HS

Issue #05 **April 2018**

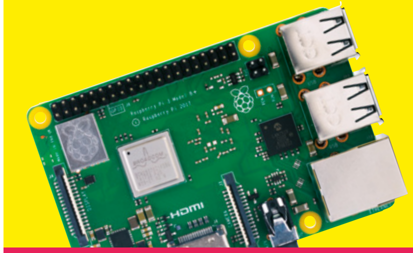
MAKE | **BUILD** | HACK | **CREATE**

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | **April** 2018 | **Issue** #05

Apr.2018
Issue #05 £6



RASPBERRY PI 3B+

Faster processor
& networking

RESIN.IO

Control all your smart
devices with one tool

DREMEL HACKING

Turn your hand tool
into a table saw

LED FASHION

Bring your accessories
to life with LEDs

LADYADA

LIMOR FRIED REVEALS **ADAFRUIT'S** SECRETS TO MAKING GREAT HARDWARE

LadyAda

LITHOPHANES **CHEERLIGHTS** PENCILS **NEURAL NETWORKS**

hsmag.cc

Welcome to HackSpace magazine

If you're reading this, then there's a good chance you like making stuff. And, that means you need bits and pieces to make your builds out of, whether it's electrical components, fully populated PCBs, or bits of wood and metal. We makers are also consumers, and we're pretty demanding consumers at that. Few other groups of shoppers are as likely to take apart, push to extremes, and critically evaluate a product as makers and hackers.

> Few other groups of shoppers are as likely to take apart, push to extremes, and critically evaluate a product as **makers and hackers**

If you want to learn how to make high quality hardware, then, the best person to speak to is someone who's built a business out of making products for makers, and that's what we've done this month as we have managed to get Limor 'LadyAda' Fried to share the tips and tricks she's learned at Adafruit with us.

BEN EVERARD
Editor @ ben.everard@raspberrypi.org

PAGE 44
SUBSCRIBE TODAY

GET IN TOUCH

@ hackspace@raspberrypi.org

f hackspacemag

🐦 hackspacemag

ONLINE

hsmag.cc

EDITORIAL

Editor
Ben Everard
@ ben.everard@raspberrypi.org

Features Editor
Andrew Gregory
@ andrew.gregory@raspberrypi.org

Sub Editors
Jem Roberts, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Dougal Matthews

Designer
Lee Allen

Photography
Brian O'Halloran,
Exposure Photo Agency

CONTRIBUTORS

Jen Botezat, Lucy Rogers, Andrew Huang, Limor Fried, Zack Akil, Cameron Norris, Mayank Sharma, Archie Roques, Graham Morrison, Sophie Wong, Bruce Santos, Alex Eames, Wesley Archer, David Floyd, Mathew Macdonald-Wallace, Mark Davies, Marc de Vinck, Les Pounder, Gareth Halfacre, Andy Clark

PUBLISHING

Publishing Director:
Russell Barnes
@ russell@raspberrypi.org

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
☎ +44 (0)207 429 4000

SUBSCRIPTIONS

Select Publisher Services
Ltd, PO Box 6337, BH1 9EH
☎ +44 (0)1202 586 848

Mann Enterprises Ltd,
Unit E, Brocks Business
Centre, CB9 8QP
hsmag.cc/subscribe

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

HackSpace magazine is published by Raspberry Pi (Trading) Ltd., Station Road, Cambridge, CB1 2JH. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised.

ISSN: 2515-5148.

Contents

18

06 **SPARK**

06 **Top Projects**
The Tate Modern of DIY projects

16 **Objet 3d'art**
Pretty (and pretty useful) 3D prints

18 **Meet the maker: Joseph Casha**
The man who's printing lost bits of Malta

22 **Columns**
Give credit where it's due (yes, including money)

24 **Letters**
Do makers even need a Guild anyway?

26 **Hackspace** Kwartzlab
Welcome to Kitchener, Ontario

31 **LENS**

32 **Inside Adafruit**
Limor Fried's guide to making great stuff for hackers

46 **How I Made:** Neural Network
Link lights intelligently to map a thinking algorithm

50 **Enable Makeathon**
Hardware hacking for the public good

54 **Raspberry Pi 3B+**
Faster, higher, stronger: the new Raspberry Pi

68 **Improviser's Toolbox** Pencils
Build weaponry and more with a humble HB pencil

Cover Feature

INSIDE ADAFRUIT

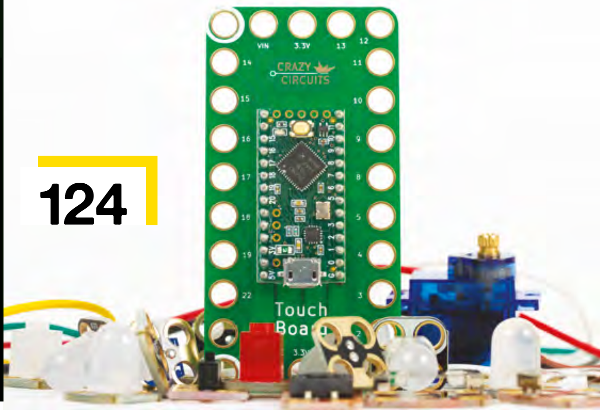
Everything you ever wanted to know about how to create your own world-beating hardware company

32

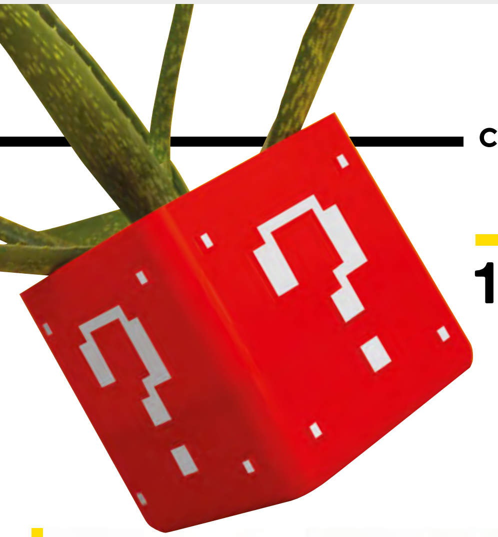
Interview

Ruth Amos from Kids Invent Stuff

58 Enter the surreal and brilliant world of kids' inventions



124



16

Review

ZX Spectrum Next Development Kit

126 This is the Z80 you've been looking for

28

84

Direct from Shenzhen

Stirling engine buggy

116 Turn heat into motion without the risk of a burst boiler

77 FORGE

78 **SoM** Seven-segment displays
Show information the old-new fashioned way

84 **SoM** Controllable LEDs
Add switching to your wearables projects

90 **Tutorial** Cheerlights
Join the world's biggest IoT project

92 **Tutorial** Dremel workbench
May your cuts be ever straight and true

96 **Tutorial** Slack-powered On-Air sign
Maintain silence the smart way

98 **Tutorial** The sheet bend
Learn the only knot you'll ever need

102 **Tutorial** Node-RED
Flow-based coding to control your home

106 **Tutorial** Resin.IO
Keep all your devices running the latest code

110 **Tutorial** Lithophanes
Turn flat images into 3D prints

115 FIELD TEST

116 **Direct from Shenzhen** Stirling engine buggy
Replace your Tesla Roadster with one of these. Go on, do it!

118 **Best of Breed**
Gesture sensors: for when commands get a little hand-wavy

122 **Can I Hack It?**
Every home needs a hacked wireless doorbell

124 **Review** Brown Dog Crazy Circuits
An accessible kit for playing with electronics

126 **Review** ZX Spectrum Next Development Kit
The home computer that launched a thousand games companies

128 **Review** Lever Axe
Upgrade your primitive iron-age technology

129 **Book Review** Making Time
Words written by Bob Clagett, who also likes to make stuff

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

Automatic herb box

By **Walter Heger**

hsmag.cc/JIfBql

The main intention to create the herb box ecosystem was to manage the humidity of my plants and connect everything up via the internet, (my herbs always went dry or suffered from over-zealous watering). I wanted to combine handcraft with electrical engineering and software development for the best result in usage and design. The box should work autonomously without any need for interaction.

The box holds up to three plants from the supermarket, to keep feeding as simple as possible. It measures the humidity of each and allows automatic or manual irrigation, based on the configuration of a website. To enable good growing also in winter, a plant growing lamp can be switched on or off. With Alexa, the status can be asked after, and all actions are controlled. □

Left
Arduino, Alexa and a bit of woodwork come together to keep Walter's herbs in good condition

CREATIVITY
HERO

DIY interactive LED coffee table

By Marija Nikolovska creativityhero.com

I've done a few LED projects so far, but this interactive LED coffee table is definitely my favourite.

My living room needed an update, so I came up with the idea to build a unique coffee table with some really interesting features.

I wanted this table to be modern, useful, and to fit my personality at the same time. So, I made exactly what I was looking for! This coffee table creates a wonderful atmosphere and takes my living room to a whole new level.

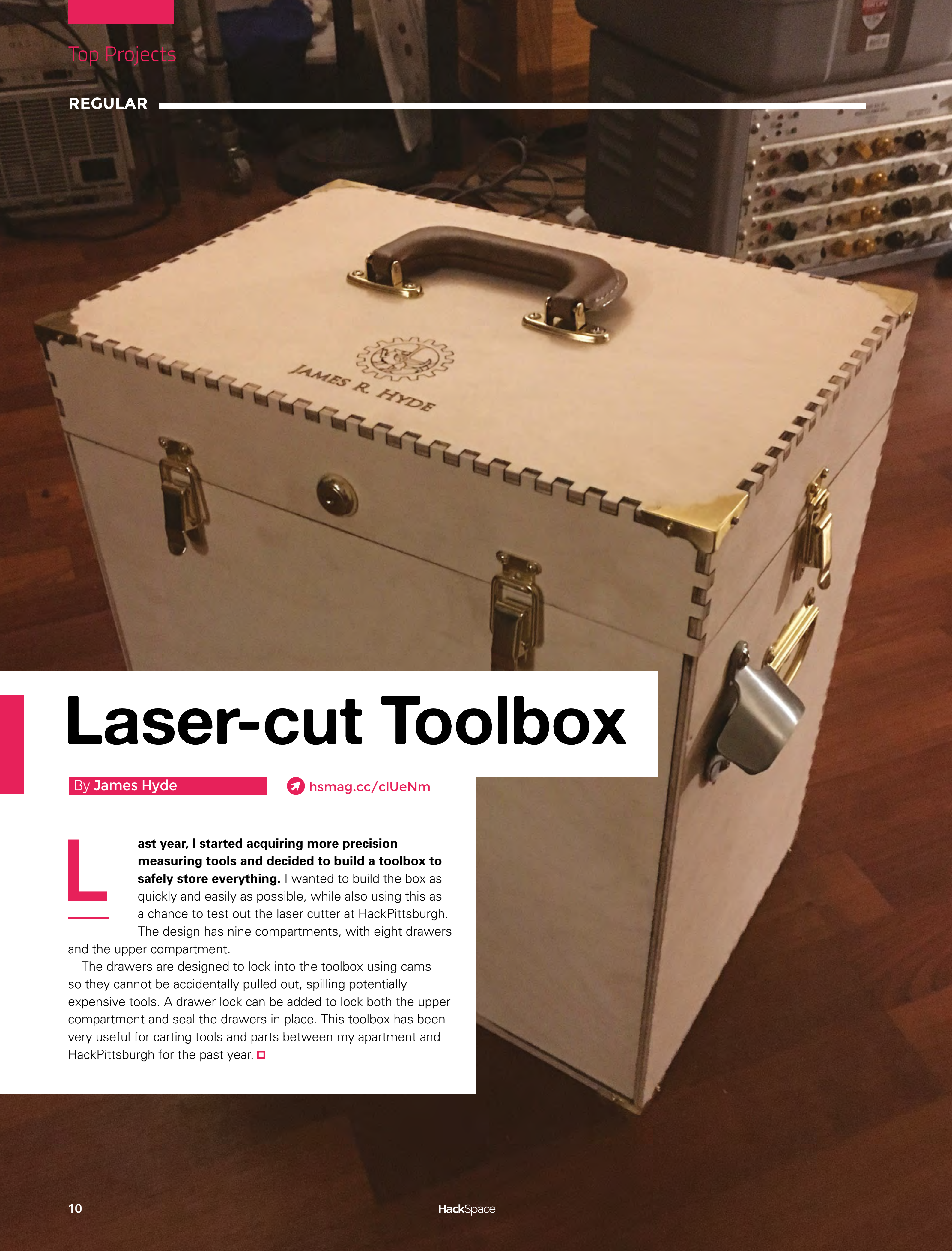
To build the body of the table, I used MDF for the box, pine for the legs, and a frosted glass for the top. The inner part consists of an Arduino board, a Bluetooth module, LEDs, proximity sensors, and a bunch of wires.

It is controlled via a custom-made Android application, so I can easily change the reactive colour, as well as the background colour, and I can even adjust the brightness.

These features are possible thanks to Dejan Nedelkovski, from **HowToMechatronics.com**, who created the source code of the program and the Android application.

My woodworking skills, and Dejan's knowledge in electronics, turned out to be a perfect combination for creating such a very cool piece of furniture.

Left
This table took ten days of hard work, including cutting, soldering, and adjusting all the parts



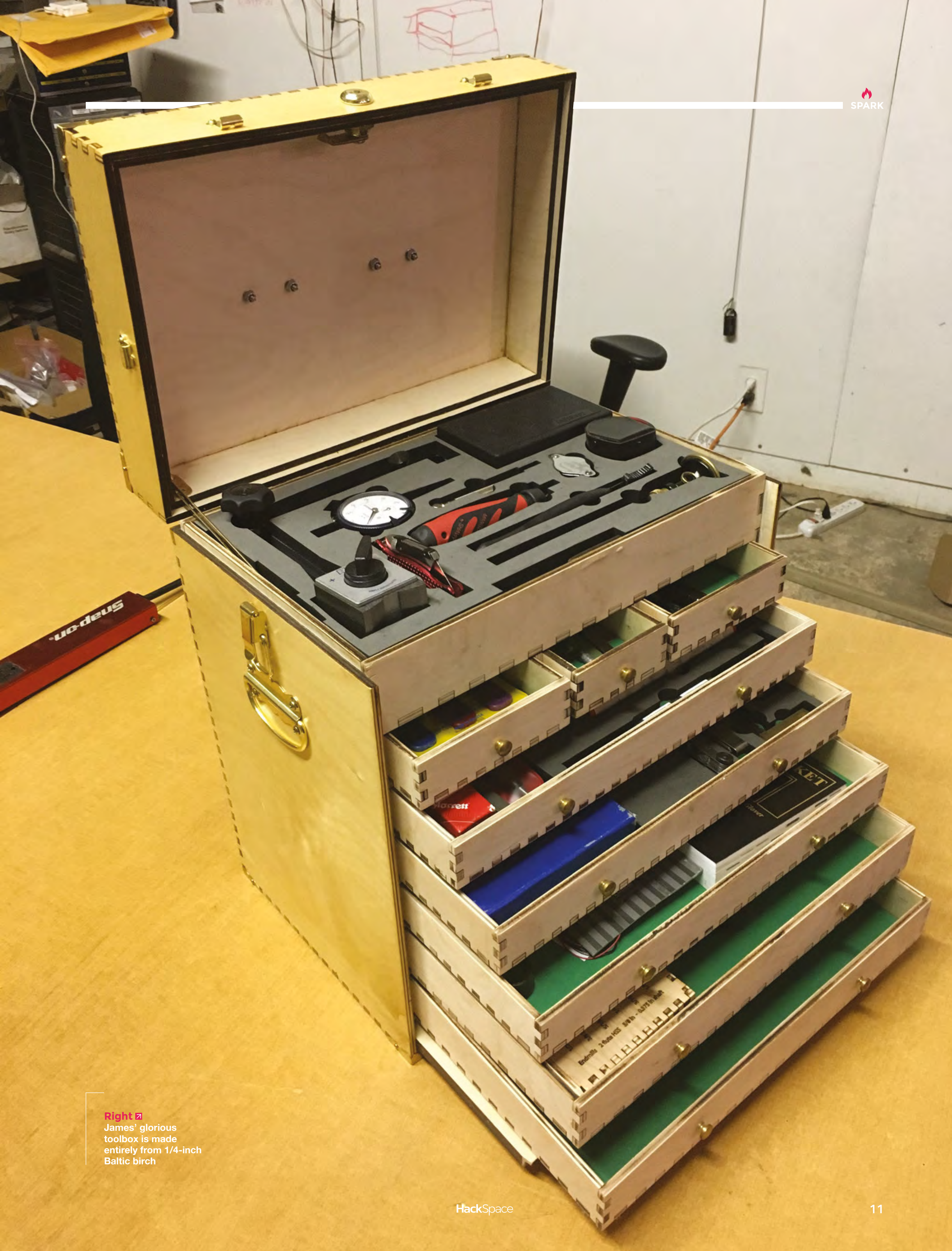
Laser-cut Toolbox

By James Hyde

hsmag.cc/clUeNm

Last year, I started acquiring more precision measuring tools and decided to build a toolbox to safely store everything. I wanted to build the box as quickly and easily as possible, while also using this as a chance to test out the laser cutter at HackPittsburgh. The design has nine compartments, with eight drawers and the upper compartment.

The drawers are designed to lock into the toolbox using cams so they cannot be accidentally pulled out, spilling potentially expensive tools. A drawer lock can be added to lock both the upper compartment and seal the drawers in place. This toolbox has been very useful for carting tools and parts between my apartment and HackPittsburgh for the past year. □



Right James' glorious toolbox is made entirely from 1/4-inch Baltic birch

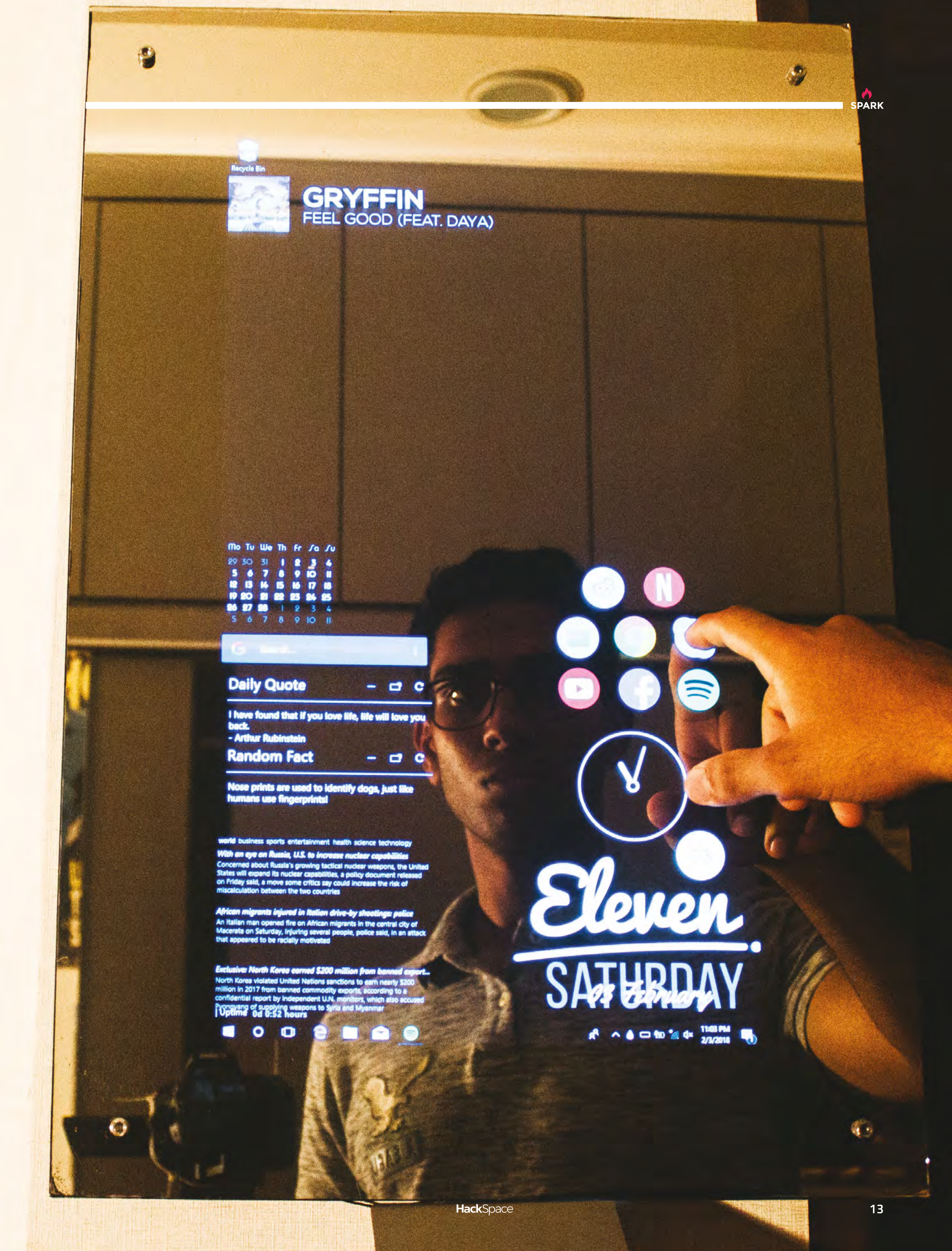
Raspberry Pi magic mirror

By Saral Tayal saraltayal

Whether you want to catch up on your day's calendar while brushing your teeth in the morning, perhaps check the weather as you head out the door, or maybe even watch a viral YouTube video as you stare at yourself, the smart mirror can do it all. Straight out of science fiction, the smart mirror is surprisingly simple to make, requiring just a one-way mirror, a screen, a computer or Raspberry-Pi, and a frame to house it all.

The smart mirror works by allowing the graphics of the display to shine through, while maintaining a mirror shine on the surface. With support for voice commands, widgets for news, Spotify, YouTube, weather, stocks, public transport timing, and any widget you want, this is the ultimate smart appliance project. □

Right ◈
Detailed build instructions for Saral's magic mirror can be found at hsmag.cc/WwOdzy



Mo	Tu	We	Th	Fr	Sa	Su
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	1	2	3	4
5	6	7	8	9	10	11

Daily Quote

I have found that if you love life, life will love you back.
- Arthur Rubinstein

Random Fact

Nose prints are used to identify dogs, just like humans use fingerprints!

world business sports entertainment health science technology

With an eye on Russia, U.S. to increase nuclear capabilities

Concerned about Russia's growing tactical nuclear weapons, the United States will expand its nuclear capabilities, a policy document released on Friday said, a move some critics say could increase the risk of miscalculation between the two countries

African migrants injured in Italian drive-by shootings: police

An Italian man opened fire on African migrants in the central city of Macerata on Saturday, injuring several people, police said, in an attack that appeared to be racially motivated

Exclusive: North Korea earned $200 million from banned export...

North Korea violated United Nations sanctions to earn nearly $200 million in 2017 from banned commodity exports, according to a confidential report by independent U.N. monitors, which also accused Pyongyang of supplying weapons to Syria and Myanmar

| Uptime 0d 0:52 hours

White Rabbit Nixie clock

By **Jack Edjourian**

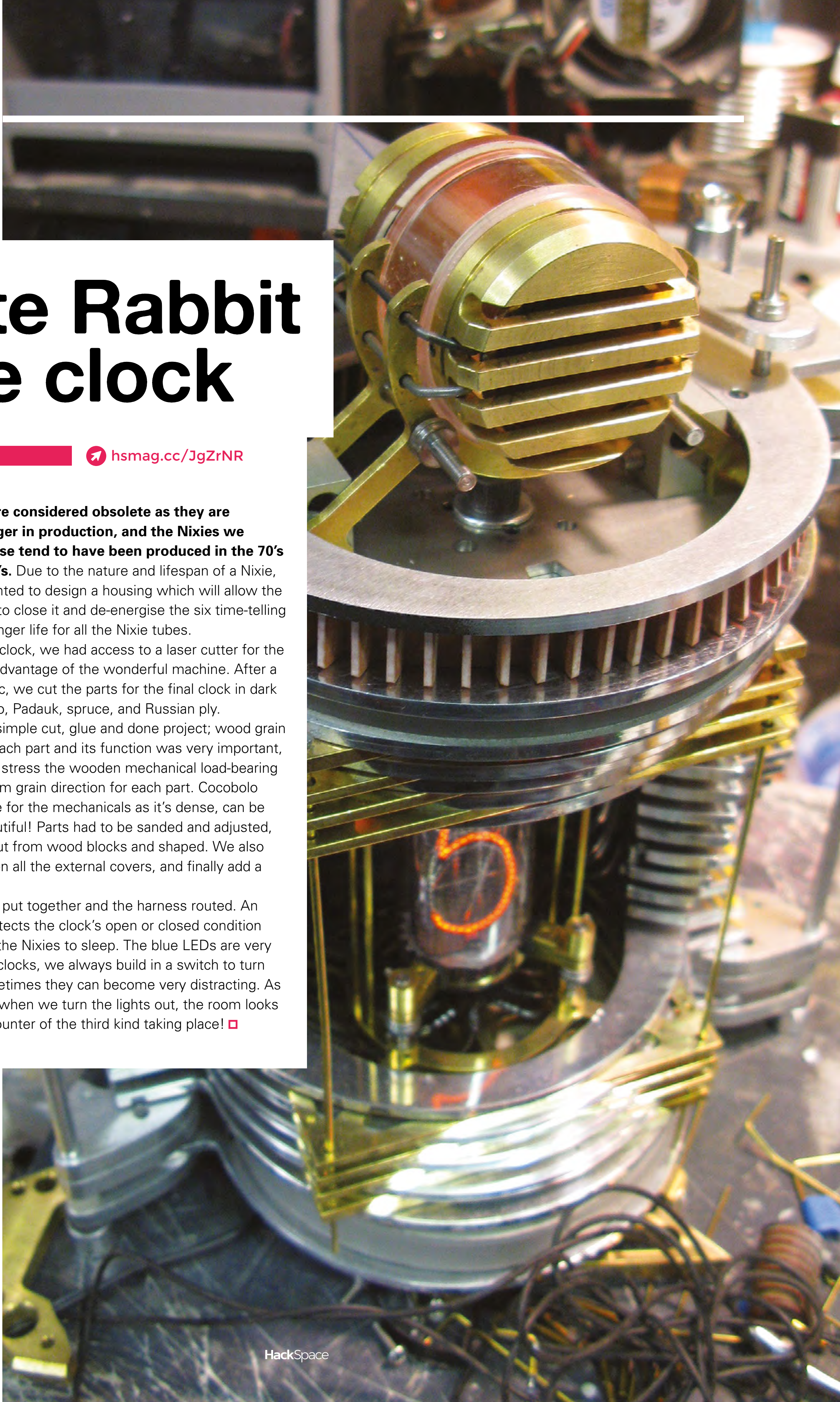
hsmag.cc/JgZrNR

Nixies are considered obsolete as they are no longer in production, and the Nixies we purchase tend to have been produced in the 70's and 80's. Due to the nature and lifespan of a Nixie, we wanted to design a housing which will allow the owner to close it and de-energise the six time-telling Nixies, allowing for a longer life for all the Nixie tubes.

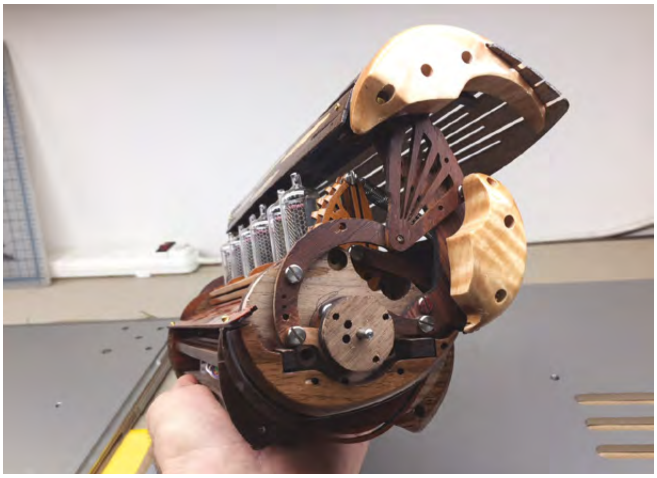
On the White Rabbit clock, we had access to a laser cutter for the first time and took full advantage of the wonderful machine. After a few prototypes in acrylic, we cut the parts for the final clock in dark walnut, maple, cocobolo, Padauk, spruce, and Russian ply.

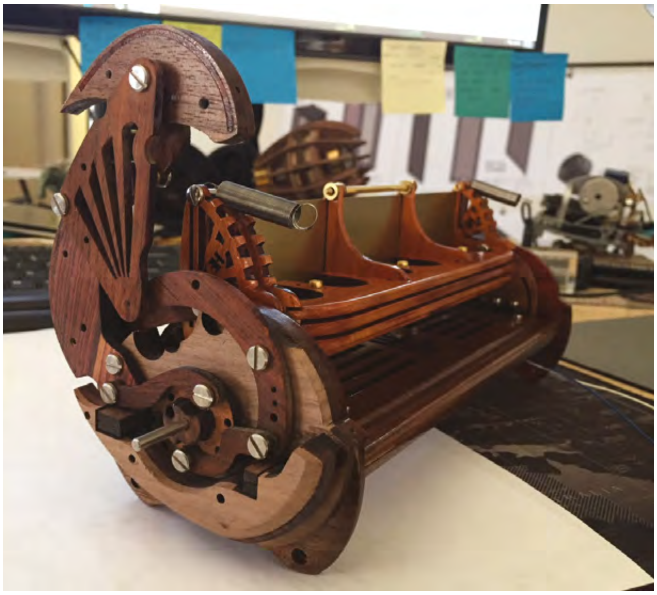
The clock was not a simple cut, glue and done project; wood grain direction in relation to each part and its function was very important, and tests were done to stress the wooden mechanical load-bearing parts to find the optimum grain direction for each part. Cocobolo was our wood of choice for the mechanicals as it's dense, can be tapped, and is also beautiful! Parts had to be sanded and adjusted, and some parts were cut from wood blocks and shaped. We also decided to inlay wood on all the external covers, and finally add a high gloss finish.

The electronics were put together and the harness routed. An internal microswitch detects the clock's open or closed condition and powers up or puts the Nixies to sleep. The blue LEDs are very common now in these clocks, we always build in a switch to turn these on or off, as sometimes they can become very distracting. As with our cyclops clock, when we turn the lights out, the room looks like there's a close encounter of the third kind taking place! □



Right
External components have three layers of catalysed automotive lacquer – internal parts are left with a natural finish to keep tight tolerances





Objet 3d’art

3D printed artwork to bring more beauty into your life

Since the dawn of time (well, since 1985), humanity has loved Super Mario Brothers. A simple tale of pipes, turtles, and mushrooms, it spoke a deep truth to us that transcends mere gamers into something more fundamental. And since the Industrial Revolution, we’ve loved to enhance our surroundings with little bits of greenery, as a reminder of what we’ve lost by cutting ourselves off from nature.

These two loves come together in 3D printed format in this Mario Bros planter, by graphic designer and Thingiverse user, Flowalistik. It’s available in dual and single extrusion versions, so you can use a different colour filament for the question mark, and of course you can make it with or without drainage holes. Best of all, it’s part of a triptych of works inspired by Crash Bandicoot, Mario, and Pokémon.

hsmag.cc/KKeaoE

Metric measurements make sense to anyone who can divide by ten, which hopefully is everyone reading this. But Imperial units are mostly nonsense. Some of them are okay, but even then, pints, and miles, and ounces come in different sizes, depending on what you're measuring and where you're measuring it.

This is exactly why we need this measuring cube, by iomaa. The cube measures a cup, 1/2 cup, 1/3 cup, 1/4 cup, 1/8 cup, 1 tsp, 1/2 tsp, 1/4 tsp, 1 tbsp, and 1/2 tbsp. If that's gibberish to you, one of these might come in handy. Just make sure that you use a food-safe printing method if you want to use this in the kitchen.

hsmag.cc/djUILG

Meet The Maker: Joseph Casha

A YouTuber bringing 3D printing to the masses

By Jen Botezat

After posting his first video just over a year ago, 3D maker Noob (less well-known as Joseph Casha), has already become one of the leading go-to channels for everything in 3D printing. He broadcasts live builds, 3D printer reviews, and all sorts of tutorials to legions of eager makers around the world. All from his home in an unlikely small town in Malta.

We sat down with the Noob himself on the beautiful island of Malta. He took off his helmet after riding his scooter across the island to meet us. With the towering presence of Vin Diesel and the smooth, measured voice of a masterful educator, it was easy to see the personality in 3D maker Noob that attracts thousands of devoted fans.

His steel-trap memory for technical details, model numbers, and programming specs were no less impressive.

Here, he shares his rollercoaster ride in the world of 3D printing in his own words.

Below
The 3D printing farm, with everything set up and ready for action

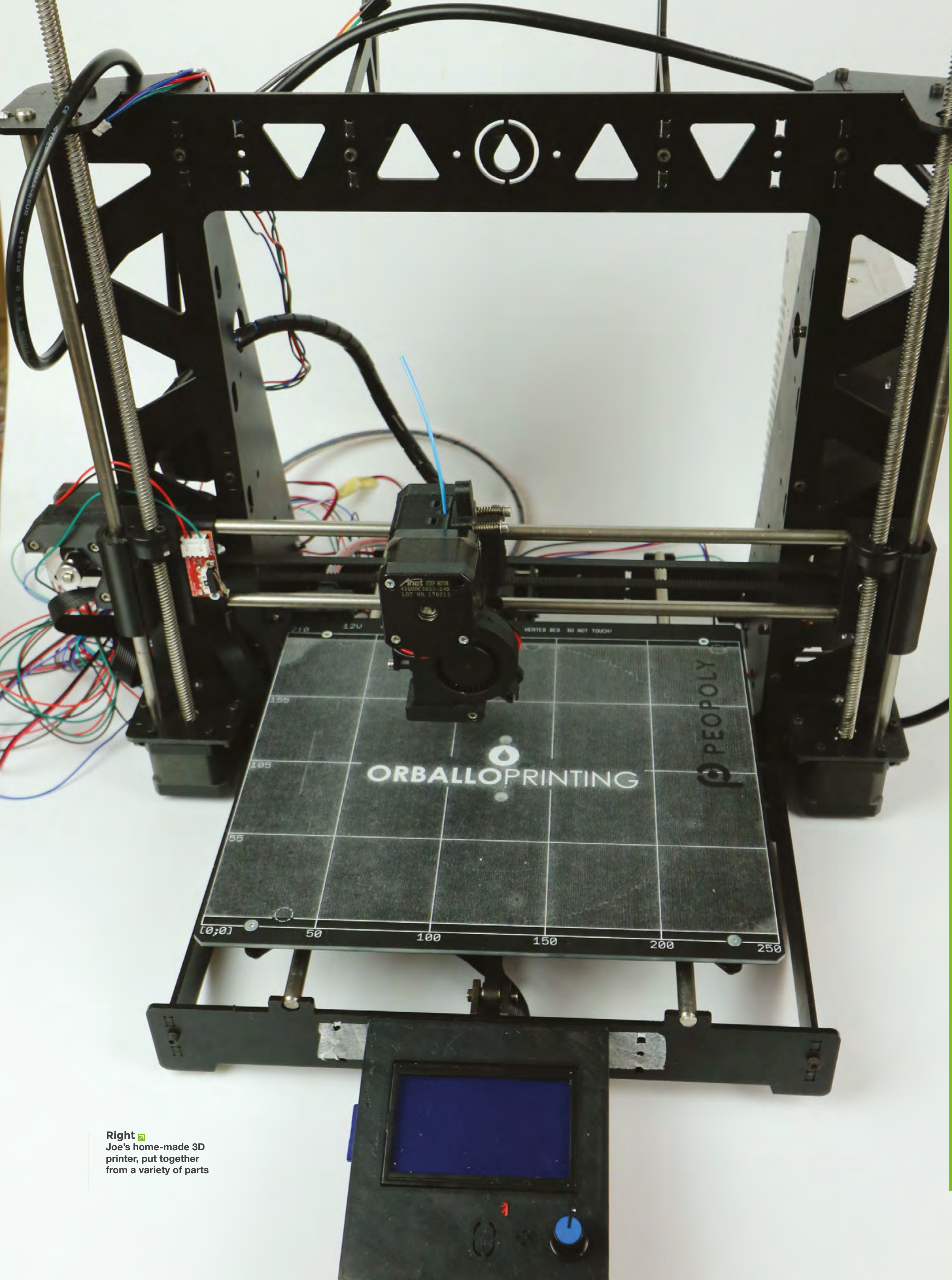
FROM THE BEGINNING

I started by watching some amazing 3D printing channels that existed already. 3D Printing Nerd (Joel Telling) offers great technical details and projects. Maker's Muse (Angus Deveson) is a fantastic designer. Thomas Sanladerer has awesome guides and reviews. They first inspired me, and I was in awe of their knowledge.

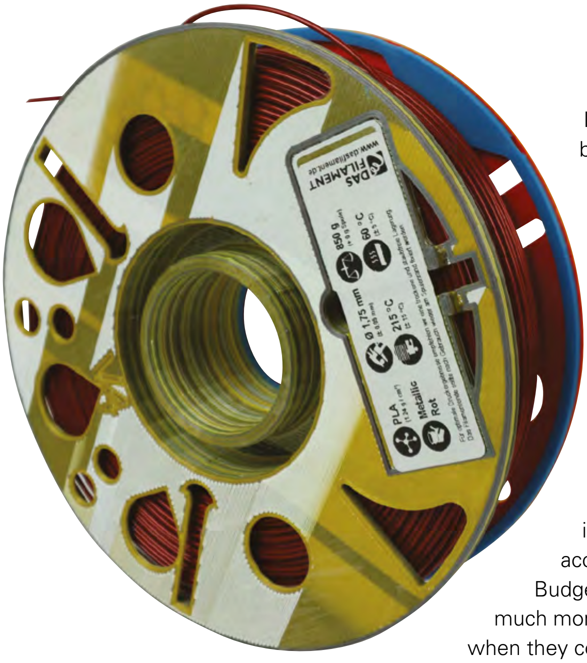
But I also felt that something was missing – a unique and less technical perspective that I could bring. I wanted to look at 3D printing through the eyes of a complete noob, someone who was interacting with the technology for the very first time, and learn about it together with my audience.

I bought my very first 3D printer, the Original Prusa i3 Mark 2 kit, and I was super-excited. I had never had a 3D printer or used one before, and it was fascinating to me. I couldn't wait to open up the boxes! So unboxing and reviewing the Prusa i3 became my first YouTube video.

Now, I have 50 printers. Most of them are not enclosed, and they are fairly affordable. Desktop 3D printing has opened up manufacturing to people in their homes, so that now they can experiment and tinker with making their own things. That's largely my audience. →



Right
Joe's home-made 3D printer, put together from a variety of parts



I mostly review budget printers, so that I can recommend them based on print quality, surface finish, ease of setup, and usability. I really want more people to get involved with 3D printing, so I am interested in printers that are accessible to hobbyists. Budget 3D printers require much more hands-on tweaking when they come out of the box. I have spent many hours tinkering with the settings and doing upgrades to make my printers print the way I'd like.

Perhaps the simplest 3D printer to get up and running out of the box that I found, has been the Prusa i3 Mark 3. I had just started my own business printing customer parts, and I had orders to fulfil. After I assembled the Prusa kit, I immediately began printing customer orders with it, and I only reviewed it after it had been printing non-stop for 1000 hours. I didn't experience any teething issues with setting it up, and the prints turned out fantastic.

> **Perhaps the simplest 3D printer** to get up and running out of the box that I found, has been the Prusa i3 Mark 3

I have noticed that manufacturers are now making better quality machines for a bit more in price, and people are willing to pay a bit more if it guarantees that the 3D printer will work out of the box. The Creality CR-10 is a perfect example of this, and it prints with amazing quality from the start.

MAKING WITH YOUTUBE

I was always into gadgets and learning how things work. I would take a Nintendo apart and look inside. With every project I have ever done, I referred to people who knew more about it than me. I have always used videos to learn from others online.

YouTube was crucial in making my projects. I am now restoring my motorcycle, and using YouTube to guide me through it. I am always amazed by the willingness of people to learn and help others there. I also have some absolutely awesome Patreon fans!

Through 'Just Joe' (a new YouTube channel), I am trying out new gadgets and technologies, and sharing my experience with the world. I'm flying stuff and occasionally crashing it – basically being a noob at everything, not just 3D printing. That's a great hobby for me. I'm still waiting for that one viral video!

It's an exciting time for me now, because I am finally moving the printers out of the house and into my new studio space. There, I will have more room to do awesome projects, film videos, and have visitors, who can interact with the technology. This will hopefully be completed soon.

I have also built my own 3D printer using different parts that I really liked. It brings together all the best things I've come across in my 3D printers. I made a test print with it recently, and it came out awesome!

I print full-time, 16–18 hours a day. I am collaborating with the local FabLab and doing all the 3D printing for their projects. Most recently, I made a full hand alphabet for an organisation that works with people with hearing disabilities. I have done projects on recycling leftover filament and plastic

Above
A master spool that helps Joe cut down on waste by using filament suppliers that don't send spools with each order of filament

Right
The Noob Rack, a storage system made out of old filament spools. The design is on Thingiverse: hsmag.cc/pZQueW

3D PRINTERS OF YOUTUBE

Watched 3D Printer Noob's videos and still on the lookout for more things to watch? Take a look at these channels as well:

3D Printer Nerd

hsmag.cc/FOlzlb

Reviews, tips and builds – Joel Telling provides a great overview of all things 3D printing-related

Thomas Salanderer

hsmag.cc/lrwpny

The Tom's Tips series is a goldmine of useful 3D printing advice

The Maker's Muse

hsmag.cc/BqhVik

With playlists such as CAD for Newbies, and 3D Printing 101, Maker's Muse is a great place to increase your 3D skills

3D print guy

hsmag.cc/jJgBmH

2D animations taken to the third dimension as the 3D Print Guy makes and prints models from popular cartoons such as *Rick & Morty* and *Samurai Jack*

spool holders – you can take a look at it on YouTube, just check out **hsmag.cc/qAdgik**.

3D printing to me is, first and foremost, fun. I don't want to think of it as a job, because as soon as you think of it like that, it becomes more of a chore. That's why I don't have a set schedule for posting videos. I only post when I genuinely come across something interesting or useful.

The YouTube channel alone isn't enough, so I do many other 3D printing projects outside it. I always keep myself busy!

Left The Royal Opera House in Valetta, Malta was reduced to ruins during WWII, but Joe hopes to recreate it as a 3D print

3D PRINTING FOR EVERYONE

It is still early days for consumer 3D printing. We are a long way from a truly plug-and-play machine that is as easy to use as, say, a microwave. 3D printers are still not attractive to people who are not makers and who will not spend time tinkering with settings. Setting up a 3D printer to print well is still a very hands-on, technical process that requires some practice.

The Ministry of Education and Employment in Malta recently invested heavily in 3D printers for public technical schools. As a result, 80 Ultimaker 3D printers made their way into schools across the country. However, the problem is that there are still very few people with the experience to run these machines.

I had the privilege of being invited to help teach students how to use the machines, design their own 3D prints, and build their own 3D printers from scratch. I have already presented a lecture on 3D printing at the University of Malta. Fortunately, nobody fell asleep!

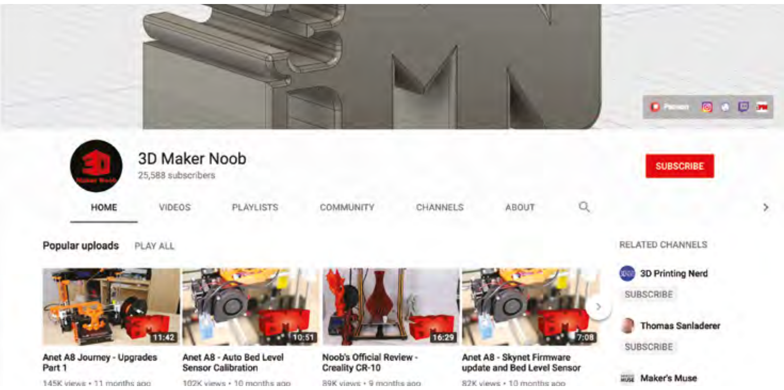
It felt weird to be at the front of the class, teaching students. I myself was very bad at school and left at the age of 15. Now, it seems like the roles have been reversed. The students raise their hands to talk to me and call me sir!

FROM MALTA WITH LOVE

There is definitely a lot of demand for 3D printing here, and 3D printing companies like Magigoo thrive here.

I am now working with designers to make a model of a Luzzu, a traditional fishing boat from the Maltese islands. I also want to recreate the Royal Opera House of Malta as a 3D model. It was one of the most beautiful and iconic buildings in the capital city of Valetta, before being destroyed by bombing in World War II.

I love this country. I want to keep promoting it and putting it on the map. □



3D Maker Noob
25,588 subscribers

Popular uploads

Anet A8 Journey - Upgrades Part 1
145K views • 11 months ago

Anet A8 - Auto Bed Level Sensor Calibration
102K views • 10 months ago

Noob's Official Review - Creality CR-10
89K views • 9 months ago

Anet A8 - Skynet Firmware update and Bed Level Sensor
82K views • 10 months ago

RELATED CHANNELS

3D Printing Nerd

Thomas Sanladerer

Maker's Muse

Below You can see all of Joe's videos at hsmag.cc/ZffKgl

Share the knowledge

Who puts all the information on the internet?

Lucy Rogers

@DrLucyRogers

Lucy is a maker, an engineer, and a problem solver. She is adept at bringing ideas to life. She is one of the cheerleaders for the maker industry, and is Maker-In-Chief for the Guild of Makers: **guildofmakers.org**

The collective term for a group of badgers is a cete. I only know this because I looked it up on the internet. Who put that information online? I have no idea. But when I want to know something, I can access most of the world's knowledge with just a few clicks.

As a maker, this has helped me enormously. Not the badger information in particular, but things like the pinout of chips, which oil is best to use on green oak, or how to lay out the frets on a cigar box guitar.

When it's just minor facts, I often just use the information, without giving a second thought to who actually put it there. But somebody out there kindly took the time and effort to contribute to the internet, and I call these people my superheroes.

I try to give due credit and attribution to those superheroes whose work I use in my projects. But I know I miss many – that one line of code that made everything work, but the source got lost among the open tabs on my web browser, that photo that inspired me, that funny tweet that I shared, without checking the origin. And I also know… I sometimes just forget.

I try to give due credit and attribution to those superheroes whose work I use in my projects. But I know I miss many

It's so easy to share things on the web – and an unscientific Twitter poll of my followers suggested that over 70% of them do already. We can all do something. It can be simple, such as answering a question or posting an inspiring photo. With a bit of planning, we can do something more complex, such as a series of how-tos or a beautifully edited film. Each contributes to the global knowledge base and could help unknown people around the world.

When we share online, we can make it easier for others to credit us correctly, while protecting our own rights – such as using Creative Commons (**creativecommons.org**) or Open Source Licences (**opensource.org/licenses**).We can obviously also ask strangers around the world for financial help by using membership clubs such as Patreon (**patreon.com**) and crowdsourcing, such as Kickstarter (**kickstarter.com**).

There are many reasons to post things online. For me it's the love of sharing, to give back, and as a record. But there's also that little ego thing too of, "Hey, look what I did!" Or as H. Jackson Brown says in *The Complete Life's Little Instruction Book:* "Share your knowledge. It's a way to achieve immortality." □

Transparency vs. Liability

How Spectre/Meltdown impacts open hardware

Bunnie Huang

@bunniestudios

Andrew 'Bunnie' Huang is a hacker by night, entrepreneur by day, and writer by procrastination. He's a co-founder of Chibitronics, troublemaker-at-large for the MIT Media Lab, and a mentor for HAX in Shenzhen.

By now you've probably heard about Spectre/Meltdown, a series of CPU bugs that affect CPUs with aggressive speculation capabilities. For better or for worse, the Raspberry Pi's CPU doesn't feature the performance optimisations that trigger these bugs, so Raspberry Pi users are safe – today. However, the public backlash triggered against many CPU makers highlights a double-standard in open source that threatens transparency in the hardware world.

Open source software developers benefit from licences that absolve them of any liability whatsoever. Should a buggy library you develop be used in a home automation appliance that later causes a house to catch fire, you get to walk away scot-free, thanks to the expansive limited-liability clauses that are baked into every open source software licence.

Unfortunately, hardware makers don't get to enjoy that same luxury. Beyond guaranteeing a product free from workmanship or material defects, consumer protection law often requires an implied or express 'fitness for purpose' guarantee – that a piece of hardware is capable of doing what it's advertised to do. The latest controversy over Spectre/Meltdown indicates that more people than not feel CPU makers like Intel should be liable for these bugs, under the 'fitness for purpose' theory.

Open hardware makers should be deeply concerned. Consider if the Raspberry Pi were found vulnerable, and the only fix was to recall hardware units and replace them with a new board. If you had baked a Raspberry Pi into your product, what would happen to your business?

