WHAT 3D PRINTER?

Find the ultimate replicator for 2020

Building a kiln

Melting glass with a Raspberry Pi

+

▶ **CIRCUIT PYTHON**

▶ **SOLDERING WITH GAS**

▶ **PICKING AN IMPACT DRIVER**

▶ **SEWING MACHINES**

Arduino data log

Store info on your microcontroller

CNC milling

Carve out objects automatically

KIM FREEBURN

Talks autonomous racing robots

PAINTING CANDLES HAIR-DRYERS INFILL

Welcome to HackSpace magazine

Automatic replicators have been the stuff of science fiction for generations. For a decade or so now, 3D printers have been on the verge of promising to make this a reality, but they've been fiddly and hard to get to work reliably at an affordable price. In the last year or two, it feels like there's been a real shift, not in the underlying technology, but in its ease of use. Non-technical (or moderately technical) users can now buy affordable 3D printers and use them reliably, and that's a huge step. Of course, it hasn't happened uniformly across the industry, so in this issue, we're looking at the best 3D printers around, and what experiences people have had in their homes and workshops. Could 2020 be the year 3D printing goes mainstream?

In the last year or two, it feels like there's been a real shift, not in the underlying technology, but in its ease of use

BEN EVERARD
Editor ben.everard@raspberrypi.org

Got a comment, question, or thought about HackSpace magazine?

get in touch at
hsmag.cc/hello

GET IN TOUCH

hackspace@
raspberrypi.org

hackspacemag

hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

PAGE **48**
SUBSCRIBE TODAY

Contents

126

Tutorial
Build a baby monitor

Cover Feature

WHAT 3D PRINTER?

Discover the results of the first in-depth community 3D printing survey

18

Direct from Shenzhen

USB power adapter

56

114 Phenomenal USB power with an itty-bitty price tag

Interview

Chan'nel Vestergaard

106

60 From self-taught Arduino tinkerer, to visiting MIT fellow, via shoes made from pineapple

116

96

Giant LED

By **Robert Fitzsimons**   hsmag.cc/OlgTmx

This giant LED is completely pointless, but as a great man once said, all art is quite useless. It is, however, completely beautiful. Everything is created by Robert, from the casting to the handmade moulds, the only off-the-shelf parts being the electronic components. He's made a series of videos documenting the process, showing a Zen-like focus on perfectionism that we think shines through in the finished article.

In the first version, for example, the legs were made of 6 mm copper tubing; in the final version, they're cast resin with brass welding rods embedded for strength and conductivity. To get the right compromise of transparency and diffusion, Robert has included 0.0075% zinc oxide (white pigment) into the resin that makes up the LED bulb, and every aspect of the build has undergone loads of tiny tweaks and modifications to get as close to perfection as possible. This is a great build. □

Left ◨
Modelling the shape of the LED was easy (says Robert): it's just a sphere, a cylinder, and another slightly larger cylinder, with a chunk taken off on the cathode (short leg) side

Dragonfly

By **Jack Spiggle** ✈ **hsmag.cc/xyEjKy**

This is a BEAM robot, which stands for biology, electronics, aesthetics, and mechanics. BEAM robots are characterised by extremely simple circuitry, controlled directly by hardware and reactions to sensors, rather than logic programmed into a microprocessor. They tend to be very simple, performing only one function – robots may move toward a light source, for example, or away from a heat source. As such, they're a way of physically modelling the behaviour of simple creatures, motivated by primal things like a need for food, water, or WiFi.

This flapping, solar-powered dragonfly was built by Jack Spiggle, aka NanoRobotGeek, using the guts of a broken radio-controlled flying toy dragonfly. Most of it has been replaced with a snazzy free-form soldered circuit made from brass rods.

The dragonfly stores energy from a solar panel in capacitors until it reaches 4 V, at which point it dumps it into the wing-flapping mechanism. You can see capacitors soldered into the dragonfly's tail – these add up to 2200 µF, which is only enough to give the wings a slight flicker. To add extra power, Jack hid an extra 4700 µF capacitor in a hole drilled into the wooden base under the solar panel. This combined 6900 µF is enough storage to power the wings through a whole cycle. ◻

