable front end for emulation projects.

Copying RetroPie to an SD card is handled in the same manner as PiPlay, by transferring the image to the card using dd or a GUI application. On first boot, it asks the user to insert a controller to configure. We found this a little confusing, as the configuration only refers to the user interface and not the emulators contained therein.

To configure the controller for the emulators, you have to drop into the terminal and hack together a config script that covers all of the emulators, which is far from ideal for those new to emulation.

Since RetroPie emulates the same consoles as PiPlay, we tested the same SNES, Mega Drive and PlayStation emulators. First, the SNES with Super Mario World and then Street Fighter 2 – both worked very well and were fluid to play. RetroPie was the only emulator to successfully load and play Star Fox. A quirk that we found with the SNES emulator was that only emulator to successfully load and play Sonic 2 preferred games to be unzipped before play.

A serious quirk that we found with the SNES emulator was that it couldn’t solve.

RetroPie boasts a particularly slick user experience and comes with a ROM scraper tool, which prettifies your library of ROMs by downloading thumbnails and information from the web. You manage your ROMs via the Raspbian desktop, and RetroPie includes a great script that detects when a USB stick is inserted into the Pi. When detected, RetroPie creates a directory structure for ROM files that mirrors what is installed on the Raspberry Pi. As all you need to do is put USB stick into your desktop PC and copy the ROMs onto it (making sure to put them in the right folder). Now, when you put that memory stick back in your Raspberry Pi, RetroPie automatically puts them in the right directory, which is both rather clever and exceptionally useful.

Ultimately, RetroPie is a highly refined product, but one that’s not aimed at the newcomer. If you’re an enthusiast who wants to make their own cabinet or home entertainment solution, it’s easily the best choice. It’s a powerful, beautiful piece of software.

Installation of Raspicade is equally as easy as the other two emulation offerings on test, and it took a grand total of ten minutes to get running from a standing start.

Raspicade is a little different to the others on test, inasmuch as it comes with a configuration script that runs every time the Raspberry Pi boots. Among other things, the script configures the audio output, enabling you to choose between the 3.5mm audio jack and the HDMI port, something lacking from the other solutions. The script also handles your IP address and, in another interesting twist, allows you to choose between three user interfaces: a simple Raspicade bespoke interface, Emulation Station 1 (which is an older interface but very light on resources) and Emulation Station 2, which comes as standard on RetroPie.

Given the latter fact, we reviewed using Emulation Station 2 to ensure a fair test.

First of all, we configured the joystick to work with Emulation Station, which seemed to go well, but found that there was no way to create a default configuration, so we needed to repeat the process for each emulator we tested. Elsewhere, we found two instances of ‘Sega Genesis’ in the interface, since the software used both PicoDrive and dgen Mega Drive emulators, we decided to test the latter, which is supported in all three options on test today.

As before, we tested all three Sonic games and Streets Of Rage 3. All played fluidly, with no issues to speak of. Next, the SNES emulator handled Super Mario World and Street Fighter 2 with ease, but Star Fox refused to play along.

While we weren’t terribly surprised at that, we were disappointed that Raspicade doesn’t offer a PlayStation emulator, so we fired up the NES emulator and tried a built-in game called Solar Wars, which – unsurprisingly – worked really well.

Finally, we look at Raspicade, which is designed to play well with the specific needs of arcade cabinets and interface with arcade-quality joystick and buttons.
SKYWRITER HAT

In our first new-look review, Les Pounder reaches for the sky to uncover the secrets behind the latest gadget from the ‘Pirates of Sheffield’

Control your Pi with a flick of the wrist

Pimoroni

How we interact with technology is constantly evolving – from the early days of clunky keyboards to the sensors we use to capture location and orientation automatically. What remains the most basic interaction is touch and, by extension, gestures. Gesture control can be achieved using technologies such as Microsoft’s Kinect or OpenCV, but these are quite resource-intensive solutions. So a cheap, simple and resource-light solution is needed; step forward Skywriter.

Near-field sensing

The Skywriter is the latest board from Pimoroni, the Sheffield-based maker of a number of quality add-ons for the Raspberry Pi. Designed for the Pi Model A+ and B+, the Skywriter is a gesture controller built around the MGC330, a 3D gesture recognition and tracking controller chip. It uses near-field sensing to locate the position of your hand in the air at a range of 5cm. The board can also detect touch input to a number of positions on its surface.

Physically, the Skywriter HAT measures 64mm wide by 56mm tall with a depth of 5mm. On the Raspberry Pi A+, it fits neatly over the top, and with a B+ it fits snugly in the space between the USB ports and the display slot (DSI).