Now consider that open review of documentation greatly increases the rate of bug discovery. Does it make business sense to share documentation with a customer base that will surely sue you for the favour?

Open source software developers benefit from licences that absolve them of any liability whatsoever

There's no right answer to this situation. We need transparency to make more secure, trustworthy, bug-free systems, but consumers also need to make sure they are not being sold a lemon. The Spectre/Meltdown question merely begs the question, but offers no guidance on the correct answer. If you think transparency is important to safety and security, rewarding hardware makers for sharing documentation by voluntarily reducing their liability can create a measurable economic incentive, thus encouraging more sharing. On the other hand, if you feel a guarantee for fitness is paramount, then you must accept that what you don't know can't hurt you – until it does. In other words, if your policy is to sue developers over bugs, you can't in the same breath blame them for making bugs harder to find by closing their source. □

Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at **hsmag.cc/hello**

WE ARE SAILING

Please extend a hearty salutation to Josh, the maker whose boat you featured in Issue 4. It doesn't look practical, or fast, or warm, but that it's put a massive grin on his face will do for me. People will always pay for things that make them happy, so good luck to him.

Amy
London

Ben says: My favourite bit about that build is that you can see the join between two pieces of wood in the hull, like two pieces of a jigsaw fitting together. Lovely.

GROW UP

Lucy Rogers' assertion that the maker movement has grown up is, in my opinion, a bit silly at best and insulting at worst. It's as if, for the years I've been working away in my shed I've been messing, rather than running a serious business, paying tax, insurance, all the rest of it.

Now there's a Guild of Makers, making is all of a sudden a proper industry? How condescending.

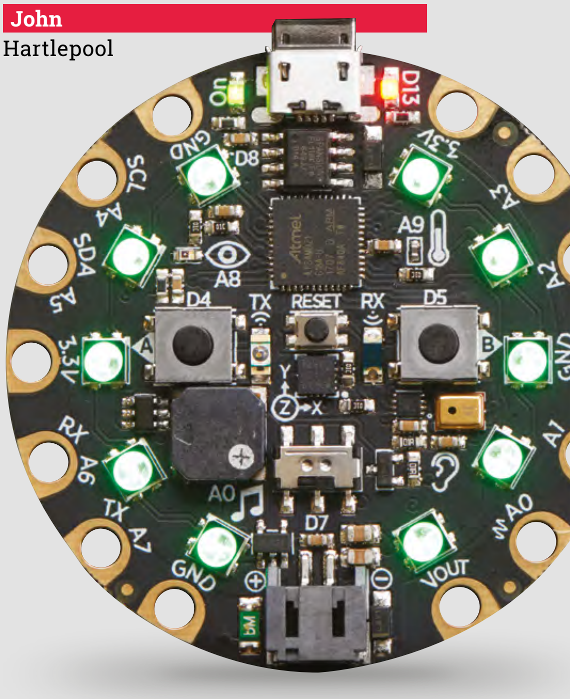
Robert
Birmingham

Ben says: I'm not sure that's the point. The Guild of Makers is there to help other people get to the point that you're at; where tax, insurance, and other business admin is just routine stuff to do, rather than terrifying and unknown. And the point about the maker movement growing up is just that: the movement growing up, not the makers.

GOODIES

Thanks for the Adafruit CPX! I subscribed from Issue 1 so I thought I'd missed out; then it arrived completely out of the blue. It's a great bit of kit, so cheers!

John
Hartlepool



CROWDFUNDING NOW

BUYER BEWARE !

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.

Mugsy

The world's first hackable coffee maker

From $150 | hsmag.cc/vxVkrm | **Delivery:** Nov 2018

While there are lots of coffee makers that can be hacked, prodded, and generally twisted to make coffee in just the way that's right for you, Mugsy is the first one we've come across that's designed from the ground up to be customisable.

While most coffee makers make espresso-style coffee by forcing pressurised hot water through the grounds, Mugsy uses the pour-over technique which, as the name suggests, involves pouring hot water over the ground coffee. This method is becoming popular among hand-made coffee aficionados, but is still a rarity in the automatic coffee maker world.

Mugsy requires just whole beans and water to create your drink, and it lets you customise almost everything about how the two are combined, from how much coffee is used to the pattern in which the hot water is poured over the grounds. Of course, not everyone knows enough about coffee to select the perfect options. Fortunately, Mugsy's linked up to a database of different coffees with recipes for each one. Just scan the bar-code of your beans and you'll get the recommended settings automatically. Alternatively, if you've got the settings you like, you can link them to an RFID tag (two keyring fobs come with Mugsy), and you can get the perfect cup each time, and not be infected with other people's inferior settings.

As well as ordering up a brew using the standard controls, you can also text, tweet, Slack, email, or Alexa yourself up a cup of hot caffeine.

All the mechanicals are controlled by a Raspberry Pi that you can control to bend the machine to your whim, whether that's tweaking parameters to the standard software, or writing your own code to take control of the hardware. □

Hackspace of the month: KWARTZLAB

KWARTZLAB

kwartzlab.ca

@kwartzlab

Chris Bruner spoke to us about Kwartzlab, a space in Kitchener, Ontario, that's at the heart of the maker community. They have lasers!

We've been around for about eight years, and have about 70 members currently. We all work on our own individual projects and some are more visual then others. We have a weekly open night, where anyone is allowed to come and work on projects and get to know us. It is our main method of getting to know potential new members. There is a membership fee of $50 a month. We try to only allow people who we think would be a good fit with us, that is makers and people who volunteer for things. If you look at the Kitchener Maker expo, almost everyone mentioned in the About section is either currently a member of Kwartzlab, or has been a member in the past, and that was all volunteer work: **hsmag.cc/mzcSjj**.

Kwartzlab had one of the first 3D printers in the area before they were 'a thing'. I don't know that the kit was ever functioning well, but well, we were early adopters. Now we have three that are fairly reliable. We also have some large tools, a wood bandsaw, a metal bandsaw, a CNC router, a Bridgeport milling machine, and a metal lathe and a wood lathe, as well as oscilloscope, and standard electronics equipment.

We have a laser cutter, thanks to the heroic James (do not look at laser with remaining eye), and have done significant modifications to make it more reliable. There's also a weekly meeting of the robot club, which is basically just robot enthusiasts. →

Kwartzlab's Artist in Residence program has seen a great range of inspiring artists come through the doors for three-month residencies

Above Yes, that's a giant game of Connect Four

HOW BIG?

Kwartzlab has around 2400 sq ft of space, split into four areas:

Front room
Event space, work tables, 3D printer, digital projector, laser printer, piano, photo booth

Laser room
Laser cutter, 3D printers, drinks fridge

Shop
Wood/metal working, shop sink, garage door

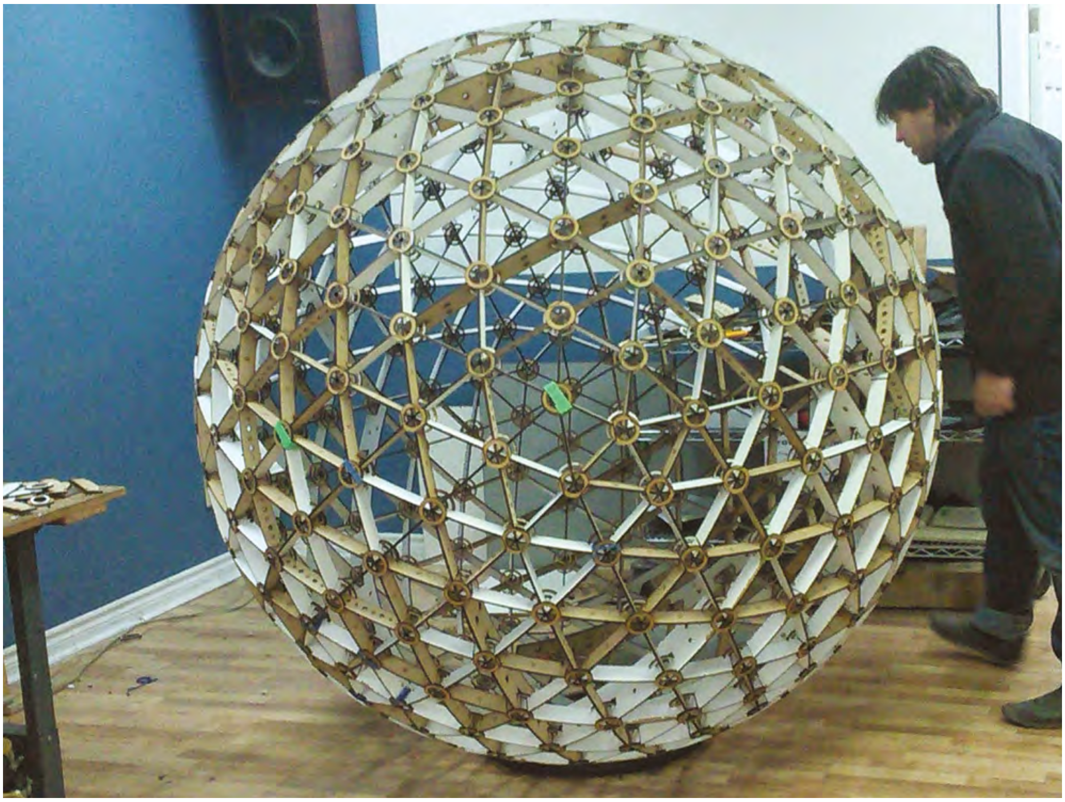
Upstairs
Kitchen, craft/sewing area, electronics bench, member storage, TV, and game systems

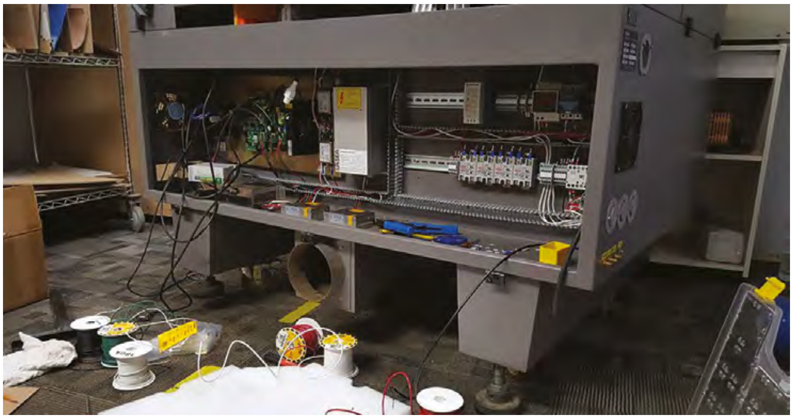
KWARTZLAB ROBOT CLUB ◈

Mike Robertson (left) and Chris Bruner of Kwartzlab working on Prometheus, which we hope to use as a standard base for hobby robot enthusiasts.

GIANT LASER-CUT PROJECT ◈

Don Leibold is creating a giant sphere out of laser-cut plywood and hair elastics. This structure will eventually serve as a sit-in pod that floats near-frictionlessly on a cushion of air. Think 'awesome flight simulator' and then think beyond that.





OUR LASER CUTTER

Not only does James (who is a hero), have a solid understanding of how the laser cutter works, he understood how the laser cutter was being used by members. He understood the deficiencies in the original design, and how to optimise its operation. James formulated a thorough plan, created schematics, a bill of materials, and secured funding from the Board, not only to fix the laser cutter, but improve it immensely as well.

THE BRIDGEPORT MACHINE

The beautiful Bridgeport mill. Kwartzlab members were looking for one of these for a while, when one was kindly donated by Clearpath Robotics: **clearpathrobotics.com**.

CONTACT US

We'd love you to get in touch to showcase your makerspace and the things you're making. Drop us a line on Twitter **@HackSpaceMag** or email us at **hackspace@raspberrypi.org** with an outline of what makes your hackspace special, and we'll take it from there.

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 32

MAKING FOR MAKERS

How Adafruit builds hardware for the most demanding users

PG 46

VISUALISING A NEURAL NETWORK

Add sci-fi lights to an artificial brain

PG 50

ENABLE MAKEATHON

Red Cross join forces with makers to improve the world

PG 54

FINGER IN THE PI

Now with an extra slice of networking speed

PG 58

RUTH AMOS

Bringing life to the creative juices of the young folk

PG 68

IMPROVISING WITH PENCILS

Build bridges, shoot your foe, and draw grey lines

PG 72

LEDs

The flashing, blinking, illuminating semiconductor

LIMOR FRIED

MAKING FOR MAKERS

Adafruit has been making great maker hardware for over a decade – here are the secrets the team has learned along the way

People have been hacking around with electronics for as long as there have been electronics to hack about with, but in the last decade it's gone from a niche pursuit of a few geeks to the edge of mainstream culture.

This popularisation of DIY technology is due to a lot of factors, but one of the largest must be the companies putting out great hardware to hack on, from kits that you can build to hardware designed for makers. Of this new breed of hacker companies, Adafruit, led by Limor Fried (aka LadyAda) must be the most iconic.

It's easy to forget that Adafruit hasn't been around that long. Back in 2005, LadyAda started Adafruit in her dorm room at MIT where she was a graduate Engineering student.

From this modest beginning the company has continued to grow. In 2009, in recognition of the fact that all Adafruit's hardware is released under an open source licence, the Electronic Frontier Foundation (EFF) awarded her the Pioneer Award.

Limor's particular genius seems to be melding the geek world of interesting hardware and open source designs with the business world that means these designs are commercially successful. Keeping money coming in means more hardware for hackers and makers. This is something that's been recognised outside of the geek world, and in 2012, Entrepreneur magazine named her as Entrepreneur of the Year.

In 2016, Limor was named a White House Champion of Change – an opportunity for the White House to feature individuals doing extraordinary things to empower and inspire members of their communities. Perhaps a large part of this success has been Limor's engagement with the wider making community, and Adafruit's live-streams are a staple part of the week for many makers. Since 2010, Limor and partner Philip Torrone have been hosting Ask An Engineer at 8pm (New York time) every Wednesday.

LIMOR WAS NAMED A WHITE HOUSE CHAMPION OF CHANGE – AN OPPORTUNITY FOR THE WHITE HOUSE TO FEATURE INDIVIDUALS DOING EXTRAORDINARY THINGS

We caught up with Limor to find out what it takes to make great hardware for makers. →

LIMOR'S 3 TOP TECH TIPS

Make hardware the Adafruit way with Limor's advice for making great products

1

Makers are a demanding market for hardware companies. While people building consumer products can hide their circuitry behind fancy plastic covers that are glued shut in an attempt to keep prying eyes away, and dodgy code can be hidden in proprietary software, when you're making stuff for makers, there's nowhere to hide. The bare circuit boards expose all your wiring, and open source software will be prodded and pulled apart.

Every limit of your hardware will be tested by people pushing more volts in, pulling excessive current from pins, trying to run on the minimal possible power and just about every other indignity an electrical component can experience will be experienced by poor hardware built for makers. Through it all, your little device has to function as well as possible in the circumstances.

We've been asking Limor to share her hard-won advice to help more people design hardware for makers. Take it away Limor:

"Pretty much since I graduated school, I've spent my time designing hardware for makers – whether they be projects for myself or kits to sell. It's been more than 12 years and I've learned a lot about what makes for a good kit. Here are some of my top tips for hardware design – from detailed power supply suggestions, to the zen of regression testing…"

1 POWER SUPPLY

Ahh, the humble power supply – this is the oft-neglected part of hardware design. Nobody loves to build the power supply: it's just something you 'have to do' so everything else works. But that doesn't mean it isn't important: most customer problems stem from power supply woes. Since it's your gateway to the world, there are a lot of ways things can go wrong here, but a really good power supply section will keep mistakes from turning into disasters.

USB power jacks Yep, even if you aren't using USB for data, a USB connector is a good way to get about 5V DC into your project. Unlike DC power jacks, you can be pretty certain of the voltage and polarity. There are wall adapters with USB out, and those USB power-packs for on-the-go power.

DC power jacks These common 2.1mm jacks are ubiquitous, low-cost and super durable. You can get them with through-hole or surface mount pads. Stick to 2.1mm, the most common size. 95% of the time, they're centre-polarity and DC, and range between 5V and 12V but some audio adapters are negative-polarity and/or AC. So…

Below
USB power packs give you a power source, charger, and standard connector

Above
2.1mm power jacks are ubiquitous, but watch out for different polarities

Stick a diode on it 1N4001's are common, inexpensive, and when placed in series will protect you from negative polarity going POOF! (You'll lose 0.7V or so from the diode – advanced hackers can use a P-FET instead.)

Fuses for safety You can pick up PTC 're-settable' fuses in a wide variety of currents and voltage ranges. Glass fuses are good for high currents, but need replacing after they blow. PTCs just need a few minutes to get back down to temperature.

Protecting from overvoltage-input is something you'll have to deal with if you have a 2.1mm plug input. You can use a PTC fuse and a zener diode to create a simple voltage clamp. Or just pick a voltage regulator with a high input voltage, like the honourable 780X or 1117 series regulators.

Don't forget a heat sink on your linear regulator These can get really hot when the input voltage and current draw is high. If you've got a standard TO-220 size regulator, and (Vin-Vout) * current > 1 Watt, you definitely need one. →

2 PICKING A BRAIN

Chances are your project uses a microcontroller or microcomputer. Good idea! You can cram a whole lotta code into chips these days. This is where the fun really is…

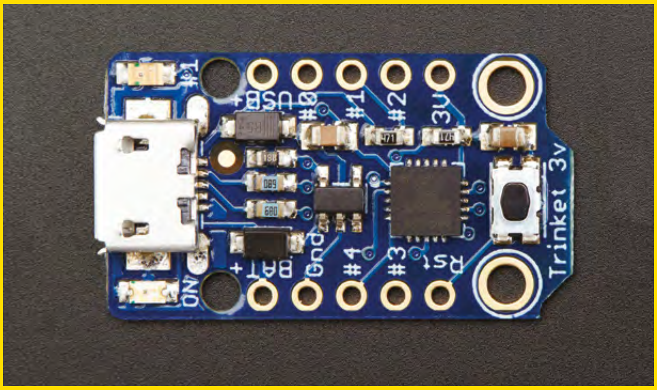
Go with what you know Are you a big Arduino Artist? Maybe a Pythonista at heart? Or an FPGA Fan… Sometimes it seems like there's a lot of choices, paralysingly so. Don't worry about getting the perfect design first round. Use whatever you've got to create a sketch of the hardware you want to build. Even if it takes too much power (and you want it to be portable) or is too tough to build, it's best to get a prototype running with what you know. Then you can…

Optimise at the end A wise man by the name of Knuth once said "premature optimisation is the root of all evil (or at least most of it) in programming." That applies to making and hardware as well. It's far easier to design your project in a high level language, with an overpowered microcontroller/computer, then slowly optimise and speed it up, to reduce the cost and parts list. I try to remind myself of this daily because the urge to optimise early is strong – resist it!

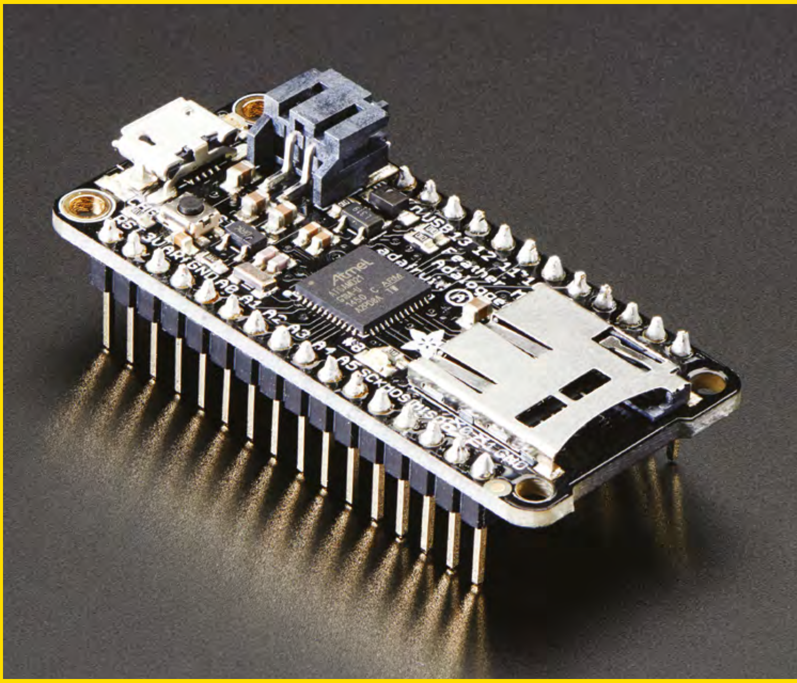
Have a way to test each part Chances are you're going to be gluing together parts from various datasheets, open source projects, product examples, GitHub repositories… each part of your project is going to have its own code and hardware requirements but those can cause problems with other parts! It's imperative you have code you can use to test each section to make sure it works. That way, if something goes amiss, you can re-load your testing suite, and check that each element at least works on its own.

Read the errata This didn't used to be a huge deal, but more often, I find the chips come with pretty big errors that don't appear in the main datasheet – so look in the errata! Likewise check that the datasheet you have is the latest – we've had datasheets with incorrect pin-out information that was then 'silently' corrected in a later version. (Shakes fist at clouds!)

Below
The ATtiny on the Trinket is a great way of adding Arduino compatibility for low cost

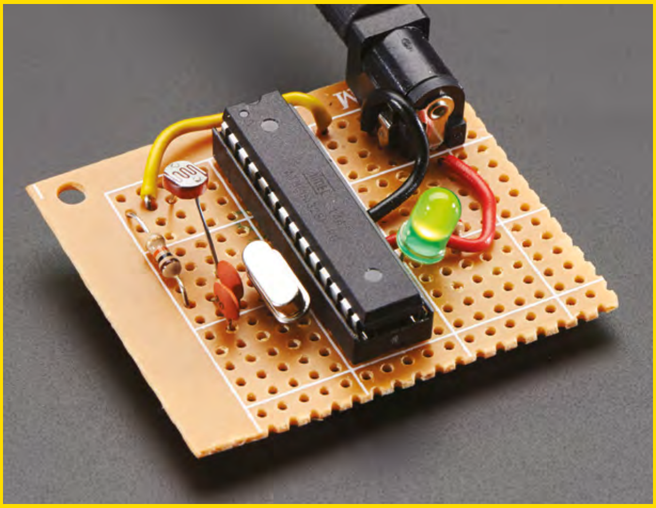


Below
The Feather range gives you a choice of different processors in the same form factor



3 MAKING IT PERMANENT

Once you've built a project and got it working, you'll want to save all the hard work you did and make your project permanent. Or at least, long lasting. If you do a good job at this stage, your invention will last for many, many years.



Above: Upgrade your breadboard designs to soldered boards for longevity

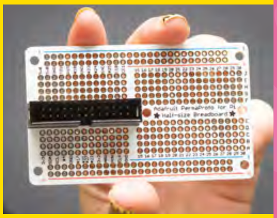
- **Solder-full breadboards** You probably built your design on popular solderless breadboards. These chunks of white plastic contain metal clips that grab your wires and components to make connections. They're great for prototyping, but they rust easily, or the springs weaken, so over time you'll get annoying flaky behaviour. Changing over to 'Perma Proto' or 'Veroboard' PCBs will give you a permanent setup. You can even try making a custom PCB! Either way I recommend getting a full new set of parts and 'rebuilding' with soldered connections, so you can check section by section.

- **Connectors** Your prototype maybe uses headers to make connections. These are okay, but not great for long-term use. If you do use plain 0.1"/2.54 mm headers, grab some gold-plated ones. Terminal blocks are a good alternative, as are proper plugged connectors like Molex or JST brand – get polarised and unique connectors to avoid mixups! Panel-mount connectors are a little more expensive but will look great when you pick out your…

- **Enclosure** Keep your project safe in its own little home. Historically, makers would peruse electronics shops for a variety of cases. The most popular types are injection-moulded ABS boxes, bent-sheet aluminium cases, and extruded aluminium tubes with caps. All of these can be cut and drilled and labelled, but they require a lot of work to look good. Get creative with plastic food storage bins, pencil cases, lunch boxes... you may already have something in your recycling you can clean and repurpose.

- **Documentation** Let's be honest, you're not likely to spend a bunch of time documenting everything you built. So, rather that set yourself up for an unreachable expectation, do the best job you can in 30 minutes. Take lots of photos, with all the parts and connections visible. Photograph part numbers on the top of chips & components. Photograph the top and bottom of any PCBs to show wiring and traces. Print out a schematic, drawn or from CAD software, and place in a ziptop bag tucked into the project enclosure. Code is a little tougher, it's probably not practical to print it out, but you can try copying it onto an SD card that goes with the schematic. I also recommend signing up for an online code-sharing account and popping it in there; as of this writing, GitHub, GitLab and Bitbucket are popular ones. →

Below: There's a wide range of protoboards available so pick one that fits the connectors you need



DESIGNING HARDWARE FOR MAKERS

How the Circuit Playground Express came together

After Limor gave us her top tips, we asked her to walk us through one of her designs, and the process that went into one piece of hardware. Here at HackSpace magazine we're massive fans of the Circuit Playground Express – it packs a huge amount into a tiny space and it's just good fun to use, so that was the obvious choice.

1 **Form Factor** One thing I noticed with the Flora is that people were sewing it into wearable projects, but far more people were using alligator/croc clips. And we were getting feedback from teachers and parents letting us know that the school couldn't afford solder tables, or breadboards were too difficult for young kids to use. I liked that thought, of skipping over breadboards completely, and I think the round shape makes it friendlier for people to hold, and the big holes are good for attachment points as well.

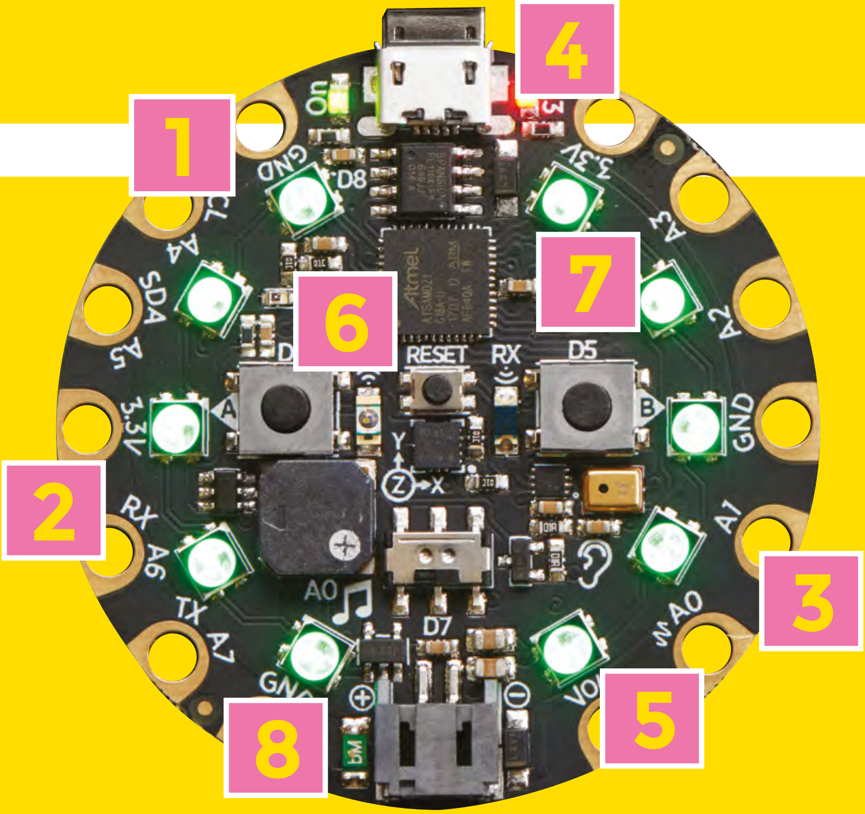
2 **Brains** The original Playground was based on the ATmega32U4, which is an 8-bit 32kB FLASH, 2kB RAM chip. It's a happy little chip but can only really run straight-compiled code like C / C++ / Arduino. But, around the time I was thinking about how I could improve it, I met Jacqueline Russell from Microsoft. She was the manager for the PXT (now MakeCode project) and really liked the Circuit Playground Classic, but MakeCode only supported ARM core chips. They were planning to support the ATSAMD21 family, and if I made a Playground with one, it would be straightforward to add support. Since I'd had experience with the chip as a core-Arduino-compatible in the Feather M0 line of development boards, it was not too hard to swap in the main processor chip. Then at the same time, MicroPython had been ported to the Micro:Bit and I was like "Wait, you can fit it into that small a chip? Well, the SAMD21 is about the same amount of memory, 256kB FLASH and 32kB RAM so let's try it!" And we were able to! So now I've got a chip that is a 'triple-threat' – it can be used with MakeCode, which is great for beginners, CircuitPython which is super fun and easy to learn, or as an Arduino with official support in the Arduino IDE when you're ready for C++!

3 **Lighting** Before there was NeoPixel, we had the Flora Pixel, which was a little sewable PCB with a chip on one side (WS2811, used for LED strips and signs) and a 5050 RGB LED on the other. They worked quite well and solved the

COMMUNICATION

We asked Limor why the CPX doesn't have any SPI pins. Was it that she doesn't think SPI is needed or didn't it fit into the design?

"Ahh, there is SPI, we just hide it. The SAMD21 has the concept of SERCOMs, which are configurable pins that can be I2C, UART or SPI. Now, as all engineers quickly find out, I2C and UART really need hardware peripheral support, they are a royal pain to bit-bang. But SPI? It's quite forgiving, you can always bit-bang it, if slowly. So, you can use any three pins, really, for SPI, or you can use one of the SERCOMs for hardware support. We just don't have to be so careful about forcing a SERCOM since bit-banging is an option…"



'how to get your clothing all blinky' problem. But they were annoying to manufacture. Eventually, the WS2812 was invented, which squeezes the WS2811 into the same package as the LED and, to differentiate it from the original 'Flora Pixel', NeoPixel was born. [Adafruit's] Resident Creative Engineer PaintYourDragon (Phil Burgess) coined this very memorable term!

4 Noise First off, the 8mm speaker with a class D amp is a total abuse of this little speaker. It's really designed for beeps only, but you can drive it a little hard and give a little audio playback device. The toughest part was finding a good tiny class D chip. We went with the PAM8301, which is a SOT-23-6 so very small, and has an enable line. The SAMD21 has a true analogue output, so pipe that in and Bob's yer uncle!

5 Accelerometer Ahh the LIS3DH, it's an accel and honestly you kinda need a three-axis one these days. Great for detection motion and tilt. It's not the cheapest accel but has some nice features: IRQ output, tap and double-tap detection, I2c or SPI support, free-fall detection, and adjustable ranges.

6 Temperature There's a built-in temperature sensor in the MCU, but I figured that it would always be a little hotter than the environment. A 10K thermistor is cheap (only two cents), so it's an easy 'YES'!

7 Light Like the thermistor, this is a very low-cost sensor, maybe 10 cents. I think light and temperature are sort of the minimum sensors to be able to do environmental sensing. This just happens to be the cheapest one I could find!

8 Microphone On the Playground classic, this is an analogue mic with an opamp. On the CPX, we got to take advantage of the I2S/PDM interface. PDM mics are very inexpensive and don't require an analogue opamp. But you have to get the data out via DMA and then perform a sinc filter to turn the digital PDM data into analogue. In Arduino you can even perform FFT to get frequency components – it's fun to have something that can react to music! →

INSIDE THE FACTORY

Limor explains how a bit of Adafruit hardware goes from idea to reality

I moved to NYC in 2006 to take part in an Artist/Engineer in Residence program at an art gallery in Chelsea. And, being lazy, I never left! I moved from a tiny apartment to a small apartment, to a larger apartment to, now over 12 years later, a 50 000 square foot factory with multiple floors in SoHo. It's been quite a journey! But, you may wonder, what on earth do we need all that space for…?

First, pretty much everything Adafruity happens here. We've ony got one location, so if you've ordered from our website, it'll ship from here. We do have a wide range of distributor shops though, so you can always order from your local electronics supplier.

THERE'S A LOT OF SET-UP EXPENSE WHEN OUTSOURCING. IT'S PERFECT WHEN YOU HAVE ONE OR TWO PRODUCTS, BUT WE MAKE 300 OR SO…

Say there's a fancy new sensor coming out… let's take the last one I did as an example, it's the SGP30 air quality sensor. We saw this on DigiKey's 'new products' webpage and the specs looked good, so I requested the Purchasing team to order up 10 samples and, after reading the datasheet, get to work designing a breakout board in PCB CAD software (we use EagleCAD & KiCad). That only takes maybe a few hours, we often have several designs in the works at once. On Wednesday we panellise up all the PCB and order prototypes.

A week or so later, the prototypes and sample chips come in, and the R&D team puts together the prototypes, hand-soldering all the components. We use fine tweezers, microscopes, syringes with paste, and hot air tools to do the job. It's good to make maybe three of each, usually one or two will make it through. Now we start 'bringing up' the design, soldering wires and headers from the breakout to a microcontroller, then writing C and/or Python code to communicate with the sensor, and learn its secrets! We'll go through this step a few times, revising until it feels done.

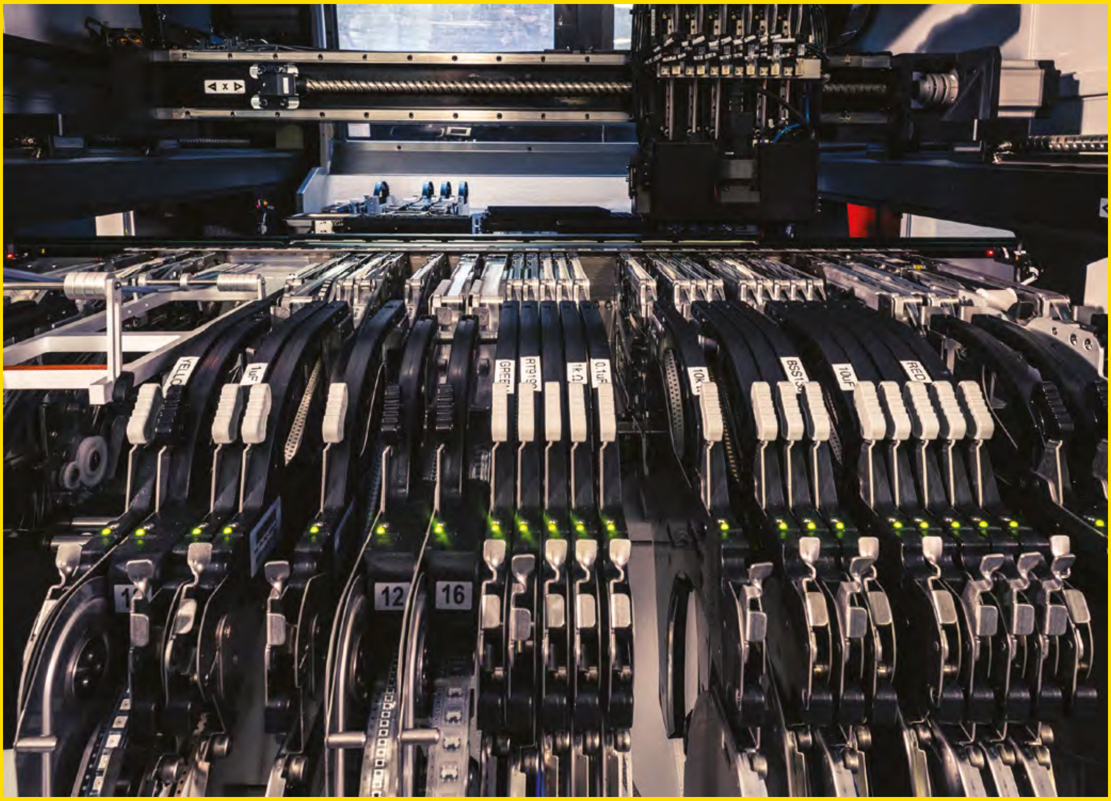
GETTING PHYSICAL

Okay, this sensor breakout design is good to go, only took two tries. Now it's time to manufacture! This is where we differ from most electronics design companies – it's much more common to do the design in-house but then outsource all the manufacturing. But that takes a lot of time, and we want to move fast. There's also a lot of set-up expense when outsourcing, so it's perfect when you have one or two products, but we make 300 or so, and you have to pay for all the changeovers. With the PCB design done, we panellise the circuit board. This way we manufacture 40 at a time, using our machinery. We'll order about 20 panels to start, so that's 800 total PCBs.

This is when the Fabrication group kicks in! The Fabrication Production Management team schedules a Work Order (WO) – let's start with 400 boards →

Right → **The pick and place machine populates bare PCBs with the components that give them life**

adafruit
SUNG
STOP
START
READY
RESET
FEEDER CHANGE
SAFEGUARD OVERRIDE
MAIN SWITCH



Left
Reels of components waiting to be loaded into the Pick and Place

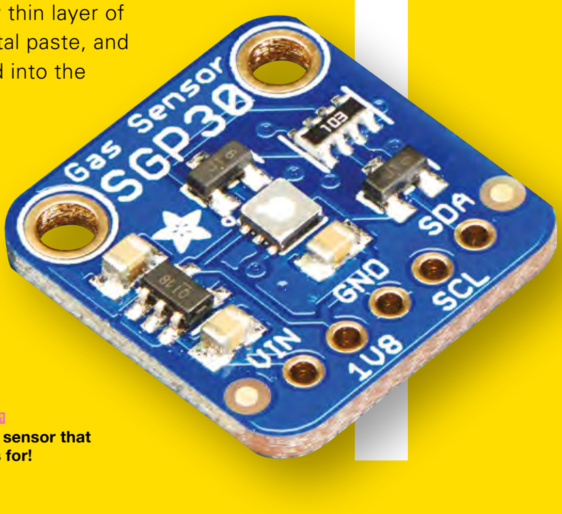
to start. Once all the SKUs for the work order are in stock, we get a notification to start the building process.

On the day of the build, the Machines team starts by loading the original CAD file for the board into the Pick and Place (P&P) machine. The P&P is the machine that will 'pick' up tiny little chips, LEDs, resistors, capacitors, and other parts, then 'place' them on the PCB. It needs to know what all the parts are and where they go – luckily you can just give it the CAD file and the software is very smart and will calculate all the movements.

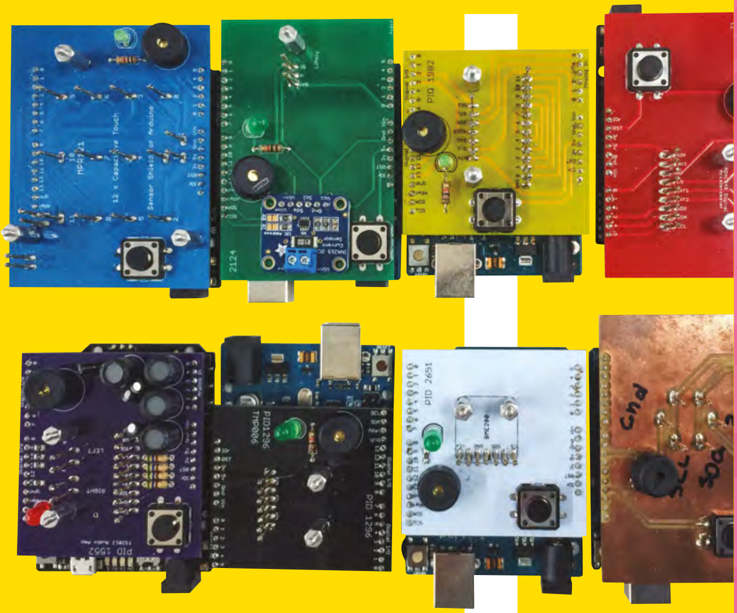
In order to pick the parts up consistently, we have to present them to the machine in reels. Each reel is just what is sounds like, a very, very long 'tape' with little pockets, each pocket containing a part. We try to schedule the work orders so that they share components throughout the day, minimising feeder change-overs. It's a little like doing an acrostic or sudoku puzzle!

Now, all programmed up, we're ready to rumble! We put the stack of panellised PCBs into a small destacking machine which will pop out one panel at a time, like a circuit-board PEZ dispenser. The panel is fed into the stencilling machine. This machine takes a laser-cut metal stencil (we have those manufactured in New Jersey right after we order the PCBs), and squeezes a thin layer of tin silver goo through the laser-cut holes. Each hole lines up with a metal pad on the PCB upon which a component will be placed. If you've ever done a spray-paint stencil, or silk-screened a T-shirt, its just like that but the goo is made of tiny little balls of metal so it looks like silver-grey icing.

After stencilling, each pad has a very thin layer of sticky metal paste, and is shuttled into the



Above
The new sensor that all this is for!



Above

Each product has a custom-built tester that flashes and beeps if the part is working properly

WHY ADAFRUIT WORKS THIS WAY

It's quite a journey, and it all happens in one building. It's not terribly common for all this kind of activity to occur within a few floors: usually design is in one location, manufacture in another, packaging can be in yet another factory, and shipping from a fulfilment centre. So why do it this way? Well, the biggest reason is speed. The maker and electronics marketplace moves very fast – new chips get announced and people want to use them ASAP! If we used outsourcing and external factories to manufacture our designs, we would have to insert six weeks into each step, as it just takes longer to schedule time on machinery when you're sharing it with others. Since we own all our equipment, we can manufacture the day we get parts, and put it into the shop within a few days. It means we can get new parts made faster, revisions done faster, and back in stock faster – every day we're not in stock is a lost sale.

There's also the benefit of better QA procedures. With thousands of parts shipped a day, there'll always be mix-ups or faulty parts – it happens! But, if your fulfilment centre is far away, it's really hard to check stock once a mistake is found, so QA takes days or weeks, instead of hours. At each step of production above, we have QA checks and verification steps, so we can find flaws early and fix them, rather than hoping for the best, and ending up with a bad yield or hundreds of order replacements.

Pick and Place. The P&P jumps into action, picking up dozens of parts a second, and with high speed cameras, checking the part and then placing it right on top of those metal-pasted pads. The paste is sticky, so once the part is pressed in place, it pretty much stays exactly there.

NO GOING BACK NOW

All the parts placed, there's a final checkpoint where a human uses their eyes to inspect the panel. This is the last point where we can still wipe off the paste and start over, so check that the components are right by looking at the markings! All good? Send it down to the oven! The oven is just like those toaster ovens at hotels and delis, where you put a bagel in the top and it goes through the body slowly and pops out the bottom complete. But this machine is 12 feet long, and heats up to 240 °C! Each board slowly goes through the many heating zones, slowing coming up to 240 °C, until the metal paste melts, the rosin and water evaporate away, and you're left with a permanent metal solder connection.

Now it's time for the Fabrication Test and Prep team! They'll break apart each PCB from the panel and use a tester to verify functionality. I design the tester early, around the time the PCBs are designed. It's usually built on a microcontroller and does one thing only, which is communicate with the part, and verify that it's functioning correctly. For this sensor, I would check that it communicates over I2C, and maybe verify the temperature and air quality readings. On success, it makes a beep and turns a green LED on to indicate success. We get about 97% yield, which means some boards don't make it – maybe they didn't melt enough, or one of the parts got shifted out of place. The Rework team will try to fix the board, if it makes sense, and then put it back through test procedure.

That's how Adafruit builds hardware, but everyone's got their own processes. Let us know how you build yours – whether it's one piece or 1000 – at **@Hackspacemag**. □

Below

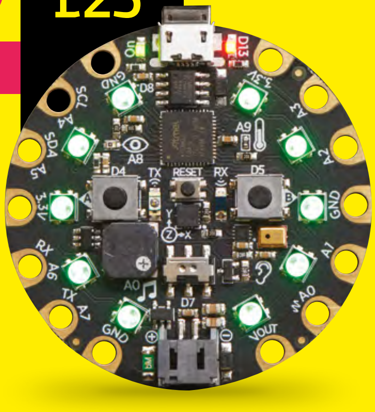
Solder paste ready to be spread onto the waiting PCBs

adafruit

FREE CIRCUIT PLAYGROUND EXPRESS

WORTH £25

WITH 12-MONTH PRINT SUBSCRIPTION



MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
April 2018
Issue #05
RASPBERRY PI 3B+
Faster processor & networking
DREMEL HACKING
Turn your hand tool into a table saw
RESIN.IO
Control all your smart devices with one tool
LED FASHION
Bring your accessories to life with LEDs
LADYADA
LIMOR FRIED REVEALS ADAFRUIT'S SECRETS TO MAKING GREAT HARDWARE
LITHOPHANES CHEERLIGHTS PENCILS NEURAL NETWORKS

FROM JUST £55

12-month subscription from £55:

→ **UK:** £55 per year

→ **EU:** £80 per year

→ **US:** £90 per year

→ **RoW:** £95 per year

Visit: hsmag.cc/subscribe

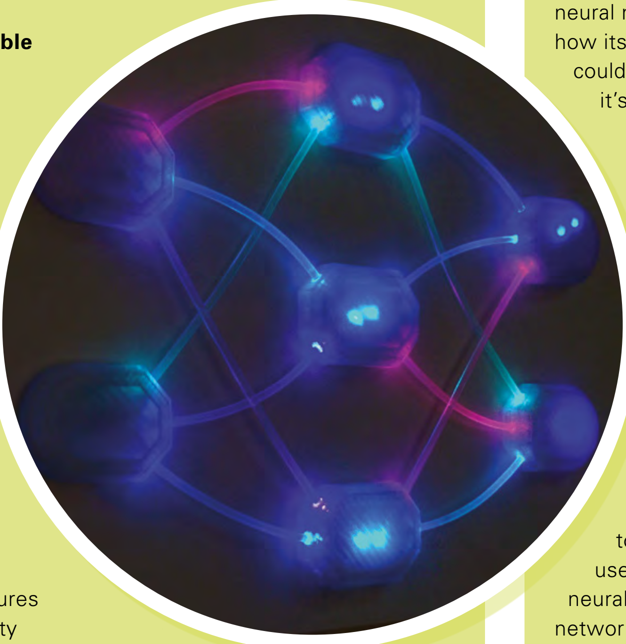
How I Made NEURAL NETWORK

Learning how we learn with robotic brains

By Zack Akil

Public interest in artificial intelligence (AI) is soaring higher than ever thanks to movies and news stories about incredible applications of the technology. The specific technique utilised in a lot of the applications we hear about is a type of machine-learning algorithm known as a neural network (a biologically inspired machine-learning algorithm that is capable of learning complex non-linear patterns). Even though these algorithms already have an aura of science-fiction about them, Hollywood has been a bit guilty of portraying the training and using of these algorithms to be more visually appealing than it actually is in real life. They tend to show intense organic fibrous structures of light to visualise AI, when in reality a machine-learning engineer will get a command-line interface (or a line graph if they are feeling particularly extravagant).

I wanted to make something that is closer to the movie representation of AI, partly because it's a lot more visually appealing than the command line, and partly because neural networks are infamously black box (see 'Black Box Algorithms'). Making a visualisation of a neural network that intuitively lets you see how its internal configuration is adapting could give some extra insight into how it's actually learning.



What I built was a physical fibre-optic neural network. As a network is being trained, the fibre optic glows different colours to represent the internal state of the neural network. The colour of the light represents the internal weights being optimised. The RGB neural network hangs on my wall and communicates with my laptop over the local WiFi network. On my laptop I have standard machine-learning tools and libraries that I normally use. When I want to use the RGB neural network for visually inspecting a network, I just import a special library and add one line of code into my training script which takes the current state of the neural network on my laptop and sends it to the RGB neural network on the wall. I'm then treated to a visual symphony of fluctuating light that is a one-to-one mapping of my neural network's state.

Above
Toggle switches add a more tactile experience than push-buttons

Below
Neural network model mapped to RGB colours

WHAT THE LIGHTS REPRESENT

Within a neural network there are connections between the nodes at each layer. When the neural network is making a prediction, the input values propagate through these connections. Each connection will multiply the value fed in at one end by some coefficient. It's this coefficient that is learnt during the training of the neural network, and it is the value of this coefficient that is represented by the RGB light in the fibres. At each node in a neural network (except the input nodes) there is also a constant value, known as the bias or intercept, that is optimised during training. The RGB neural network represents these values by the colour of the node's body.

SYSTEM ARCHITECTURE

At the heart of the project is a Raspberry Pi Zero W which runs a small Python server that receives web requests for what LEDs to change. There is very little code in the RGB neural network itself: most of the project's complexity is in the special Python library developed that plugs into my day-to-day machine-learning stack. This library is responsible for extracting the internal values from a neural network model, mapping them to RGB values along with the specific LEDs they map to, and then send off the requests to the RGB neural network. The RGB neural network can be aptly described as just an internet-connected LED array.