Code control

Along with a neat hardware solution, the team have invested a great deal of time producing a Python API (application programming interface) for both Python 3 and 2.7, installable via the Pip package manager. Examples of how to use this board can be found in Skywriter’s GitHub repository (github.com/pimoroni/skywriter-hat). We found the contents of test.py in the python/examples folder very interesting, since it puts out lots of interesting debugging data, as well as helping confirm it’s working properly.

How can you use Skywriter HAT in your next project? Instantly, gesture-controlled gaming comes to mind. Using Skywriter as a gesture controller for a game of Pac-Man perhaps, controlling your Big Trak with just a flick of your hand or launching a rocket with a tap. We look forward to waving our hands around like Tom Cruise in Minority Report to take control of our next project, using nothing more than a gesture and some Python code.

Last word

The Skywriter HAT is a small and well-constructed board that sits neatly on top of your Pi (as any HAT should). It’s a really capable and versatile board that adds a novel form of input to any project and the easily understood Python API will benefit coders of all abilities.

£16/$21

shop.pimoroni.com

Easy configuration

Once it’s on, it can be configured via the browser of any device connected to your router. One of the tools enabling anyone to stay anonymous online is Tor (The Onion Router), which provides a series of relays that bounce your connection around the world via an encrypted network. Tor is configured per machine, so the Anonymebox’s differing approach is more convenient.

Tor made easy

The Pi-powered Anonymebox connects to your home network so anyone can connect securely with any computer. It comes with an Ethernet cable to connect to your router and then insert the Wi-Fi dongle before powering up.

An easy solution to anonymise you online

Les Pounder

Since George Orwell wrote 1984, the fiction contained therein has become fact in our society. Big Brother is indeed watching you. Your ISP can throttle your speed and governments can get hold of your browser history. One of the tools enabling anyone to stay anonymous online is Tor (The Onion Router), which provides a series of relays that bounce your connection around the world via an encrypted network. Tor is configured per machine, so the Anonymebox’s differing approach is more convenient.

Easy configuration

Once it’s on, it can be configured via the browser of any device connected to your router. The web interface is sparse, with just an overview showing the devices connected to the Anonymebox, and a settings menu to change the admin password and configure the access point details. Your first post-setup activity involves updating the default password and, optionally, changing the default name of the access point. Connecting to it over Wi-Fi is the same as connecting to any other router, but the Anonymebox will completely anonymise your online presence via Tor. Visiting a site such as whatismyipaddress.com will show that your location is in a completely different country.

Tor needs to be regularly updated to ensure protection,

£75/$115

anonymebox.com

The Anonymebox is a good tool for those who don’t have the skills to configure Tor on every machine they use, though its ease belies its true power.

Last word

The Anonymebox is a good tool for those who don’t have the skills to configure Tor on every machine they use, though its ease belies its true power.

£80/$108

oldfart.com

Easy configuration

Once it’s on, it can be configured via the browser of any device connected to your router. The web interface is sparse, with just an overview showing the devices connected to the Anonymebox, and a settings menu to change the admin password and configure the access point details. Your first post-setup activity involves updating the default password and, optionally, changing the default name of the access point. Connecting to it over Wi-Fi is the same as connecting to any other router, but the Anonymebox will completely anonymise your online presence via Tor. Visiting a site such as whatismyipaddress.com will show that your location is in a completely different country.

Tor needs to be regularly updated to ensure protection,

£80/$108

oldfart.com

Easy configuration

Once it’s on, it can be configured via the browser of any device connected to your router. The web interface is sparse, with just an overview showing the devices connected to the Anonymebox, and a settings menu to change the admin password and configure the access point details. Your first post-setup activity involves updating the default password and, optionally, changing the default name of the access point. Connecting to it over Wi-Fi is the same as connecting to any other router, but the Anonymebox will completely anonymise your online presence via Tor. Visiting a site such as whatismyipaddress.com will show that your location is in a completely different country.

Tor needs to be regularly updated to ensure protection,

£80/$108

oldfart.com

Easy configuration

Once it’s on, it can be configured via the browser of any device connected to your router. The web interface is sparse, with just an overview showing the devices connected to the Anonymebox, and a settings menu to change the admin password and configure the access point details. Your first post-setup activity involves updating the default password and, optionally, changing the default name of the access point. Connecting to it over Wi-Fi is the same as connecting to any other router, but the Anonymebox will completely anonymise your online presence via Tor. Visiting a site such as whatismyipaddress.com will show that your location is in a completely different country.

Tor needs to be regularly updated to ensure protection,
Review

BOOKS

RASPBERRY PI BESTSELLERS

Wiley’s top three bestselling PI books shouldn’t be missed...