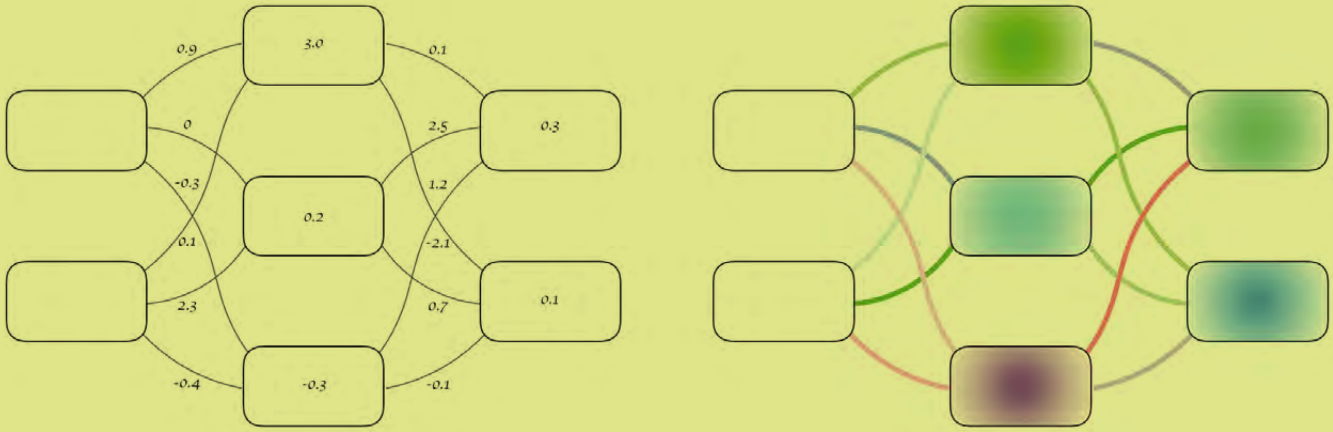
When I started building the project, I already had an addressable RGB LED strip in my inventory. The specific model of the LED strip was WS2813, and at the time I could not find a library that would allow the Raspberry Pi to control it. To overcome this, I used the FastLed library with an Arduino and connected that to the Raspberry Pi using a USB. The Raspberry Pi can then change the LEDs by sending serial commands to the Arduino. The Arduino is loaded with a small program that reads messages from the serial connection and parses them into actions to send to the LED strip. In a future version I'd like to replace the Arduino-Raspberry Pi combo within a single control board; for example, the Arduino Yún combines the robust hardware compatibility of an Arduino and the robust web interfacing of Linux.

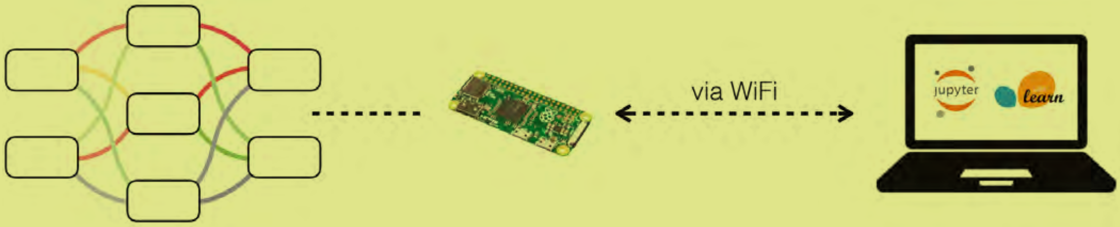
The Raspberry Pi runs a small Python Flask app that accepts web requests for which colours to change the LEDs to, and relays those commands onto the Arduino. The body of the web request sent to the Raspberry Pi is JSON, containing a list of objects describing the RGB values and the number describing which LED to change.

→

Black Box Algorithms

When it's hard for a human to examine and extract out the insights that a machine learning algorithm is finding in the data, it is called a 'black box' algorithm.





Left
RGB neural network system diagram

LED MAPPING

One of the more tedious parts of the project was writing the code that maps a weight in the neural network being trained on my laptop to the LEDs it corresponds to on the RGB neural network. I say 'LEDs' because a single weight maps to multiple LEDs. The reason for this is because I wanted the fibre-optic tubes to glow as bright and as uniformly as possible. To achieve this there needs to be a light source at both ends of the fibre-optic tubes. There are also multiple LEDs in the body of the nodes to increase their brightness, as shown in the LED map above. One thing that is also shown in the LED map is that there are redundant LEDs that are not linked to any fibre optic tubing e.g. LEDs 11 and 16 on the diagram. This is because I wanted to keep the building process simple and just make multiples of the exact same component, even if it meant wasting a couple of LEDs.

> Sending serial messages to an Arduino from a Raspberry Pi is surprising painless

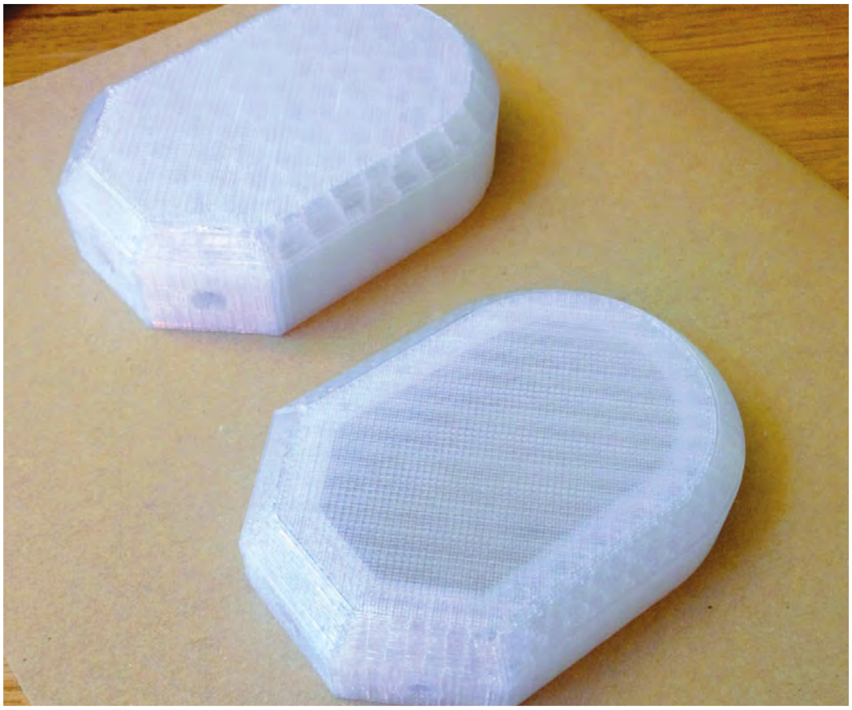
THE NEURAL NETWORK LIBRARY

The software used to train the neural network on the laptop is 'Sci-Kit Learn', which is a widely used open source machine learning Python package. The whole package follows the ethos of Python and it is therefore easy to extend and inspect the objects created by the package. Extracting out the weights of a neural network is done by calling '.coefs_' on the neural network object, which will return a list of the weights for each layer in the neural network. Mapping a single number value to an RGB colour is handled by a Python library called 'colorsys'. The coloursys library has the functionality to convert HSV (hue saturation value) to RGB, so I just create a new HSV colour, using the weight as the hue value, and converted that value to RGB using the coloursys.hsv_to_rgb function. To give more flexibility in the choice of colour mapping, I could have spent time integrating compatibility for the colour mapping presents from matplotlib, which is a widely used Python chart-plotting library. With the current system, the colour map is hard-coded to be red for low values, blue for close to zero values, and green for high values.

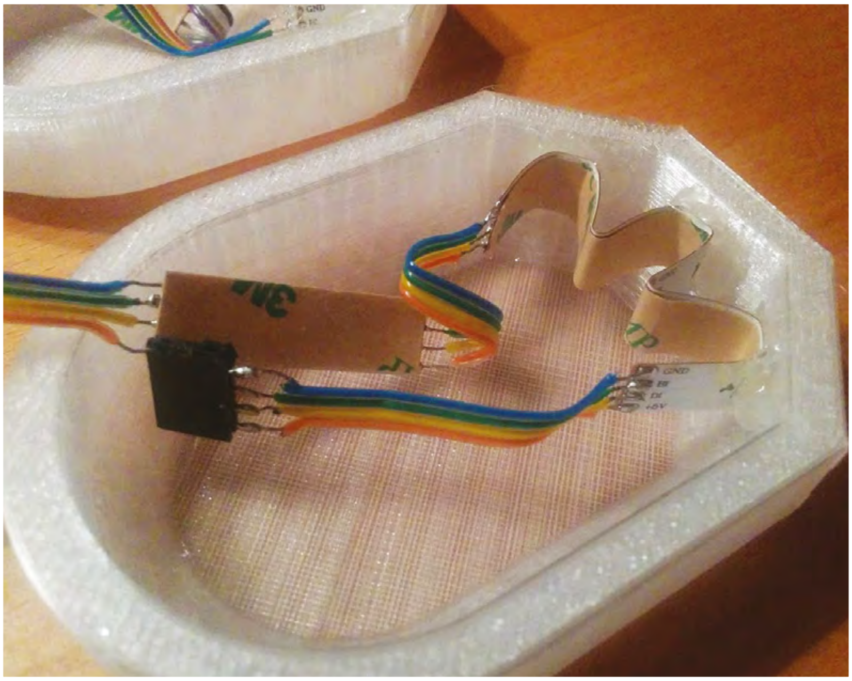
HARDWARE

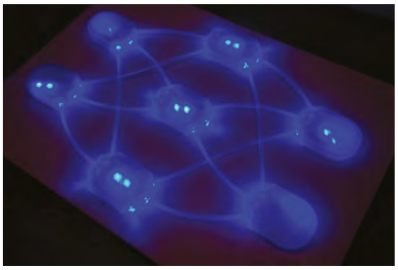
The physical nodes are 3D printed in transparent PETG plastic. I designed and printed multiple versions of the node in order to find the best one for letting light shine through. The final version was printed with just three solid layers of plastic on the top window section, which lets plenty of light diffuse through whilst still holding its shape.

The LEDs were cut from the original strip and mounted inside each node with hot glue. In an effort to make my life in



Left
First iteration of node design, bottom – final design with thin body window

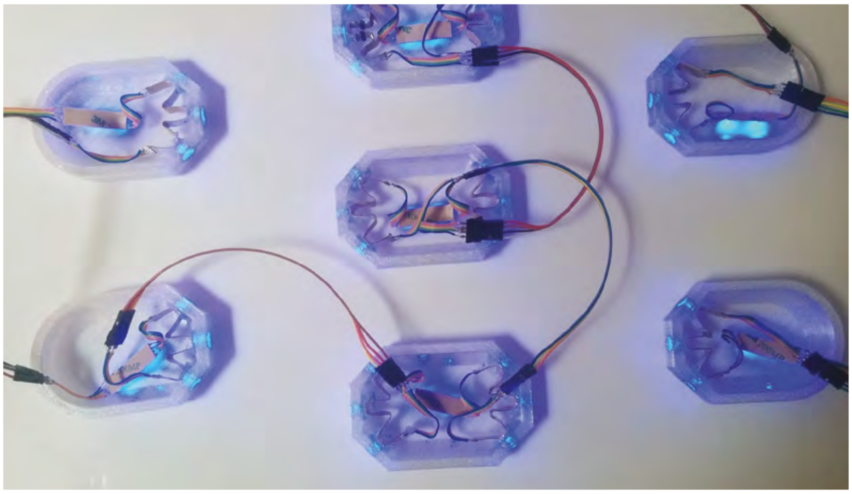




Above
All nodes connected with fibre optic

Left
Node fitted with addressable LEDs

Below
Nodes daisy-chained for testing



assembling and debugging the system easier, I soldered each node with female and male header pins. This allowed me to quickly test each node in isolation and, because they are all addressable LED segments, I could progressively daisy-chain all of the nodes together as I was building to test the system.

The final step of the project was just to stick all of the nodes to some poster board in the formation I wanted and then, using a razor, to cut the fibre optic tubing to size. The specific type of fibre optic I used is known as side glow fibre optic. The side glow aspect of the fibre optic means that, when you shine light through it, it emits the light along the length of the fibre. This makes it perfect for sci-fi props and neural network visualisations.

HOW TO SEND MESSAGES BETWEEN RASPBERRY PI AND ARDUINO

Sending serial messages to an Arduino from a Raspberry Pi is surprisingly painless. It just requires you to connect them via a USB and install the Arduino library to the Raspberry Pi.

Then, in Python on the Raspberry Pi, you can send serial messages to the connected Arduino using the 'pyserial' library.

Then it's the case on the Arduino side to read from the serial the messages sent and parse them into commands for the LED strip. I used a 12-character message to describe which LED to change to which colour.

The Arduino would parse out the values and, using the FastLED library, prep the desired LED in the strip to then be ready to change to the desired colour. Then, after I've sent a message for each LED I want to change, I send a single character message (e.g '0/n') to the Arduino and that actually changes all of the LEDs to their new colours. □

Flask

Flask is a lean Python library that lets you build web APIs with a few lines of code. Services built can run locally in your private network or be uploaded to a server hosting service for the world to use.

FEATURE

Making the WORLD Accessible

The Enable Makeathon aims to help 20 million disabled people living in rural India

Cameron Norris

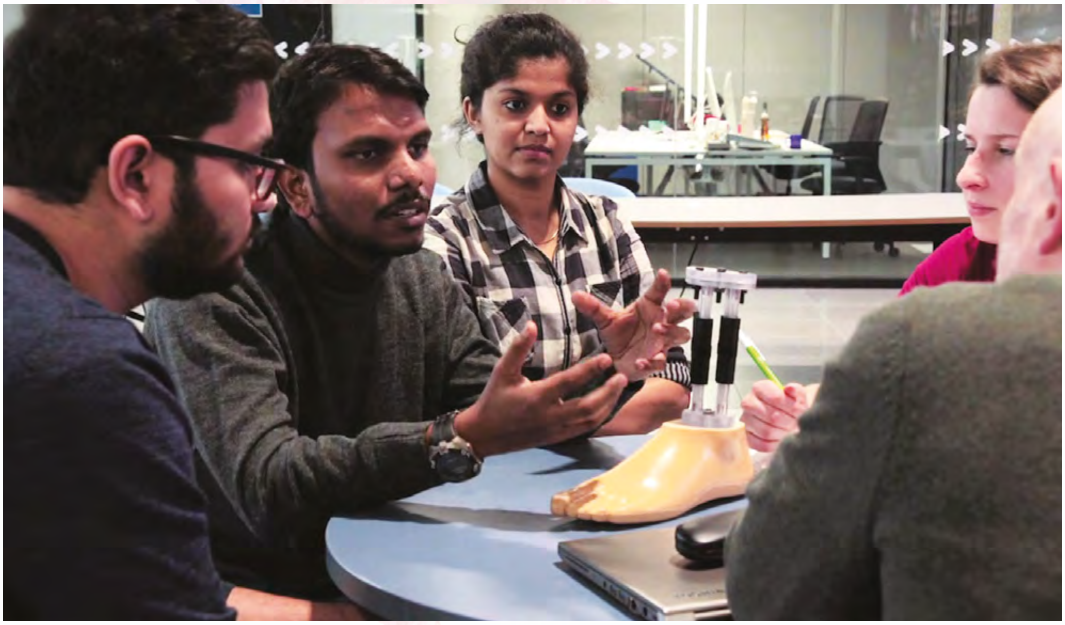
@cameronsnorris

Cameron is a technology and communications specialist, passionate about the use of open source hardware for social innovation. He has spent the last four years providing community building and digital marketing expertise to tech startups, while developing new technologies to improve the lives of disabled people from low-income and rural communities in developing countries.

India has seen phenomenal growth of the maker movement over the last decade, with the community garnering support from universities, non-profits, corporations, and the government. Hundreds of makerspaces and hackathons have appeared across the country, helping makers to pursue their ideas with enthusiasm. In addition to this, social entrepreneurship has attracted ever-growing amounts of talent, money, and attention – with extraordinary people coming up with brilliant new products and services that dramatically improve people's lives, while driving social change that creates a lasting, transformational benefit to society.

The Enable Makeathon is an intensive innovation programme set up by the International Committee of the Red Cross (ICRC), supported by the Global Disability Innovation Hub and other partners, to develop affordable solutions to the challenges faced by persons with disabilities in India, with a particular focus given to those living in rural areas.

The programme aims to crowdsource innovative new ideas in assistive devices, while sling-shotting the most promising concepts from ideation to action through a co-creation camp with mentors, experts, and makers to develop a solid business plan and prototype in just 90 days.



Right Good products are the result of good collaboration

Credit © R&A Collaborations

Worldwide assistance

Disabled people in more than 48 countries have benefited from the ICRC's physical rehabilitation services and the provision of over 1.4 million assistive devices, including repair and maintenance, since 1979.

Above
Build, tweak, improve, repeat

Credit
© R&A Collaborations

The ICRC supports a large number of people with disabilities across the world, and has active programmes in more than 40 countries. Most of these people, including thousands of children, are from low-income and rural communities. ICRC set up the Enable Makeathon to bring together humanitarians, startups, engineers, academics, makers, and people with disabilities on one platform, to co-create products suitable for scaling up and manufacturing for use across the globe.

COLLABORATION COLLABORATION COLLABORATION

Usually working in partnership with the National Red Cross and Red Crescent Societies, the ICRC provides humanitarian aid and expertise, most notably in the areas of emergency response, international humanitarian law, orthopaedic services, as well as restoration of family links, water, and habitat. The ICRC has a proven record and long history being able to reach and meet the needs of vulnerable persons in Asia.

"Enable Makeathon is a unique social movement. It brings together a multidisciplinary team of experts with persons with disability to find solutions for the real challenges that they would like to address." Tarun Sarwal, Head of Innovation at ICRC.

The Enable Makeathon 2.0 co-creation camps recently took place in two simultaneous locations; Bengaluru, India hosted by the ICRC, and London, United Kingdom organised by the Global Disability Innovation Hub and University College London. 15 teams, selected from over 100 applications, competed against each other for grants and the opportunity to join a one-year incubator programme to further develop their solutions with support from industry leaders, mentors, and disabled people, in order to create impactful low-cost solutions with the potential for global impact. A jury selected ten teams from India and five teams from the United Kingdom to tackle 12 priority challenges related to disability issues, including access to education, access to tourism, and access to transport. Bengaluru co-creation camp attendee, Deepa Malik, who won Silver at the 2016 Summer Paralympics said, "It is important that stigma around persons with disabilities is removed. Social inclusion in every sphere of life is important to ensure that those with disabilities progress."

Enable Makeathon is a unique social movement. It brings together a multidisciplinary team of experts with persons with disability to find solutions for the real challenges

A panel of judges, which included representatives from the National Institute of Design, announced the three winners at a ceremony in Delhi on 6 February 2018. Bleetech from India emerged the overall winners with their low-cost encyclopaedia for the hearing-impaired, which enables persons with hearing impairment to ask questions and receive responses in sign language using their mobile phone. →

Left
Team GameAble's gesture recognition device makes computer games accessible for more people

Credit
© R&A Collaborations

The Previous Edition

The first edition of the Enable Makeathon was also a success, with 186 applications being filtered down to 32 teams, eventually leading to 15 prototypes. The competition resulted in five solutions selected for the finals, and three grants provided to the top three teams. The first prize went to Mobility India, who designed a low-cost standing frame to help children with cerebral palsy correct their posture. Other winning ideas included a wheelchair attachment to increase mobility over uneven terrain, as well as a low-cost, energy-storing 3D printable prosthetic foot.

"From what I have seen so far, I don't think anybody could have obtained such a wide range of viable solutions for such little money," says Tarun Sarwal, ICRC's Head of Innovation. "We want to try to find ways to do this not just for disability but also in the areas of water and sanitation for displaced people," he says."If you can solve local problems in India, you are creating solutions for another billion end-users outside India as well."

Not just another hackathon...

And if you think that the Enable Makeathon is just another event where the winners and participants are gone and forgotten the following year, you'd be wrong. Participants of the first Enable Makeathon have gone on to successfully test their finished products with patients, and continue to receive guidance and support from program mentors and supporting organisations.

One such organisation, the Global Disability Innovation (GDI) Hub, is a research, teaching and innovation centre based in the Queen Elizabeth Olympic Park in London, where the Olympic and Paralympic games were held in 2012. The GDI Hub is partnered with University College London to promote collaboration between governments, researchers, entrepreneurs, local communities, and non-profits to promote accessibility and inclusion and remove barriers faced by disabled people.

INCLUSION AND IMPACT!

Jeremy England, Head of the ICRC's Regional Delegation for Bhutan, India, Nepal, and Maldives, explained in his welcoming speech, "We're excited about what the teams have come up with, and we can see them really making a difference with their innovative solutions. Inclusion, collaboration, and impact are at the heart of the Enable Makeathon. It is about looking for solutions with persons with disabilities".

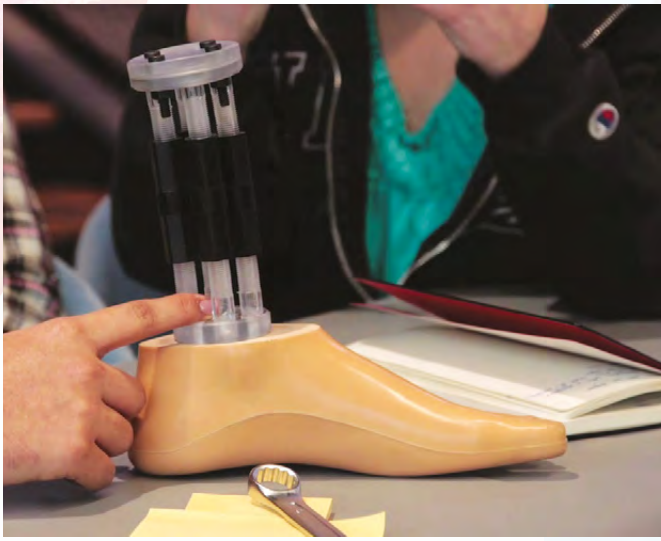
Amputees often have to wait up to two years or more to receive a custom-made prosthesis

Team GameAble from the UK came second for their gesture recognition-based control software that enables persons with disabilities to play video games, while Team Nonspec from the United States emerged third for their light-weight, low-cost, rapidly adjustable below-knee prosthetic system.

Nonspec co-founder, Erin Keaney, explained how "there are approximately 54 million amputees worldwide", and although the majority of them do have access to clinicians, there simply isn't enough time or resources to provide every amputee with an appropriate assistive device. Amputees often have to wait up to two years or more to receive a custom-made prosthesis, as designing and fabricating the best possible prosthesis and socket for each amputee is incredibly time-consuming. Furthermore, the first fitting is often uncomfortable and requires the patient to repeatedly return to their clinic to continuously make adjustments until the discomfort subsides.

In response, Nonspec has developed an affordable and easily adjustable prosthetic limb kit for amputees worldwide. The team explains how they are different from existing solutions because their 'off the shelf' kit can be adjusted in an hour or less by a minimally trained clinician. The kit also allows the same components to be used by patients of eight years old and above, expanding as required to compensate for growth, while maintaining comfort. This unique ability to adjust the prosthetic limb's length increases the usable life from six–eight months up to four years.

At the core of the device is a 'pylon' system, based on the same aerospace technology that is used in helicopter rotors. The 'pylons' can be adjusted to accommodate for any patient height, as well as simple daily changes, such as footwear, so that patients are no longer required to have repeated fittings or "different legs just because they switched from trainers to a nice pair of shoes", explained Nonspec co-founder, Jonathan De Alderete. "The Enable Makeathon had two prosthetics groups, which allowed us to learn from one another and figure out ways that we can work together moving forward to meet patients' needs."



Right ◈
Nonspec team's prosthetic lasts up to eight times longer than traditional prosthetics

Credit
© R&A Collaborations

Left Working hard doesn't mean you shouldn't be enjoying yourself

Below Aesthetics can be an important part of functionality, as it can help children feel more comfortable using the devices

NOBODY GETS LEFT BEHIND

Carley Booker from the GameAble development team explained how there are over 600 000 children in the UK with fine motor control issues, preventing them from being able to play computer games with their friends. "That's one in twenty children in the UK who can't play computer games with their friends. Our product can help change that by turning large movements into keyboard presses, so children without fine motor control aren't left out from play."

25-year-old Nupura of Bleetech explained how, "one of the biggest hindrances people with hearing disabilities face on a regular basis is inaccessible knowledge." Through their app, Nupura, Jahnavi, and Bleetech team provide answers to questions in Indian Sign Language. "The queries we receive are spread across a plethora of subjects. From general knowledge and politics to current affairs and English learning, we're asked anything and everything," explains Nupura. "And we ensure that we respond to every query."

Nupura and Jahnavi are already planning their next steps: "we aim to collaborate with corporate companies across the banking, healthcare, and e-commerce sectors, and make their content accessible for our audience," says Nupura.

During the co-creation event in London, Iain McKinnon, Head of Inclusive Design and Co-founder of the Global Disability Innovation (GDI) Hub, described how millions of people remain in need of better quality assistive devices and that people in low-income countries were at a particular disadvantage. "There is no reason that this should be the case, every individual should have the assistive devices required to meet their needs," he said. "The challenge is doing that in a way that is appropriate, affordable, and accessible. That's what this program is all about."

"By having a competition-based process where you filter out a lot of applications to get the best quality ones you can, you can put them in a very intensive period with a lot of support to get them to a stage where they can deliver. It can seem like quite a harsh filtration process, but that's the only way you can do it to get the best ideas and the best chance of success. The exciting thing about the Enable Makeathon is the encouragement of grassroots innovation and getting many more people to realise their ideas can make a difference."

You can find out more at **enablemakeathon.org**. □

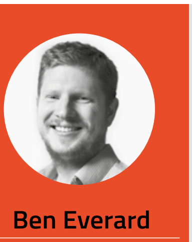
Voices

"The vision of the GDI Hub is to build a movement to accelerate disability Innovation, and the Enable Makeathon is exactly the type of partnership and programme that will achieve this."

Cathy Holloway
Academic Director, Global Disability Innovation Hub

The Raspberry Pi 3B+

With more power comes more possibilities



Ben Everard

@ben_everard

Ben has a secret stash of every model of Raspberry Pi. He hopes they keep making them this small because he's running out of space.

There's a new Pi in town: the Raspberry Pi 3 Model B+. It's an upgrade to the previous version 3 Model B. For followers of this little board, its form factor will come as no surprise – it's the layout that's remained more or less unchanged for almost four years. However, within this form, there's some important developments.

The most obvious changes on the new board are the shiny bits of metal hiding a couple of chips. The largest of these covers the System on a Chip (SoC). While this contains the same quad-core processor as on the Pi 3B, there's a change in the design of the substrate, and more efficient heat dispersal through the heat spreader. These improvements mean it's now clocked at 1.4 GHz (up from 1.2 GHz on the Pi 3B).

The second big change is also hidden behind metal. The Pi logo conceals a CY43455 chip, which controls the wireless networking. This chip works with both 2.4 GHz and 5 GHz networking (802.11 b/g/n/ac), meaning that the Pi 3B+ can connect to a wider range of WiFi networks than earlier generations, as this allows you to avoid the more congested 2.4 GHz range, giving you the potential for a network speed-up (take a look at the WiFi Channels boxout for more details).

The third big change is the introduction of Gigabit Ethernet thanks to a Microchip LAN7515 controller. This network connection runs over the board's internel USB 2.0 bus which limits the top speed, but if your networks router and other hardware supports Gigabit Ethernet you should find that the new board runs up to three times faster than the previous generation of Raspberry Pi.

These new specs give a snappier performance to the Pi as a general computer, and the 17% speed-up on the processor may well be enough on its own to convince people using the Pi as a desktop to part with $35 to upgrade. However, we're more concerned with what the new Pi means to our makes.

The GPIO connection remains unchanged, so as far as physical connections go, there's no difference, and any wiring you could do on the previous Pi will still work here.

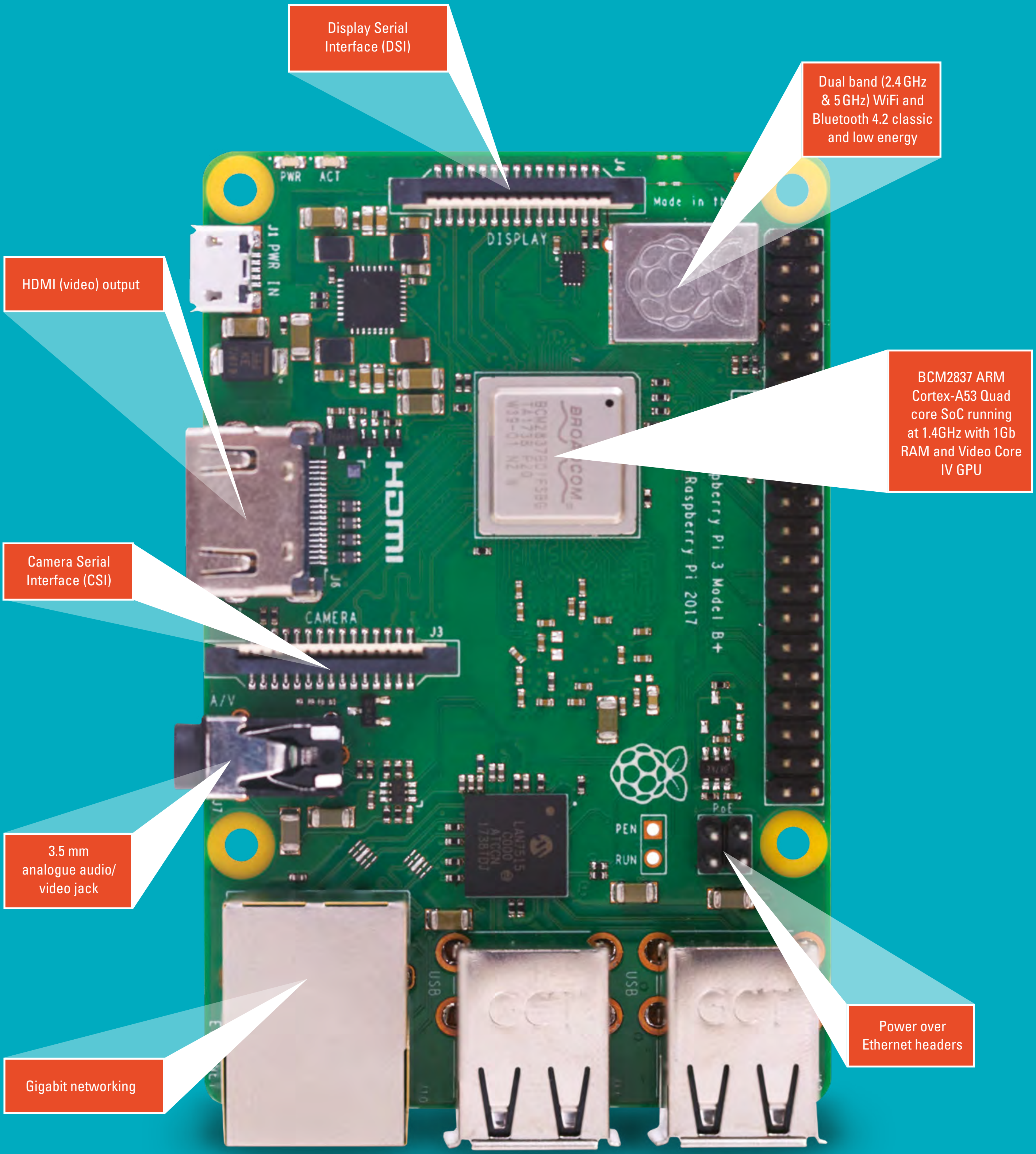
The extra horsepower will come in useful if you need a computationally intensive brain for your make (image processing or artificial intelligence, for example). Take a look at the next few pages for ideas on how to put the extra power to good use. →

WIFI CHANNELS

WiFi networks run on a 'channel' or small range of frequencies within a particular band. Originally, WiFi ran in 14 overlapping 40 MHz channels between 2412 MHz and 2484 MHz. With all the various WiFi networks in the area competing to use this fairly narrow frequency range, there can be quite a lot of interference. In network terms, interference means slow speeds. Throw in the fact that microwaves and cordless phones also use a similar frequency range and, in a busy area like an office or apartment block, the result can be a dramatic slow down of the network.

The solution to this overcrowding is to open up more frequencies for WiFi networks, and this is where 5 GHz comes in. There's less demand for frequencies this high, so there are more channels available – 23 non-overlapping channels in fact. This means less overcrowding.

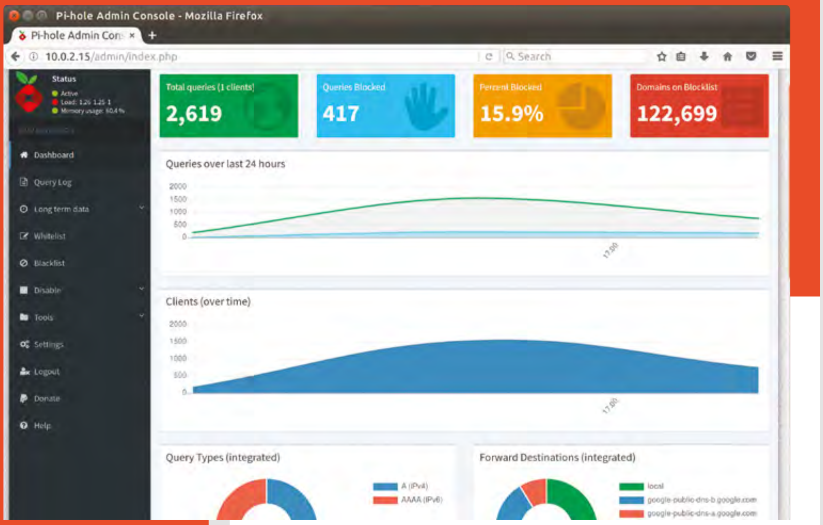
There is a slight complication in that you can't just double the frequency and expect the properties of the signal to be the same. The biggest difference most people will experience from the different frequencies is that 2.4 GHz is likely to transmit further than 5 GHz. We say likely because there are no hard and fast rules for WiFi range, and it depends on physical objects, atmospheric conditions, and a myriad of other conditions.



Display Serial Interface (DSI)
Dual band (2.4 GHz & 5 GHz) WiFi and Bluetooth 4.2 classic and low energy
HDMI (video) output
BCM2837 ARM Cortex-A53 Quad core SoC running at 1.4GHz with 1Gb RAM and Video Core IV GPU
Camera Serial Interface (CSI)
3.5 mm analogue audio/ video jack
Power over Ethernet headers
Gigabit networking

Pi-hole

Make the most of the 3B+'s faster networking



Total queries (1 clients) 2,619

Queries Blocked 417

Percent Blocked 15.9%

Domains on Blocklist 122,699

Queries over last 24 hours
2000
1500
1000
500
0
17:00
Clients (over time)
2000
1500
1000
500
0
17:00
Query Types (integrated)
A (IPv4)
AAAA (IPv6)
Forward Destinations (integrated)
local
google-public-dns-b.google.com
google-public-dns-a.google.com

Above Point your web browser to the IP address of your Pi-hole and you can see graphs of how many ads you've blocked

Most people these days run a home network with a router distributing a single internet connection to all their devices, from mobile phones to TVs to computers. Most internet packages come with a router that does all the basics to get your house online. However, you don't have to blindly use the default settings you get from your internet provider, and with a little of your own infrastructure you can control how your devices access the internet. One such option is to run your own ad-blocking DNS server, such as Pi-hole. While that may not sound that exciting, it does one important thing: it allows you to block web adverts on all your devices with only one setting.

The DNS provider is what converts a domain name (such as **hsmag.cc**) into the IP address which uniquely identifies the computer that serves up the relevant content for that website. Most internet adverts are served up by domains associated with particular advertising agencies. Pi-hole does this converting of domains to IP addresses normally, except any time a website requests content from a domain flagged as belonging to an advertiser, when Pi-hole simply serves up blank content.

The result is that all devices connected to this network get ad blocking without any additional software needing to be installed. Given that such a large proportion of internet traffic can be adverts, using Pi-hole can speed up your internet browsing speeds on any Pi. However, the new model should give you DNS results quicker, especially if you live in an area where there's a lot of traffic on the 2.4 GHz WiFi network. □

HOME SERVER

The low power usage of the Raspberry Pi makes it a great option for a home server and can run a lot more than Pi-hole. A few of our favourites are:

Nextcloud

Nextcloud is a suite of tools for collaboration and sharing. There's email, file synchronisation, web-based document editing (via Collabora Office Online), calendars, and more. All this can run on a Raspberry Pi, and allows you to take control of your own data.

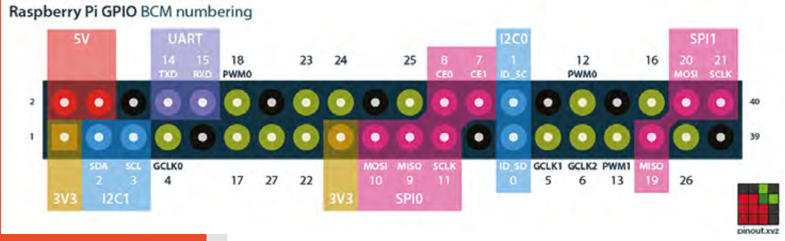
FreeNAS / OpenMediaVault

Network-attached storage (NAS) systems provide shared disk space available to all computers on your network. It's a little bit like Dropbox or Google Drive but run on computers you control. There are a few options for this on the Raspberry Pi, but FreeNAS and OpenMediaVault are a couple of the most popular.

Web server

You can run most modern web software on a Raspberry Pi, so anything you want to put on the web can probably be run off a Raspberry Pi provided the number of visitors doesn't go too high. You can even run this from your home network, as long as you set up port forwarding on your router, and use a dynamic DNS system if necessary. If you do make it publicly available though, you'll need to ensure that you properly secure the Pi to protect your network.

Below The pinout.xyz website has an interactive guide to GPIO pinouts for the new Raspberry Pi



RetroPie

Get the extra processing power playing games

Most maker projects don't tax the processor too much, but retro computing is one of the exceptions. Emulating hardware is a complex process, and it can take a surprising amount of computing power to recreate the effects of older hardware. The new Pi's extra clock speed should give retro-gamers a little more headroom for running older games and adding effects.

Many retro computer emulators are open source and can be compiled on the Raspberry Pi; however the easiest way to get started is with RetroPie. This is a software package (also available as an SD card image) that bundles together emulators for 52 classic games consoles, including the best games system of all time, the Super Nintendo Entertainment System [rather than the Commodore 64, as has been erroneously reported elsewhere], and gives them an easy to use interface.

You'll need a controller, and RetroPie can work with most USB game pads, or you can build your own. Essentially, a controller is just a set of buttons and the circuitry needed to translate these into commands that your computer understands (usually via USB). You can build a one-off microcontroller board, or use a specially designed controller such as PetRockBlock's GamepadBlock, which does most of the work, leaving you to design the housing and attach the buttons.

While it's not essential to build your system into a period-accurate case – maybe by utilising the casing from an old console, or perhaps an arcade cabinet – it does add a certain extra pleasure to do so. □

ROMS AND LEGALITIES

Playing older games can put you in a legally ambiguous position. While there are vast packs of ROMs (game files) available online, these are usually posted without the copyright-holder's consent. Legal jurisdictions vary and most of the issues around playing old games exist in a legal grey area of 'Fair Use' in copyright law that's not been tested in court.

You can still play retro-style games without having to worry about the intricacies of copyright law by playing games that have been released for emulators. You can find a wide range of such games at **pdroms.de.**

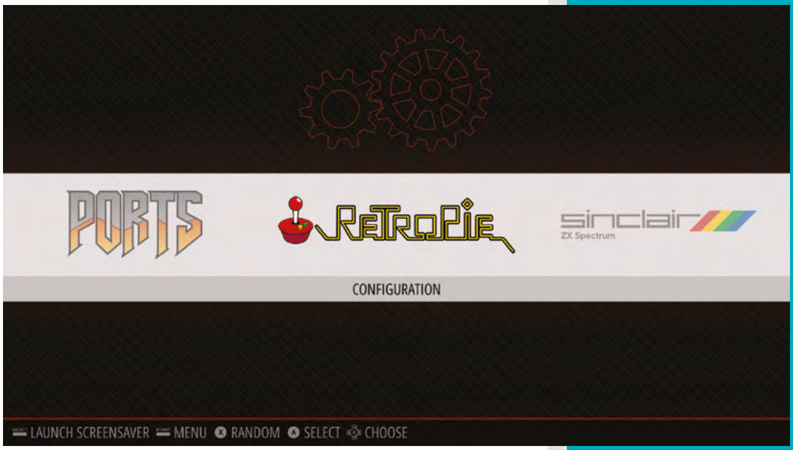
Perhaps the best technical solution to this legal quandary is to read the games directly from the cartridges. You can get a device called a Retrode, which lets you read console cartridges via USB from **dragonbox.de** (they were out of stock at the time of writing, but a new shipment should have arrived by the time you read this).

Many old games cartridge connectors are simply address buses to the memory, and you can read in the game data bit-by-bit as you need it. For example, there's code and circuitry to read SNES roms on a Raspberry Pi here: **hsmag.cc/DxWffc**.

The most challenging bit is usually physically connecting to the cartridge, as this can often require the original connector desoldered from an old console. This level of hacking can feel a little sacrilegious, but can mean that your game collection will find a new home.

Above ⇱
It's not essential to disguise your Raspberry Pi as a retro computer, but it does feel good

Below ⇲
The RetroPie display is easy to navigate, even without a keyboard or mouse plugged into the Pi



HackSpace magazine meets...

RUTH AMOS

Engineer, entrepreneur, and custard superhero

Take two engineers, add the creativity of countless 5–11-year-olds, then sit back and watch the fun. That's Kids Invent Stuff.

Kids Invent Stuff is a YouTube channel that does exactly what it sounds like it should: it's a showcase for kids to show off their ideas. Best of all, the ideas get turned into real gadgets built by two real, grown-up, professional engineers: Ruth Amos and Shawn Brown.

We spoke to one half of this team, Ruth Amos (who, like Shawn, is a former winner of the Young Engineer of the Year Award), to find out what the hardest thing is about building kids' creations, why it wouldn't work with adults' ideas, and the ridiculous fun of using an angle grinder. →

Above

Time and space are difficult, so we only got to talk to Ruth. Sorry Shawn!

HACKSPACE Kids Invent Stuff is such a brilliant idea that it's hard to imagine nobody had thought of it before. How did you get started bringing the inventions of children to life?

RUTH AMOS First, we're not the only ones doing this: Dominic Wilcox has Little Inventors (**www.littleinventors.org**). They do projects and then he has a team of designers who bring the inventions to life. We weren't aware of what Dominic was doing until after we started Kids Invent Stuff, but anything that encourages kids to invent is fantastic.

I personally fell into engineering. Growing up I wanted to be a barrister. Part of my GCSE Resistant Materials project, I invented something called the StairSteady, which is a specialist handrail to help people walk up and down their own stairs. With that product I won the title of Young Engineer of the Year, which totally changed my career path. Part of that is working on different STEM initiatives, and being a trustee of a charity that supports engineering, and working with the Government on getting more women into engineering. I tried to think about things that we could do that weren't necessarily just classroom-based.

I met Shawn in 2010. He'd won the same award that I'd won and we met through that competition. We got talking about how there are some great STEM initiatives and showcases on engineering, but some of them are very much set in the dark ages. There's some great work, but we felt that there could be more.

We realised that some of the things that influenced us were in the media. I've ended up being an engineer, my sister's an engineer, my brother's an engineer, but I never saw myself as being an engineer, because in all the TV shows that I watched, there were very few women who were practically getting on and making. That's one of the things that we wanted to look at.

We were talking about this for probably about seven years on and off… discussing things that we could do, projects, we looked into starting an engineering venue… all sorts of things. In 2016, Shawn and I sat down and started brainstorming what we could do. One of the things that had always held us back from doing something on the scale of Kids Invent Stuff, in the sense of video, was getting the right platform. Through our research, we realised that 90% of 5–11-year-olds were watching content on YouTube. We thought that we should be using that, putting our content on there and – for them, that's their TV.

I'd love to have a great story about how we came up with Kids Invent Stuff, but it just came out of probably five or six meetings where we decided that kids invent the best things… why not set a challenge each month for us to build, and we could film it, and put it on YouTube?

> "I've ended up being an engineer, my sister's an engineer, my brother's an engineer, **but I never saw myself as being an engineer**"

That's kind of where the idea came from. In June of 2016, Shawn and I reached out to friends and organisations that we knew.

We set the challenge for superhero gadgets, got inventions sent in, and we decided to make Lucy's (aged seven) custard-firing super suit. We filmed it, as we would do the normal series, as a pilot to show people the concept. When that was ready, we shared it out to different people to try and raise sponsorship to be able to continue the series. We launched in January 2017 and, since then, every single month we've set a different invention challenge. 5—11 year-olds have submitted loads of invention ideas.

HS I'd never realised that you'd have to make a pilot to get people on board. I think people forget that just because you're on YouTube you still have to jump through hoops (unless you've got deep pockets).

RA Having done that sort of thing before for other production companies, I realised that people don't usually just take your word for it that you can build a kids' invention. Most people think you can't, and so we knew that we needed to have an example to show people, because if companies and foundations were going to give us money to do this STEM outreach work, they had to know we could do it.

It was also for our benefit; we weren't entirely sure of the format exactly. We filmed the first one ourselves and didn't really have any filming help, which made us realise that we needed another person to help us film. It was useful to help us work out the logistics of the process.

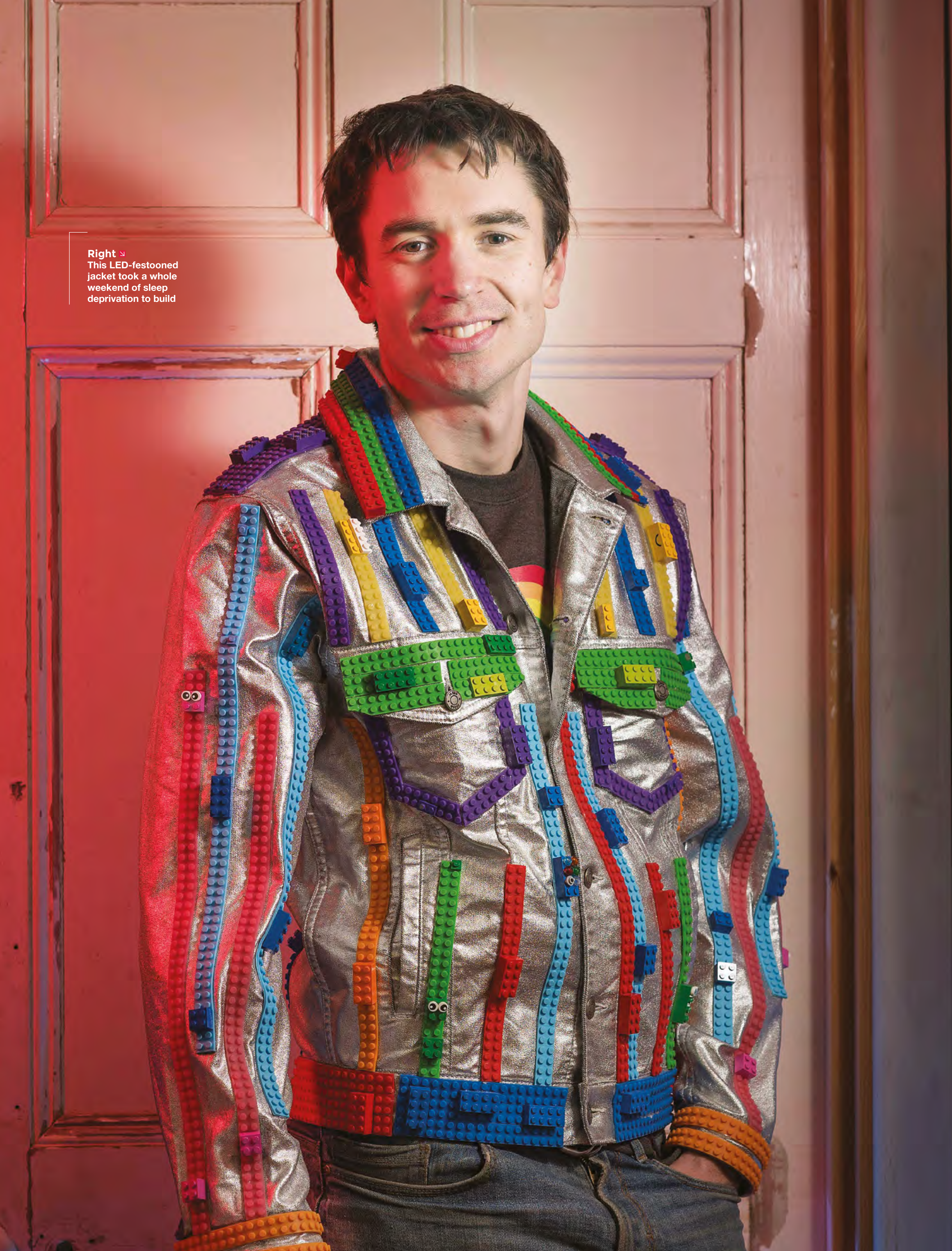
Also it's a funny cycle in the way that we release videos. We do a weekly video, but obviously kids need three weeks to submit their invention ideas. We do a video to invite idea submissions, then there's the video where we choose what to make, the video where we do the build, and the video where we test the invention out. Sometimes there's an extra Saturday, and we'll do either stuff that kids can make at home or… we're in the process of trying to create some content around some of the stuff that we've already built. We're building on that. It's interesting and very highly responsive.

HS You've got 2,700-odd subscribers now. We just assumed you'd been going for a lot longer. Also it's highly awesome stuff. The custard video, even though it was the first one, it really looks like you're having loads of fun.

RA It was ridiculous amounts of fun. We went and bought loads of custard from Asda, went to an old disused quarry, and ran around firing custard at each other all day long, on probably one of the hottest days ever in Cornwall.

Every single build and test is loads of fun. We want it to be a spectacle. →

Right
The Kids Invent Stuff duo work out of Concept Shed, a studio in Falmouth



Right

This LED-festooned jacket took a whole weekend of sleep deprivation to build

We want kids to look at it and think "That's my idea", or "that could be my invention", to inspire them to really think about inventing and making.

For example, the fire and water shooting piano: where do you test a piano that shoots fire and water up into the air when you play it? Obviously you test it on a clifftop in Cornwall and get a drone to fly overhead filming it. The logistics of that were challenging: if anyone's ever moved a piano you'll know it's not the easiest object to move from A to B.

The builds have to be quick and responsive. We never really know what we're going to be making next. For example, our most recent challenge closed yesterday, and we're in the process of working out which one of these inventions we're going to make, ordering in parts. We'll be filming tomorrow and a little bit on Saturday, we'll release on Saturday, and then next week we'll start that build process, so we don't have a lot of time to think about what we're going to build and how we're going to test it. But so far it's all managed to work.

The day before we did the underwater dolphin, we didn't even have anywhere to test it. Luckily in Cornwall a lot more people have pools than they do in other parts of the country. We sent a plea out at 6pm the night before we were due to film the test – we were really flying by the seat of our pants to get stuff sorted and made.

And things don't work, we have a very limited amount of time to build. Just a handful of days when we're building everything. With the crazy jackets we made, we were going to sew the LEDs, and then they didn't work on the jackets and it was a Saturday night, no shops were open, so we were frantically wiring through the night. That build took 48 hrs solid to get it ready to test on the Sunday. It's lots and lots of fun, but it is also very 'interesting'.

HS Is this your full time job? It sounds like a lot of pressure and time to get things done.

RA No, it's not. I'm still involved with the StairSteady, and I do lots of speaking and other bits of consultancy work as well. Shawn works on other projects; he's currently involved with the Man Engine, which is the largest mechanical puppet in Britain.

That's based in Cornwall, but it's going on tour across the UK this year. He also does loads of STEM workshops in schools and things like that. So it isn't a full-time job for either of us. When we sit down to work out the projects — and in a way the logistics of where we live make it work better — we have to have set amounts of time that we can devote to working on things, which works really well. We set ourselves some really key targets

Although it isn't our full-time job, that's how we manage to get stuff done. We manage all the social media, all the editing. We build and test but we also do everything else, apart from the filming.

> "We manage all the social media, all the editing. **We build and test but we also do everything else,** apart form the filming"

HS That sounds like a hell of a lot of work. How do you choose between the ideas that the kids put forward?

RA That's the hardest bit. It really does depend on the competition; sometimes we have people who regularly submit, but we also have some different challenges, or maybe it fits in with what a school class is doing, so a whole class will send their ideas in. It can really fluctuate, especially if a teacher submits 30 kids' work, or a whole year's worth of drawings, and suddenly that can add lots of other inventions into the mix. It's the hardest bit because we have these pictures and we have videos, and we keep an eye on inventions coming in.

It's the hardest part because we like to choose inventions that maybe adults would look at and be like: "Oh no, why would you make that? Why would you want to create shoes that walk on water? Or, why would you want to make a jam-firing rocket?" But they're the sort of things that really capture kids' imaginations. We want to celebrate that.

There's the added thing of being able to physically make them and doing justice to them as well. We have a set amount of time and budget, and want that child to feel like we've made their invention properly. That's the hardest thing when we're making it. We're constantly aware that we don't want to let that child down. You don't want to have them disappointed with what you're creating. We want to capture other kids' imaginations, but we want the child whose idea it was to know that we've done a good job.

HS That's the most heartwarming thing I've ever heard.

RA It's funny, because that's the biggest pressure when we're building. [After we're finished] we'll get a nice email from a parent of that child and be so relieved. The interesting thing about choosing the invention is that it's not about the neatest drawing or the person who has explained it the most, it's about the idea. We often have parents who say that their child doesn't succeed academically or doesn't enjoy school, but they're really creative and this has encouraged them to feel like they are able to contribute and their ideas are valued, and that's amazing.

It's something that's really important to us. Shawn's dyslexic, and he really struggled at school, and we really want to celebrate those creative skills that aren't celebrated as much within the school system. →

Left
Shawn's working on the School Spinner – an invention by 9-year-old Alisha

HS I bet the younger kids are more creative than the older ones because they've not had it bashed out of them yet, by having to conform to exam standards.

RA You do see, as people become teenagers, that our creativity in some areas disappears. I'm often asked why we don't do an adult one. Unfortunately adults over-think things, they're not as creative as kids are.

HS That's when you get really mundane things like boxes and desk tidies being made in schools. At our school we churned out loads of useless plastic boxes, because we had a machine for bending acrylic.

RA I understand why schools have to do that. It's really hard to say, should it be down to the parents? Should it be down to the school? How do we support creative kids? Really letting them have that space to do those things. It's a hard one to answer. We wanted Kids Invent Stuff to be as open as possible, if you've got an idea and a pen and paper — and we do have kids who scribble ideas on the back of receipts — it's not about having all the fancy gear, it's about having the idea. We want any child to be able to join in, and not feel that they were excluded because they couldn't buy certain kit.

HS I'm not going to ask you to pick a favourite, because that's not fair, but was there one build that was the hardest, or the most dangerous?

RA In most of the builds something goes wrong at some point, so they all have their own challenges. The biggest build that we've ever done was the crazy car. We were looking for inventions to make long car journeys more exciting, and the invention we chose was a seat that came up *Thunderbirds*-style from the top of the car, and you ride along 3 m in the air, on the top of the car. That's the most expensive build we've ever done. We were very lucky to get an extra sponsor with that one to help us.

The components involved were pretty simple, but they were big and expensive, and it was so much fun to do something that big. I mean, why would you ever cut the roof off a car? I think I'd used a grinder maybe twice in my life, and to be able to grind a roof off a car was like: "This is impressive. This is ridiculous". That was probably one of our favourites, because it took so much work to make it happen.

HS Can we ask you about Girls with Drills? That's another thing you've been involved with, and one that's picked up a fair bit of attention. What's it all about?

RA Girls With Drills is something that me and my friend Kisha Bradley did at the end of last year. She does something called Bright Box, which is a charity that helps disadvantaged children access maker sessions, and puts on sessions in libraries and community spaces for free. She's also based in Sheffield.

We got talking and this phrase emerged – we wanted to see more girls with drills. Not with a view to turning them into engineers; just to give them the confidence to not be scared of them, so they can put up shelves and hang pictures, because quite often we see tools as a stereotypical bloke thing.

We wanted to highlight all these amazing women making and designing stuff, and really celebrate inclusive making. As the conversation went on, we thought it would be really cool to make some #girlswithdrills tops. So we thought we could open up for pre-orders and anyone who buys one could support and pay for a disadvantaged child to attend a maker session. It's a bit tongue-in-cheek, and it's a conversation starter: "Why is it just about girls?" Well, it's not just about girls. It's about inclusive making.

We launched this programme, and people really liked it. People started to buy straight away. There was a set amount of time for the first run, because we knew we wanted get the tops made and sent out before Christmas. We were really lucky when some of the bigger names in the maker community retweeted us. Robin Ince signed up to it, and the campaign just grew and grew.

So we closed the campaign, got everything printed, and sent out. We were really lucky, we had people like Colin Firth buying the tops. Shawn sent me a picture over Christmas and Simone Giertz had worn one of the tops on one her videos. That started it all up again.

We didn't want to start up a clothing company, but it did make us want to start another campaign.

So we're going to launch again on International Women's Day (8 March). We've got an illustrator to do a unique design for the tops this time, and again we're funding maker sessions for disadvantaged children. People really liked it last time – it's really important to keep having those conversations. □

Left

Attention: safety equipment is essential when building kids' inventions

pi-top

Inspiring inventors and creators to seek the skills of tomorrow and create their future, today.

POWERED BY RASPBERRY PI

OCR
Oxford Cambridge and RSA

The modular laptop with sliding keyboard

8HR BATTERY LIFE

14" FULL HD 1080P SCREEN

180° HINGE

CUSTOM PASSIVE COOLING BRIDGE

MODULAR RAIL

pi-top

Colors 🟩
Raspberry Pi 3 optional

AWESOME INVENTOR'S KIT INCLUDED

20+ projects to explore

Explore beyond the screen and keyboard by creating with the all-new **pi-top** modular laptop.

Get started with 20+ inventions in the inventor's guide booklet. There are 3 inventor's journeys - Smart Robot, Music Maker and Space Race.

The modular desktop

14" FULL HD 1080P SCREEN

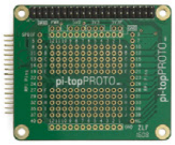
MODULAR RAIL

ADJUSTABLE VIEWING ANGLES

pi-topCEED

Colors 🟩⬛
Raspberry Pi 3 optional

pi-topCEED is the plug & play modular desktop. It's the easiest way to use your Raspberry Pi. We've put what you love about our flagship laptop in a slimmer form factor. Join hundreds of code clubs and classrooms using **pi-top**CEED as their solution to Computer Science and STEAM-based learning.



pi-topPROTO

pi-topSPEAKER

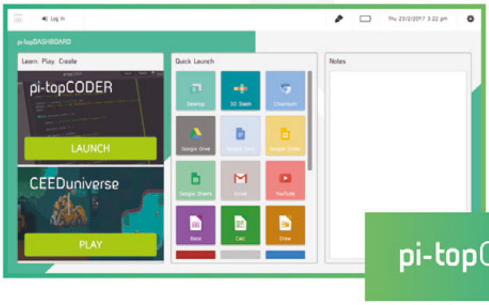
pi-topPULSE

Modular **Accessories**

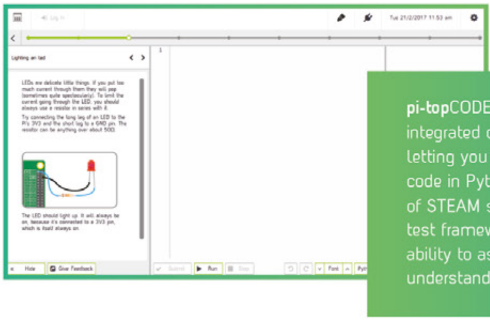
www.pi-top.com @GetPiTop /GetPiTop Stay up to date with our latest news by following our social media

pi-top is an award-winning ecosystem designed to make experimenting, coding and building electronics, simple, affordable and fun. **pi-top**OS is here to guide you through the world of making!

The OCR* endorsed **pi-top**OS (Operating system) platform comes pre-installed on the SD card shipped with every unit. **pi-top**OS software suite lets you - browse the web, - check emails, - create and edit Microsoft Office compatible files. Gain access to dozens of hands-on learning lesson plans with **pi-top**CODER and have fun learning to code with CEED**universe**!



pi-topOS

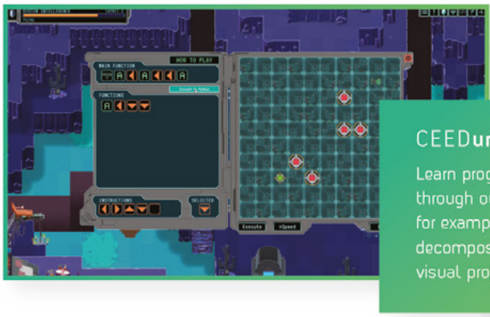


LEDs are delicate little things. If you put too much current through them they will pop (sometimes quite spectacularly). To limit the current going through the LED, you should always use a resistor in series with it.

Try connecting the long leg of an LED to the Pi's 3V3 and the short leg to a GND pin. The resistor can be anything over about 50Ω.

The LED should light up. It will always be on, because it's connected to a 3V3 pin, which is itself always on.

pi-topCODER has a fully integrated coding environment letting you program hardware, code in Python and learn lots of STEAM skills! Our integrated test framework gives you the ability to assess your own understanding as you learn.



CEED**universe**

Learn programming concepts through our minigames, for example, learn problem decomposition by solving visual programming puzzles.

IMPROVISER'S TOOLBOX

PENCILS

Dubbed the ultimate tool for expression, it's often said that all you need is a pencil and a dream to go anywhere and build anything

Mayank Sharma

@geekybodhi

Mayank is a Padawan maker with an irrational fear of drills. He likes to replicate electronic builds, and gets a kick out of hacking everyday objects creatively.

Pencils are amazingly versatile. They can scribble underwater and even in zero gravity, and have been fulfilling our impulse to record, create, and share knowledge for over 400 years.

It's a common misconception that the core of the pencil is made of lead. The writing bit of a pencil is actually made with a non-toxic mineral called graphite. Modern-day pencil cores are, in fact, a mix of graphite and clay, fired together in a kiln at more than 800 degrees Celsius. The misnomer originated in 1564, when some shepherds from Borrowdale, England stumbled upon a massive deposit of pure graphite. Back then, the material was misidentified as a rare form of plumbago, which is the Latin word for lead ore. It wasn't until three centuries later, in 1789, that it was rechristened as graphite using the Greek word 'graphein' meaning 'to write'.

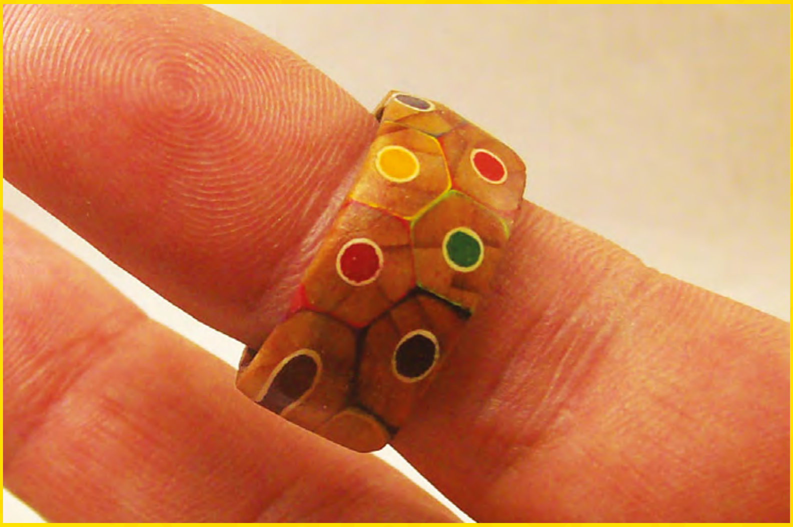
Graphite is very soft and brittle, and back in the day early users wrapped it with string to keep it together. Later on they started sawing it into thin rods placed between pieces of wood. When France declared war in the 1790s, it lost access to Borrowdale's excellent graphite. With pencil supplies dwindling in 1795, the French Minister of War asked one of his officers, named Nicolas-Jacques Conte, to come up with a substitute. He came back with a kiln-baked mixture of graphite powder and clay, which turned out to be so good that it's still in use today.

Modern-day pencil cores are produced by first processing the graphite and clay to remove any impurities. A machine then shapes the mixture into large square blocks, which are then either baked in a kiln or squeezed in a machine to remove any moisture. The mixture is then ground into a fine powder and made into a soft paste by adding water. This paste is then shaped through a metal tube into spaghetti-like strands that are cut into pencil-length pieces and dried. Finally, they are soaked in oil or wax to add smoothness. The outer shell of the pencils is most often made from wood – the most common woods used for pencils being incense-cedar and basswood.

PENCIL GRADES

So, we know that pencil cores are primarily made up of graphite and clay, and it is the formulation of this mixture that determines its lead grade. The hardness or lightness of the mark made on the paper is dependent on the ratio of the two. So if there's more clay, the pencil will be harder and lighter. Similarly, if there's more graphite, the pencil will be darker and softer. There are two grading systems for pencils – American and European. The American system is numerical and ranges from 1 to 4. The European system uses the letters H and B to note how hard or soft a pencil is. The HB pencil is the standard writing pencil today. The HB grading is named for Hardtmuth and Budejovice, after the location of the company that produced the graphite.

COLOUR PENCIL RING



Left
Peter says it took him three tries to get that one ring turned and filmed

Project Maker
PETER BROWN

Project Link
hsmag.cc/piXVGn

Pencil Fact

Ebenezer Wood was the first to manufacture hexagonal-shaped pencil barrels in the 19th century.

If you want a unique piece of jewellery to show your love for pencils, follow Peter Brown's lead. Peter glued twelve pencils together in two rows of six. He then measured the size of his finger and drilled a hole between the glued pencils using a drill bit. Next he cut a rough shape using the scroll saw to take the edges off and make it easier to turn.

To mount the pencil cut-out on the lathe for turning, he made a quick jam chuck from a piece of pine. While turning, he used chisels to shape the wooden ring. Once he was done shaping, Peter used sandpaper to get a nice smooth feel to the ring and to trim it down. Peter advises you to be careful while sanding the ring so as to not smear the colours. He finished the ring with two coats of spray lacquer.

After the video of him doing this went viral, Peter says he received more than 1000 orders for coloured pencil rings on his Etsy shop. But he hasn't made another one because he could never figure out how to make them thin enough to be pretty, yet strong enough to last. □

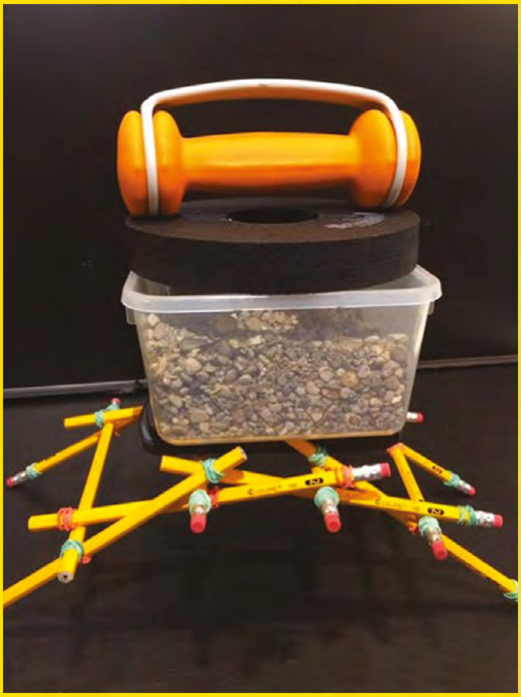
DA VINCI BRIDGE

Project Maker
IGAMEMOM

Project Link
hsmag.cc/RrfYfu

iGameMom is a useful blog that's chock full of STEM (Science, Technology, Engineering and Maths). After witnessing a demo of the Da Vinci bridge, the blog's creator decided to replicate the marvel of mechanical engineering design using pencils.

They start out the blog post with a link to a video of a father and son duo building a Da Vinci bridge with pencils. When they started to build their own, they failed because the pencils didn't have enough friction to hold them in place. So they looped rubber bands on the spots where the pencils intersect to create the required friction. They've detailed the assembly, along with images as well as tips, to help you build one. You'll need a minimum of twelve pencils to build the bridge. Once you've got the hang of the basic construction, you can create a longer bridge by adding more pencils. However, even with twelve pencils, the assembly will require two pairs of hands. When completed, the bridge is impressively sturdy: even their twelve-pencil version is strong enough to hold upwards of 3 kg. □



Above In addition to engineering design, the Da Vinci bridge familiarises you with the concepts of force and centre of gravity

Pencil Fact

The first patent for a pencil sharpening device was issued to French mathematician Bernard Lassimone in 1828.

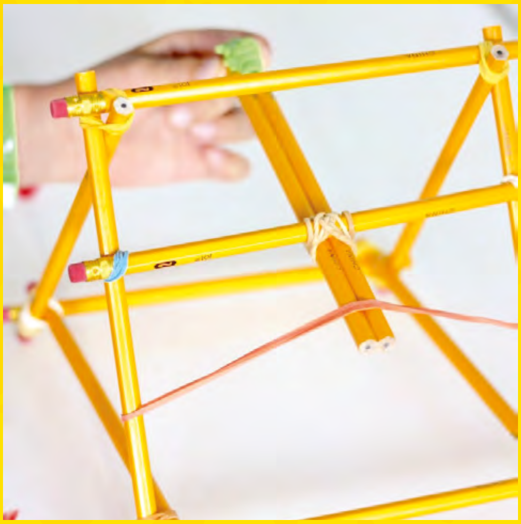
PENCIL CATAPULT

Pencil Fact

Sharpeners were outlawed during the Second World War due to concerns over wastage of wood and lead.

Project Maker
SARAH McCLELLAND

Project Link
hsmag.cc/fpOZFK



Left
Besides pencils, Sarah has made catapults from LEGO, plastic spoons, popsicle sticks, and even marshmallows

Sarah McClelland also blogs about STEM projects over at her blog called Little Bins for Little Hands. The goal of her blog is to help her readers realise how cool science can be. To demonstrate some basic concepts of physics and mathematics, Sarah built an easy-to-assemble catapult from some pencils and rubber bands.

Sarah takes a handful of number 2 pencils (as explained, the American equivalent of the HB pencil), and ties them together with rubber bands to create a catapult. She has detailed step-by-step instructions on her blog, along with images to help you craft the various components of the catapult and assemble them together in the correct order. In her build, Sarah has used eraser pencil toppers to launch from the catapult, but you can use something else like the cotton balls that Nicola Duffy used when she created the catapult in her craft class. □

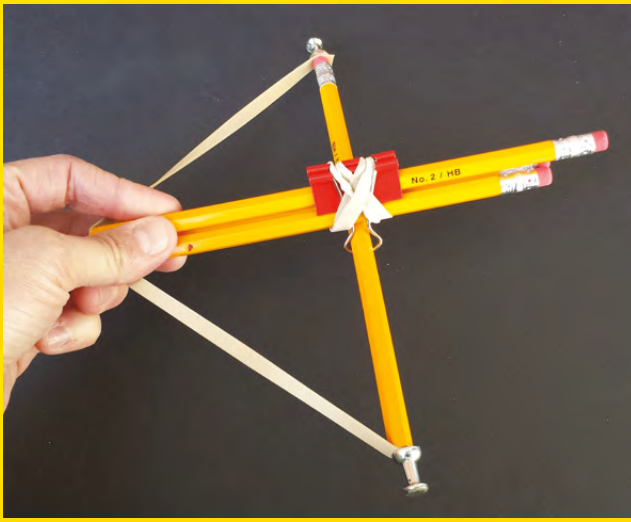
PENCIL CROSSBOW

Project Maker
MARIE

Project Link
hsmag.cc/BvLBjP

To keep a handful of boys aged three to eleven occupied during a family trip, Marie decided to build mini crossbows to help their target practice. While she had seen several mini crossbow builds on Pinterest, she decided to make one using pencils for simplicity. Each crossbow is made with four pencils, two thick rubber bands, two push pins, and one binder clip.

Marie has detailed the construction and assembly on her blog with images for every step. The design is simple and can be replicated by anyone in under half an hour. Be careful during the step where you attach the rubber band to the pencil using pins. Marie has used pencils with erasers at the end, so pushing the pin on one side will be easier than pushing it into the wood on the other side. Her crossbow uses a pencil as an arrow, and she advises you to always make sure the eraser points towards the target. □

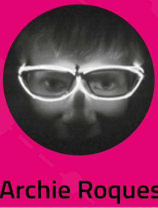


Right
You can vary the length and stretch of the rubber bands to change the distance and speed of the projectiles

LEDs

THE GOOD, THE BAD AND THE BLINKY

They light up our projects, but how do they work



Archie Roques

@archieroques

By day a humble A level student, by night a hardware engineer, Norwich Hackspacer, and general projects man. Also blogs at **roques.xyz**.

The venerable LED, aside from its place in everyday life across the world, has had a strong career in the maker industry. Few electronics enthusiasts' benches are without drawers, pots, tins, or bags of the ubiquitous luminaires we know and love. They illuminate crazy cosplay creations and innocent blinking 'hello world' electronics projects alike, as well as being responsible for meeting a significant portion of the world's lighting demands.

THE BASICS

LEDs are a kind of diode, and diodes are a kind of semiconductor. So in order to understand how LEDs work, we need to dive down to the atomic level.

The basic premise of semiconductors is that a semiconducting element like gallium (as is used in many LEDs) is 'doped' with some atoms of another element. These other atoms either introduce extra electrons or take them away, creating two types of semiconductor with either an excess of electrons or an excess of 'electron holes' (a lack of electrons).

A SEMICONDUCTOR SANDWICH

LEDs, like all diodes, are basically a sandwich of the two types of semiconductor: 'n-type' allows negative charges to flow and 'p-type' allows positive charges to flow. Current will only flow from the side with lots of electrons to the side with few electrons, which is why diodes only let current flow one way, and why your circuit won't work if you connect an LED backwards. These P-N junctions are used in all semiconductors – transistors, integrated circuits, and processors. You've probably got thousands of them in your house – or even your pocket.

These produce light when the electrons jump between energy levels, which happens when they flow from the n-type to the p-type and begin to fill the holes. This involves some pretty complex quantum physics, but simply, an electron jumps up an energy level, and then falls down again, and when it falls down again it releases a photon. Photons are the tiny packets of energy which make up visible light.

Depending on how much energy the electrons lose when changing energy levels, different wavelengths of photon are emitted, and different wavelengths of photon are what make up different colours of light. Each doped semiconductor has its own 'band gap energy' (the difference in energy between the p- and n-type), so different energy levels produce different colours. Almost all diodes produce light, but to get visible light out of them, and ensure it isn't absorbed by the diode itself, requires certain materials.

In order to make other colours (you might sometimes see pink and purple LEDs), a variety of methods can be used. Sometimes, phosphors can be added to the semiconductor to alter the wavelength of light given out, and sometimes multiple LEDs are combined to make new colours (more on that later).

USING LEDs

Putting theory into practice

LEDs are great for a number of reasons. Traditional filament light bulbs generate light by incandescence – using heat to generate light. This works, but is woefully inefficient: usually, just 2% of the energy input into an incandescent bulb is output as useful light. LED bulbs use electroluminescence meanwhile, and their efficiency is closer to 50% – which, although it could be improved, is still a marked difference.

LEDs are also low-voltage and use direct current (DC), which, whilst it's an inconvenience for mains lighting (where AC must be converted to DC), is very useful for mobile applications. An LED can run off a couple of AA batteries for many hours, which makes old incandesant bulbs pale in comparison.

LEDs are also much more robust than conventional bulbs, and can be made in much smaller packages. The average LED is as small as just 0.8 mm in diameter, and yet smaller sizes are available. Also, they aren't made of glass, so can be dropped without too much of an issue.

They're also significantly cheaper at the component level than traditional bulbs. At the time of writing, a well-known component supplier could supply its cheapest LEDs for less than a penny each when purchased in quantity.

> “ LEDs are much more robust than conventional bulbs, and can be made in much smaller packages ”

As mentioned before, LEDs only work one way round. On most through-hole LEDs, the longer lead is the anode, which should be connected to positive, and the shorter lead is the cathode, which should be connected to negative or ground.

Due to the difference in materials between each LED, some colours need a higher voltage to run than others. You can try this out – if you connect a red LED and a blue LED to the same coin battery, only the red LED will light. This is because the red LED has a forward voltage of only 1.8 V, whereas the blue LED has a forward voltage of 3 V or more. The current will flow at 1.8 V, but this isn't enough for the blue LED so it won't light.

NEVER GIVE AN LED MORE VOLTAGE THAN IT LIKES

If you give an LED a voltage that is too high, however, it won't like it and is likely to burn out. Whilst this is fun to try once, it can get in the way of projects! A resistor is normally needed when connecting LEDs to other things, to limit the current passing through it. The value of resistor you need varies depending on your LED and what it's connected to – there are lots of helpful LED resistor calculators online.

You can connect LEDs in series or in parallel. If you connect them in series, you will need to add the voltages of each LED, but the current will be the same as for one LED. If you connect them in parallel it's the opposite way round – you'll need a higher current but the same voltage as for one single LED. →

Voltages

Different colours of LEDs have different forward voltages. Usually, red and yellow LEDs have forward voltages of around 1.8 V, with blue, green, and white closer to the 3 V mark.

Below
There are few things that can't be made to look better with the addition of a copious quantity of LEDs

MULTIPLEXING

Getting more lights for your pins

Multiplexing is a way of driving multiple LEDs with a reduced number of control pins (on an Arduino, Raspberry Pi, or your development board *du jour*). It's all very well controlling a couple of them with their own pin, but if you want to do that 8×8 matrix you've always promised yourself, things start to get a little tricky. That's where multiplexing comes in.

Multiplexing uses the fact that LEDs will only work in one direction to its advantage. When we say voltage, what we really mean is potential difference. This isn't really an important distinction for most makers, but the word 'difference' is fairly vital here. If you were to plug in a 3V LED to a supply with 230V and 233V, it would still run fine. That's because the difference in electrical potential is 3V, which is what the LED needs. Conversely, if you were to connect both the anode and the cathode to 3V, it wouldn't do anything, because the difference is 0V.

You can connect two LEDs between two microcontroller pins. If you put them both low or both high, nothing will happen as there is no difference. If they are in different states, however, different LEDs will light. This is pretty useless for just two LEDs, but you can easily connect more and more LEDs between more and more pins. This technique – known as Charlieplexing – allows you to control huge numbers of LEDs with few pins. There's even an equation to work this out! The number of LEDs you can control = number of pins × (number of pins minus 1). So, with a normal Raspberry Pi it would be possible to control over 750 LEDs (if you were insane enough to wire 750 LEDs in this way and didn't exceed the current limits).

Problems start to arise if you try to light two LEDs at once (other LEDs will light up too), but this is easily solved by pulsing through the rows and columns faster than the eye can see, so it looks like all the required LEDs are on all of the time. There are even specially available chips that do this for you.

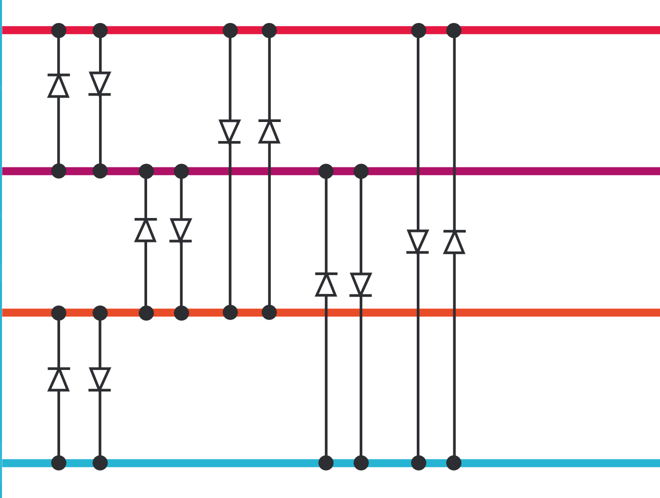
Right The internals of a NeoPixel with the microcontroller and three LEDs

> **With a normal Raspberry Pi it would be possible to control over 750 LEDs (if you were insane enough to wire 750 LEDs)**

Voltage as potential difference

Think of a building block, placed on the floor, about 25mm tall. If you put it on a table, it'll still be 25mm tall, but it might be a metre above the ground.

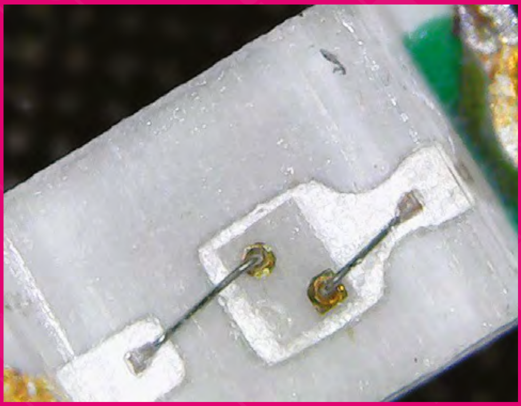
This is true for other components too – it's why if you get a shadily engineered phone charger, it could still electrocute you (because you're connected to the ground) with mains voltage, even if your phone is fine because it's just using a 5V potential difference to charge.



Right Charlieplexing twelve LEDs from four data lines

ADDRESSABLE LEDS

When you absolutely have to have the most blinking lights



Left
The back of an APA102 blinkenlight under the microscope

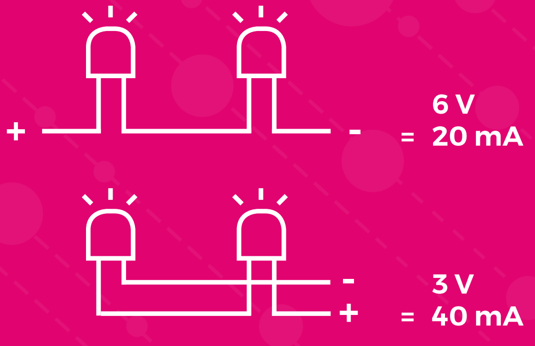
The coveted WS2812B LED strip is an often-used staple of many creative electronics projects. More commonly known by the trade name NeoPixel, along with its distant cousin the APA102 it makes up a large majority of the blinkenlights in the maker world.

Addressable LEDs combine a teeny driver chip with three LEDs (red, green, and blue), which can be controlled to make any colour you desire. The basic premise of addressable LEDs is simple: imagine a bus full of passengers travelling down a road. A person only gets on or off the bus if it's their bus at their stop, and the same sort of thing happens with the data that addressable LEDs use. The data is passed from the microcontroller down a data bus (yes, it's really called a data bus!), and then from LED to LED, and the driver chip only picks up data relevant to that LED in the chain. It's possible to hook hundreds and hundreds of LEDs together in one long chain, and send one string of data to program them all.

ALL LIGHT IS MADE UP FROM THREE CONSTITUENT COLOURS

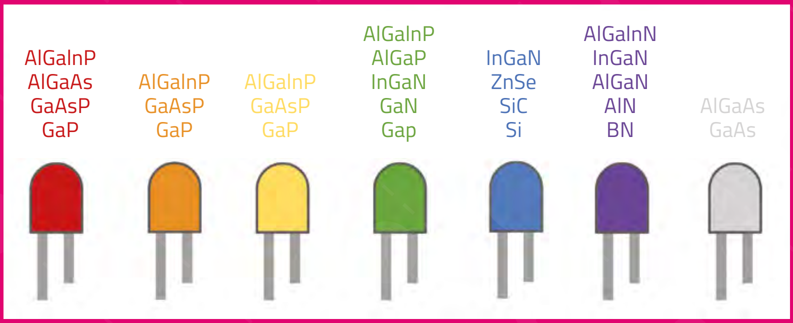
All light that we see is made from three constituent colours – red, green, and blue (hence RGB). By mixing these in different amounts, you can reach almost any shade in the world. However, it's not easy to produce true-white light, and the white light emitted from RGB LEDs often has a purplish tinge to it. There are variants of these addressable LEDs available that also contain a white LED (RGBW addressable LEDs), which are more suited to lighting applications.

There is one crucial difference between APA102s and WS2812s. The former use a two-wire protocol, akin to the SPI (Serial Peripheral Interface) used for many other small electronic modules. It requires a controller device that's capable of sending SPI data. The WS2812s, on the other hand, use a one-wire protocol. This sounds great – only one wire, and the LEDs are cheaper too – but it has a few caveats. WS2812 LEDs take a little longer to change than APA102s (though this is only noticeable in high-speed applications like persistence of vision) and require super-precise timing on their data line. This is fine when using a microcontroller, but when working with single-board computers like the Raspberry Pi, the timing can become a bit of a sticking point.



Left
Different wiring configurations have different electrical requirements

Below
The quantum properties of different materials manifest themselves in different colours



MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
April 2018
Issue #05
RASPBERRY PI 3B+
Faster processor & networking
DREMEL HACKING
Turn your hand tool into a table saw
RESIN
FASH
LADYADA
LIMOR FRIED REVEALS ADAFRUIT'S SECRETS TO MAKING GREAT HARDWARE
LITHOPHANES CHEERLIGHTS PENCILS NEURAL NETW
WEARABLE TECH
ELECTRONICS MEETS FASHION
LASER CUTTING HOT WATER HACKABLE DISPLAYS

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG 78
SCHOOL OF MAKING
Start your journey to craftsmanship with these essential skills

78 Seven-segment displays
84 Light up your wardrobe with LEDs

PG 90
CHEERLIGHTS
Join hackers around the globe in the world's largest IoT project

PG 92
DREMEL WORKBENCH
Keep your cuts straight with a miniature table saw

PG 96
ON-AIR
Stop workmates interrupting your meetings and downtime

PG 98
THE ONLY KNOT YOU NEED
Keep everything shipshape and Bristol fashion

PG 102
NODE-RED
Flow-based coding to control your home

PG 106
RESIN.IO
Keep all your devices running the latest code

PG 110
LITHOPHANES
Turn flat images into 3D-printed objects

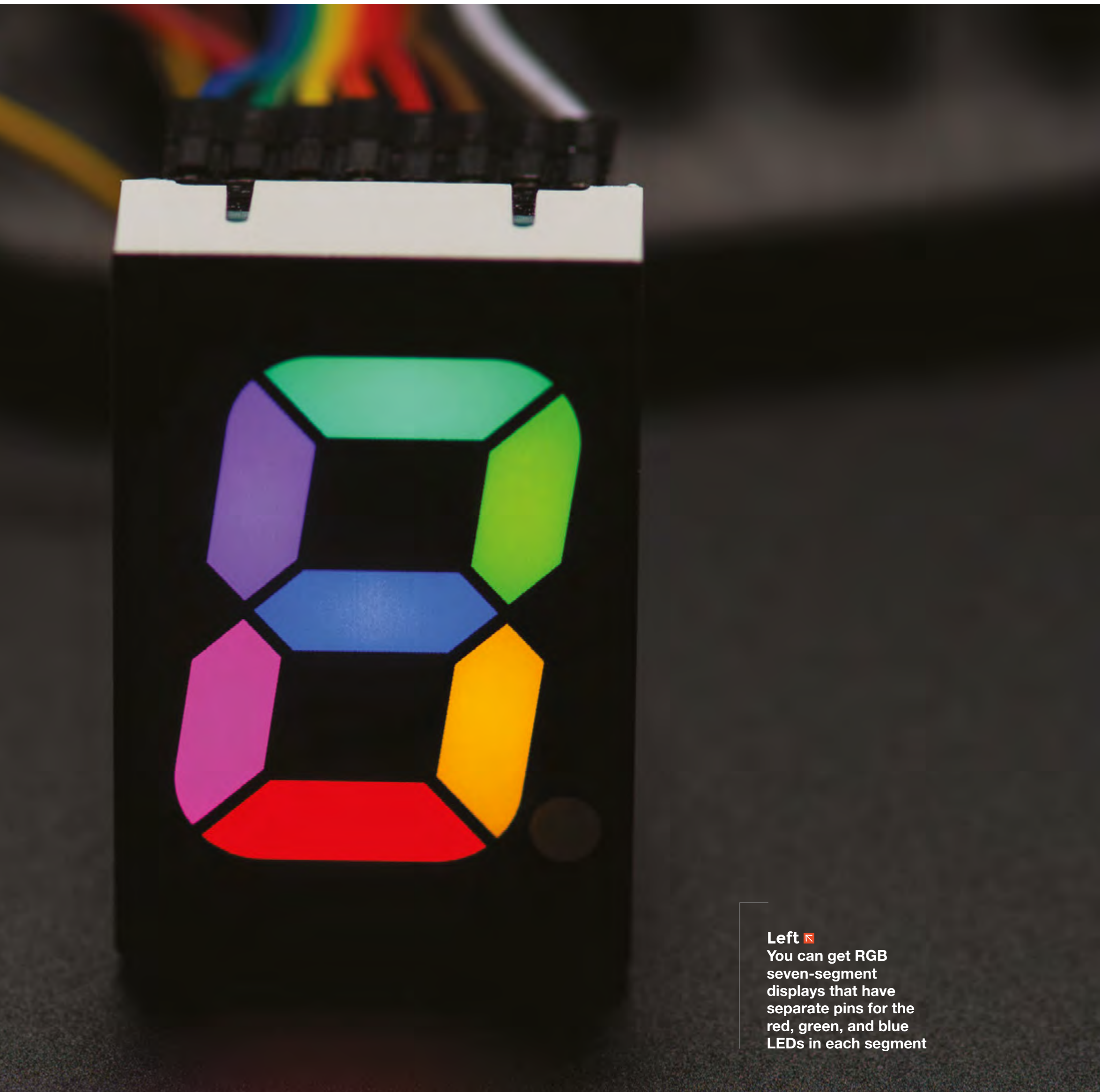
Arduino programming: Seven-segment displays and multidimensional arrays

Get meaningful output out of your projects and master interdimensional data storage

Graham Morrison

@degville

Graham is a veteran Linux journalist who is on a life-long quest to find music in the perfect arrangement of silicon



YOU'LL NEED

- Kingbright SC10-EWA
- Arduino Uno
- 7 × 220 ohm resistors
- Breadboard
- Connectors

Left You can get RGB seven-segment displays that have separate pins for the red, green, and blue LEDs in each segment

When it comes to programming the Arduino, one of the most important skills to master is taking a physical problem and then constructing a solution that can be expressed efficiently in code. This gets easier with hardware and code experience, but it's important to note that whether you're an expert or a complete beginner, your first solution is highly unlikely to be the best, and in most cases a project may be completely rewritten once, twice, thrice, even four times. Each successive rewrite will incorporate the experience learned from the previous version, as you begin to better understand how to make your solution work.

This is what makes variable types, and the related subject of data structures, so important. Not only do they enable you to write code that makes the most efficient use of your hardware, they allow you to more accurately define your solutions in code. For example, it's perfectly acceptable to use an 'int' type to store which Arduino digital pin to use for an LED. But, as the typical Arduino only has around 14 digital pins, using a variable capable of holding any whole number between -32 768 and +32 767 is considerable, especially when Arduinos have so little RAM. And there's a variable type that works the same and takes less space: the 'byte' type holds an 8-bit unsigned number and, if you can remember your binary mathematics, this works out to be a number between 0 and 255. It's not perfect, but it's more memory-efficient and easier for readers of your code to understand and modify because you've set limits on how the variable should be used. Creating structures that have limits is one of the cornerstones of object-oriented programming.

To put this idea into practice, and explore further how variables are used in a working example, we're going to create a simple foundation project that can be used at the heart of many more ambitious projects. The reason why this project can be used in so many others is because it takes the simple idea behind every basic Arduino LED example and expands upon this to build a fully fledged output device capable of representing many different alphanumeric characters. The simple component used to perform this magic is the humble seven-segment display, as used on the Apollo spacecraft, pinball machines, and microwave ovens. With a seven-segment display, your devices can communicate with the outside world, whether that's a temperature or volume reading, or the radiation level on your Geiger counter. In fact, a seven-segment display would be the perfect upgrade to the temperature sensor Arduino project described in HackSpace issue 3.

A seven-segment display is really little more than seven LEDs in a single package, or eight if you include a decimal point. Pins alongside two edges will correspond to either the negative or positive input pins for each LED, dependent on whether your display uses a common anode or common cathode path. This type difference dictates whether a segment/LED is lit by either grounding the pin, or by providing it with 5 volts (respectively). In our example, we'll be using the more usual 'common cathode' type, but the wiring can be simply inverted if you find yourself with the opposite kind of display.

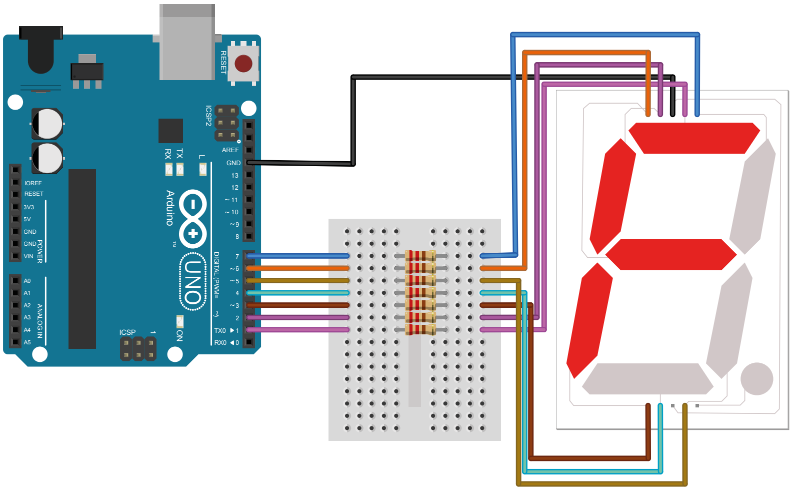
> " A seven-segment display is really little more than seven LEDs in a single package, **or eight if you include a decimal point** "

WIRING UP

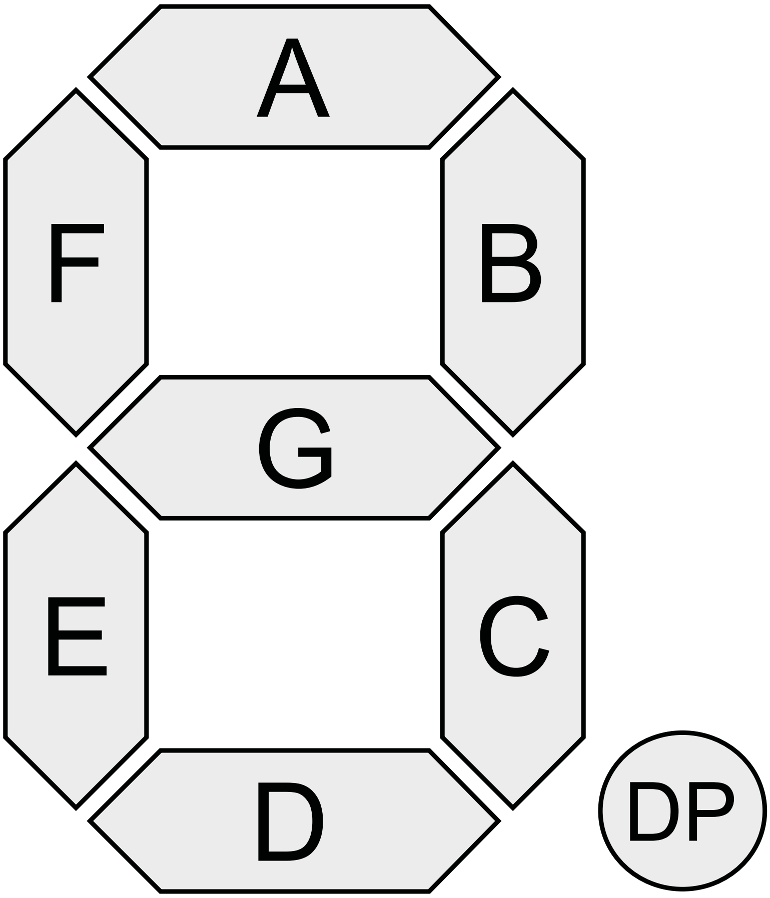
Getting the polarity of an LED correct is vital, and the same is true with a seven-segment display. The only real difference is that with a seven-segment display, all seven of those LEDs are fused into either common anode or common cathode, and you need to get this correct for the whole thing to work. Only the specification of your displays will tell you which way around you'll need to wire the display and which pins are used for common ground or power, but it should still be a very simple circuit to wire. With our specific hardware, one pin connects to ground (GND) on the Arduino while the majority of the remaining pins connect to Arduino →

QUICK TIP

As the specification for our display allows for 5V per segment, we don't need to use any resistors. But yours might not be the same.



Below For a common cathode display, Arduino digital pins 2,3,4,5,6, and 7 should connect with segments a,b,c,d, e,f,g respectively, plus ground



Above All seven-segment displays use the same lettering for the same segments, so that characters can be shared between them. Image credit: CC-BY-SA h2g2bob/ Wikimedia.org

digital outputs 2–7 via 220 ohm resistors (that prevent too much current flowing through the LEDs). The usual configuration sees the pins wired clockwise from the top, but this should also be described in your display's specification document. Don't worry if you can't make sense of which pin is responsible for which segment – see the 'Which segment is which?' box on page 83 for how to work this out manually.

Which leaves us with the code. Writing the code for lighting an LED connected to a digital pin on the Arduino has been covered many times before, and specifically in HackSpace #3. A single variable holds the pin number which is used as an argument within a function called **digitalWrite** to send either an on or off signal to the pin. We could approach a seven-segment display in exactly the same way, creating seven separate variables to hold the pin numbers and then writing seven different function calls to either turn on or turn off the specific elements within the display. And this is where both knowledge of the programming language and experience with design comes in, because computers and their programming languages are developed to solve this exact type of repetitive problem.