PYTHON PROJECTS

Author: Laura Cassell & Alan Gauld
Publisher: Wiley
Price: £14.99
ISBN: 978-1119016762
tinyurl.com/jggwzaq

You’ve completed the Python tutorial – or begin a book, or MOOC – and you’re ready to move on. If you learned programming in order to join or begin a particular project, no problem, but if you didn’t, then where to go next to learn what you can really do with Python is a problem. So, in RASPBERRY PI USER GUIDE, the official guide remains the right place to go.

BLACK HAT PYTHON

Author: Justin Seitz
Publisher: No Starch
Price: £23.50
ISBN: 978-1593275907
tinyurl.com/v8d7em6

Python is a popular choice in the field of information security, and penetration testing in particular. Seitz, a senior security researcher at Immunity, presents a broad range of security topics, touching on tools traditionally used, then pointing the way to build your own Python equivalents. Like most good security books, it reveals what an insecure place our computer networks are, but provides you with the tools to do something about it, building Python replacements for many everyday tools like Netcat. This leads to stronger knowledge not just of the network security topics, but also where you can take Python. Along the way, Seitz respects the intelligence of the reader, but doesn’t assume detailed networking knowledge – introducing relevant information if it is necessary for progressing through the book. For example, SSH tunneling is explained, but the reader is left to look up any extra information she may want on the Address Resolution Protocol (ARP) on ‘Web Hacking’ will be of particular general interest. Anyone with a Joomla, Drupal, Word Press or similar site can feel justifiably alarmed about their security after a few pages of Python brute-forcing scripts, discovering leftover files and scripts on the server to gain admin login. A useful eye-opener.

ESSENTIAL READING: PYTHON

Python makes a great first language, but choose a book that matches your learning style...

Learn Python the Hard Way (Third Edition)

Author: Zed Shaw
Publisher: Addison Wesley
Price: £24.99 (free online)
ISBN: 978-0321849565
learnpythonthehardway.org

The hard way is typing 8 all in until you absorb all of Python and don’t make mistakes. Works well, but Shaw doesn’t cover Python 3.

Program Arcade Games: With Python and Pygame

Author: Dr Paul Vincent Crown
Publisher: CreateSpace
Price: £14.99 (free online)
ISBN: 978-1505825560
programarcadegames.com

Balances games and programming exercises to keep the learner going. Very popular: available in several languages on the website.

Writing Idiomatic Python 3.3

Author: Jeff Knupp
Publisher: CreateSpace
Price: £13.97
ISBN: 978-1612102870
tinyurl.com/pb8zqxp

Get Pythonic from the start. Concise guide to idiomatic code, best after another text, but suits some brave learners.

Dive Into Python 3

Author: Mark Pilgrim
Publisher: Apress
Price: £15.49
ISBN: 978-1430224150
diveintopython.net

Dives straight into code, then the explanations follow. A concise but comprehensive start that will appeal to independent study types.

Learning Python – 5th Edition

Author: Mark Lutz
Publisher: O’Reilly
Price: £14.92
ISBN: 978-1449353739
tinyurl.com/mx3hphp

Comprehensive doorstop (1,600 pages) great for programmers new to Python and object orientation. Covers Python 2.7 and 3.3.

BOOKS

PYTHON PROJECTS

Author: Carrie Anne Philbin
Published: Wiley
Price: £14.99
ISBN: 978-1119016762
tinyurl.com/oov562q

RASPBERRY PI USER GUIDE

Author: Eileen Upton & Gareth Hallscro
Publisher: Wiley
Price: £14.99
ISBN: 978-1119024605
tinyurl.com/k4qd8rl

Updated for the Model B+, the official guide remains an invaluable introduction to all things Pi, particularly physical computing. Aimed at beginners, the enthusiasm and depth of knowledge give something to every reader.

RASPBERRY PI ADVENTURES IN PYTHON

Author: Eben Upton
Publisher: Wiley
Price: £23.50
ISBN: 978-1118921661

For Pi users, while physical computing is left to some pointers at the end of the book, there’s still a wealth of material. Following a recap of core Python, it’s straight into scripts to access the operating system, then managing data. Useful exercises, along with summaries of what’s been learned, cement the education experience. After desktop and web applications come the tools you need for working on larger projects: testing, debugging, tuning, structuring, and releasing – all essential information. The book closes with a look at areas you may want to try next: SciPy, Pygame, drawing modules and animation, as well as the aforementioned physical computing. Plenty to inspire you to further Python adventures.