We'll start off by introducing an array. The good news is that if you've done any kind of programming before, you'll already be familiar with arrays. An array is a series of values, all of the same type, encapsulated into a single variable. By defining an array, you don't have to go through the lengthy process of creating and assigning values separately, plus the compiler that turns your code into a binary file can usually make more efficient use of an array. It can ask for ten consecutive chunks of memory, all of the same size, for example, rather than ten individual requests that may be scattered about in memory. The consecutive nature of data stored within an array is often reflected in the way a programming language will let you automatically step through one, or access values within an array via an offset.

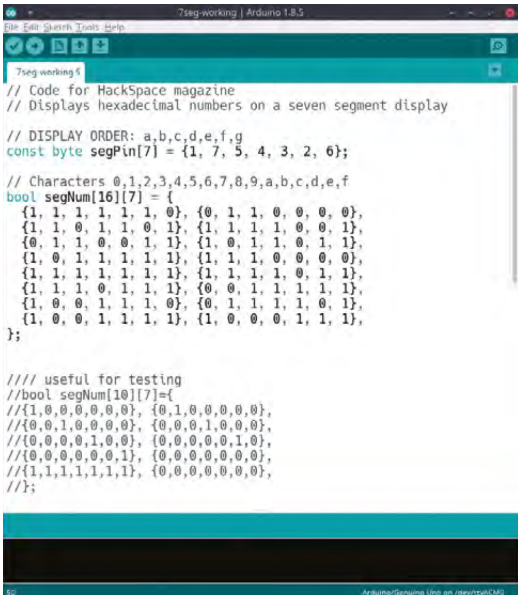
ARRAY WE GO

You define an array just like any other variable, except you need to specify the size of the array (so that the memory can be reserved for the correct number of values), and the values for each position within the array.

For example, the following creates an array called **segPin** that holds seven values, each of the type 'byte':

```
const int segPin[7]={1,7,5,4,3,2,6};
```

As you can guess, **segPin** holds the number of each Arduino digital pin that's connected to the seven-segment display, following the clockwise wiring of the segment order. Pin 1, for instance, is connected to the pin that activates the segment labelled 'a'. The reason why our example isn't a sequential set of numbers is


7seg-working | Arduino 1.8.5
File Edit Sketch Tools Help
7seg-working §
Arduino/Genuino Uno on /dev/ttyACM0

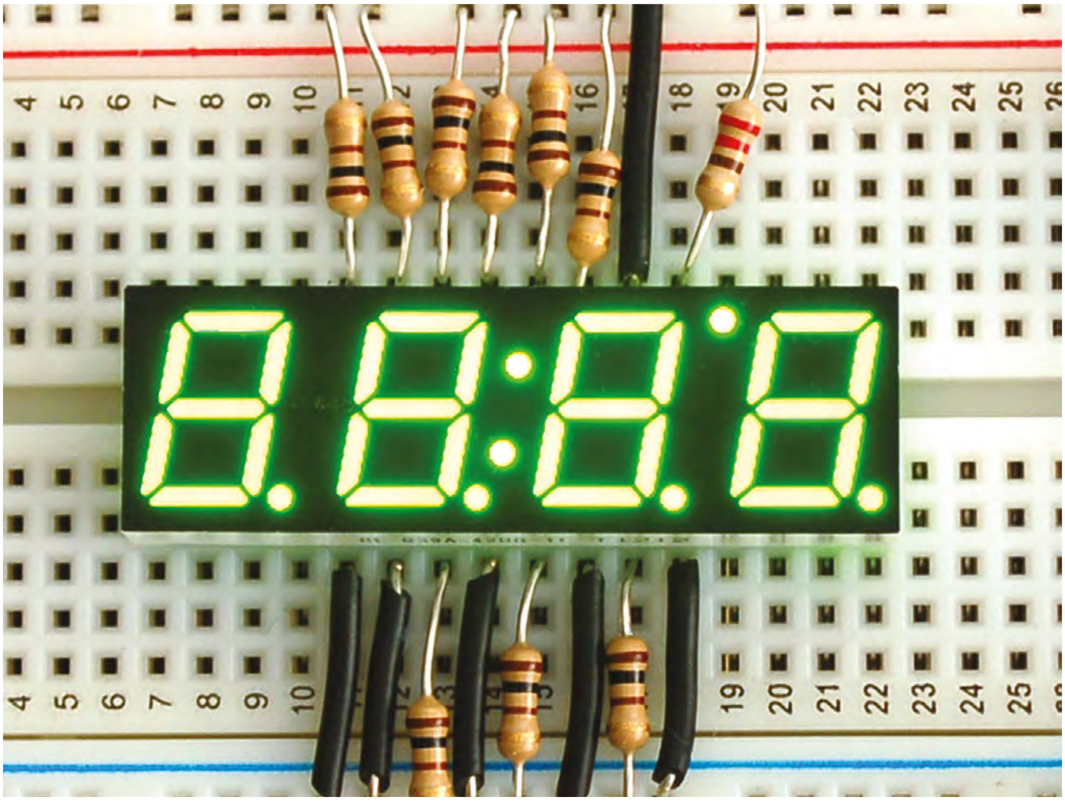
```
// Code for HackSpace magazine
// Displays hexadecimal numbers on a seven segment display

// DISPLAY ORDER: a,b,c,d,e,f,g
const byte segPin[7] = {1, 7, 5, 4, 3, 2, 6};

// Characters 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
bool segNum[16][7] = {
  {1, 1, 1, 1, 1, 1, 0}, {0, 1, 1, 0, 0, 0, 0},
  {1, 1, 0, 1, 1, 0, 1}, {1, 1, 1, 1, 0, 0, 1},
  {0, 1, 1, 0, 0, 1, 1}, {1, 0, 1, 1, 0, 1, 1},
  {1, 0, 1, 1, 1, 1, 1}, {1, 1, 1, 0, 0, 0, 0},
  {1, 1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 0, 1, 1},
  {1, 1, 1, 0, 1, 1, 1}, {0, 0, 1, 1, 1, 1, 1},
  {1, 0, 0, 1, 1, 1, 0}, {0, 1, 1, 1, 1, 0, 1},
  {1, 0, 0, 1, 1, 1, 1}, {1, 0, 0, 0, 1, 1, 1},
};


//// useful for testing
//bool segNum[10][7]={
//{1,0,0,0,0,0,0}, {0,1,0,0,0,0,0},
//{0,0,1,0,0,0,0}, {0,0,0,1,0,0,0},
//{0,0,0,0,1,0,0}, {0,0,0,0,0,1,0},
//{0,0,0,0,0,0,1}, {0,0,0,0,0,0,0},
//{1,1,1,1,1,1,1}, {0,0,0,0,0,0,0},
//};
```

Right Defining all your data at the start of an Arduino sketch makes it easy to find and update if your hardware changes



Left Multiple seven-segment displays may be multiplexed, and this allows them to run on fewer pins at the expense of more complex code

purely to do with the way we wired the circuit, and more organised builders would surely connect 1 to a, 2 to b, and so on. We, however, got our cables crossed at some point and this is reflected in the order of the array. If you wire them in order, simply replace the array with {1,2,3,4,5,6,7}. And because these pin assignments aren't going to change while the code is running, we've made the type 'constant', as covered in HackSpace #4.

An array can be used just like any other variable except that rather than using the name of the array alone, you need to also target a specific element within the array within square brackets. To set the pin mode of the first element within the array to 'OUTPUT', for instance, you'd use this:

```
pinMode(segPin[0], OUTPUT);
```

Infamously, arrays and lots of other sequential programming elements start at zero rather than one, so the above code is setting the pin mode of the first element (coincidentally, digital pin 1) to OUTPUT. So far, this is no different to using a regular variable. We could copy this line seven times and update the array reference number to run through the list of pins, just as we would with variables. But the array reference number is a clue. By making this a reference to another variable which we then increment to step through every element of the array, we can construct a much smaller and more efficient loop. Here's the code that does exactly that:

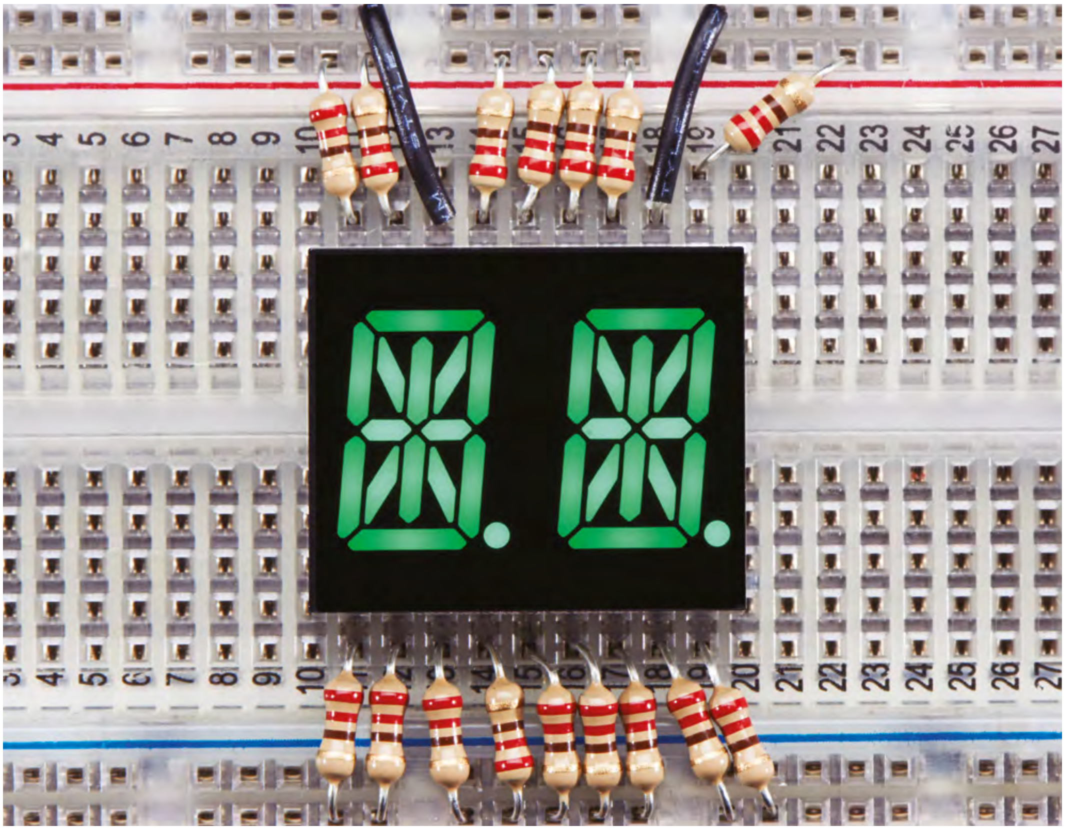
```
void setup(){
 for (int i=0; i<=7; i++){
  pinMode(segPin[i], OUTPUT);
 }
}
```

We've tucked the above code within the `setup()` function, as this is called automatically when your sketch starts. It's perfect for doing initialisation, such as setting pin modes, which is exactly what we're doing here. We've replaced the specific element value of the array, 0, with a variable called `i`. This variable is initialised within the arguments of the `for` command, which is probably one of the most common logical constructs of any programming languages. The `for` statement will simply repeat through the code that follows within the curly braces for as many times as defined by an increment counter initialised within the brackets. This initialiser always seems a little arcane, but regardless of language, it's only ever really saying, "take this variable, check it doesn't meet these requirements, and increment (or decrement) until it hopefully does."

In our example, we're creating the variable `i` with an initial value of 0. The `for` loop will then run while `i` remains less than 8 (our array holds elements 0 to 7, so the loop will stop before `i` gets to 8) and after each run will increment `i` by 1. This is what `i++` means; `++` and `--` are special kinds of operators, known →

QUICK TIP

A seven-segment display is actually capable of representing 127 different patterns – enough to create your own alphanumeric code!



Right ◈
Seven segments not enough? You can display a full range of alphanumeric characters on a 14-segment display

> A seven-segment display is capable of generating lots of recognisable output, **easily showing the numbers 0–9 and the characters a–f**

as compound operators, that take a single operand and either increment or decrement the value by 1. They're almost shorthand for `i = i + 1` or `i = i - 1` with one exception: if the `++` is placed after the variable, the variable is incremented after any evaluation. If the `++` comes before the variable, the variable is incremented before any evaluation. The following code should make this clearer:

```
1. i = 1;
2. j = i++;
3. j = ++i;
```

On line 2 above, `j` is assigned the value of `i` before `i` is incremented, making `j` equal to 1 while `i` equals 2. On line 3, `i` is incremented before any evaluation and then assigned to `j`, making both `i` and `j` equal 3.

The only code executed within the curly braces after the `for` definition is a single line, almost identical to the line we used earlier to set the mode of pin 0. The difference is an individual character where we've replaced the absolute value of 0 for the first element in the array with `i`. It shouldn't take much to guess that as `for` loops over each iteration of `i`, this value will step through 0,1,2,3,4,5,6, and 7, assigning all the Arduino pins we've configured to output with a single line. That's why arrays can be so powerful and why, as your projects become more complex, you can save yourself a lot of time and pain by simply choosing the best data structures. Such as the one in our next step – two-dimensional arrays!

THE SECOND DIMENSION

So far, we have used an array to store the pin allocation for the connections to the seven-segment display. The next step is to send on and off signals to the various elements within the display to create some meaningful output. As you will already know, despite being only a grouping of LEDs, the layout and design of these means a seven-segment display is capable of generating lots of recognisable output, easily showing the numbers 0–9 and the characters a–f. This perfectly corresponds to the base 16 or hexadecimal numeral system, with characters (a–f) representing the values (10–15) respectively, and this is what we're going to code our display to show.

We could easily use an array to store each of these 16 characters. For example, the following creates an array of type 'bool' to hold either an on (1) or off (0) value for each pin connected to the display:

```
bool segNum[7]={1,1,1,1,1,1,0};
```

If you were to display the above using a **for** loop similar to the one we created earlier, you would see the number 0, which you can guess because there's only one element not lit – the middle element of the seven-segment display.

We could go on and create arrays for every character we want to display along with **for** loops and functions to handle them. But this would be horribly inefficient and tedious to implement and maintain. You might think that we've already played the array card, but they have the answer once again. Just like a line on a single axis is said to have a single dimension, an array has a single dimension if it only has one set of elements. But like a line with two dimensions, x and y co-ordinates for example, an array can have two dimensions and even more.

Here is the code for an array with two dimensions, the first for the 16 characters we want the array to store and the second for the seven on/off pin configurations for each character:

```
bool segNum[16][7]={
{1,1,1,1,1,1,0}, {0,1,1,0,0,0,0},
{1,1,0,1,1,0,1}, {1,1,1,1,0,0,1},
{0,1,1,0,0,1,1}, {1,0,1,1,0,1,1},
{1,0,1,1,1,1,1}, {1,1,1,0,0,0,0},
{1,1,1,1,1,1,1}, {1,1,1,1,0,1,1},
{1,1,1,0,1,1,1}, {0,0,1,1,1,1,1},
{1,0,0,1,1,1,0}, {0,1,1,1,1,0,1},
{1,0,0,1,1,1,1}, {1,0,0,0,1,1,1},
};
```

As you can see if you follow the curly braces, the first set holds the outside array of 16 elements, each held within its own smaller seven-element array. You can even add more dimensions to an array, but like multidimensional space-time, these arrays become very difficult to conceptualise.

The only problem we now have left to solve is augmenting our **for** loop to handle all this interdimensional space. This is easy if we put the whole thing into its own function:

```
void displayNum (int number) {
   for (int i = 0; i < 8; i++) {
     if (segNum[number][i]) {
       digitalWrite(segPin[i], HIGH);
     } else {
       digitalWrite(segPin[i], LOW);
 }  }  }
```

The above code expands on the earlier **for** loop in several ways. Firstly, it's encapsulated the logic within a function. This means we can call **displayNum(4)** whenever we need the number 4 displayed, rather than repeating the same old code. Within the function, the **for** loop steps through a counter for each pin, only this time there's an extra **if** and **else** command. These reference our two-dimensional **segPin** array to check whether a pin should be set to on (HIGH) or off (LOW), and they do this using the two sets of the same square brackets used to create the array. Only this time, rather than setting the size of the array, they're used to reference a specific element. Keeping with our two-dimensional line theory, this is equivalent to a specific x and y location. The trick is that this location is defined by the number passed to the function, used to point at the character we want to draw, and the value of **i** which is being incremented by the **for** loop so that each pin can be set separately.

All that is now left to do is write the central **loop** function that the sketch calls automatically and use this to call the new **displayNum** function, ideally stepping through all the characters we can now step through on our seven-segment display:

```
void loop() {
  for (int i = 0; i <= 15; i++) {
    displayNum(i);
    delay(500);
} }
```

WHICH SEGMENT IS WHICH?

The schematics for elements like a seven-segment display can be difficult to follow. For this reason, you might find it easier to work out which pin goes where with a brute force approach. This is actually what we had to do and why the array that holds the order of pin connections is in a strange order.

The easiest way to do this is to take the code from this month's article and replace the two-dimensional array holding the characters with the following:

```
bool segNum[10][7]={
{1,0,0,0,0,0,0}, {0,1,0,0,0,0,0},
{0,0,1,0,0,0,0}, {0,0,0,1,0,0,0},
{0,0,0,0,1,0,0}, {0,0,0,0,0,1,0},
{0,0,0,0,0,0,1}, {0,0,0,0,0,0,0},
{1,1,1,1,1,1,1}, {0,0,0,0,0,0,0},
};
```

When you run this code, the seven-segment display should light up each element in order, a–f. You just need to change the pin array variable so that what you see follows the same order and then everything else will work automatically.

QUICK TIP

The code for this project can be found at the following URL: **git.io/vAS8Y**

Light up your wardrobe

Use the same simple circuit to add a splash of colour in three different ways

Sophy Wong

@sophywong

Sophy Wong is a designer, maker, and avid creator. Her projects range from period costumes to Arduino-driven wearable tech. She can be found on her YouTube channel and at **sophywong.com**, chronicling her adventures in making.

With a few modifications, one circuit can make many different projects. Here are three wearable ideas all built from one simple circuit. First, we'll make an LED headband from start to finish. Then, we'll use the same circuit to make both a *Star Trek*-inspired combadge, and a light up hoodie.

In all three builds, we'll play with capacitive touch and make a button using conductive fabric. This is a great technique for wearables, and the Gemma M0 from Adafruit makes it easy to incorporate into projects. Let's get started!

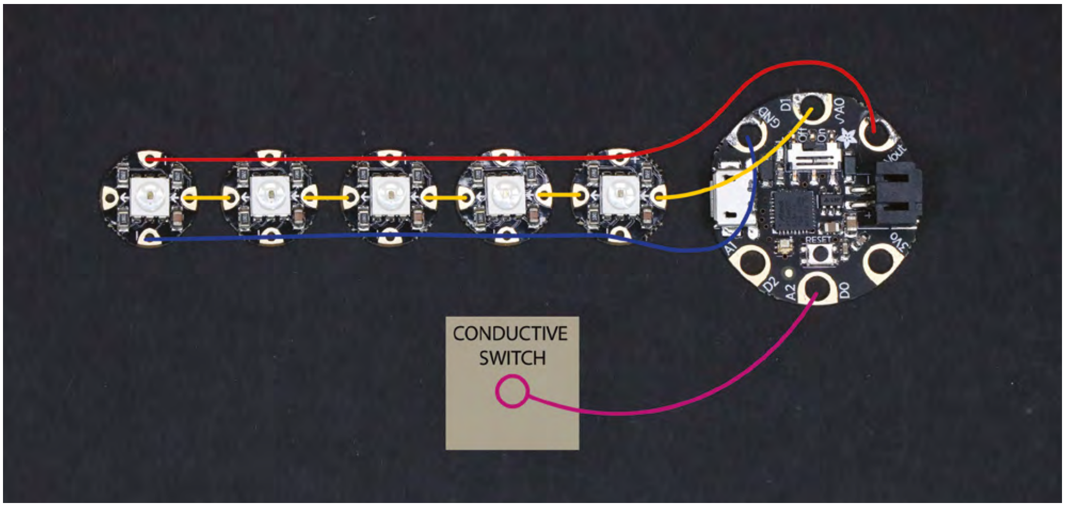
THE CIRCUIT

This circuit is basically a strand of NeoPixels controlled by a Gemma M0 microcontroller. It's simple enough to be used for many different projects, but we've given it a fun twist and added a soft 'button' made of conductive fabric. Touching the fabric changes the colour of the NeoPixels from red to green to blue.

NeoPixels come in several different forms: strips, circles, sewable singles, and more. In most cases, the different form factors are interchangeable. With simple projects like this one, you'll just need to update the number of pixels specified in your project's code.

The Adafruit Gemma M0 microcontroller is great for wearable projects, as it comes with an on/off switch and a JST battery connector on board. We'll make use of the Gemma M0's built-in capacitive touch function, and program it using beginner-friendly Circuit Python. →

Right NeoPixels are great when you've only got a small number of GPIO pins



YOU'LL NEED

- Gemma M0
- 3.7 V 150 mAh LiPo battery
- Conductive fabric
- Conductive thread
- Silicone-coated stranded wire
- Velcro
- NeoPixels for your project
- Soldering iron and soldering supplies

FOR THE LED HEADBAND

- 5 sewable NeoPixels
- Wide headband

FOR THE COMBADGE

- 2 NeoPixel sticks
- Pin back
- Thin craft foam

FOR THE LED HOODIE

- 10 sewable NeoPixels
- Hoodie
- Needle and sewing thread
- Clips or clothes pins

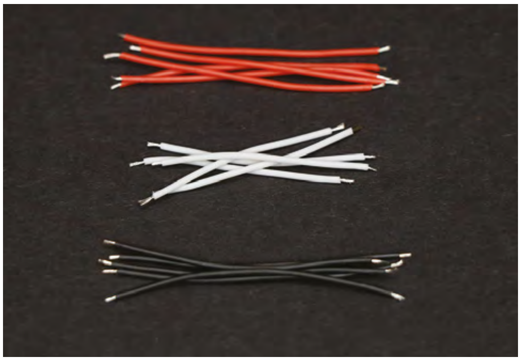
LED HEADBAND – BUILD THE CIRCUIT

We'll start by building a chain of NeoPixels. Notice the directional arrows on each sewable NeoPixel. All NeoPixels pass data in only one direction, so always look for these arrows when soldering strips, sticks, or single NeoPixels together. The arrows should always point in the same direction.

Cut five sets of power and ground wires in 3 cm pieces. Cut five sets of data wires in 2.5 cm pieces – these need to be shorter than your other wires or the strand will not lay flat. Using different colours of wire keeps your build organized, but for a sleeker look you can use just one colour of wire. Strip and tin the ends of each wire.

Solder one data wire to the in and out pins of the first NeoPixel. Solder two power and ground wires to their pins, as shown. This is the beginning of your chain.

Solder a wire from the data out pin of the first NeoPixel to the data in pin of the next NeoPixel. Then solder wires to the power, ground, and data out pins as with the first NeoPixel, and chaining the power and ground from the first NeoPixel to the second. Continue building the chain, making sure the arrows on the NeoPixels all point in the same direction.



ADD THE GEMMA M0

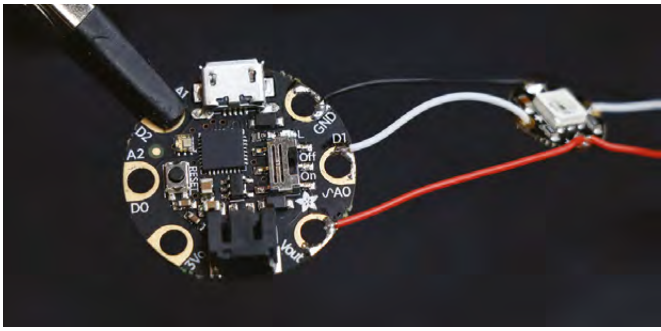
When you've completed your chain, it's time to add the Gemma M0 to the input side. Solder the connections as follows: Ground to GND, power to Vout, and data to D1.

MAKE THE CONDUCTIVE SWITCH

To make the conductive switch, start with a piece of wire about 6 cm long. Strip about 1.5 cm of insulation off of one end, and coil it around an awl or a skinny knitting needle to make a loop. Tin the loop with solder to keep it in shape.

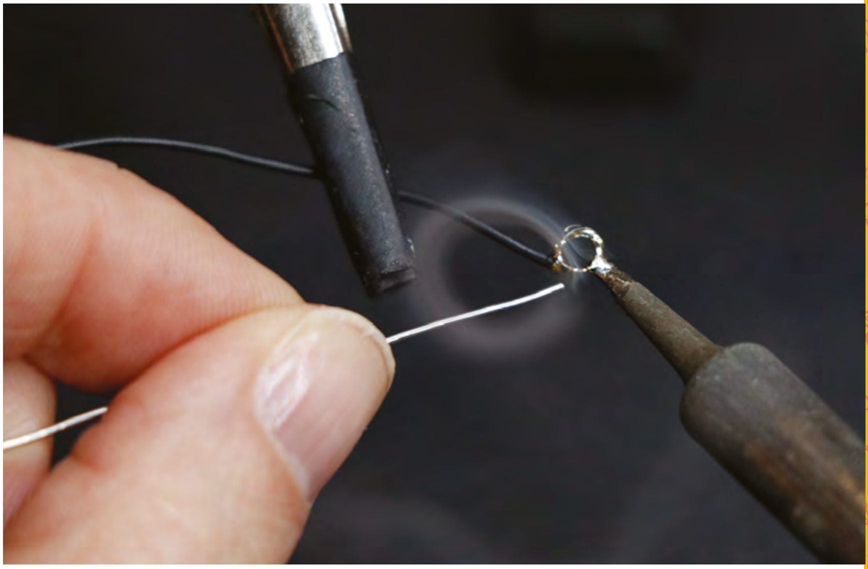
Cut a small piece of conductive fabric, about 1.5 cm square. Place the wire loop in the centre of the back side of the fabric and sew the loop onto the fabric using conductive thread. You can use regular thread for this too, just be sure to pull your stitches snug to keep the wire loop tight against the conductive fabric.

Your conductive switch is complete! Solder the end of the wire to A2 on the Gemma M0. Now we're ready to code.



Left Get everything ready before you start and you won't need to mess around with wire-cutters when soldering

Above The small size of the Gemma makes it a great choice for projects like these



Above
A drop of solder helps the loop on the conductive switch keep its shape

PROGRAM THE GEMMA M0

For in-depth setup help for the Gemma M0 and CircuitPython, visit **hsmag.cc/NCwBLc**.

Connect your Gemma M0 to your computer using a micro USB cable. The Gemma M0 appears as a drive called CIRCUITPY. You'll need to install the **neopixel.mpy** library for the code to work. The library is available from **hsmag.cc/AYaScW**, along with more information about using Gemma M0 and Circuit Python. Install the **neopixel.mpy** library by copying it into the **lib** folder on the CIRCUITPY drive.

Once your library has been installed, you can move on to programming the Gemma M0. With Circuit Python, you can do this by editing a text file on the board. Find the file on the drive labelled **main.py** and open it in your favourite text editor. Replace the text in that file with the text downloadable at **hsmag.cc/issue5** and save the file. This will update the board, and all of your NeoPixels should light up and be red.

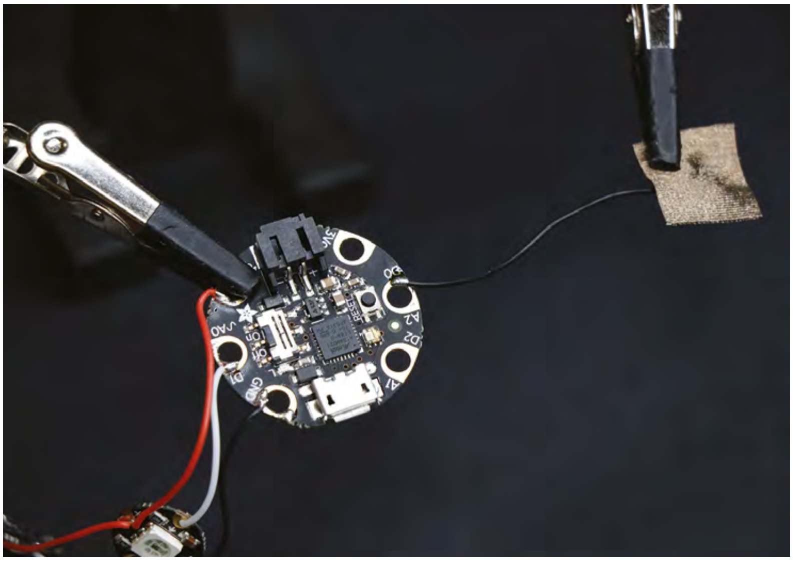
Tapping the conductive fabric switch should change the colour of your NeoPixels to green. Another tap should turn them blue. If the colours are cycling by themselves, or the capacitive touch doesn't behave as expected, try pressing the tiny black reset button on your board to recalibrate it.

Unplug the Gemma M0 from the computer (it's a good idea to eject the drive first). Cover the LiPo battery with electrical tape. This will help protect it from scrapes and sharp things, as well as create a barrier of insulation between the battery and the conductive fabric. Glue the conductive fabric to the front of the battery – don't use hot glue for this. Now plug the battery into the Gemma M0 and make sure everything still works.

PUT IT ALL TOGETHER

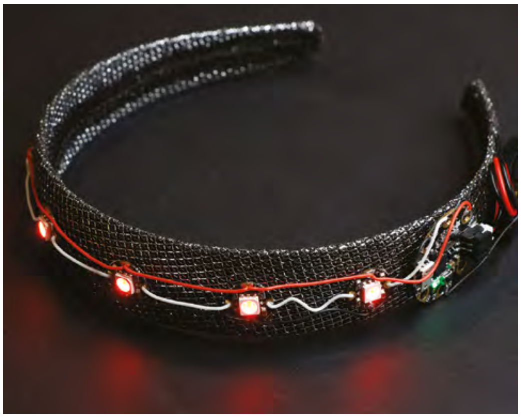
Centre the NeoPixels on top of the headband and glue them in place with hot glue. You can also use hot glue to attach the Gemma M0 at the side of the headband. Use adhesive-backed velcro to attach the battery to the headband. This makes it easy to replace the battery or remove it for charging.

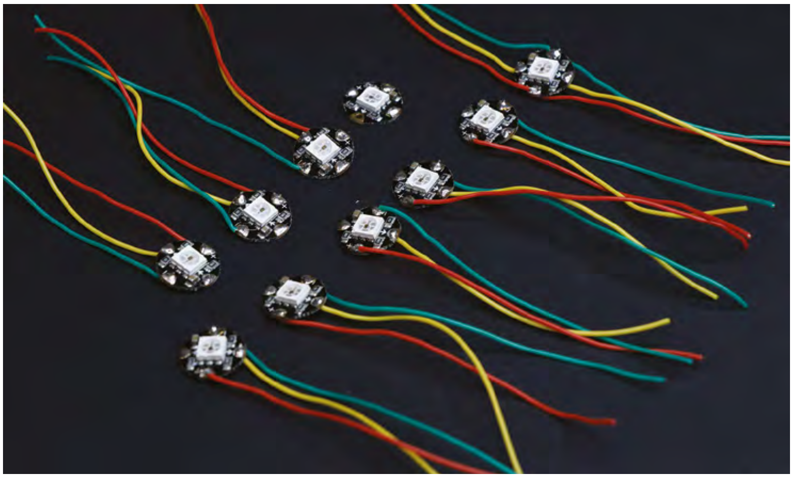
This headband is great on its own, or you can add material on top of the NeoPixels for more impact. Fabric flowers, feathers, or tulle work well for diffusing light. Just tap the conductive switch to change colour modes. Or, get creative with code and make new light animations! →



Left
Conductive fabric allows you to sew capacitive switches into loads of wearables

Below
You can choose wire to stand out or blend in, depending on your fashion taste





Above Soldering wires to each NeoPixel first can help avoid tangles

LIGHT-UP HOODIE

Although the size and shape of this project is very different, this project uses exactly the same circuit and components as the LED headband. We've simply doubled the number of sewable NeoPixels and sewn them into a hoodie. There's just one small change to make to the code to make this work.

Lay the NeoPixels out around the inside edge of your hood and decide how far apart you want to space them, about 7 cm or so works well. Build the NeoPixel strand as we did before, remembering to keep the data arrows pointing in one direction down the chain.

Connect the Gemma to the input side of the chain. Make the conductive switch and connect it to the Gemma M0, as described in the headband build. Glue the conductive switch to the battery as well.

MODIFY THE CODE

We can use the exact same code we used for the headband project, with one small change: we'll need to change the number of NeoPixels specified in the code. This is a common update to make when using code from NeoPixel tutorials and examples, so it's good to get comfortable with the concept.

Connect the Gemma M0 to your computer and program it using the method described in the headband build. Only five of your NeoPixels will light up. To fix this, open your **main.py file** and look for this line in the code:

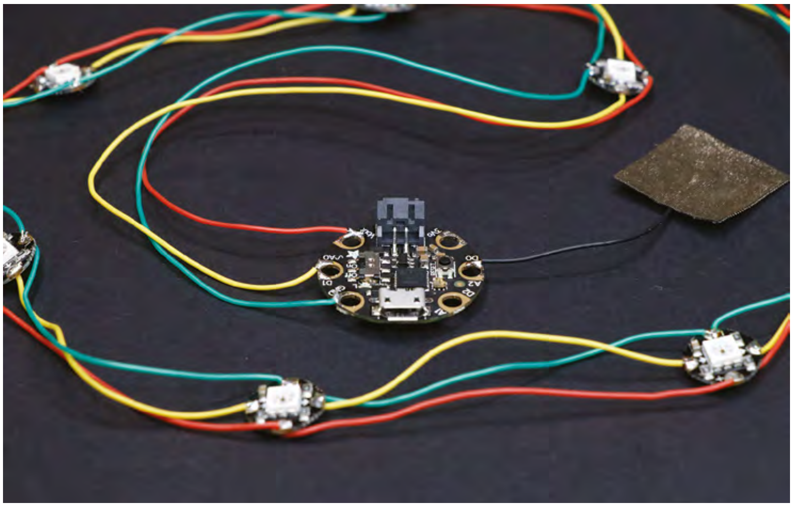
```
numpix = 5
```

This line creates a variable for telling the microcontroller how many NeoPixels we are using. Since we have doubled the NeoPixels in this project, change the five to ten. Save the file and your Gemma M0 should automatically update. Now, all ten of your NeoPixels should light up! When everything lights up and works correctly, you're ready to sew the circuit into your hoodie.

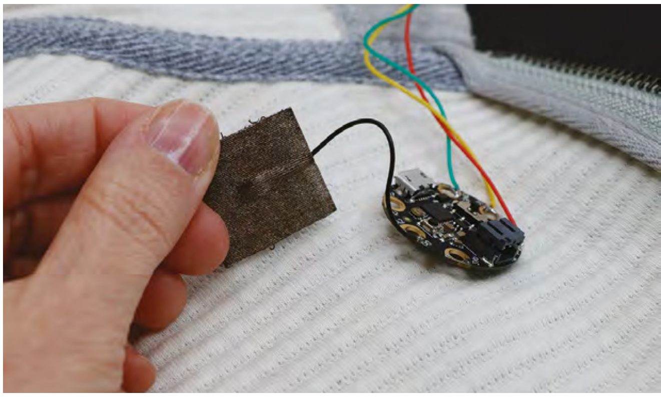
Use clothes pins to hold your circuit in place on the inside of the hood. Then hand-sew the strand of NeoPixels into place with strong sewing thread. Use large stitches about 1/2 cm apart. Consider your NeoPixel strand removable – to wash your hoodie, cut away your stitches to remove the strand for laundering. The rest of the circuit attaches with Velcro for easy removal.

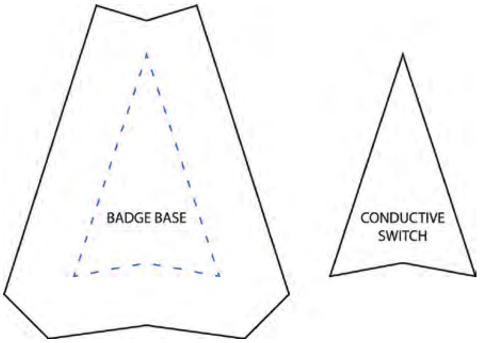
Cut three small squares of velcro loop (the soft side), one each for the conductive fabric pad, the Gemma M0, and the battery. Position the velcro squares near the end of the Gemma M0 so that the conductive switch is easy to reach, and the battery can plug into the Gemma M0 comfortably. Attach the Gemma M0, the battery, and the conductive fabric switch with the hook side velcro, and your light up hoodie is complete!

Below The hoodie uses the same circuit as the headband, but with longer wires



Below Attaching the Gemma and switch with velcro makes it easier to wash your hoodie





Above
Use this design in foam to make a firm base for your combadge

COMBADGE

This *Star Trek*-inspired combadge pin is a fun way to use the conductive fabric button. Tapping the centre of the badge changes the colour of the LEDs, which is satisfyingly next generation. The circuit is built in the same way as the previous two projects, but instead of chaining together individual NeoPixels, we'll combine two NeoPixel sticks to make two sides of the triangular badge.

Cut both triangles out of foam, and cut the smaller triangle out of conductive fabric as well. Use the same method described in the headband tutorial to create a sew loop in a small piece of wire, and attach it to the back of the conductive fabric triangle.

Glue the conductive fabric to the smaller foam triangle, making a hole for the wire to pass through to the back side.

Chain the two NeoPixel sticks together in the same way that we chained the individual sewable NeoPixels together: solder wires between the sticks to connect power, ground, and data (data out to data in). Solder wires to the input side for data in, ground, and power, but don't connect the Gemma M0 just yet.

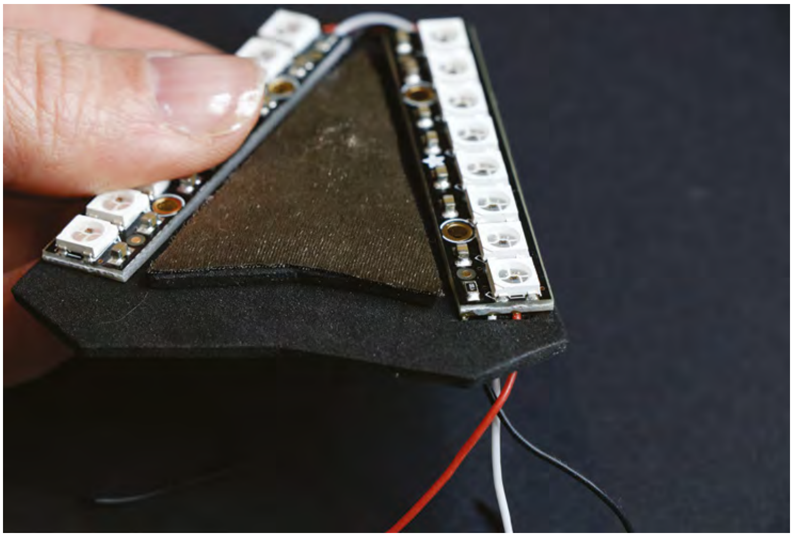
Glue the small triangle in place, making a hole for the wire to pass through to the back of the large triangle. Glue the NeoPixel sticks in place, making another hole for the wires to pass through to the back.

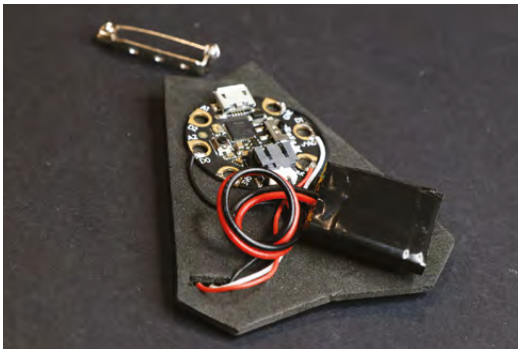
On the back of the badge, solder the input wires to the Gemma. Attach the Gemma M0 with hot glue. Use small squares of adhesive-backed velcro to attach the battery to the badge, and plug it into the Gemma M0. Then, attach a pin at the top of the badge with hot glue.

The code for this project is exactly the same as the code we've been using, but again, we'll need to update the number of NeoPixels specified in the code. Go back into your code and change the 'numpix' value to 16. Save the file, and your badge should light up completely.

Pin the combadge on your chest and enjoy tapping it like they do on the Enterprise NCC-1701-D!

What else can you make this universal circuit into? Show us your builds at **@HackSpaceMag**! □





Above
The conductive fabric should fit snugly between the NeoPixels

Left
Select the right-sized LiPo and everything will fit behind the badge

Below
Ready to boldly explore the universe – just don't wear a red shirt, obviously

Connect to your fellow makers with CheerLights

Learn to use Python and collaborate in the biggest IoT project in the world

Bruno Santos

@feiticeir0

Linux and Raspberry Pi lover, systems administrator, programmer, kind of a nerd! Loves to create things that may or not work, but could help the world someday!

YOU'LL NEED

- **A Raspberry PI** (any will do – we're going to use a Pi Zero W)
- **NeoPixel** (a stick or a strip – we're going to use a NeoPixel stick)
- **3 × jumper cables, female-female**

We're going to dive into IoT and collaborate in one of the biggest IoT projects on the internet – CheerLights.

An IoT project created by Hans Scharler, CheerLights is a network of devices people have created that set their colour based on an API. You can see the current colour at **hsmag.cc/dYalRJ**. This can be updated by anyone via Twitter: just tweet to **@cheerlights**, including a colour in your tweet, and suddenly devices all over the world will change to that shade. It's part art project, part a way of learning about the Internet of Things, and part a way of feeling a little more connected to the global maker community.

You can use almost any hardware with a network connection and the ability to change colour in a CheerLight, but we're going to use a Raspberry Pi and a strip of eight NeoPixels.

NeoPixels are RGB LEDs that include a simple controller and networking so that you can connect a strip of them to a single GPIO pin and drive it. The actual networking protocol is a little complicated, but there are libraries for most common maker controllers including the Arduino and Raspberry Pi. We're going to use Adafruit's NeoPixel library that you can download and install from **hsmag.cc/kWkKii**.

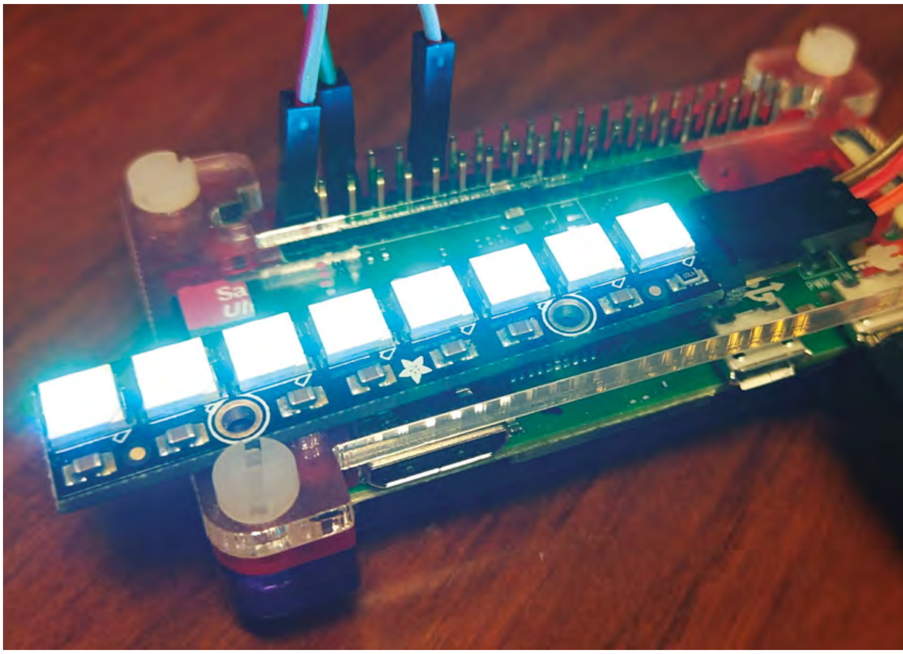
This library allows quite a lot of control over the lights – you can change the colour of individual lights in the strip in any order you like to create quite complex effects using the `setPixelColor(pixel_number,color)` method. We're using this to set all the NeoPixels to the same colour, but for other projects you could use any combination of colours you like.

With NeoPixels, you need to be a little careful of power draw. Each pixel contains three LEDs, each of which can draw up to 20 mA. While this isn't a lot, it can quickly add up. We're using eight pixels, which can draw up to 480 mA. This can be drawn from

Right Cheerlights running on two different projects. It's up to your imagination to come up with ideas for enclosures

GETTING COLOURFUL

With the CheerLights API, we can get the latest colour in two ways: hexadecimal value or the name of the colour. Each one can come in three possible formats: JSON, XML, or text.



Above Raspberry Pi Zero W with Adafruit NeoPixel stick

the 3.3 V pin on the Raspberry Pi, but if you wanted to add many more you'd need a dedicated power supply, and large displays can require quite careful power management.

The code for our project is:

```
# import libraries
import time
from neopixel import *
import requests

# define color_of_strip
# taken from NeoPixel library
def stripColor (strip, color):
    for i in range(strip.numPixels()):
        strip.setPixelColor(i,color)
        strip.show()
        time.sleep(50/1000.0)

strip = Adafruit_
NeoPixel(8,18,800000,10,False,255,0,ws.WS2811_
STRIP_GRB)
strip.begin()
stripColor(strip,Color(0,0,0))

# store old color
oldcolor = ‘’

try:
    while True:
        # get cheerlights color using requests
        lastcolor = requests.get(‘http://api.
thingspeak.com/channels/1417/field/2/last.txt’)

        # store request text value
        hex = lastcolor.text
        # remove the # using ltrip
        hexcolor = hex.lstrip(‘#’)
        r,g,b = bytearray.fromhex(hexcolor)

        if oldcolor != hexcolor:
            stripColor (strip, Color(r,g,b))
            oldcolor = hexcolor

        time.sleep(10)
except KeyboardInterrupt:
        print (‘Exit’)
        stripColor(strip,Color(0,0,0))
```

That's all there is to it. Go out and create your own CheerLight and join this colourful worldwide community. □

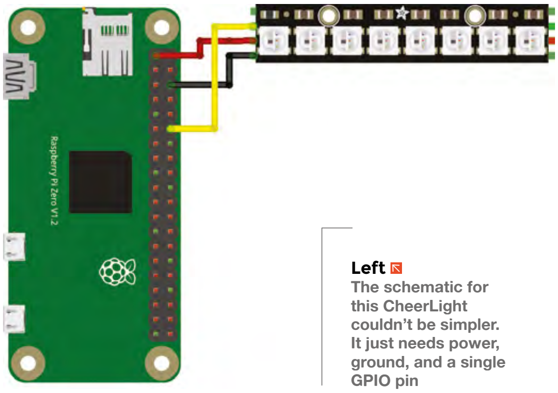
QUICK TIP

To set a colour, just tweet @cheerlights, or include the word 'cheerlights' somewhere in the tweet with the name of one of the possible eleven colours.

POKING THROUGH PROXIES

If you're behind a proxy server, requests can also use a proxy server. Just set a variable named `proxies` (you can name it whatever you like) and define your proxy server.

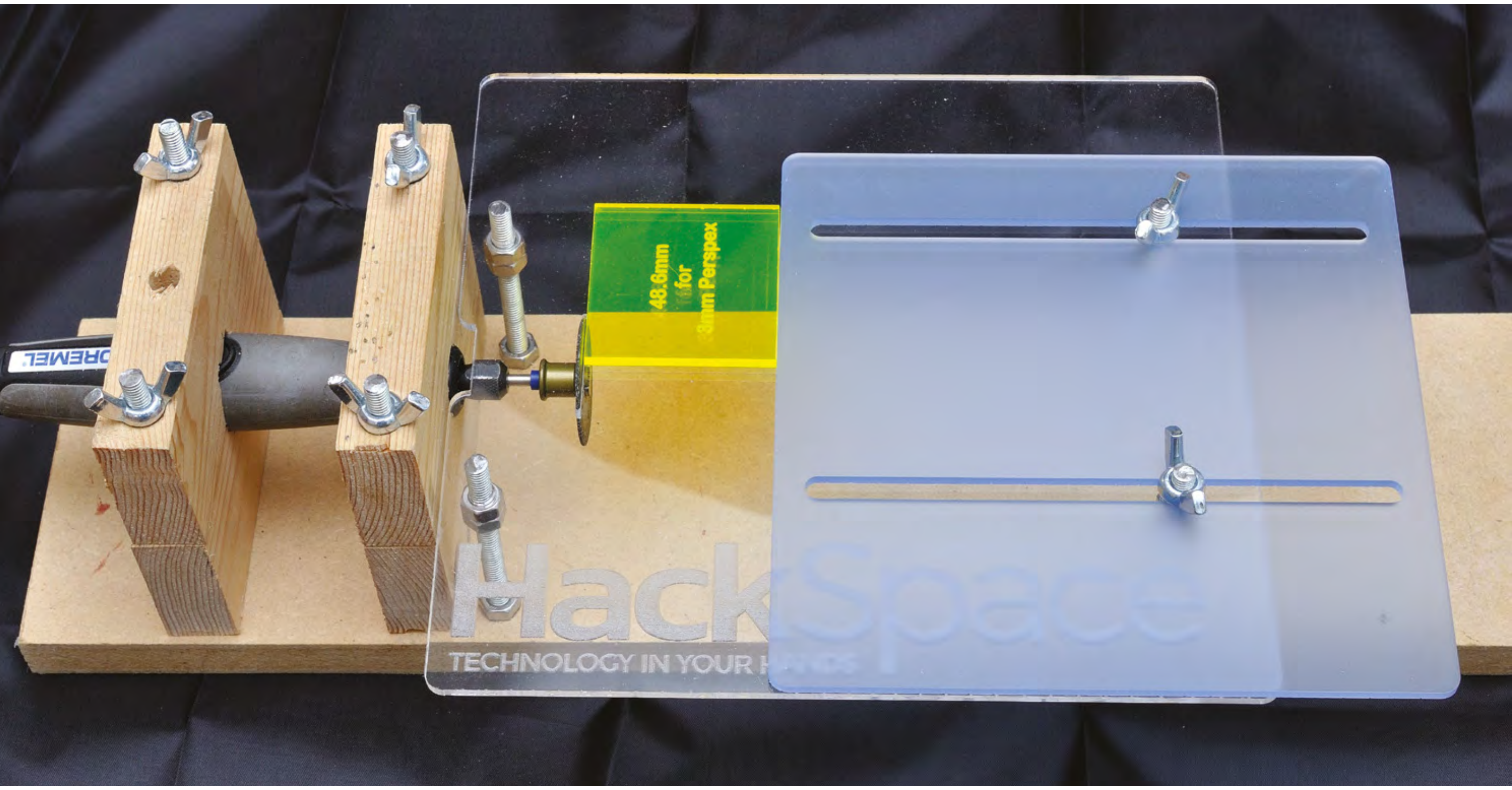
```
proxies = {
        ‘http’   : ‘http://<your_proxy_server>:<port>’,
        ‘https’  : ‘https://<your_https_proxy_server_ip>:<port>’
}
Replace the requests.get instruction with this one (replace the url):
r = requests.get(<api_url_in_the_code>,proxies=proxies)
```



Left The schematic for this CheerLight couldn't be simpler. It just needs power, ground, and a single GPIO pin

Make a Dremel table saw

Upgrade your multi-tool to make perfect, straight, repeatable cuts



Alex Eames

RasPiTV

Alex Eames loves making things and regularly blogs/vlogs at RasPi.TV He makes a living designing and selling RasP.iO products.

Accurately cutting small female pin headers, PCBs, threaded rods and suchlike with a hand-held Dremel can be difficult, hand-achingly tedious, and a bit uncontrolled.

Using just a piece of MDF, a small amount of wood, and some bolts, you can clamp down your multi-tool to keep it rock-steady. This is also brilliant for grinding and polishing small items.

Then add some 3mm acrylic (or thin MDF), threaded rods and nuts and you can go the whole hog, converting your multi-tool into a small table saw. Just add a fence for perfectly repeatable, straight cuts.

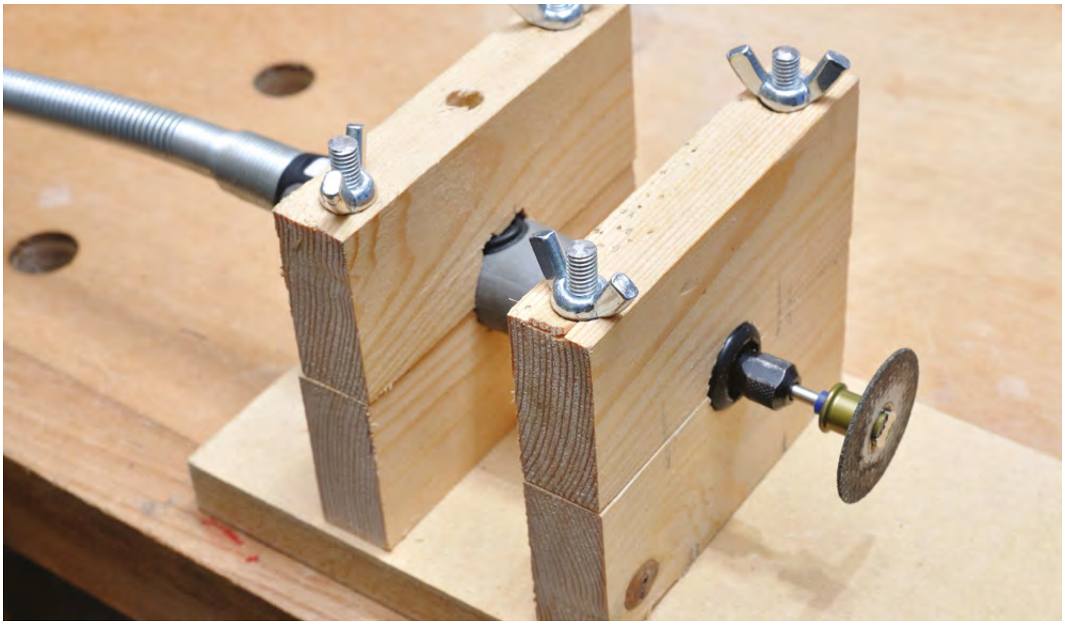
If you already have a multi-tool, you can convert it to a small table saw for about £8-worth of hardware, £8-worth of materials, and some workshop time.

Above
Your completed tool might not be quite as powerful as a full table saw, but it can make precision-cutting thin material easy

LET'S GET STARTED

You'll need to make two wooden clamps to hold your multi-tool down on the base. One should go close to the spinning end of the multi-tool, but clear of the collet/chuck. The other clamp should be a few cm further back along the tool at a convenient place to match your available drill sizes. If you have a Dremel flexible shaft adaptor, the holes can be exactly 19mm (3/4") and 25mm (1").

Drill out the large central holes for the two clamps by clamping or taping the two wood pieces together and drilling through the centre. Your drill bit should



Left Ensure the wood securely holds your Dremel in place

be aligned so the cut spans both pieces of wood perfectly symmetrically, with the centre point of the drill right in the middle.

YOU KNOW THE DRILL

You're going to use lengths of M6 threaded rod to hold the clamps to the base and retain the Dremel securely, so we need to drill holes for this. With the two wood pieces correctly aligned and clamped or taped together, drill 6 mm holes (~1 cm from each end) right through both pieces. Do this for both clamps. If you have a long enough drill bit, you can do this in one operation. Otherwise drill as far as the bit will allow, then remove the top part, then carry on through the bottom part.

Next, place the bottom half of the first (outer) clamp on the base about 4 cm from the left-hand end. Trial-fit the Dremel and use it to place the bottom half of the other clamp correctly. Once satisfied, use a few blobs of hot glue to hold the bottom halves to the base temporarily. Drill four 6 mm holes through the MDF base, using the bottom halves of the wooden clamps as a guide.

Gently remove the clamps, flip the base over, and counter-bore the four holes you just made using a 9.5 mm (3/8") diameter bit to a depth of 5–6 mm. If you have a 3/8" Forstner bit, these make a lovely flat-bottomed hole. If not, a normal drill bit will do. These four recesses will each hold an M6 nut.

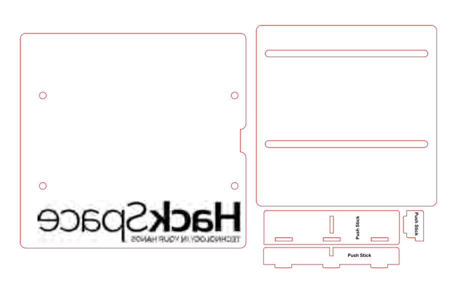
CAPTIVATING NUTS

The 9.5 mm hole is deliberately a bit too small for the nut. This lets us make it a 'captive nut' so it won't try to rotate when in use. **Here's how to insert it in the MDF base…**

PROJECT FILES

The .DXF and .SVG design files for the acrylic (or MDF) table parts can be found at **hsmag.cc/cfOFsq**. They are eminently tweakable to suit your own needs. If you're using a different multi-tool or have no flexible adaptor, you might want to adjust the table cut-out to better fit your tool.

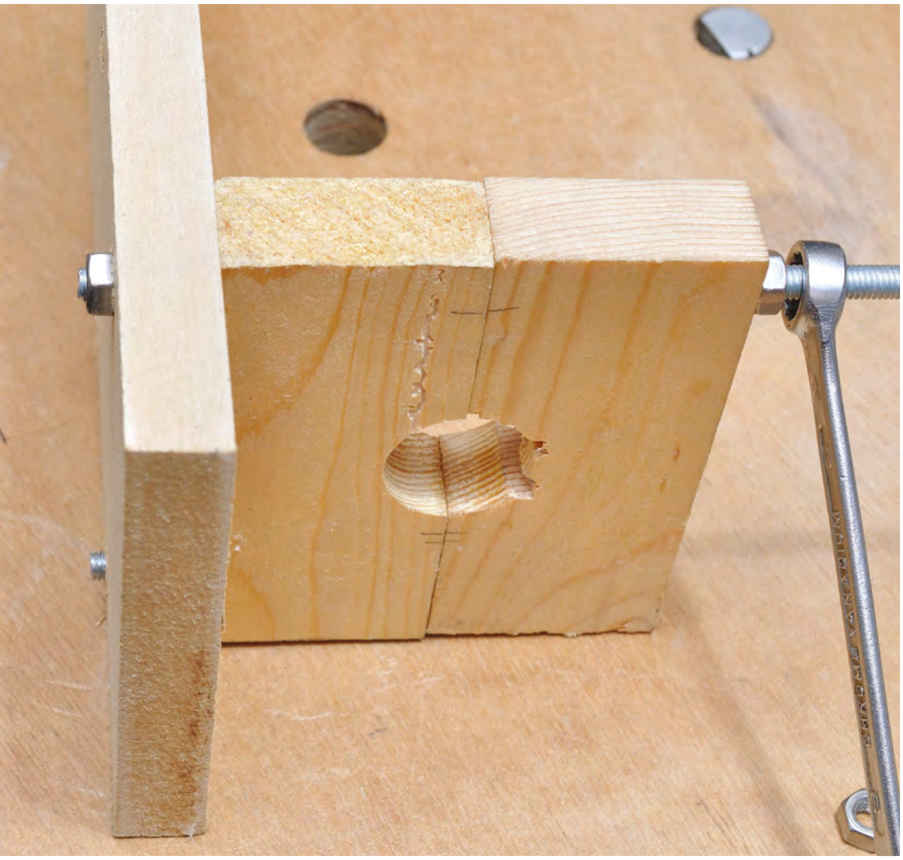
See below for the default version: a layout of the table and fence SVGs…



Left Securely clamp your wood before drilling to ensure clean holes

YOU'LL NEED

- MDF 40 × 11 × 1.2 cm
- 4 × Wood 10 × 4.4 × 1.8 cm
- 4 × M6 threaded rod 30 cm
  screwfix.com
- 20 × M6 nuts
  screwfix.com
- 6 × M6 wingnuts
  screwfix.com
- 1 × M6 washer
  screwfix.com
- 2 × A4 sheet of 3 mm Acrylic or thin MDF
  hobarts.com
  kitronik.co.uk
- Dremel or multitool (ideally with a flexible 'snake' adaptor)
- Drill + 6 mm (1/4"), 9.5 mm (3/8"), 19 mm (3/4"), 25 mm (1") drill bits
- 10 mm spanner, adjustable spanner, saw, glue gun, hacksaw, file, clamps and general workshop tools
- Router with 1/4" cutter or laser cutter (for cutting slots in acrylic fence)



Above You can use this style of clamp to hold all sorts of round objects securely in place

QUICK TIP

To mark your clamp block for carving, use a felt-tipped pen on protruding parts of the Dremel, then press the blocks together, thus marking them.

Shorten the threaded rods so that they stick up proud of the clamps by only ~15 mm.

Once cut, you may need to dress the ends of the threaded rod with a file. Then use M6 wingnuts to secure the clamps conveniently.

- Take one of the M6 threaded rods
- Thread an M6 nut onto the end so it's completely on the rod
- Push the other end through the underside of the base
- Add the two halves of the wooden clamp (this aids alignment)
- Place an M6 washer over the rod and on top of the wooden clamp
- Thread an M6 nut all the way down to the washer
- Tighten the top nut with a 10 mm (or adjustable) spanner until the nut at the bottom of the base has been pulled completely into the base
- Make sure that neither the bottom nut nor the threaded rod end are protruding from the base (or your work surface will be scratched)
- Repeat for each of the remaining three clamp holes

Now trial-fit your Dremel or extension. If necessary, you can make any adjustments to the clamping blocks by carving out excess material for a perfect fit. If it's loose, or isn't secure, don't continue until it's fixed. If the tool comes out when running it could be dangerous, so make sure this isn't going to happen.

Once you have the main table top part, place it on the base, lining up the cut-out on the left-hand side with the Dremel. It should be touching the wooden clamp, but not overlapping the Dremel. Mark the locations of the four holes and disassemble the clamps ready for another round of drilling and counter-boring.

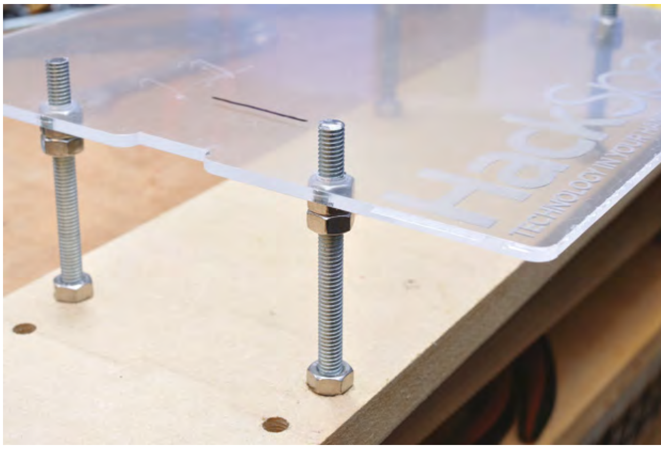
Drill four 6 mm holes through the base in the marked places. Then counter-bore to 5–6 mm depth with the 9.5 mm drill and embed an M6 nut in each of the counter-bores using the same method as before.

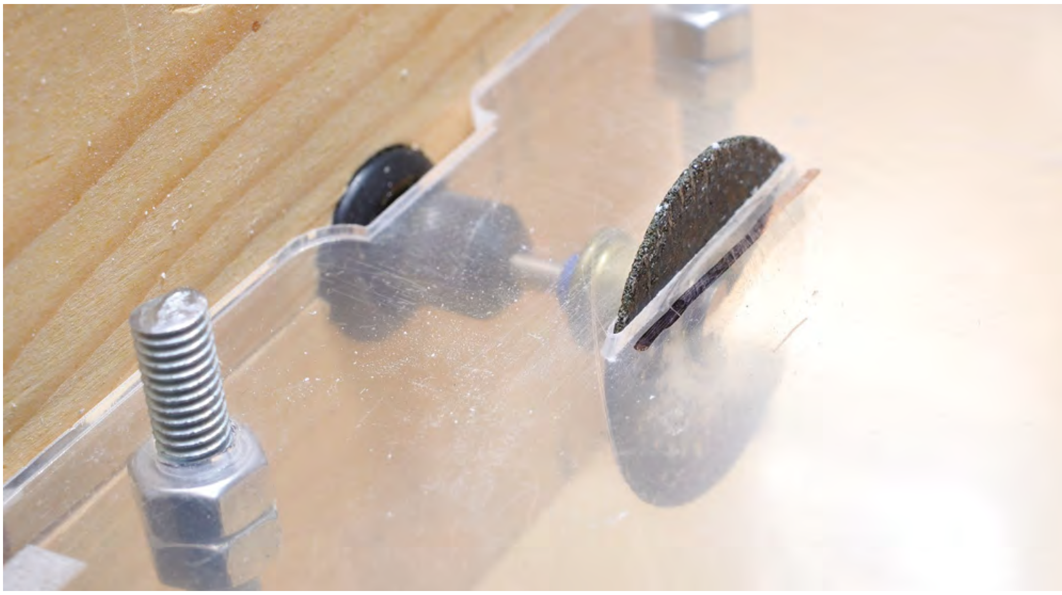
If you're following the plan exactly, you will need four M6 threaded rods 8.5 cm long for the table supports. If you're changing the design, the length will change accordingly.

MULTI-THREADING TIME

Thread an M6 rod into each of the four table support holes and add an M6 nut to 'lock' the rod in place. Don't fully tighten it yet. Repeat for all four table supports. Thread a further two M6 nuts onto each rod. The top of the uppermost nut should be 50 mm from the base. Tighten the two nuts together to lock them in position. This sets the table height 50 mm above the base.

Below The 'table' for our mini table saw is held in place with M6 thread and nuts





QUICK TIP

If one of the threaded rods gets in the way when you're cutting, shorten it, remove it or lower it down through the base.

Left The encased blade is held securely for more accurate cuts

Check that the table top fits over and lines up correctly with the threaded rods. If needed, you can push and 'persuade' the threaded rods slightly into place to correct small alignment errors. Once positioned correctly, tighten the bottom lock nuts at the base, ensuring the rods are not sticking out underneath.

Remove the table top, reinstall the Dremel clamps and Dremel, with cutting disc fitted.

Now place the table top back on the supports but not in contact with the cutting disc. Switch on the Dremel to a medium speed. Now use the cutting disc to cut its own 'zero clearance' slot in the table top by gently pushing the table top down onto the supports. It may be necessary to stop part-way through and clear out any accumulated debris. Repeat this process until you have cut a slot for the disc. Avoid contact with any other part of the Dremel other than the disc. Set the table height by adjusting the top two nuts on each support as required.

NOT LASER-CUTTING?

3 mm acrylic laser cuts beautifully, but can also be worked very nicely with woodwork tools if you prefer. The table top could just as easily be made out of MDF, but it must be fairly thin or you won't get much cut depth with your 38 mm diameter Dremel discs.

If you're not laser-cutting the table parts, print out the files and use them as a template for drilling and cutting.

If you're not laser-cutting the fence part, you'll need to rout the two slots carefully (using a fence, guide or router table) with a 6 mm or 1/4" bit. It's advisable to do this in two passes. If you try to cut full depth in one pass, breakages can occur. (Been there, done that!)

> If you're following the plan exactly, you will need four M6 threaded rods 8.5 cm long for the table supports. **If you're changing the design, the length will change accordingly**

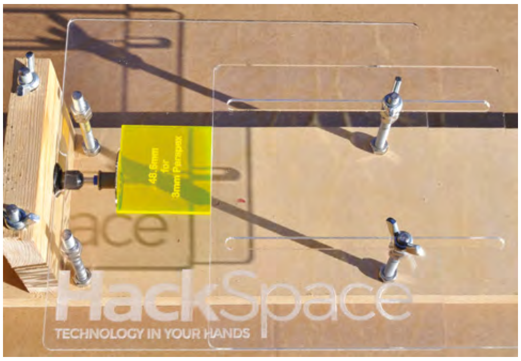
Peel off any protective film from the acrylic and enjoy the glorious look of your shiny new table surface.

In use, you should set the final table height so that enough of the disc is protruding for you to make your cut.

ADD A FENCE

Next attach the fence using the two right-hand table support posts and two M6 wingnuts. The fence position can be adjusted according to your required cut length.

If you want to use a push stick to protect your fingers, there's a 3-piece 'slot-and-glue' design in the project files: **hsmag.cc/cfOFsq**. Enjoy the Dremel table saw and may your cuts be ever straight and true. □



Left You need never have wonky cuts again with your encased Dremel table saw

Slack-controlled On Air sign

Build your own Particle Photon-powered 'On Air' sign that can be controlled via Slack, to alert friends and colleagues that you are busy

Wesley Archer

silktide.com

Digital Account Manager for Silktide by day, and maker in his spare time. Also a regular contributor to The MagPi Magazine.

We used a local company to create a custom vinyl sticker for the sign

The sign, including colour, is controlled via slash commands sent via Slack

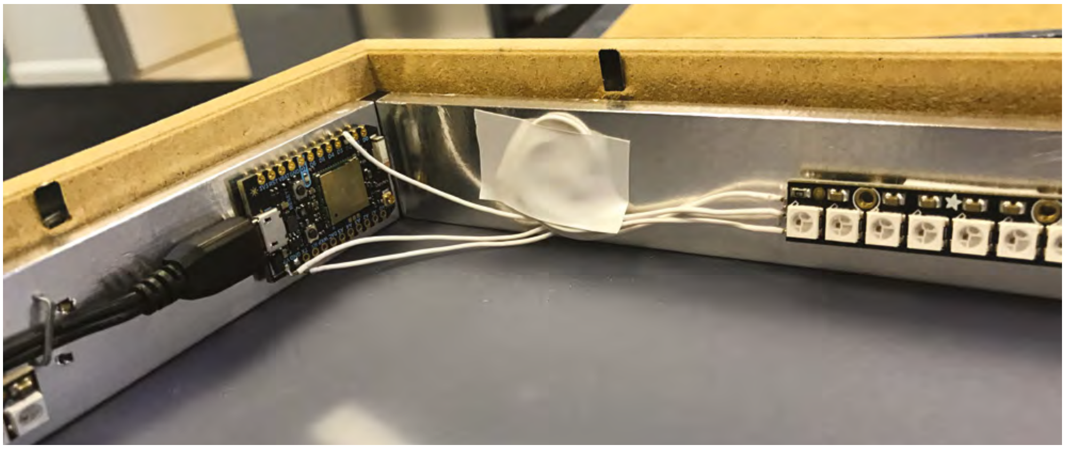
In a busy office, it can be difficult to let your colleagues know that you are on an important phone call other than 'shushing' them, which is not very polite. This is particularly challenging when you have a team of developers working on the floor above, who regularly come to the kitchen on the ground floor to get their caffeine kick or lunch! In this guide, we will show you how you can build your own 'On Air' sign, powered using a Particle Photon and controlled via Slack.

STEP 1
PREPARE THE FRAME

The RIBBA frame is cheap and easily available from IKEA. We did not need the glass front, as our custom-sized, frosted-acrylic cover would replace it. The vinyl sticker was designed in Photoshop and printed by a local sign company, then applied to the acrylic to complete the look. To maximise the light output, we applied some foil tape to the inside of the frame, and added a 20 × 20 cm acrylic mirror, found on eBay, to the back plate so all the light would be reflected out through the lettering, making the sign more visible.

STEP 2
CONNECT YOUR NEOPIXEL STRIPS

The NeoPixels strips are connected in series using three wires (GND to GND, DOUT to DIN, and 4-7 V to 4-7 V), then stuck to the frame using double-sided foam pads. The last NeoPixel is connected to the Particle Photon. We used the D2 pin for the data (any D, or digital, pin can be used, but be sure to update the code to use the pin you chose), along with the GND and 5 V pins. We tried to keep the length of wire used to a minimum, as we did not want too much excess wire inside. You can just use



Left The Particle Photon is attached using a double-sided foam pad, and wires taped in place

CODE

Code
onAirSlackLambdaTrigger.js

Language
JavaScript

GitHub
hsmag.cc/WymPgM

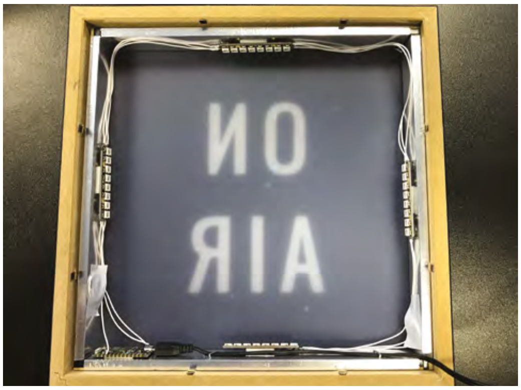
some electrical tape to safely secure the wire to the frame, as we did.

STEP 3
FLASH THE CODE TO THE PHOTON

Thankfully, there is a very good NeoPixel library on the Particle Photon IDE already. We adapted this to make it work with our sign, including set colours and a breathing pattern when the sign is on. You can change the colours by adjusting the values in the code, (you'll need the **breathing-slack-on-air.ino** file from our GitHub repo), but we chose simple colours for ours. You can flash the code as per the official documentation: **hsmag.cc/UcJwkA**. Once uploaded, the NeoPixel sticks should be ready.

STEP 4
SET UP SLASH COMMAND IN SLACK

You can add custom integrations with Slack – it's one of the best things about the tool – and we needed to create a custom slash command (which is called like /somecommandhere in any Slack chat). You can create these commands at **hsmag.cc/WmcnHT**. You will be given a token (needed in the next step), and you will be asked for a URL that will perform the action of the command. This can be any web app, but we'll host ours as an Amazon Web Services (AWS) Lambda function (see next step). This is a commercial service, but you should stay in the free tier provided you don't switch the sign excessively. Set the method to 'POST' and save the integration.



Above The NeoPixel sticks are attached to the sides, then connected to the Photon

> "The vinyl sticker was designed in Photoshop and printed by a local sign company, then applied to the acrylic **to complete the look**"

STEP 5
THE LAMBDA FUNCTION

In order to control the Photon via Slack, we need to configure a Lambda function to translate it. Ours was hosted on AWS and can be found in the GitHub repo (called **onAirSlackLambdaTrigger.js**), but be sure to update the relevant parts in the code, such as your Slack token, your Particle device ID (found in the Particle IDE) and your Particle Access Token (also found in the Particle IDE). You will need the URL of the function, so copy that and add it to the URL field in the slash command from the previous step.

STEP 6
CHECK THAT IT WORKS!

By now, you should have a custom slash integration within your Slack account and a Lambda function to provide the translation. If all has gone to plan, you can now control the On Air sign by typing in your slash command (e.g. /sign) followed by a colour, for example '/sign red' in any Slack channel. Our code includes commands for various colours ('on' (white), red, blue, green, yellow, purple, and 'off'), but feel free to change the code and corresponding commands to suit. □

YOU'LL NEED

- **Particle Photon** store.particle.io/
- **23 × 23cm RIBBA frame** (IKEA)
- **4 × NeoPixel Sticks (8 LEDs per stick)**
- **Micro USB PSU** 5 V 2 A for a Raspberry Pi works just fine
- **Prototyping wire**
- **23 × 23 cm frosted acrylic square** online / eBay
- **Vinyl lettering**
- **Our code via GitHub:** hsmag.cc/hutnnZ

Sheet bend: The most useful knot

One knot that can perform most tasks you ever need

Ben Everard

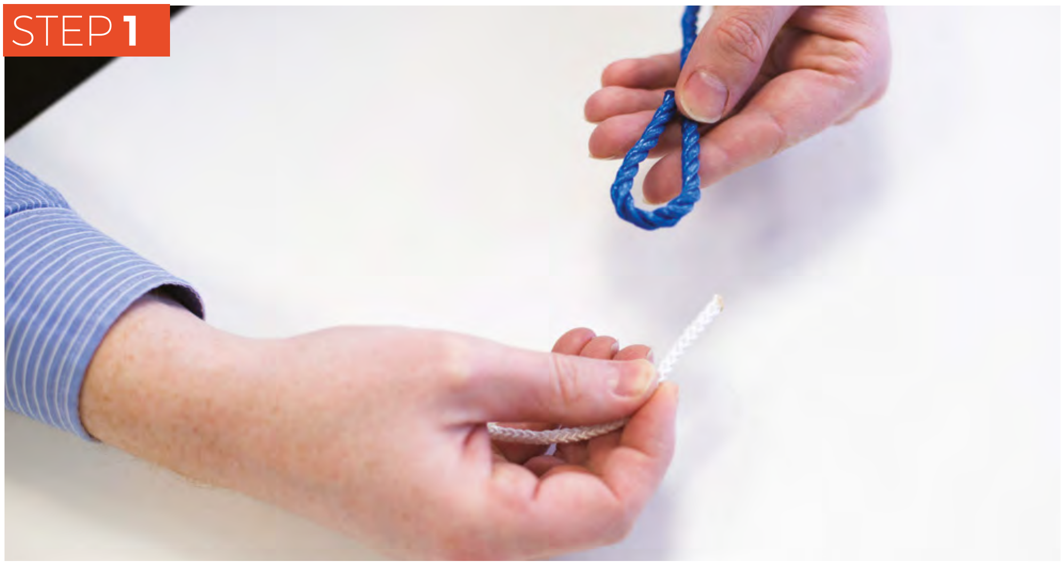
@ben_everard

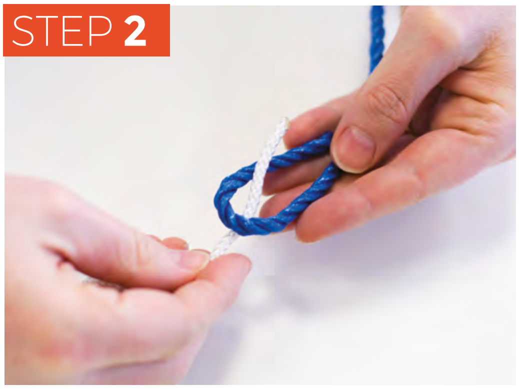
Ben has very strong opinions on knots, and people tend to avoid mentioning anything involving rope or string in his presence so they don't get lectured in the finer points of cord typing.

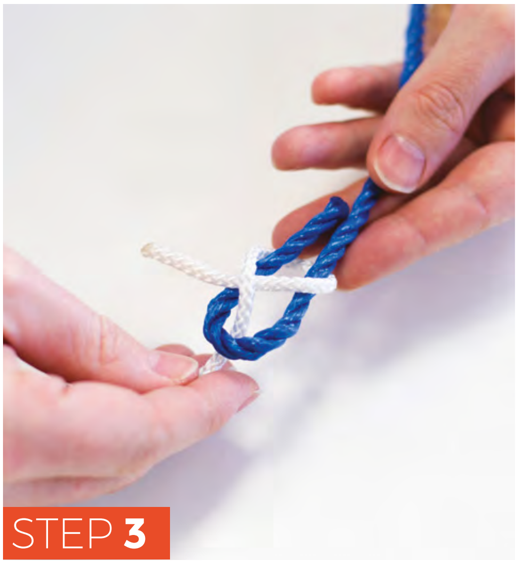
There are several thousand different knots, each of which serves a particular purpose in a unique way, however, there is only one great generalist knot: the sheet bend. This knot can perform almost any job you need of a knot, and it's secure and easy to tie. In other words, if you only know one knot, it should be this one.

The most basic use of the sheet bend is joining two ropes together. Unlike the more famous reef knot (a knot which, while widely taught, doesn't actually do any jobs well, other than securing the reefing lines on a sailing ship), the sheet bend doesn't require both ropes to be the same. It can join thinner ropes to thicker ropes, although there is a limit on how different the ropes can be, the only real way of telling is to try to tie the knot and see if it slips or not.

We're going to look at three ways of using this versatile knot that cover most cases when you need to knot rope.







STEP 1
Start with the stiffer rope (which will usually be the thicker rope) – we'll call this rope 1 – and bend it into a U shape, with a little bit of rope left over on one end.

STEP 2
Take the other rope (which we'll call rope 2) and push it up through the base of the U.

STEP 3
Wrap rope 2 behind both uprights of the U in rope 1.

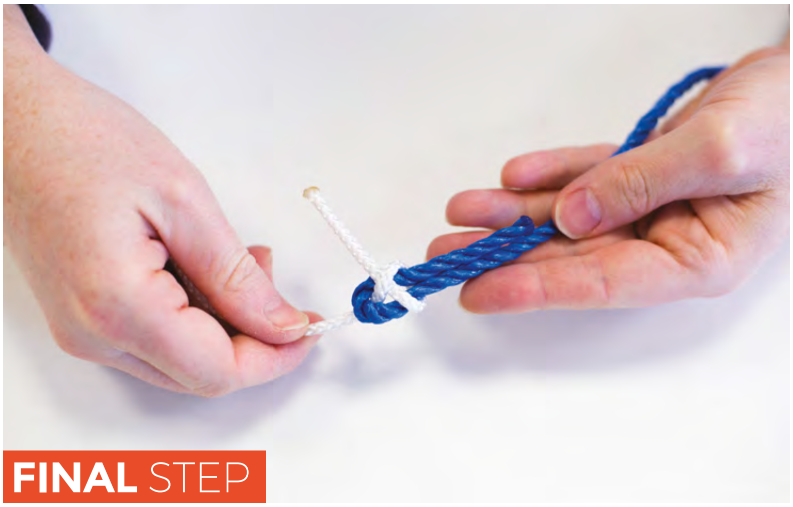
FINAL STEP
Tuck rope 1 under itself, and pull the knot tight.

A COUPLE MORE USEFUL KNOTS

While the sheet bend will serve you well in almost every scenario, there are a couple of extra knots that are easy to tie and useful to know.

Round turn and two half hitches
This knot is for attaching a rope tightly to an object. It's similar to the running bowline knot on the next spread, but instead of starting with a bowline, you start by hooking the rope fully around the object, then tying the two half hitches. As you pull on the rope, the knot will tighten so it will always be securely fastened.

Figure of eight
This is tied by taking a U of rope, twisting this U through one complete turn, and pushing the loose end through the loop created by this twist. If it's done correctly, this should look like a figure of eight. The figure of eight is a good way of keeping the end of a rope tidy and stops it slipping through any pulleys or eyes. It can be tied on a doubled-up section of rope to create a secure loop that's often used by climbers.



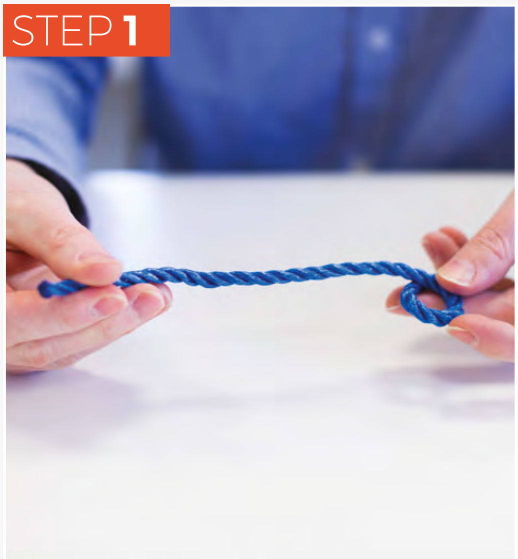
Going loopy

Tying the sheet bend back on itself to create a bowline

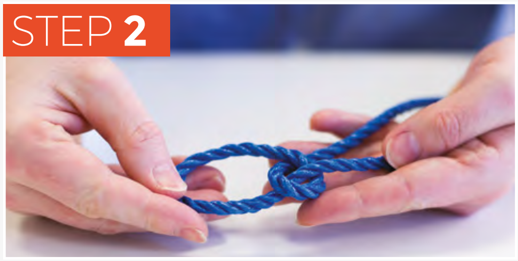
Perhaps the most famous use of the sheet bend isn't for joining two ropes, but for tying one rope back on itself to create a loop. In this configuration, it's known as a bowline, but it's exactly the same knot.

We do have to tie this knot slightly differently because one end of the rope isn't accessible, (as it's not an end at all but a part of the loop).

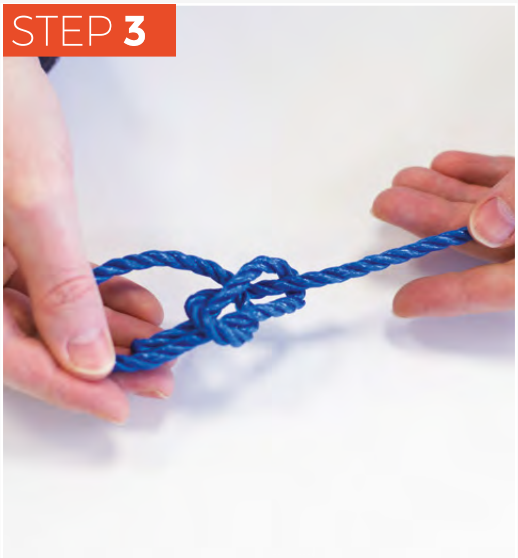
STEP 1



STEP 2



STEP 3

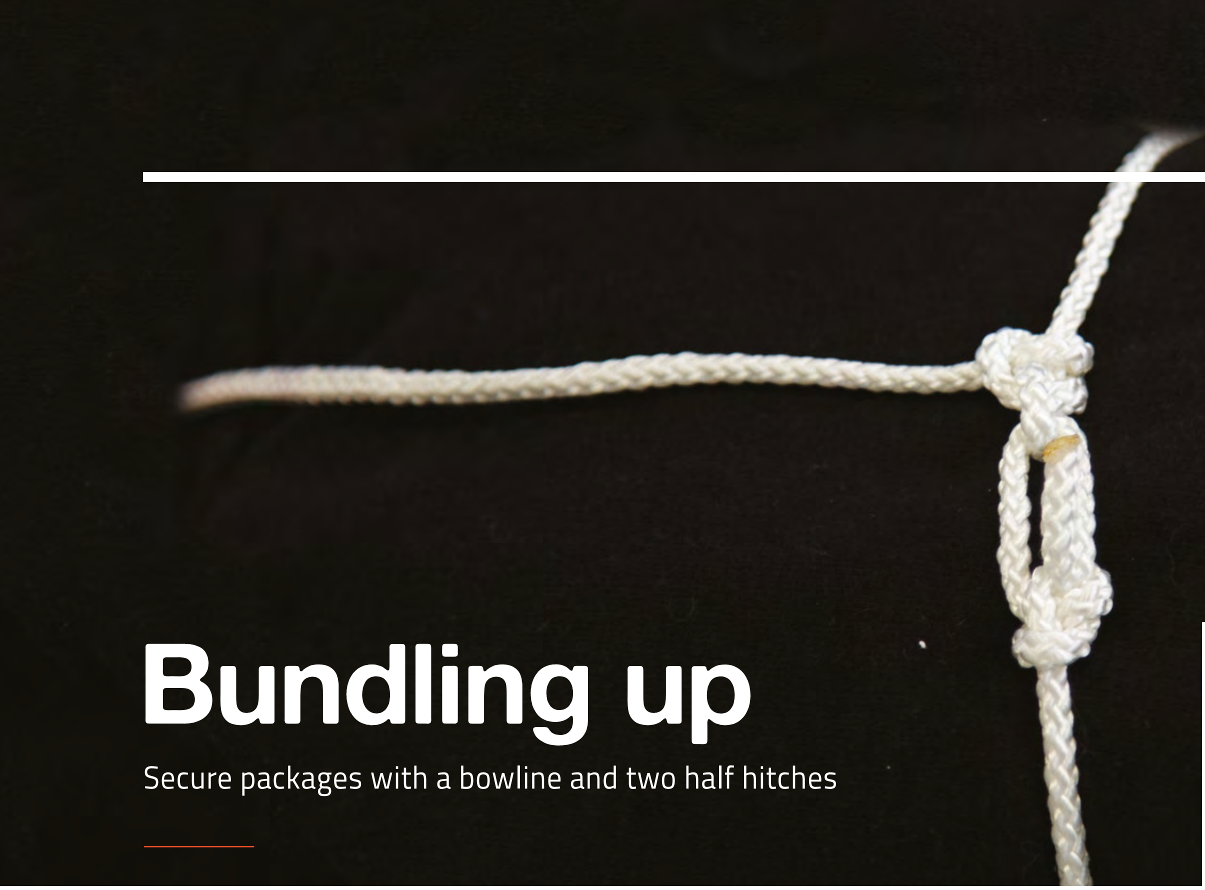


STEP 1
Create a loop in the end of the rope with the loose end going over the long end. This isn't the final loop we're creating, it's part of the knot, so keep this around 2 or 3 cm in diameter.

STEP 2
Take the loose end and push it up through this loop. This creates a second loop (which is the loop we're making), so adjust it to the size you want.

STEP 3
Move the loose end under the long end and back down through the loop. It should sit alongside the rope that went through the loop in the previous step.

FINAL STEP
Pull the knot tight and you now have your completed bowline.



Bundling up

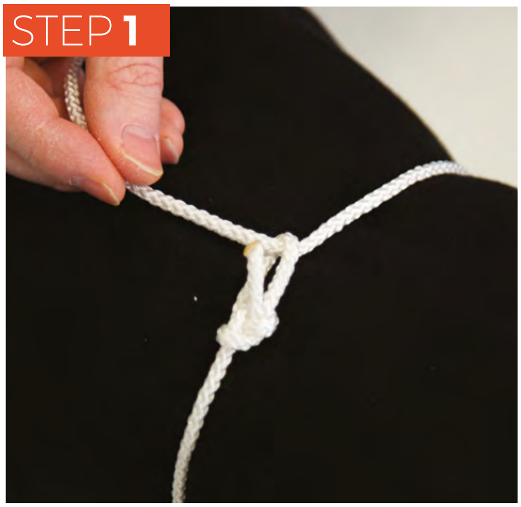
Secure packages with a bowline and two half hitches

While the bowline is great for creating a loose loop, it's very difficult to tie tightly around an object; this makes it useful for bundling together material, tightly attaching one object to another, or properly securing a package with string.

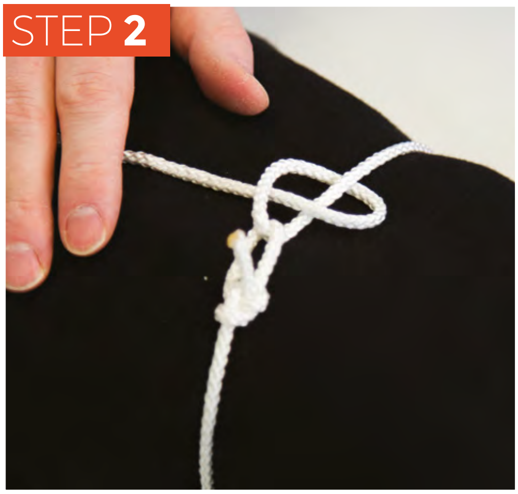
This is similar to a butcher's knot that's used to secure a roasting joint, though the butcher's knot is quicker to tie but less secure.

We're cheating a little with this one as it's a sheet bend with an additional two half hitches. In this configuration, the bowline acts both as a pulley (allowing you to tie the bundle tightly), and as something to tie the half hitches against. The result is a way of securing the bundle, but not too tightly.

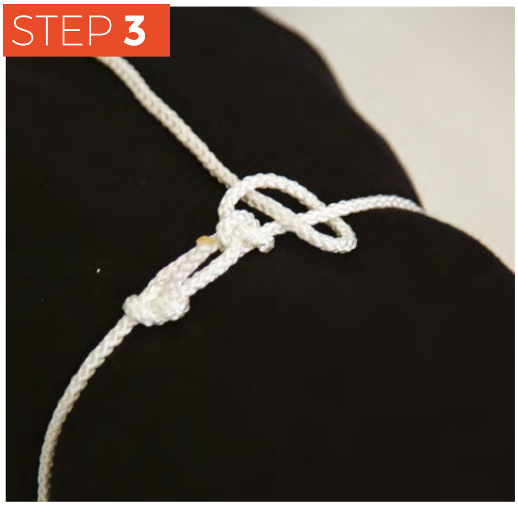
STEP 1



STEP 2



STEP 3



STEP 1

Tie a small bowline. It doesn't really matter how small – if you're using string you might want a diameter of a centimetre or so, if using rope you might find that about 5 cm is as small as you can manage. Loop the long end of the rope around the objects you want to bundle then though the bowline to create a noose.

STEP 2

Pull the noose tight, then wrap the loose end around the now-tight rope going into the noose, and back over itself to create a half hitch.

STEP 3

Repeat step three to create two half hitches. This second knot makes the bundle far less likely to slip than if you leave it with a single half hitch.

FINAL STEP

Pull the knot tight and your bundle is now secure.

Building smart devices with Sonoff switches and Node-RED

Turn devices on and off the lazy way with Node-RED and a Raspberry Pi

David Floyd

GeekDayUK

David always loved technology, started with a KIM1, still had a Mac 128K (but hacked to 512K), and an original IBM PC. He loves the description ‘polymath’, and would like to live up to it.

WARNING

This project uses mains electricity. Make sure you understand the implications of this before undertaking this, or any other project, involving high voltages.

Never connect your Sonoff device to the mains while connected to another computer or with the case off.

Sonoff produces a range of WiFi-enabled mains switches that are powered by hackable ESP8266 modules. They’re available from Amazon, eBay, and direct-from-China shops for just a few pounds each.

They’re in-line relays, which means that you connect them directly to the power cable of what you want to control, and they turn the power on or off. This way they can be used to link almost anything up to your network, provided that it can be controlled with a simple switch. Lamps (and other lighting) work really well, but anything that needs some input after being powered on will be more difficult.

In this tutorial we’re going to look at how to control these using Node-RED, a browser-based programming system that’s available for the Raspberry Pi, Windows, and macOS.

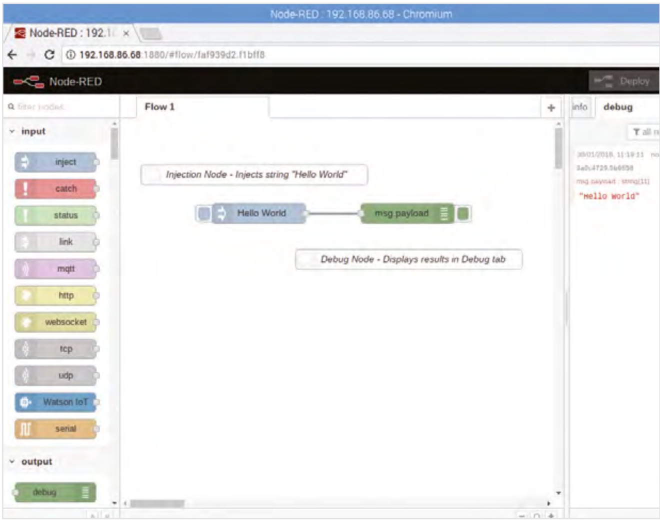
Node-RED programs are created with drag-and-drop nodes in your browser, joining them together with links. The nodes are effectively pre-written customisable subroutines. It is then easy to create programs that read sensors or website feeds, and then do different actions based on the results.

By default, the Sonoff relays are controlled via a phone app that uses the manufacturer’s services, so the first step is to flash new firmware onto them to allow us to control them via our own Node-RED server instead.

There are a wide variety of alternative firmwares for the Sonoff devices, but we’re going to use one called Tasmota (**hsmag.cc/ZcTDuo**), as it provides a comprehensive set of features and a good set of support documentation.

STEP 1
Uploading firmware

We are going to use some of the GPIO pins on the Raspberry Pi to communicate with the Sonoff board and load our new firmware.



Above
Hello World with Node-RED is an injection node that’s used to create the data, and a debug node that displays the message

First we need to enable the GPIO serial port. To do this, we need to edit the **/boot/config.txt** file.

In a terminal window, enter:

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

At the end of the file, add:

```
enable_uart=1
```

and then save the file.

By default, the serial port is used for the console communications (a hangover from serial terminals) and we need to disable this. To do so we will need to enter the following commands.

```
sudo systemctl stop serial-getty@ttyS0.service
sudo systemctl disable serial-getty@ttyS0.service
```

Finally, we need to edit **/boot/cmdline.txt**:

```
sudo nano /boot/cmdline.txt
```

Scroll down the file; we are looking to remove:

```
console=serial0,115200
```

Save the file and then reboot your Pi.

To upload the new firmware, we are going to use a tool called esptool (**hsmag.cc/lYvsve)**. This is available as a package for Raspbian, and can be installed using the command-line package-handling tool apt-get. To install esptool, we use the following commands in a terminal window:

```
sudo apt-get update
sudo apt-get install esptool
```

Now we can download the download Tasmota firmware from GitHub: **https://github.com/arendst/Sonoff-Tasmota/releases**.

Here you will find a number of different language versions. We used the **sonoff.bin**, which is the English language version.

From this point onwards, following the tutorial will invalidate any warrantee you have on your Sonoff device. Let's get hacking!

Now we are ready to open up the Sonoff case. Remember that this case is needed to protect users from mains voltages, so the project should be returned to this case before it's plugged in.

Remove the board and solder in some 5-pin female header strip (**Figure 1**). You could also use male headers if you prefer.

Now, using our four jumper wires, we need to connect up the Sonoff board to the Raspberry Pi.

Once we have the Sonoff connected to the Raspberry Pi, we need to get it ready to accept an upload. This is done by disconnecting the power Pin 1 on the Raspberry Pi and the Sonoff, then holding down the Sonoff button, and reapplying the power.