LAUREN IPSUM

Author: Carlos Bueno
Publisher: No Starch
Price: £15.50
ISBN: 978-1593275747
laurenipsum.org

Wow! The number of great coding books to appear in the last few years, aimed specifically at children, has been a very encouraging part of the movement to get young people coding. But coding – although fun, creative and rewarding – is only a part of computational thinking, which is a set of problem-solving skills including, but not limited to, algorithms, data modelling and logical thinking – invaluable in today’s world.

A computer science book with no computers in, written in the grand traditions of Lewis Carroll, it follows the adventures and encounters of Lauren Ipsum, lost in Userland. From Reversion Junction to the Push & Pop Cafe, meeting characters like Hugh Rustic and the Wandering Salesman, Lauren’s journey takes the reader through a history of ideas and logic. Bueno has an engaging style and the lessons are so integral to the characters like Hugh Rustic and the Wandering Salesman, Lauren’s journey takes the reader through a history of ideas and logic. Bueno has an engaging style and the lessons are so integral to the

from the start. Concise guide to idiomatic code, best after another text, but suits some brave learners.

Dive Into Python 3

Author: Mark Pilgrim
Publisher: Apress
Price: £15.49
ISBN: 978-1430224150
diveintopython.net

Dives straight into code, then the explanations follow. A concise but comprehensive start that will appeal to independent study types.

Learning Python – 5th Edition

Author: Mark Lutz
Publisher: O’Reilly
Price: £14.92
ISBN: 978-1449353739
tinyurl.com/mx3hphp

Comprehensive doorstop (1,600 pages) great for programmers new to Python and object orientation. Covers Python 2.7 and 3.3.
RASPBERRY JAM EVENT CALENDAR

Raspberry Jams are community-organised events that help you meet other Pi enthusiasts, share knowledge, and learn new things.

PUT YOUR EVENT ON THE MAP
List your forthcoming events at raspberrypi.org/jam
WHAT WOULD YOU DO WITH A RASPBERRY PI 2?
Let us know for your chance to win!

How to enter:
For your chance to win a Raspberry Pi 2 with a Pimoroni case of your choice, tell us what your dream Raspberry Pi project is. Simply email competition@raspberrypi.org with no more than 100 words detailing your perfect project, hack, or make.

Terms & Conditions
Competition closes 30th March 2015. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don’t like spam. Participants’ details will remain strictly confidential and won’t be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.
THE FIRST OF MANY STEPS

One of the newest members of the Raspberry Pi team, Matt Richardson, closes the issue with memories of his first experiences with the world’s favourite credit card–sized PC...

Since this is the start of a new era for The MagPi, I wanted to use it as an opportunity to reflect on my beginnings with Raspberry Pi. Admittedly, when I first heard of it before it was released, I was sceptical. “A small Linux computer for only £35? I’ll believe it when I see it,” I thought. I predicted it would either go up in price or down in history as another example of vapourware. I’m glad that I got it so wrong.

I was shocked when the Foundation started accepting pre-orders for the Model B. The hype was reaching fever pitch and it forced me to take this scrappy little computer a lot more seriously. At that time, Arduino was making hardware development more accessible than ever. What if all the powerful tools developed for Linux could be blended with what people were creating with Arduino? For example, writing or reading a file with Arduino was possible, albeit difficult. The same was true for networking, timekeeping, video display, and multitasking. I knew exactly what I wanted to do with Raspberry Pi, but I didn’t think much further than that.

Beyond physical computing

Physical computing projects were only my initial motivation to try Raspberry Pi. Since then, I’ve started learning how to work with Wolfram and Mathematica, which is included with Raspbian. I’ve also used Sonic Pi to experiment with synthesizing electronic music. I used a Raspberry Pi to set up a network server for my 3D printer. And now I’m learning low-level assembly programming by following one of the ‘bare metal’ resources available for Raspberry Pi. It’s teaching me a lot about how computers and operating systems work. After that, perhaps I’ll hook a Pi up to a TV and use it as a media player.

What’s important is that the Raspberry Pi is so versatile that there’s not just one possible first step with it. Whether you want to use it to learn basic Linux, acquaint yourself with Python, or you need a local file server for your family, Raspberry Pi fits the bill quite nicely. And if you’re anything like me, after Raspberry Pi gets a tiny toehold in your consciousness, soon you won’t stop thinking of ways that you can use it.

Clearly there’s a host of possible motivations for using Raspberry Pi. I encourage you to think about yours as only a starting point for what you ultimately do with it.
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.

Breakout Pi Plus
The Breakout Pi Plus is a useful and versatile prototyping expansion board for the Raspberry Pi.

ADC Pi Plus
8 channel analogue to digital converter. I²C address selection allows you to add up to 32 analogue channels 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.

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.

We also stock a wide range of expansion boards for the original Raspberry Pi models A and B.

ABelectronics UK
www.abelectronics.co.uk