When reconnecting the power to the Sonoff, you may find that the Raspberry Pi reboots. This is because the power supply cannot supply enough power. In this case we found rebooting the Raspberry Pi, with the Sonoff button pressed until the reboot has completed, solved the problem. If this doesn't work, you may need a more powerful power supply.

We can upload the new firmware using the command:

```
sudo esptool -vv -cb 115200 -cp /dev/serial0 -ca
0X00000 -bz 1M -bm dout -cf /home/pi/Downloads/
sonoff.bin
```

PIN CONNECTIONS FOR FLASHING THE FIRMWARE

Description	GPIO Pin	Sonoff Pin	Wire Colour (Figure 2)
3.3 V	Pin 1	Pin 1 (Nearest to switch)	Red
TXD->RXD	Pin 8 (TXD)	Pin 2 (RXD)	Orange
RXD<-TXD	Pin 10 (RXD)	Pin 3 (TXD)	Brown
GND	Pin 6	Pin 4	Black

We used the `-vv` option so you can see the progress and any relevant error messages. The options to this command mean:

- `-vv`: Verbose mode
- `-cb 115200`: The baud rate of the serial connection
- `-cp /dev/serial0`: The serial port to use
- `-ca 0X00000`: Start address to upload to
- `-bz 1M`: Size of the flash chip
- `-bm dout`: Flash chip interface mode
- `-cf …`: The file to upload

If all has uploaded successfully, we need to power-cycle the Sonoff device. →

YOU'LL NEED

- **Raspberry Pi 3 running Raspbian**
- **Sonoff Basic WiFi switch**
- **Four jumper wires** (Male–male)
- **5-pin male / female header strip**
- **Fine-tip soldering iron and solder**

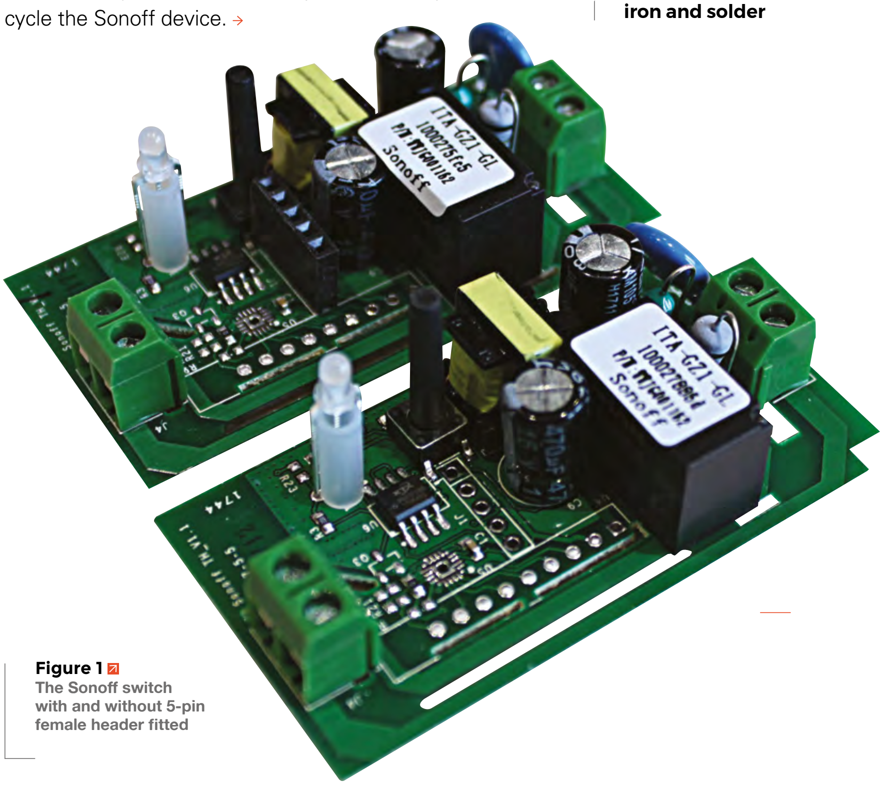


Figure 1
The Sonoff switch with and without 5-pin female header fitted

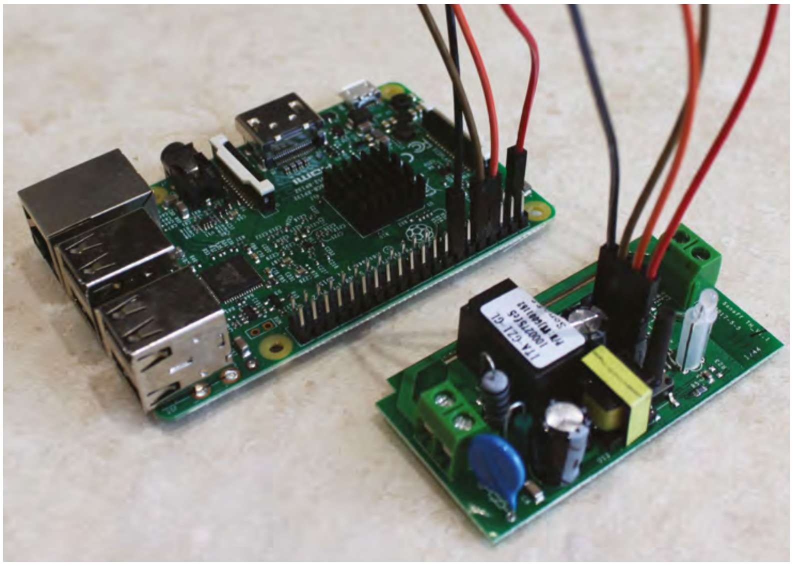


Figure 2 The Sonoff switches are programmable using the Raspberry Pi's universal asynchronous receiver-transmitter or UART

STEP 2
Tasmota WiFi setup

With the new firmware loaded and with the Sonoff-Tasmota device still powered by the Raspberry Pi, we are going to need to configure the Sonoff to connect to your WiFi. We start the process by pressing the button on the Sonoff four times quickly – this will put the Sonoff into WiFi access point mode.

Remember that you should never connect your Sonoff device to the mains while connected to another computer or with the case off.

The LED on the Sonoff will blink during configuration; a single button press at this point will abort the process if you need to start again. See **hsmag.cc/fmquaD** for more information on this and other configuration options.

On your Raspberry Pi, click on the WiFi icon to display the list of local WiFi access points; we are looking for an access point called sonoff -???? , Where ???? is a four-digit number for your device. You many need to refresh the list a few times before it joins the list, but once it is available, click on it to connect.

Once connection is established, open your web browser and enter the IP address **192.168.4.1** and press **RETURN**. It should open up a page called Sonoff Basic Module.

On this page you can enter your required WiFi network name (SSID), and password. In our setup this is homenetG, but you need to enter your own here. You can also enter a second alternate SSID and password if needed. Now press the Save button and the device will restart after a few seconds.

> **You can add more Sonoff switches to your network** – just remember that each one will have a different IP address and you'll need to create separate HTTP Request nodes for each one

If this all goes horribly wrong, just hold down the button on the Sonoff device for over four seconds, and you can then restart this part of the process.

Now reconnect to your local WiFi network and with the new device hopefully connected, we now need to find its IP address so we can talk to it.

The process for getting this depends a little on your network setup. Most home routers use DHCP to allocate a unique IP address to each device, and you can see this by logging into the router's web interface (this is often found at **http://192.168.0.1**, though you may need to check your router's manual if this doesn't work).

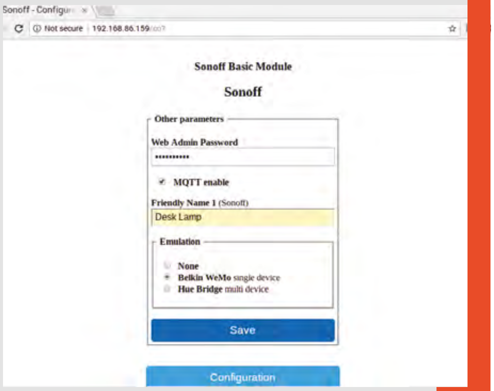
If you can't get the IP address details from your router, you can scan your network to find all the devices attached to the LAN using software such as Fing (**hsmag.cc/uMHJIA**), or an app on your smartphone.

Our device was allocated **192.168.86.159**, but yours will probably be different. Enter the IP address of your device into your browser and you should then get the following web in **Figure 3**.

ALEXA AND WEMO EMULATION

If you'd rather control your devices with your voice, you can link your Sonoff devices to an Amazon Echo (or other Alexa device), rather than using Node-RED. With the Tasmota firmware, you can do this using Belkin WeMo emulation. You will find this option in the 'Configuration > Configure Other' menu. Do be careful with your selection on the 'Friendly Name', as this will be the name you need to use with Alexa to control the device.

We found 'Alexa Desk Lamp on' and 'Alexa Desk Lamp off' work perfectly.



Sonoff - Configur

Not secure | 192.168.86.159/co?

Sonoff Basic Module

Sonoff

Other parameters

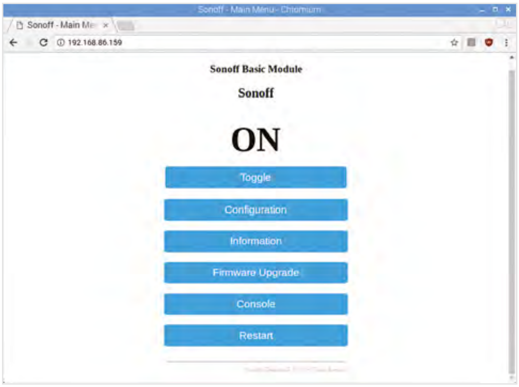
Web Admin Password
••••••••••

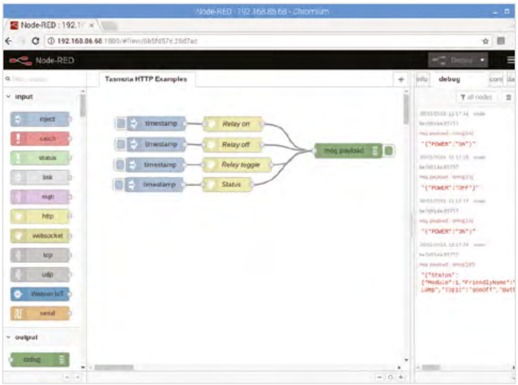
☑ MQTT enable

Friendly Name 1 (Sonoff)
Desk Lamp

Emulation

○ None
● Belkin WeMo single device
○ Hue Bridge multi device





```
"{"POWER":"ON"}"
"{"POWER":"OFF"}"
"{"POWER":"ON"}"
"{"Status":
{"Module":1,"FriendlyName":"
Lamp","Topic":"sonoff","Butt
```

Far Left – Figure 3 The web-based configuration for the Tasmota firmware

Left Node-RED set up to control our Sonoff relay

STEP 3

Remote control from Node-RED

You can control the switch using the Tasmota website, but you can get a little more power by linking it into a programming environment. You can use any that can make HTTP requests, but we'll be using Node-RED, as this allows us to easily create flow-based programs that can turn things on or off based on inputs.

While Node-RED is available for a variety of devices, we'll be using the Raspberry Pi (Node-RED comes pre-installed in the latest version of Raspbian). Make sure you're using an up-to-date image, then launch Node-RED with:

```
node-red-start
```

The web-based interface will now be available at **http://localhost:1880**. Here, you'll have the ability to drag and drop nodes into flows that perform actions.

We'll set up our basic Node-RED test using injection nodes, HTTP request nodes, and a debug node. The injection nodes are used to first trigger the action. You'll need four injection nodes, with each one linked to a HTTP Request node. The link in the HTTP Request should point to a command on the Sonoff. The links are:

- http://<ip-address>/cm?cmnd=POWER ON
- http://<ip-address>/cm?cmnd=POWER OFF
- http://<ip-address>/cm?cmnd=POWER TOGGLE
- http://<ip-address>/cm?cmnd=STATUS

These, as you may have guessed, are to turn the relay on, off, toggle the state, and get the status, respectively.

Linking these to a debug node allows you to see what's going on. This is the basic setup for controlling the Sonoff via Node-RED, and you can build it into any project you like. This will let you control your relay, but to make it do any work you'll have to put it back in its case, then put it into the powerline of the device your want to control. That means cutting the power cord and wiring it up following the manufacturer's instructions – remember that you're using mains voltage, so don't plug it in until you're happy that it's securely housed and safe. Consult a suitably qualified or experienced person if you're not sure.

You can add more Sonoff switches to your network – just remember that each one will have a different IP address, and you'll need to create separate HTTP request nodes for each one. Using these, you can create complex flows to turn your devices on and off using any form of input you can get into Node-RED. □

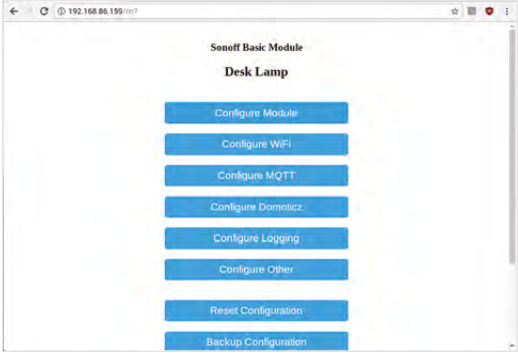
SECURITY

The Tasmota web interface allows you to add a password to your device, and then you need to append 'user=admin&password=password' to your HTTP commands. For example:

```
http://<ip-address>/cm?user=admin&password
=password&cmnd=POWER ON
```

However, the security conscious of you will notice that this is transmitted on the plain-text HTTP protocol, so this doesn't really add much security. Essentially, any device on your WiFi network will be able to see these requests and take control of your Sonoff switches.

How much this concerns you depends on your attitude to security and what devices the Sonoff switches are powering. We would recommend running the switches (or, for than matter, any smart devices) on a separate WiFi network and bridge the two using a trusted, secured device, such as a Raspberry Pi running Node-RED.



Left There are more features to Tasmota than we've explored here, including the ability to control the hardware with MQTT

Update all your Raspberry Pis with one command

Deploy at speed to get your home running like clockwork

Matthew Macdonald-Wallace

@proffalken

Matthew Macdonald-Wallace is an unashamed geek. He runs Mockingbird Consulting (**hsmag.cc/PBmQpy**), which helps communities and organisations use the IoT in their data-driven decision making.

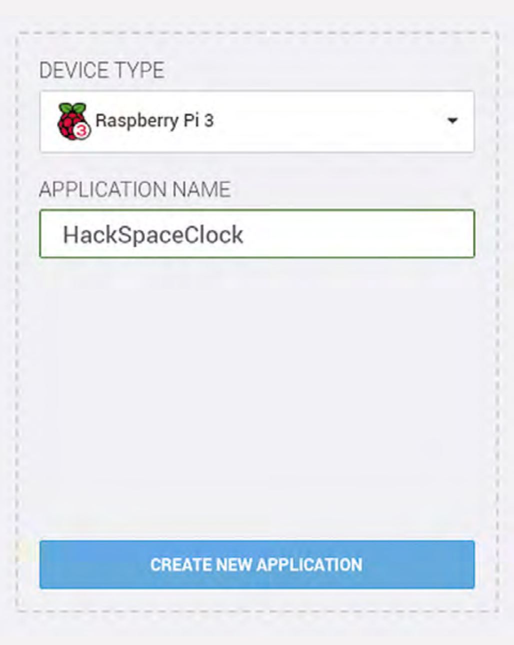
Raspberry Pis and other small Linux computers allow you to create a fantastic range of smart gadgets, but keeping them all updated can be a chore. Wouldn't it be great if there was a tool to take the pain out of managing a fleet of these little computers? Well, fortunately there is: Resin.io

In this article we'll show you how to use Resin.io to manage any number of Raspberry Pis, and give you a futuristic-looking clock at the same time. This same technnique can be used to push any code to the computers, so you could use it to keep anything in your smart home or fleet of robot assassins running on the latest code.

You can use any of the devices supported by Resin.io, which include the Raspberry Pi; BeagleBones Black, Blue, and Green; an Intel NUC; and others – see **hsmag.cc/tCpXWY** for a full list of devices. With its on-board WiFi, the Raspberry Pi 3 is one of the easiest to get started with, so we'll use that. We should probably let you know at this point that, whilst Resin.io runs well on the Raspberry Pi Zero W, it does restrict what you can do with it as the resources are more limited than usual.

The first thing to do is to create an account at Resin.io. This is reasonably straightforward so we won't document it here; however, once you've got an account we'll need to set up a new project, so let's do that now.

Once you're logged in to the Resin.io Dashboard, create a new project by selecting Raspberry Pi 3, (or other appropriate device if you're not using a Pi 3), from the Device Type drop-down, and give your application a suitable name.


DEVICE TYPE

Raspberry Pi 3

APPLICATION NAME

HackSpaceClock

Above
Selecting a good application name will make it easier to keep track of multiple apps

This will present you with a screen asking you to add a new device, so let's do that next. Click on the 'Add Device' button, and then fill in the box that pops up.

Click on 'Download resinOS', and save it somewhere suitable on your PC. You'll need to keep this for the next step.

BURN BABY, BURN...

Now you'll need to burn the image you downloaded to the SD card for the device you're using. For this we'll show you how to use Etcher, which was created by Resin.io for exactly this task. Download Etcher from **Etcher.io** and load it up.

Insert the SD card into your PC, and then follow the simple steps on your screen. First, select the file you downloaded, then select the SD card you're going to use, and finally, click 'Flash'. Once the program has finished, remove the SD card and place it into your Pi 3.

Now for the exciting bit! Make sure that your SD card is firmly in the Raspberry Pi, that the Pi is connected to the monitor, and plug the power in. You should see the Resin.io logo fill the screen, and after a few minutes, the device should show up in your Application Dashboard.

> "You can use any of the devices supported by Resin.io, which include the **Raspberry Pi; BeagleBones Black, Blue, and Green; an Intel NUC and others**"

LET'S PUSH IT FURTHER!

Now that your device is registered with the Resin platform, it's time to create some code and push it up to the device.

Resin.io is based on a technology called Docker and uses some software called Git to get the code from your computer into the right format, and then downloaded onto the Raspberry Pi for execution.

The aim of this project is to help you understand how Resin works, so instead of writing out the code for the clock application ourselves, we're going to use an existing project and make a few changes in Resin itself to change what is displayed.

Using your Git client, clone the code with:

```
git clone git@github.com:shaunmulligan/resin-
html5-kiosk.git
```

This will download the code for our clock, so run it from a suitable place on your computer. Have a look through the files (especially the **Dockerfile.template** and **src/launchBrowser.sh** ones) to get an idea of what is happening in the background, but don't worry too much if you don't understand them: the magic happens once the application is deployed.

Now log back into your Resin Dashboard and click on your application. In the top right-hand corner of your screen, you'll see a command with a URL, something like this:

```
git remote add resin <yourUsername>@git.resin.
io:<yourUsername>/hackspaceclocks.git
```

→

WHAT IS RESIN.IO?

Resin.io is a platform for developing, deploying, and managing IoT applications using modern cloud workflows and architectures. Resin.io enables you to push Docker containers to the edge, remotely update your code with a simple git push, and monitor your entire fleet of devices.

For more information on Resin.io, please visit its website at **resin.io**.

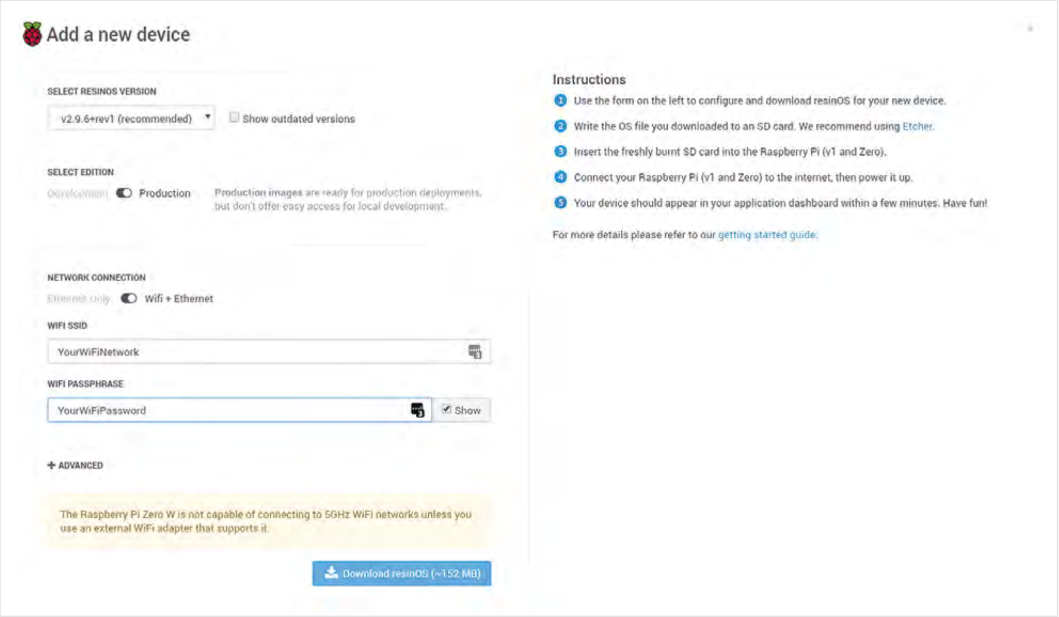
YOU'LL NEED

- **A Raspberry Pi** (any kind with internet access will work, including the Pi Zero W)
- **A screen connected to the Raspberry Pi** via the HDMI port

QUICK TIP

If you don't want to use Etcher to flash the SD card, then you'll need to unzip the file you downloaded and follow one of the system specific guides at **hsmag.cc/DCCtrS** to flash the resulting .img file to the SD card.

Right Make sure that you click the ‘Ethernet + WiFi’ switch, and then enter the details of your WiFi network for the place you’re going to deploy the clocks


Add a new device

SELECT RESINOS VERSION

v2.9.6+rev1 (recommended) ▾ □ Show outdated versions

SELECT EDITION

Development ● Production — Production images are ready for production deployments, but don’t offer easy access for local development.

NETWORK CONNECTION

Ethernet Only ● Wifi + Ethernet

WIFI SSID: YourWiFiNetwork

WIFI PASSPHRASE: YourWiFiPassword ☑ Show

+ ADVANCED

The Raspberry Pi Zero W is not capable of connecting to 5GHz WiFi networks unless you use an external WiFi adapter that supports it.

Download resinOS (~152 MB)

Instructions

1. Use the form on the left to configure and download resinOS for your new device.
2. Write the OS file you downloaded to an SD card. We recommend using Etcher.
3. Insert the freshly burnt SD card into the Raspberry Pi (v1 and Zero).
4. Connect your Raspberry Pi (v1 and Zero) to the internet, then power it up.
5. Your device should appear in your application dashboard within a few minutes. Have fun!

For more details please refer to our getting started guide.

> **If you’re not using the command line,** use your client of choice to push the code up to the resin remote destination

If you’re using the Git command-line tools, then simply cut and paste this command into a window in the existing Git project directory on your PC. If you’re using another Git client, then add:

```
<yourUsername>@git.resin.io:<yourUsername>/
hackspaceclocks.git
```

…as a remote destination for the repository that you’ve created.

Now we should be all set. From the command line, type the following:

```
git push resin master
```

If you’re not using the command line, use your client of choice to push the code up to the Resin.io remote destination.

Lots of code will scroll past your screen, and then, if all goes well, you should be presented with a unicorn and a message of success. Shortly after this, the status of your device in the Resin.io control panel should change to ‘Updating’ and the Raspberry Pi will start to download the image you’ve just created.

Give it a few minutes to download, and then you should see the Raspberry Pi restart, and finally the

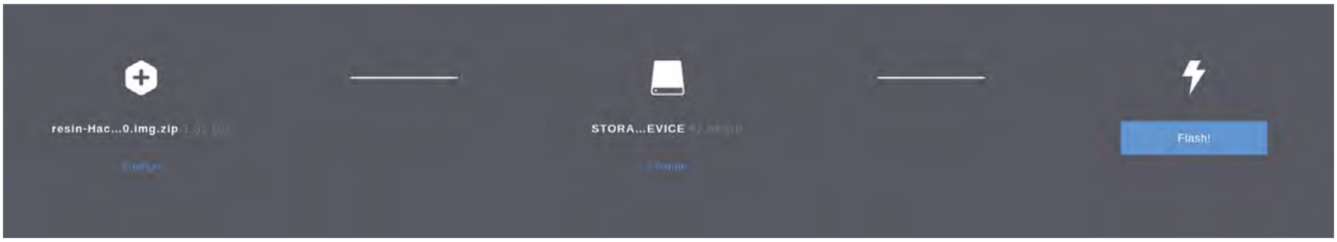
GIT READY… GO!

Git is a way of working with colleagues and friends on the same code, even if you’re not sat in the same room.

It’s used by hundreds of thousands of teams in thousands of companies, from two-person startups to large enterprises, and is invaluable when you need to keep track of who did what, when they did it, or how to roll back a change that’s broken things.

There’s an excellent interactive tutorial at **try.github.io** if you want to learn more about how to use it.

Below Etcher makes creating SD cards absolutely trivial!



browser should launch a website with an audio player in the middle.

MIXING IT UP

The code that we've taken and pushed up to Resin.io had the following important lines in it the file src/launchBrowser.sh:

```
url=$URL
default=’http://www.noiseaddicts.com/samples_1w72b820/3712.mp3’
...
epiphany-browser -a --profile /root/.config ${url:-$default} --display=:0
```

These lines say to the system 'Look for an environment variable called "URL"; if it's set then use it as the address to open in the browser, otherwise use the value of "default".'

Resin.io gives us the opportunity to set and change variables at both the application and device level, so let's set this now.

We want all of our devices to show the same URL, so from the main page of your application screen in the Resin.io dashboard, click on 'Environment Variables'. Add a new variable called 'URL' (the case is important – this should be all capital letters!), and set the value to

```
http://www.clocktab.com/#fullscreen
```

Your application will now restart, and you should see a clock on the screen instead of the audio player – congratulations on launching your first device with Resin.io.

IT'S ALL JUST A LITTLE BIT OF HISTORY REPEATING

Okay, you've got a clock that works and it's really cool and everything, but the power here is managing multiple machines, not just one.

Simply burn the image you downloaded at the start of the program to another SD card, connect up a new Raspberry Pi and screen, insert the SD card, and turn on the second device.

The Resin.io OS (resinOS) that's on the SD card will connect to your internet connection, download the existing code, and automatically execute it without you needing to get involved at all. Within minutes, you'll have a second clock up and running.

SHIPS, WHALES, AND DOCKS?

Docker is a way of packaging up an application into something called a 'container' which can then be deployed pretty much anywhere the container management service can run.

In industry, Docker is used to ensure that an application developed on a Mac or Windows computer will run successfully on a Linux host such as in Amazon Web Services, or to provide very quick feedback to developers as part of a test loop.

If you're interested in knowing more about Docker, there's an excellent tutorial at **docker-curriculum.com**.

PRICING

Resin.io is open source, but you do need an account to use the software. Your first ten devices are free, and this may be enough for many hobbyists. If you need more devices you'll need to pay for an account (starting at $99 a month for 20 devices), which includes support. This may put it out of reach for personal projects, but the time savings can be huge if you're wrangling lots of computers.

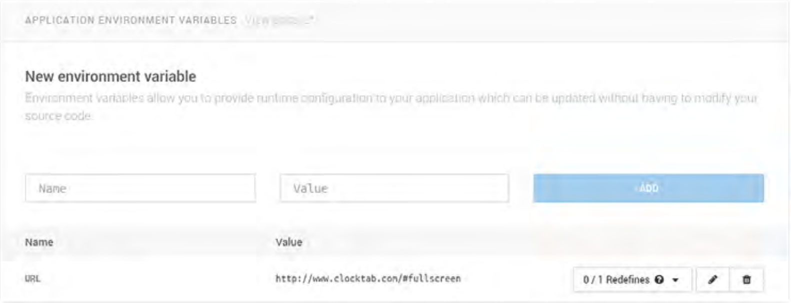
> "Your application will now restart, and you should see a clock instead of the audio player – congratulations on launching your first device with **Resin.io!**"

Now just repeat the process of connecting the hardware, burning the SD card, and turning it on until you have the number of clocks you desire – simple!

CLOSING TIME

The real power here isn't in setting up clocks, but the tools that let you update the software on many devices remotely. You can use this same method to push software for any smart devices to your various Raspberry Pis. Get in touch on social media to show us the projects you've made based on this article! □

Below Resin makes it really simple to get up and running



APPLICATION ENVIRONMENT VARIABLES

New environment variable

Environment variables allow you to provide runtime configuration to your application which can be updated without having to modify your source code.

Name	Value
URL	http://www.clocktab.com/#fullscreen

Lithophanes – 3D printing images

Turn your pictures into 3D prints, in a few easy-to-follow steps

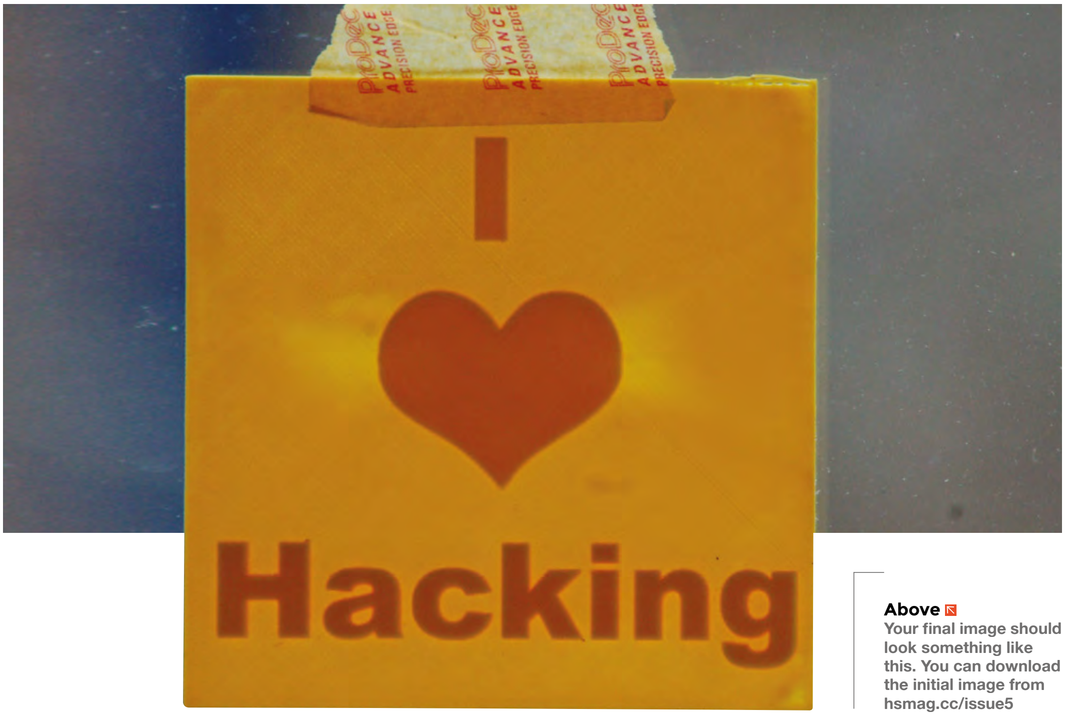
Mark Davies

@markblue777

Mark Davies is a software consultant by day, maker and tinkerer by night. If not making, then he is typically doing some form of DIY - markwilliamdavies.co.uk/blog

YOU'LL NEED

- **A 3D printer**
- **3D printer filament** (white is best, but lighter colours can work)
- **Slicer software** (Cura is used in this guide)
- **Experience of using your 3D printer** and slicer software
- **The image to accompany this guide** (Figure 1)
- **Internet connection to access the image to STL software used in the guide**



Above Your final image should look something like this. You can download the initial image from hsmag.cc/issue5

3D printers are best known for printing things that are, well, 3D, but that's not the only use for them. Lithophanes are 3D images created in varying transparency. This changing transparency creates a monochrome image that looks particularly impressive when back-lit. This article sets out to teach you the basics to get you started, and then enable you to move onto grander prints.

NOT ALL IMAGES ARE EQUAL!

The first step is image selection. In the world of 3D-printed lithophanes not all images are created equal. **Here are some key points to remember:**

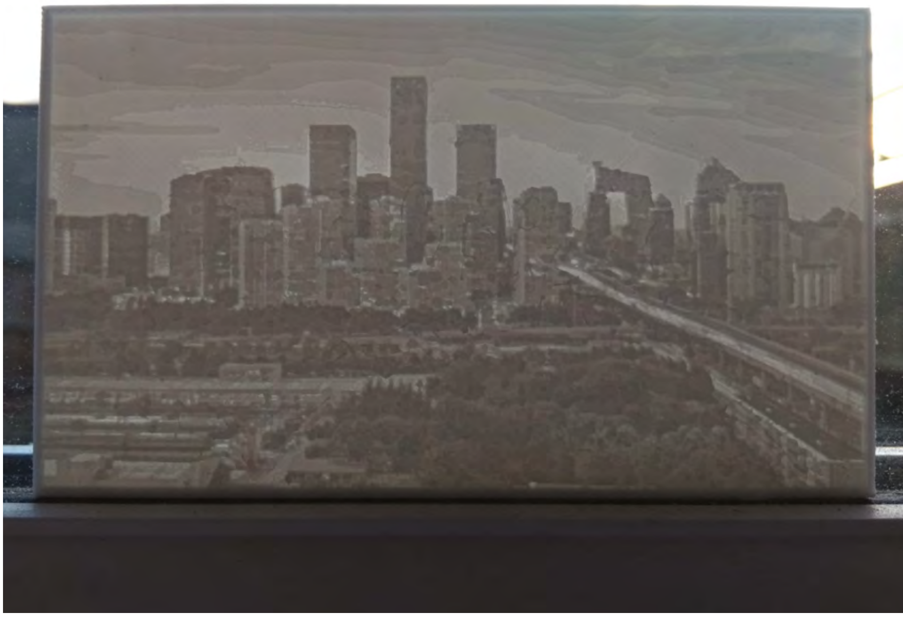
- High contrast pictures work best. A lithophane is basically printing in greyscale. So, images that rely on colour do not work well.
- Typically, the more detail there is in an image, the worse it will look. Go for simplistic images with a uniform background.
- A lower resolution image can have better results than a high-res image.
- Select an image that is close to the print size you want to make.

There is a variety of software on the market that can transform your photos into STLs, some are free and some cost. We will be using lithophane, which is hosted on **3dp.rocks**; it's a browser-based converter, so there's no need to install anything.

Open a web browser and navigate to **hsmag.cc/nFHdOx.**

STEPS FOR SUCCESS

- Ensure the printer bed is levelled and clean
- Use lighter colours. White or natural filament works best but, as you have seen, a yellow filament prints well too
- Ensure you use the correct print temperature for the filament you're using
- For our ABS prints, we need to increase the print temperature by 5 degrees for them to stick better
- Image selection is key; choosing the right image is half the battle
- Think about the filament type you are using. PLA could melt or warp if placed near a window or if it gets too hot
- A tea candle can give a nice light to illuminate the lithophane (don't burn the house down though)



> "This changing transparency creates a monochrome image that looks **particularly impressive when back-lit**"

To help you follow along, we have made a simple image that will show the basic principles around printing your own lithophanes.

Navigate to the Images tab of the website to upload your image. When the image is imported, you want to make sure the lighter elements of the image are the thinnest (or else you will end up with a negative print)

SLICE AND DICE

We're using Cura, but other 3D printing software should also work. Load the model into Cura and select your desired filament setup.

Selecting the correct filament is key: go for lighter or natural colour filaments, and think about where you will be displaying the finished piece. If the final print is going to be mounted on a window, maybe use ABS or PETG to avoid the warping that can happen with PLA when it gets too warm.

To achieve better results when printing your lithophanes, you want to:

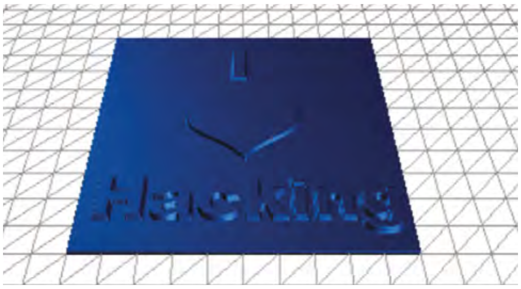
1. Use smaller layer heights (we go for 0.1 mm). You can go lower, but this will increase print times, and in our opinion, it does not give you much of a quality gain.

2. Slow your prints down – this will reduce the risks of blobs and stringing with the print. It's best to err on the side of caution with more detailed prints and slow the print speed down. We've found 40 mm/s works well.

3. Infill density should be set to 100%. If you don't do this then you will see a honeycomb effect (or the infill type you use) when you raise the lithophane to the light.

Once you have set the settings, save the gcode for the model and get printing

So that is it, you have printed your first lithophane; congratulations. With the skills learned from this guide, you should now be able to move onto some of your own custom prints. Go mad and decorate your house, office, cave etc, and remember lithophane prints make great little gifts. □

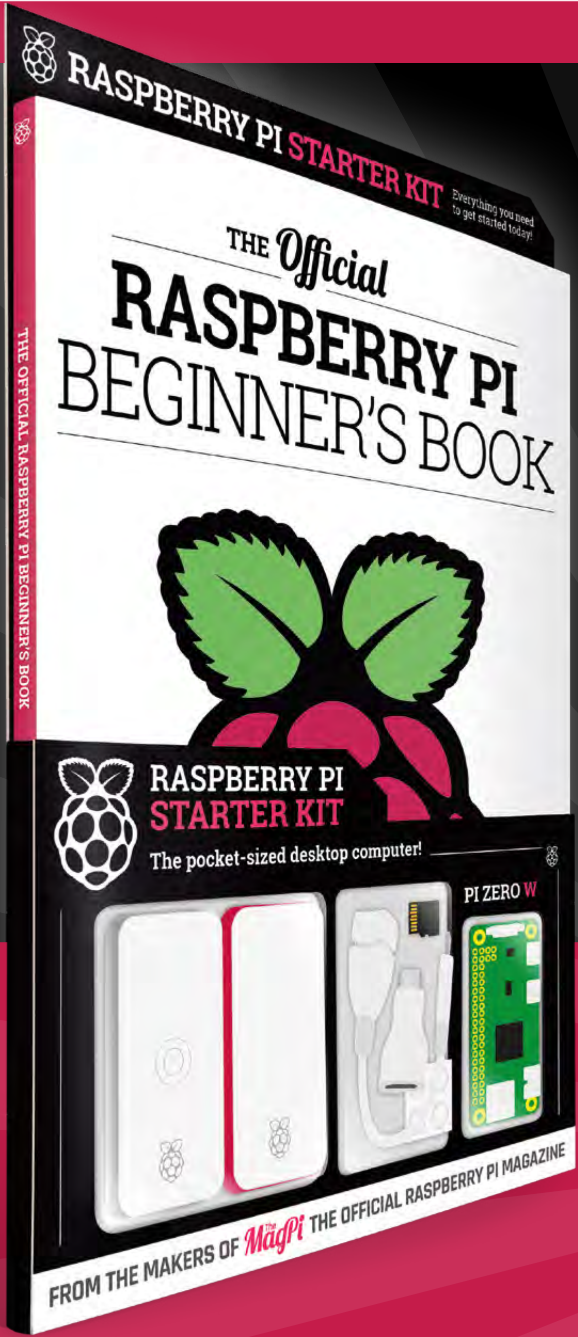


Above Starting with a good monochrome image is the key to creating good lithophanes

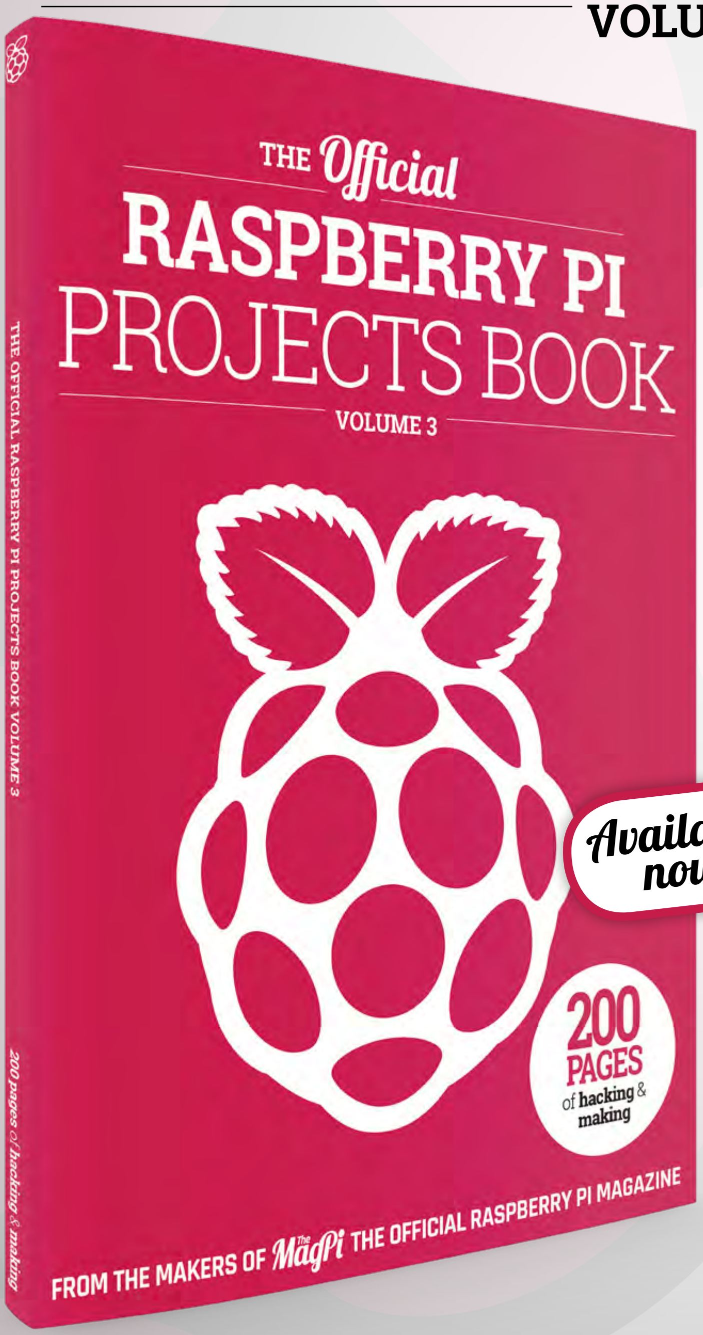
Below A positive image has the darker parts thicker to obscure more light

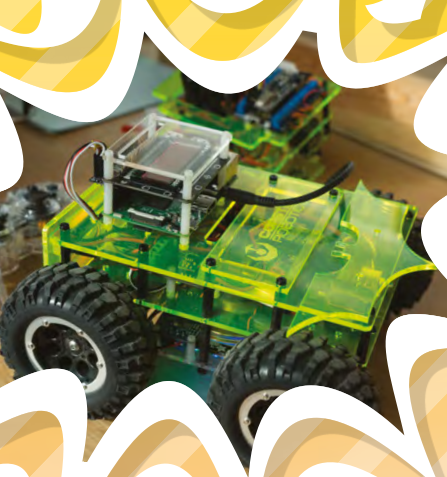
QUICK TIP

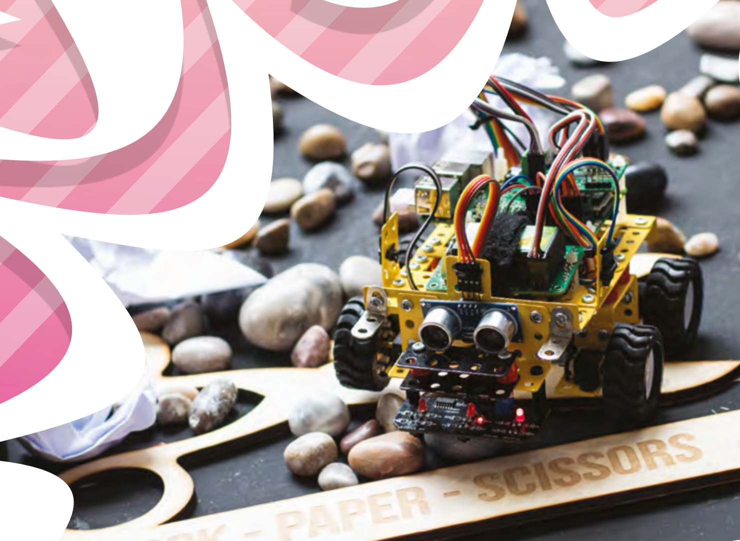
If you have the option of USB or SD card printing, go with SD card printing. Lithophane STLs can be quite large and USB transfer can be slow, causing unneeded stops and blobs whilst instructions are transferred.


RASPBERRY PI STARTER KIT
Everything you need to get started today!
THE Official
RASPBERRY PI
BEGINNER'S BOOK
THE OFFICIAL RASPBERRY PI BEGINNER'S BOOK
RASPBERRY PI
STARTER KIT
The pocket-sized desktop computer!
PI ZERO W
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

Available now


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 3
200 PAGES of hacking & making
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 3
200 pages of hacking & making




PAPER - SCISSORS

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 116

DIRECT FROM SHENZHEN: STIRLING ENGINE

Ditch the batteries and power your next build with flames

PG 122

DOOR BELL

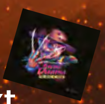
Pulling apart a modern knocker to see if you can hack it

PG 118

BEST OF BREED

Gesture Sensors: Getting user input with the wave of a hand

REVIEWS



124 Crazy Circuits

126 Spectrum Next

128 Leveraxe

129 Making Time

DIRECT FROM SHENZHEN

Stirling engine buggy

Convert heat into motion with a simple external combustion engine

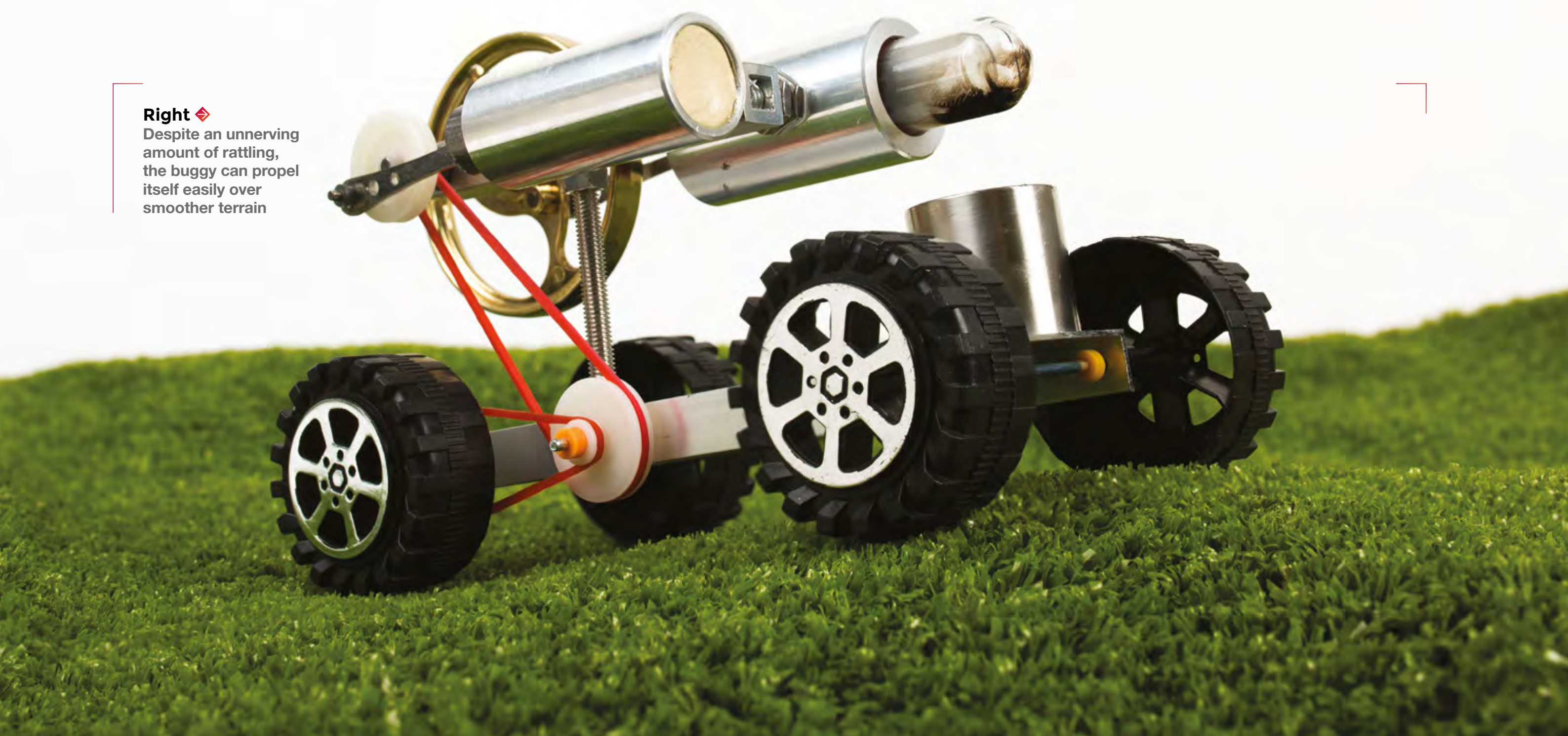
By Ben Everard

@ben_everard

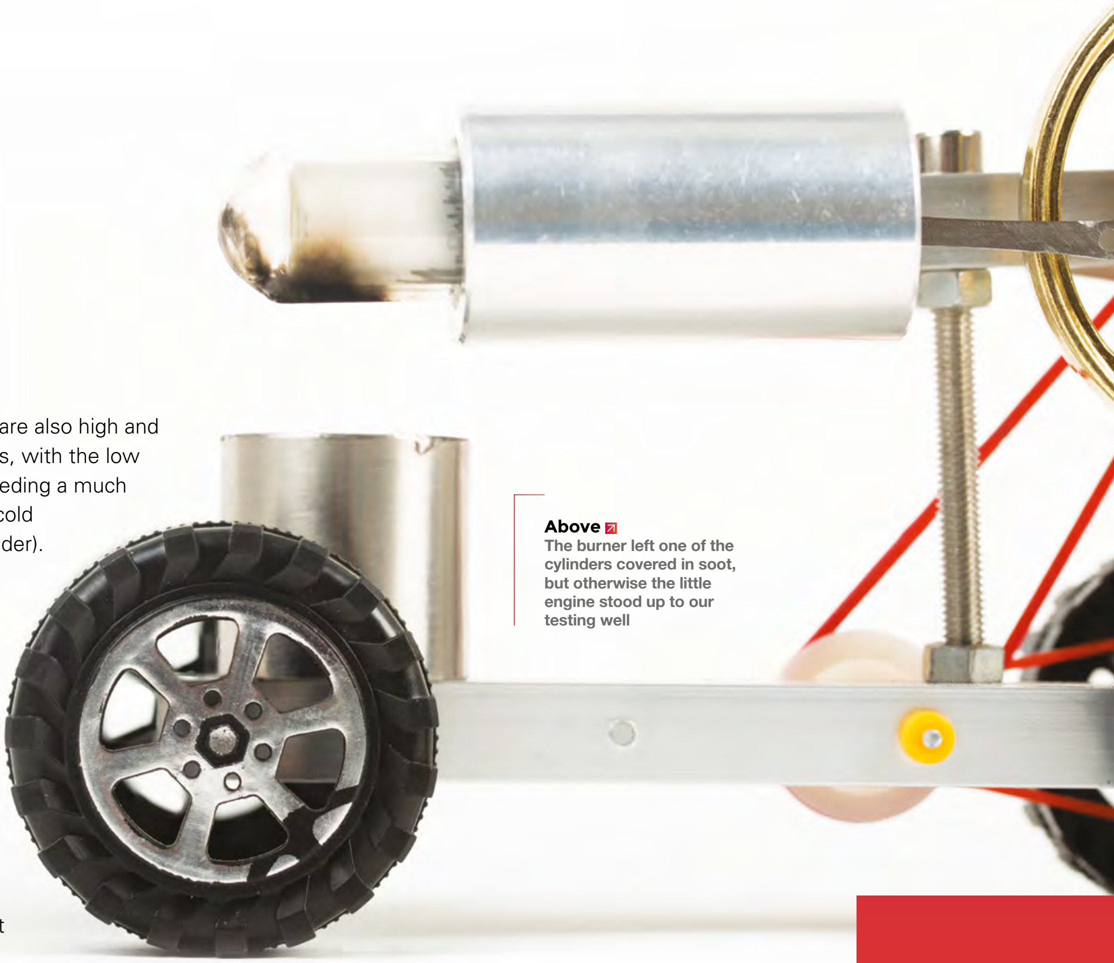
In 1816, Robert Stirling created a closed-cycle air engine which worked by heating one side of the engine while the other remained cool. Air in the hot side heats up and expands, and is shuffled through to the cold side where it contracts. This expanding and contracting drives pistons, which in turn drive a fly-wheel which itself keeps the pistons in both sides (the hot and cold parts) moving in synchronisation. It's an external combustion engine in that the fuel and heat source are entirely outside the main engine, in a similar way to a steam engine. However, unlike a steam engine, it's a closed-loop system so you don't need to top up the water (or any other fluid) – just apply heat and you're ready to go.

Typically the hot side is heated by a flame, while the cold side is left to cool to room temperature, though this doesn't have to be the case. Theoretically, the hot side can be left at room temperature and the cold side cooled – provided you can get it cold enough, the engine will still run – or some combination of both heating and cooling.

There are a few different configurations available – the most mechanically significant difference is whether there's one cylinder that holds both the cold and hot sides (beta type), or two cylinders, with one hot and one cold (alpha type). Beyond this, there are dozens of different ways of mounting them, though most are purely aesthetic, including propeller mounts and setups designed to look like an



Right Despite an unnerving amount of rattling, the buggy can propel itself easily over smoother terrain



Above The burner left one of the cylinders covered in soot, but otherwise the little engine stood up to our testing well

internal combustion engine. There are also high and low temperature differential options, with the low temperature differential options needing a much larger surface area on the hot and cold sides (leading to a short, wide cylinder).

We got an alpha-type engine driven by a small spirit burner mounted on a small buggy from the Cassiel Alice store on Ali Express for £30.81, (a wide variety of similar models is available on many direct-from-China sites). The unit came fully assembled, with the engine mounted via a threaded bolt on a short section of U-profile aluminium. A couple of axles support plastic wheels that allow the buggy to move.

A STIRLING EFFORT

We applied a flame and waited. It took around 30 seconds for the engine to heat up enough that a flick of the fly-wheel spurred it into life. The speed of the engine depends on the size of the flame, but we'd put it at between 100 and 500 RPM.

The drive system comprises of a couple of gears joined by elastic bands to the front axle. Frankly, it does not inspire confidence. However, it worked and drove our little buggy along at a fairly sedate pace. We have no doubt that this drive system could be improved to power the buggy at a higher speed, but if you're just after movement, this elastic band system does the job.

Unlike internal combustion engines with their repeated ignitions, there's not an inherent source of noise in a Stirling engine, and they can operate very quietly. However, this buggy is quite noisy in operation. This appears to be from intolerances in the fly-wheel and attachments. There's quite a lot of slack and wobble in the system, which causes it to judder about more than this type of engine really should. This doesn't really affect the buggy too much (though it may shortened its life), but it does limit what other projects you could build off this engine.

We were slightly disappointed that the spirit burner is difficult to remove – this makes it quite difficult to hack the system to work with a different heat source. Although it's bolted on, the nuts appear to be glued in place, so removing this to allow space for a different burner is difficult. An alternative hack would be to direct the power from the engine into some other mechanism. This should be perfectly possible, and the aluminium base of the buggy could be used as the base of other makes, or simply kept as a mounting for the engine.

Despite these limitations, this Stirling engine is still easy to work with, and in general, Stirling engines are probably the simplest of the mechanical engines to work into a build. They can run at much lower powers than internal combustion engines, and don't have any high-pressure parts liable to explode if pushed to their limits, (as with steam engines). Electric motors might be an easier option, but there's something just undefinably nice about building something with a mechanical engine driven with real fuel. Whether you want it for pure interest, or you're looking to add a steampunk aesthetic to the power source in your next make, Stirling engines are a fascinating option. □

ONLY THE BEST

The almost magical gesture sensors roundup

Control your next project with the wave of your hand

By **Marc de Vinck** @devinck

Most electronic projects require some kind of user interface or feedback. It might be as simple as a button, a potentiometer, or switch, which allows you enter a form of data or bit of feedback. But what if you want something with a little more flair? Or how about the ability to hide the interface altogether, yet allow for incredibly complex user feedback? If so, a gesture sensor might be just what your project needs.

You might be wondering exactly what a gesture sensor is and how it works? It's a good question, that involves a lot of variables. We'll start with what a gesture is, as it relates to these sensors in our Best of Breed review. A gesture is the action, or movement, of your hand, and with one of the following sensors or breakout boards your electronics project can interpret that movement to various degrees of accuracy. A simple example would be the ability to wave your hand from right to left indicating something similar to a button press. Where it gets more interesting is the ability to recognise 3D movements in space or gestures.

Most of us have seen at least one sci-fi movie where a character controls a computer interface with a simple, although typically exaggerated, swipe of their hand. And although we don't quite yet have the technology to make those computer screens appear in thin air in front of us, as they typically do in the movies, we do have the ability to interpret those waving hands to a fair amount of accuracy. It's one step closer towards bringing those sci-fi fantasies to life.

Keep in mind there are many different ways to have a gesture input integrated into your project. There are sensors and breakout boards like the ones we are reviewing, but there are also commercial products like the Leap Motion Controller, which packs some pretty amazing technology. You could also use a simple webcam to capture your movements with fairly high precision, or a more specialised camera that can capture both movement and depth. Whichever way you decide to add the ability for a gestural user interface in your project is ultimately up to you. The future is here – now go and make something.

Below ↙
Just add a holographic display for your own sci-fi computer interface

Skywriter HAT and 3D Gesture Sensing Breakout

SKYWRITER: 3D GESTURE SENSING BREAKOUT ◈ $23 | shop.pimoroni.com

SKYWRITER HAT ◈ $18 | shop.pimoroni.com

If you're interested in adding gesture control to your project, it's hard to go wrong with either one of these Pimoroni boards. Both the Skywriter HAT and the Skywriter: 3D Gesture Sensing Breakout board feature well-documented code and simple controls. The brains behind both of the boards is the Microchip MGC3130 sensor. Microchip claims it's the world's first electrical field-based, three-dimensional tracking and gesture controller. This integrated circuit (IC), coupled with its patented GestIC technology, allows for precise hand and finger tracking.

According to the Microchip datasheet, this IC should be capable of detecting movement up to 20 cm away, at a resolution of 200dpi. Although in practice, the range was a little closer to 5 cm for the Skywriter HAT and 10 cm for the Skywriter 3D board. Those are still pretty impressive numbers that will also allow you to embed, or hide, the boards behind something non-conductive like a sheet of paper, acrylic, or even a wood veneer. Think about the look of awe from someone interacting with your project without any visible electronics.

Both boards have great documentation that can be found on GitHub, including a nice tutorial on how to install the libraries and code. It's not every day that you see a developer include screengrabs and pinouts on their GitHub page, and I really appreciated the convenient single location for everything I needed to get the sensors up and running.

One thing that is very different with the boards is their levels of compatibility. The Skywriter HAT was designed to work specifically with the Raspberry Pi, Pi 3, 2, B+, A+, Zero, and Zero W, while the Skywriter 3D is compatible with all Raspberry Pi boards, and it's also just as happy working with your Arduino. The Python code for both of these sensor boards is well documented, and so is the library when you want to use the Skywriter 3D with your Arduino-compatible board.

Wiring up either board is fairly simple, although the HAT version simply plugs into your Raspberry Pi. The full-size Skywriter has only six connections; GND and VCC, along with four pins, GPIO 3, 2, 27, and 17. After you wire the board, or plug it in directly, all you need is to run the installation script to get up and running. It's really simple, even for someone with just basic electronics knowledge. And if you are interested in using the full-size board with your Arduino, you can easily modify the code supplied with the library to be up and running in no time. Pimoroni, the distributor for these boards, is doing a great job not only developing its new products, but also in developing all the code and documentation that makes using the products easy and fun. □

Left ◈
The Pimoroni Skywriter: 3D Gesture Sensing Breakout is a great choice for anyone working with Raspberry Pi and Arduino platforms

Below ◈
The Pimoroni Skywriter HAT is a little less versatile, but still a great choice

Credit
Pimoroni LTD

VERDICT

Skywriter: 3D Gesture Sensing Breakout

A great choice in gesture sensors that supports Raspberry Pi and Arduino

10/10

Skywriter HAT

Not as versatile as its big brother, but still a great choice

8/10

BEST OF BREED

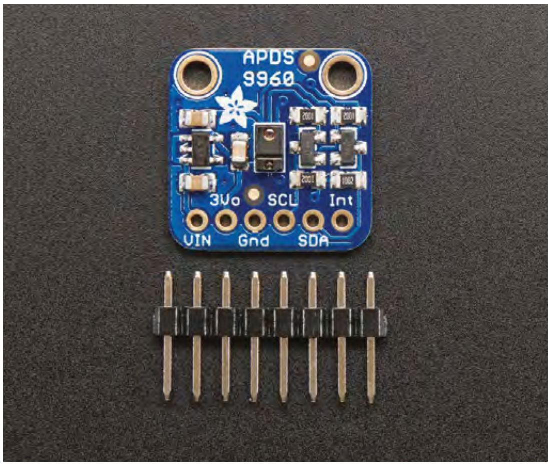
Adafruit APDS9960 Proximity, Light, RGB, and Gesture Sensor

ADAFRUIT APDS9960 ◈ $7.50 | adafruit.com

The Adafruit APDS9960 Proximity, Light, RGB, and Gesture Sensor is useful for gesture sensing and a whole lot more. This little breakout board uses an Avago Technologies APDS-9960, which includes an integrated IR LED and driver, along with four directional photodiodes that are able to detect the reflected IR energy from the onboard LED. It can accurately sense movement up to a few centimetres and at an 8-bit resolution. Coupled with the included library, you can easily detect motion from left to right, right to left, or up and down. That should cover most of your basic needs. If not, you could always dive into the examples and datasheets and write your own code for interpreting movements like zig-zag motions and more.

In addition to the gesture and proximity sensing, there is a light sensor, colour sensor and IR sensor. You can detect ambient light and detect the colour of some wavelengths of light. To be able to easily hook up the board, Adafruit also included a 3.3 V regulator and level shifting so it can be easily used with your favourite variety of 3.3 V or 5 V microcontroller. That's a lot packed into a board that is only 17.8 mm square.

Wiring up the board is simple since it uses the popular I²C interface. That means you only need four wires, power and ground, along with SCL, and SDA. It doesn't get much easier than that. In addition to the simple wiring, you will need some code, and Adafruit did a good job there too. To get up and running on the code side of things, you can either download the Adafruit_APDS9960 library for Arduino, which includes sample code, or a similar packaged code that works with Circuit Python. □



Left ◈ Don't let the size of this sensor fool you: it offers a lot of features

> " To be able to easily hook up the board, **Adafruit** also included a **3.3 V regulator** and **level shifting,** so it can be easily used with your favourite variety of **3.3 V** or **5 V microcontroller** "

USING A CAMERA

While these breakout boards make gesture recognition easy to do, even on low-powered microcontrollers, if you've got a bit more horsepower to play with, you don't need specialist hardware. Using a regular webcam and some clever software, you can see how a user is moving and respond accordingly.

To see how this works, head to **hsmag.cc/KUtbMq**, where you can play the 2048 game using a camera and a few flicks of the wrist. This uses the gest.js library (**github.com/hdmchl/gest.js**) to do the heavy lifting.

VERDICT

An extremely low-cost sensor with lots of features

9/10

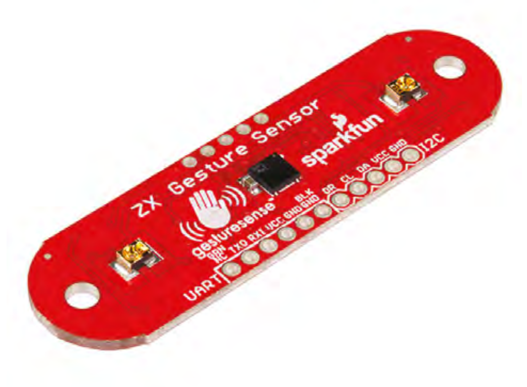
SparkFun ZX Distance and Gesture Sensor

SPARKFUN ZX ◆ **$24.95** | **sparkfun.com**

The ZX Distance and Gesture Sensor from SparkFun is another type of touchless sensor that can easily be added to your next project. The sensor was developed by SparkFun in conjunction with XYZ Interactive, using its GestureSense technology to recognise not only the distance an object is from the sensor, but also its location from side to side across the sensor. What is particularly interesting about this breakout board is the range of movement, which is 30 cm above and 15 cm side-to-side. This is a substantially larger area of sensing compared to some other products.

The way the ZX sensor works is straightforward. The board has two IR LEDs, one on each end, which send out IR light towards the object to be sensed. The IR light bounces off the object and is reflected onto the sensor in the middle of the board. This sensor can then calculate the reading and give feedback to your microcontroller or computer via I²C or UART communications.

SparkFun has done a fantastic job documenting this board. There's a very well written step-by-step guide that will get you up and running fast. You start by setting a few jumpers on the back of the board and then, like most other breakout boards, solder in a few header pins or wires as described in the tutorial. Once you have finished the simple soldering procedure, you can download their library, and well-commented examples, to get started sensing. Although this board is limited to two directions of sensing, it makes up for it with a great range and well documented code for easy setup. □



Left
Simple I²C or UART makes this sensor easy to use

VERDICT

Great for sensing objects up to 30 cm away

8/10

LEAP MOTION CONTROLLER

LEAP MOTION ◆ **$79.99** | **leapmotion.com**

You might be wondering about the commercial applications of gesture sensors, or a few of you might have played with them already. The Leap Motion Controller is a popular gesture sensor that hooks directly up to your computer, allowing for some amazing gesture controls. This little piece of hardware is much more sophisticated compared to the DIY boards. You can get gesture controls down to such a fine resolution that it can detect each finger of your hand and its location in a 3D space. It's a really amazing piece of consumer tech that really has to be tried to fully understand just how amazing its gesture sensing capabilities are in the virtual world.

Left
Leap Motion is adding its technology to a variety of different VR headsets

Credit
Leap Motion Inc.

Can I Hack It?
A WiFi-controlled doorbell?

Can a WiFi-controlled doorbell be hacked into a project?

Les Pounder

@biglesp

Les Pounder is a maker and author who works with the Raspberry Pi Foundation to deliver Picademy. He also helps teachers/learners to become creative technologists. He blogs at **bigl.es**

This month we take a look at a smart home device which promises to automate letting people into our homes, stream live video from the front porch, and catch the last ever Yellow Pages being delivered to our door. The DANMINI 720P Smart Doorbell can detect movement, see in the dark, and let us communicate with the post person from the comfort of our sofa thanks to a handy app.

While rated for IP55 water resistance, this smart doorbell has a few open spaces where moisture can intrude, even when used with the included cowl, so it would be prudent to ensure it is sheltered before use. The smart doorbell can be safely secured to a wall via the included screws and wall plugs. Around the back we have a removable metal plate that is used to secure the camera to a wall and enable the camera to be removed for maintenance. The plate also covers three ports, 12V power, a relay connection, and a 'network' connection.

To power the smart doorbell, we have an included DC 12V 1A power supply with a 5.5mm male jack, centre positive. As it's only 12V, we can easily power the unit from alternative sources (solar batteries, bank of 18650s, and suitable boost converter).

The included power supply has a woefully short lead of around one metre, so users will need to position the smart doorbell near a socket, or create their own power lead.

Below Measuring just 14.5cm × 6.5cm × 3.5cm, the unit can be easily attached to most walls and doors using the included mounting kit

YOU'LL NEED

DANMINI 720P WiFi Video Smart Doorbell

COST

£43.89

WHERE

amazon.co.uk

SENSING CHANGE

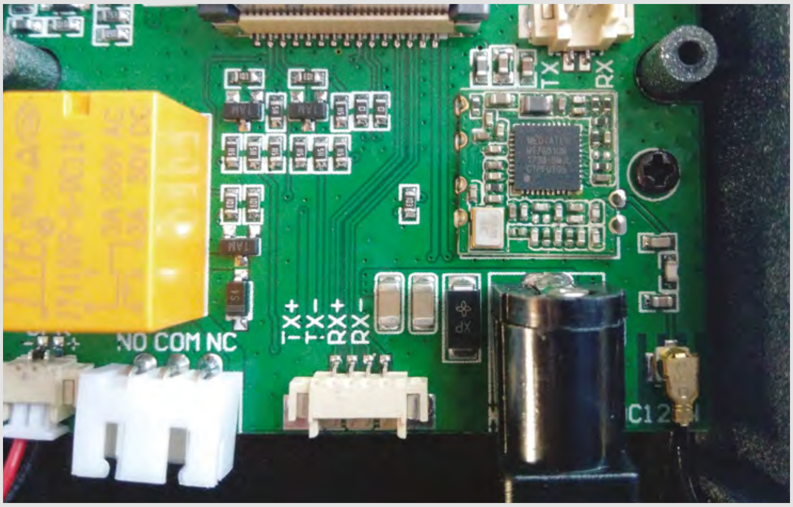
The camera is a two-part mechanism. First we have the lens, a 2.8 mm fixed-focus lens – fixed using a thread lock-type glue which can be easily broken off, if careful. The sensor is part of a GM8135S system on a chip (SoC) made for CCTV / IP cameras. The SoC has a built-in ARM 600MHz CPU and 512MB DDR RAM. The GM8136S also has IO ports for USB OTG (On The Go, as used with mobile devices), SD interface, I^{2}S (audio), I^{2}C and limited GPIO pins. These pins are not broken out for general use, but that will not stop the determined hacker. So if you are handy with SMD soldering, then you could quite easily attach your own devices to the camera. The camera board also features a photo resistor that uses light levels to alter its resistance and form a crude motion sensor with around a two metre range. Finally, around the camera we have four infrared LEDs used to illuminate the visitor at night, ready for the camera and its ISO6400 sensor to relay video. Also present is a connector for a two-pin electret microphone which is used to interact with any callers. The speaker connection, to enable the caller to hear us via the companion app, is located at the bottom of the unit and connects to the power board PCB.

INTERNAL EXAMINATION

The internal design of the smart doorbell is modular, with two boards present. These boards are for the camera, and the relay / power supply. The boards

A DIY APPROACH

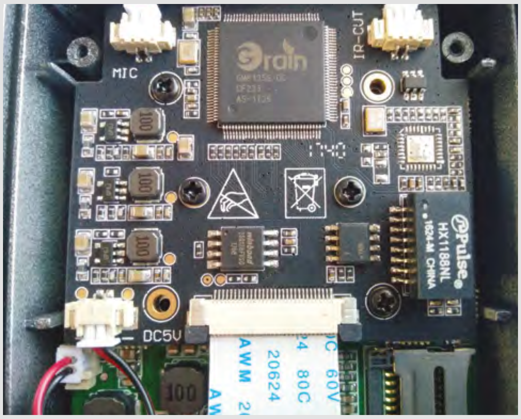
Smart doorbells are not that difficult to build. All they really are is two forms of input: a sensor and button. Once pressed, these trigger the camera to life and send video to a waiting device on the network. A maker could fabricate their own version for around the same price using a Raspberry Pi Zero W and camera, and the web streaming video is easy to do, thanks to the Picamera Python library: **hsmag.cc/qunoja** – take a look in the Advanced Recipes for 'Web Streaming' for a starter script.



Above The power board houses the DC power interface, a relay for our door lock, network access, and a MediaTek WiFi chip, enabling the device to connect to your WiFi

are connected using small two-pin connectors for power and basic input / output, as well as a ribbon connector similar to those used for the Raspberry Pi's camera port. The power supply board has the 5.5 mm female DC input, a four-pin 'network' connection, and a connection to a relay. The relay is rated up to a maximum of 250 V 3A AC, and 30 V 3A DC, which is useful as it's there to control remote door-locking mechanisms such as magnetic locks, commonly used with automated entry systems. Using the relay connections, we can have the relay keep a circuit closed until the door release is triggered, which breaks the circuit and the power to a lock for around 3 seconds. The reverse is also true for an open circuit, closed when the lock is triggered. The relay can be connected to any device that you see fit, acting as a simple switch mechanism, so we could use an ESP8266 to send a tweet or anything else using IFTTT (If This Then That). Also present on the power board is a microSD card slot used for recording images and video locally for security purposes, but the companion app also records video and images to your suitable mobile device.

Below The camera PCB uses a GM8135S system on a chip (SoC) powered by an ARM CPU. It also offers a USB interface and I2C for those with the soldering skills to hack!



MAKING CONNECTIONS

WiFi connectivity is thanks to a MediaTek MT7601U chip offering A,B,G and, N connectivity.

A smart doorbell is not very smart without an app. The DANMINI companion app is called WiFi DoorBell and it is used to set up the doorbell. When first set up, the smart doorbell acts as a wireless access point, enabling us to directly connect to it and then configure WiFi, passwords etc. Once configured, the app has two basic notifications: detecting movement around your door and indicating that a person has rung the bell. When a visitor arrives you can talk to them, ignore them, watch them without their knowledge, and then let them in (by triggering the relay), all from the app.

So can it be hacked? In a word, 'yes'. The camera would be trivial to hack if you had access to the GM8135S datasheet and a steady soldering iron. The modular nature of the smart doorbell also lends itself to further hacks with replacement, with enough space for small microcontrollers such as the ESP8266, Arduino Micro, ATtiny85, and access to a stable 5V power supply via the two-pin connectors dotted around the boards. So as well as a notification on your smartphone, a clever hacker could have the image of the caller sent to them via DM over Twitter.

The unit could also find its way into being used as an interactive prop for *2001: A Space Odyssey* fans, as the unit does bear a resemblance to HAL and, with a little 3D printing and painting, could be hacked into a more suitably inspired case.

The unit isn't cheap, but it offers an interesting insight into how consumer smart devices can be made using cost-effective components from the big manufacturers. □

INFO

- ARM 600 MHz CPU
- MicroSD card slot used for secure images
- Enough space for ESP8266

REVIEW

Brown Dog Crazy Circuits

From $50 | browndoggadgets.com

By Ben Everard @ben_everard

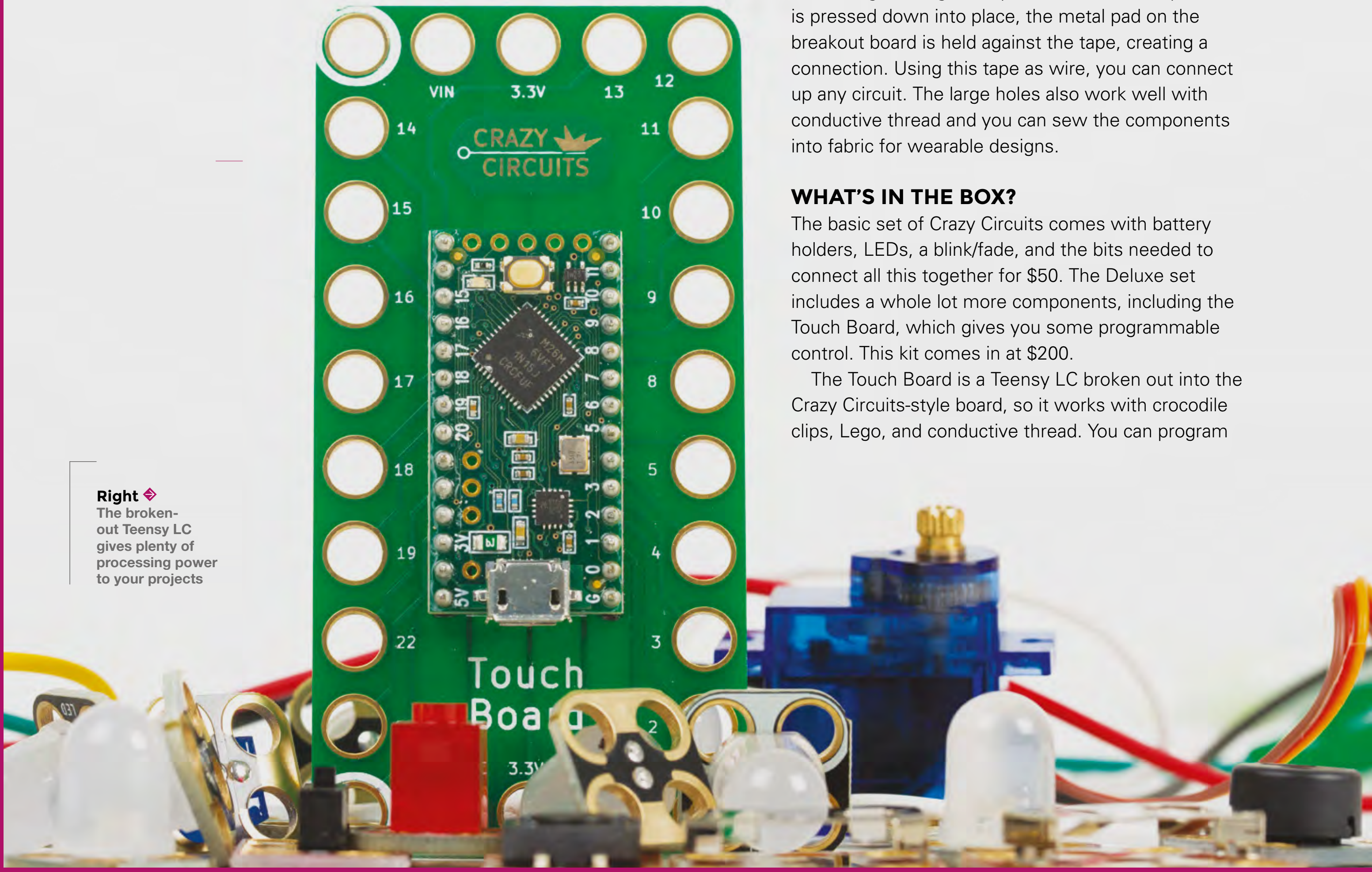
Brown Dog Crazy Circuits is a set of components on breakout boards, designed to help young makers learn the basics of electronics. The breakouts have large circular connectors that you can use to connect up your circuit in a number of ways. The most basic option is using crocodile clips. This enables you to quickly prototype your circuits. Once you've got everything working the way you like, you can then move on to incorporating this in a physical product in a variety of ways.

The connectors are arranged in the same pattern as the bumps on the tops of Lego bricks, and the kit includes the connectors needed to wire up your brick-based masterpieces. The Lego connections work with braided tape made of nylon, nickel, and copper. This runs along the Lego bumps and, when a component is pressed down into place, the metal pad on the breakout board is held against the tape, creating a connection. Using this tape as wire, you can connect up any circuit. The large holes also work well with conductive thread and you can sew the components into fabric for wearable designs.

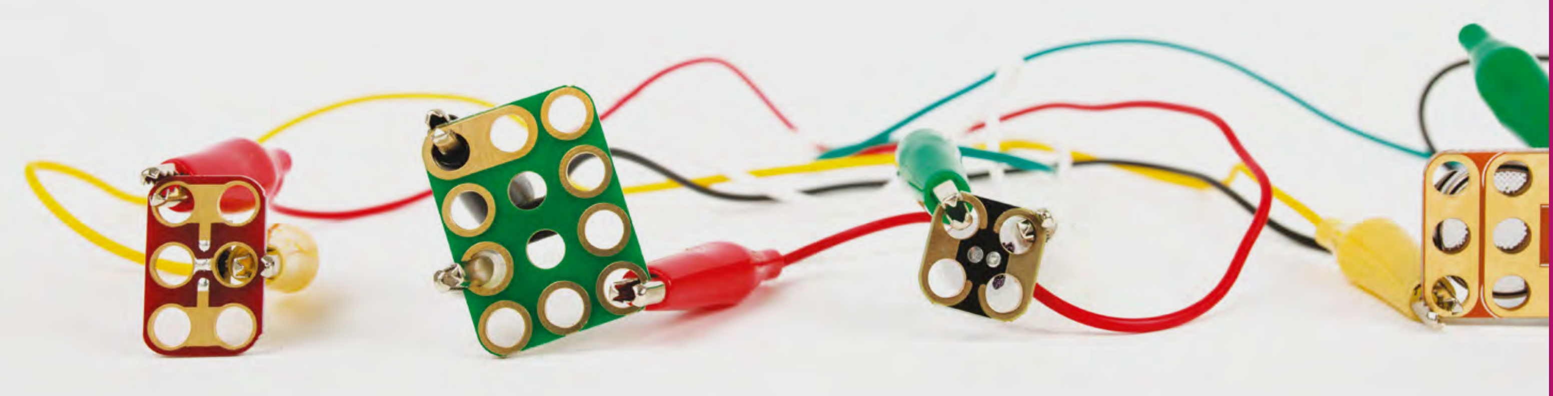
WHAT'S IN THE BOX?

The basic set of Crazy Circuits comes with battery holders, LEDs, a blink/fade, and the bits needed to connect all this together for $50. The Deluxe set includes a whole lot more components, including the Touch Board, which gives you some programmable control. This kit comes in at $200.

The Touch Board is a Teensy LC broken out into the Crazy Circuits-style board, so it works with crocodile clips, Lego, and conductive thread. You can program



Right The broken-out Teensy LC gives plenty of processing power to your projects



Above Crocodile clips are the easiest way to prototype with Crazy Circuits

this via the Arduino IDE in the same way you can any other Teensy board. It's powered by an ARM Cortex-M0 processor, which should provide enough power for most maker projects.

As the name suggests, the pins on the Touch Board can take capacitive touch input on eleven of its pins. This gives you a wide range of options for manipulating your builds with your fingers.

Perhaps the most exciting kit for us is the Robotics Kit ($150) that comes with a broken-out Arduino Nano and a set of Lego-compatible servo brackets for the four servos (two continuous rotation). This gives you quite a powerful set of tools for automating things you've made with Lego.

KEEP IT SIMPLE

Brown Dog also makes kits with squishy circuits, sewable, robotics, conductive paint, and others, but these are built on the same technology as the other kits.

We'd love to have some networking options, as this would give you options for building an IoL (Internet of Lego) network. There are some pins available via male headers, which does give you the option to hook this up to, for example, an ESP8266 for network access, but this adds another level of complexity to your projects.

None of the methods for working with Crazy Circuits is particularly suitable for complex schematics. Whichever method you're using, (crocodile clips, Lego tape or conductive thread) tends towards chaos as you make more and more connections. This is the price paid for circuits that integrate well with Lego and clothing, and it's a price worth paying for getting young people excited about intermingling electronics and physical things.

Crazy Circuits may look expensive when compared to electronics kits that come with just bare components, such as those for working with breadboards, but this extra expense is justified by the extra options that the form-factor allows.

Above The storage box keeps everything tidy when you're not using it

While the Crazy Circuits hardware is all based on existing technology like the Teensy LC and Nano, it's great to see that Brown Dog has put together a comprehensive set of documentation specifically for this product. As well as user guides that cover the basics of how to use the hardware, there's a projects section on the website, with a good selection of projects for you to try. These show off the different features and provide a good basis for planning your own projects.

We really like the effort that Brown Dog has gone to to make it easy to work with, from a simple circuit demonstration to integrating this into a device – of course, device in this context is something built by young makers. A simple circuit making an LED flash may not be that exciting on its own, but when it's powering, for example, the lights on top of a Lego police car or making stars twinkle on a Christmas jumper, it's much more attractive to learners. □

VERDICT

An electronics kit with great options for incorporating circuits into physical builds

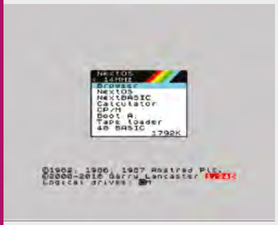
9/10

REVIEW

ZX Spectrum Next development kit

£142.80 (inc. VAT) | specnext.com

By **Gareth Halfacree** @ghalfacree



```
NextOS
  14MHz
Browser
NextOS
NextBASIC
Calculator
CP/M
Boot A:
Tape loader
48 BASIC
            1792K

©1982, 1986, 1987 Amstrad Plc.
©2008-2018 Garry Lancaster [illegible]
Logical drives: [illegible]
```

Above Warning: The ZX Spectrum Next may cause feelings of extreme nostalgia

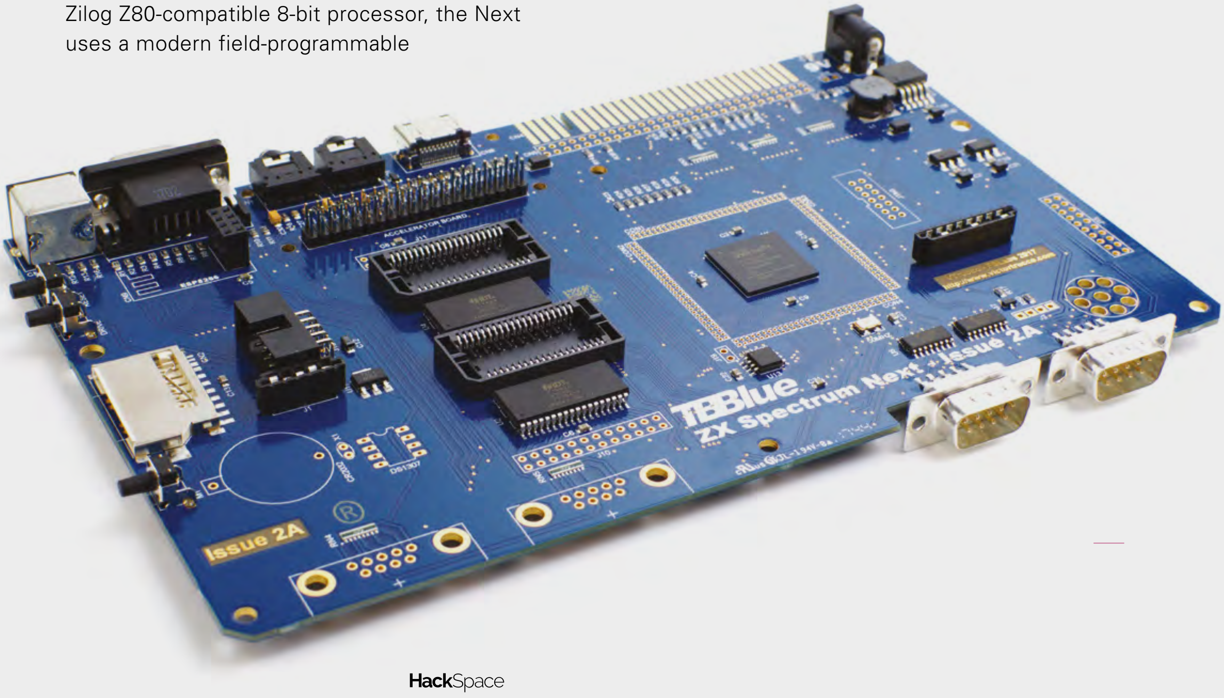
Sinclair's original ZX Spectrum, launched in 16kB and 48kB flavours in 1982, was the great success story of the British computing industry. Selling five million units over its lifespan, the low-cost colour microcomputer is credited with launching the careers of some of the biggest names in gaming.

The ZX Spectrum Next, by contrast, is a device unlikely to sell five million units. The brainchild of Henrique Olifiers, its production funded by a barnstorming Kickstarter campaign last year, the ZX Spectrum Next is, at its heart, a clone of the ZX Spectrum – one officially licensed by Sky In-Home Services, the current holder of the Spectrum and Sinclair Computers intellectual property – with some serious upgrades.

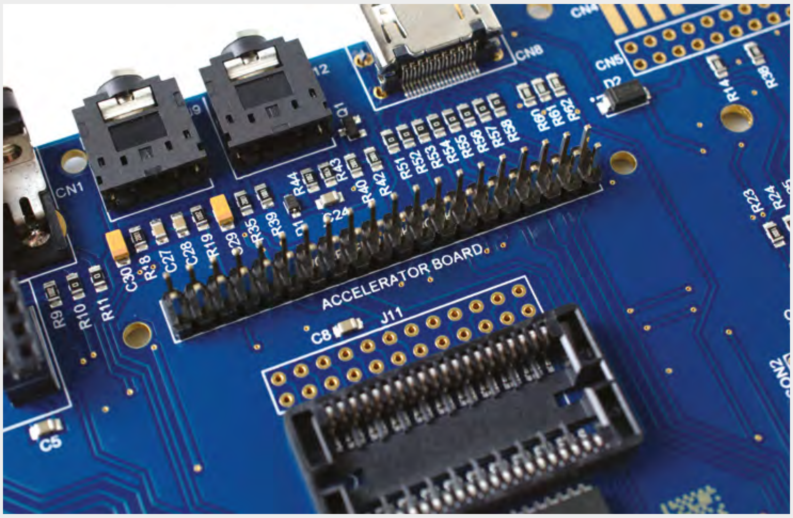
Where the original ZX Spectrum was based on a Zilog Z80-compatible 8-bit processor, the Next uses a modern field-programmable gate array (FPGA), a chip that can be programmed to replace almost any other chip. Using files known as 'gateware,' this FPGA acts as the Z80 along with the other components that made up the original Spectrum and, with the press of a key, any of its various successor devices.

With plenty of room to spare – the Z80 being a relatively simple device by modern standards – the FPGA is also used to power some of the Next's additional features, which range from an SD Card slot from which programs can be saved and loaded, to a trio of FM sound synthesis chips, which upgrade the original Spectrum's beeps and boops to multi-channel audio.

The Next isn't an emulator, though. Each of these FPGA 'cores' is a functional recreation of the original chip, running exactly like real hardware.



Right The ZX Spectrum Next is a dream machine for fans of Sinclair's original microcomputer



Above The Next can be upgraded with additional RAM, a WiFi module, and even a Raspberry Pi Zero co-processor

The result is a device which is, with very few exceptions, completely compatible with both software and hardware made for the original Spectrum – including plug-in cartridges and, with time taken to cut holes for the additional ports present on the Next, the original ZX Spectrum 16/48kB casing and keyboard.

The Next's upgrades don't end with the FPGA, though. Where the original Spectrum had low-quality RF-modulated video output which could, with a small modification, be converted to slightly improved composite video, the Next includes crystal-clear digital HDMI and analogue VGA outputs. There are two joystick ports as standard, with room for another two, and connectors for the original 'dead flesh' Spectrum keyboard or a more modern external PS/2 keyboard. There's even 1MB of RAM, upgradable to 2MB, which sounds limited by modern standards, but is a vast improvement on the 128kB maximum available on a stock Spectrum.

EXPANDING HORIZONS

If that list weren't enough, hardware designer Victor Trucco decided to keep on adding more: a header on the board makes room for a low-cost ESP8266 WiFi module for network connectivity, while there's space for a real-time clock module to be soldered in place. An 'accelerator' header even allows a Raspberry Pi Zero to be connected to the Next for use as a co-processor board – albeit one which dwarfs the system's primary hardware in speed and memory capacity.

Using the Next is by and large a pleasure. FPGA cores and the custom-written NextOS operating system are loaded onto a full-size SD card and inserted into the Next, which is then connected to a 9 V power supply – not, sadly, included in the bundle, and woe betide anyone trying to use an original Sinclair power pack, as a switch in polarity means you'll be left with a dead Next.

With power, display, and a keyboard connected, the Next boots into its operating system and displays a friendly menu through which the BASIC programming language, CP/M operating system, or image files from tapes or disks can be loaded. As a fully-compatible Spectrum clone, there's also an option to load original media: the 3.5 mm audio jacks are present and correct, allowing for easy connection to a tape deck.

Original Spectrum software runs perfectly on the Next, and two accelerated modes – which switch the Z80 core from its stock 3.5 MHz to 7 MHz or 14 MHz, with a planned 28 MHz mode temporarily removed for stability reasons – make games that would have struggled originally, such as those based on the Freescape 3D engine, tick along impressively quickly.

LOOKING AHEAD

Some of the Next's promised features, though, are not yet available. Software to make use of the optional Raspberry Pi Zero accelerator board or ESP8266 network module has yet to be written, and while the Next has the ability to display 256 colours at once and control hardware sprites, these features are limited to simple demos.

> **Using the Next is by and large a pleasure.** FPGA cores and the custom-written NextOS operating system are loaded onto a full-size SD card

A bigger issue than unfinished software is a hardware flaw in the board revision – 2A – sent to crowdfunding backers, whereby a lack of smoothing capacitor on the 9 V regulator causes noise on the power lines bad enough, in some cases, to cause instability. Thankfully, that's a relatively simple fix and one which is fully resolved in the 2B variant, which will be shipped to new customers and to those picking up the cased version of the Next, housed in a keyboard chassis custom-designed by Sinclair's own industrial designer Rick Dickinson.

At £142.80 uncased, and £252 cased (both inc. VAT), the Next is an expensive trip down memory lane – but one which ticks an awful lot of boxes and, with a little work on the software front, could bring joy to hackers of a certain vintage for many years to come. □

VERDICT

A promising, though niche, device that just needs a little longer in the workshop.

8/10

Leveraxe Classic MX

€89 | leveraxe.com

By Ben Everard ben_everard

The axe is possibly the world's first power tool – depending of course on how you choose to define 'power' and 'tool'. It builds up kinetic energy as it travels downwards and imparts this in a small strip of wood that's unfortunate enough to be in line with the blade. As the arc continues into the wood, the wedge shape of the axe uses the mechanical advantage of the crack to lever the two sides apart. It's a brilliant design that's survived almost 50 000 years of technological advancement more or less intact. However, can this basic design be improved?

The Leveraxe works in a fundamentally different way to previous generations of axes for splitting wood (known as mauls). It still has a sharp edge that comes swinging down and embeds itself into the wood. However, unlike previous axe designs, the head isn't symmetrical. The blade travels down a few centimetres into the wood, then stops and starts to rotate. The user has to let the handle twist in his or her hand to allow this to happen. The crack is levered apart, and this splits the wood more efficiently than a traditional axe – at least, this is the theory.

Handling the Leveraxe is a bit of a struggle at first because it feels so unnatural to let the handle twist in your hands. Years of axe handling have led us to instinctively grip the handle tight, which doesn't allow the Leveraxe to do its trick, but after a little practise, we're able to get into the routine of relaxing the grip on impact.

The twisting of the axe-head can knock the wood over, so it's necessary to hold it in place as you split it. The recommended technique is to fill a car tyre with wood you want to split and use this to hold it all in place as you work. Without this, you'll be constantly putting the wood back upright.

We tested the wood with some straight wood (which should have been easy to split), and some twisted and knotted wood for an extra challenge, from a range of different tree species. We found it split the straight wood without too much difficulty, but struggled at the tougher jobs. When something really needs a big wallop, the Leveraxe just didn't seem to be able to do the job.

At 6 lb (2.7 kg), the Leveraxe Classic MX is on the light side of splitting axes (and there are lighter versions of this axe), and performs reasonably without too much force. However, for anyone comfortable using a heavier maul, we remain unconvinced that this design is a significant improvement over the old tried-and-tested models. □

Below The asymmetric axe-head is unlike any other axe and it transfers the downward swing into a rotation to push the sides of the log apart

> " The twisting of the axe-head can knock the wood over, **so it's necessary to hold it in place as you split it** "

VERDICT

Easier to swing than a traditional maul, but struggles on twisted or tough wood

6/10

Making Time

Bob Clagett ◆ Kindle £3.74 Paperback £7.47 | iliketomakestuff.com

By Andy Clark @Workshopshed

Making Time looks at the journey from hobby maker through to professional YouTuber, and how Bob runs his channel and business.

Bob Clagett is a professional maker on YouTube, with the channel 'I Like To Make Stuff'. He's made over 200 videos about his projects which include CNC, tools, furniture, and a Raspberry Pi Arcade machine. He also has the Maker 101 series in which he looks at the basics of electronics, Arduino, and 3D printing. Bob also co-hosts the bi-weekly 'Making It' podcast, along with Jimmy Diresta and David Picciuto. Until 2015, Bob balanced all of this with a full time job in software.

The first section of the book is an autobiography. As a child, Bob liked to build things from cardboard boxes. He also used his construction toys to enhance other toys, such as making spaceships and forts. His grandfather was an inspiration, and Bob used to watch and help him in the workshop. Bob's high school and college life was also influential and, as well as helping him achieve a software career, it also provided some of the skills he'd need to be a professional maker.

One of the things we enjoyed about Bob's story is that you can see how he's picking up skills along the way. Each of these small pieces fits together to provide him with a complete toolkit to achieve his end-result.

The second part of the book covers Bob's thoughts on his current maker business. He looks at how he deals with feedback and interacting with the community, and how he stays productive and produces quality content. Sometimes he needed to stick with an idea and some he changed if they were not working. Bob shares his purpose and the impact he has had on others.

The process of making good videos every week requires planning, so Bob takes us through his process and what tools he uses.

There are useful thoughts on how to make money from being a maker, and when to say no.

Like many makers, Bob is continuously learning as he goes. But he's also getting in help where it makes sense. Bob sees this journey as the first step for his new business, and finishes the book with a few inspirational thoughts on what you, the reader, might do. □

VERDICT

An inspirational book for anyone looking to take their hobby to the next level.

9/10

next month

issue #6

ON SALE
19 APRIL

FEATURING PAPER ENGINEERING

ALSO

- → GLOBAL VILLAGE CONSTRUCTION SET
- → READING GLASSES THAT READ ALOUD
- → AUTOMATIC LIGHTING
- → MAKE YOUR OWN GIN
- → LOTS OF LEDS
- → AND MUCH MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

WARNING
HACKERS AT PLAY

hsmag.cc