Right ↗
We've tried free-form soldering before, and it's harder than it looks to get anything that works, let alone that looks good
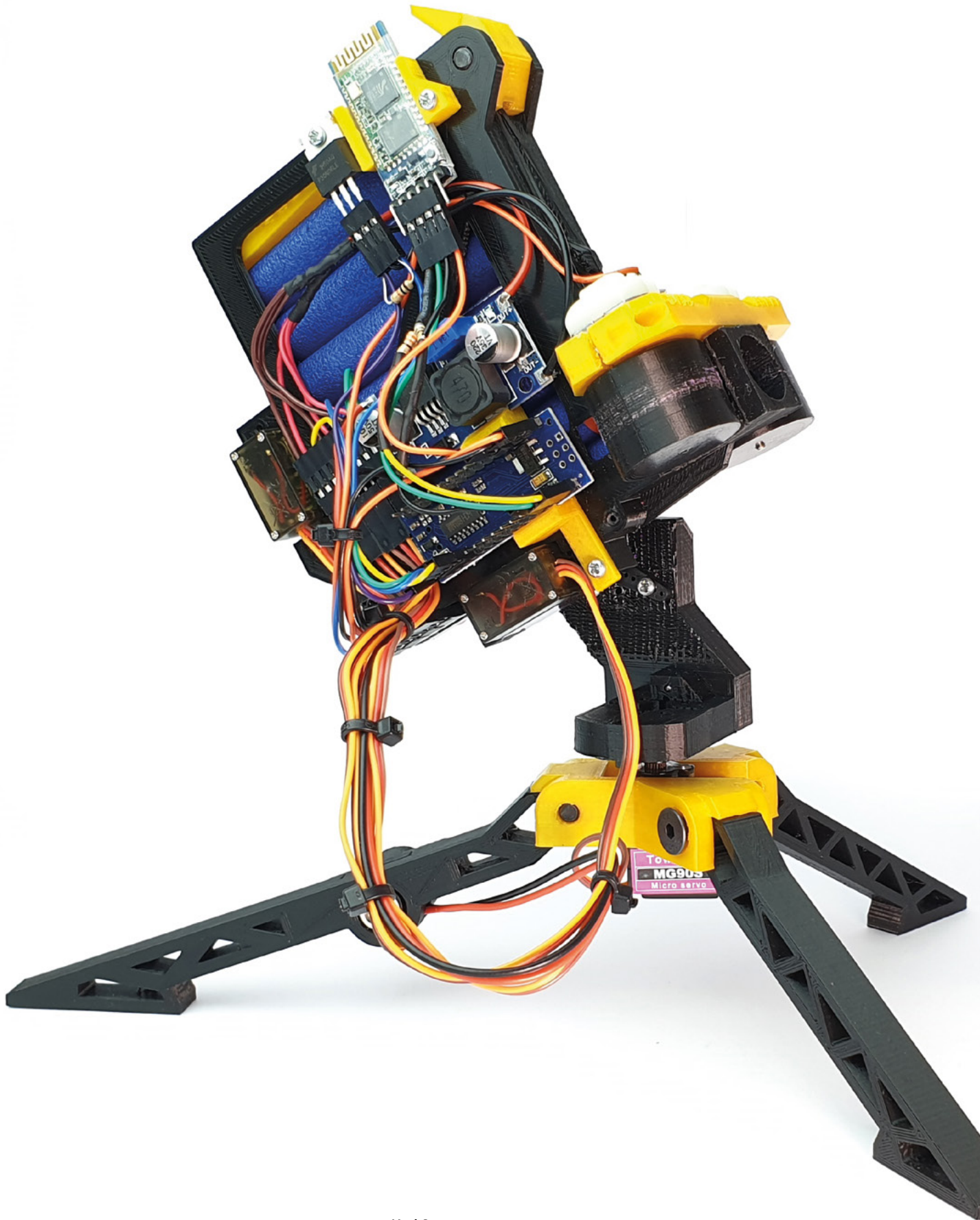
Bluetooth Nerf gun turret

By **Kevin Drouglazet**

hsmag.cc/9fnxr5

This security-conscious build is a completely 3D-printed Nerf gun turret, controllable by the user over Bluetooth. It uses an Arduino Nano to control two servers (for tilt and pan); two DC motors to fire the darts, and there's a servo to push the darts into the rollers when the order to fire is given.

Kevin's written Windows and Android versions of the control software; just move the cursor on your screen to control the turret and keep pesky kids off your lawn. The software, bill of materials, and design files for the 3D-printed parts are all available from Kevin's website, so anyone can make their own version. □

Left
They're going to rise up one day. Why can't you people understand?

Right ↗

Wes rewired the
charging status
LEDs to the outside
of the new/old case.
It's really a stunning
piece of work

NES project

By Wes Swain   🡕 Geeksmithing.com

I **n this build, Wes Swain has reanimated the dead husk of an old Nintendo Entertainment System and replaced the guts with… a NES Classic Edition, the new version of this classic console that Nintendo released in 2016.**

Wes has made several improvements. He's added a projector, so you don't even need a screen to play – a friend wearing a white shirt will suffice. He's added conductive charging to the wireless controllers and to the console itself, so the whole thing is portable, rechargeable, and battery-powered.

The build had a few ground rules, the first of which, do no harm, is what Wes calls the geek Hippocratic Oath. No vintage hardware was killed in the making of this project – in fact, while sourcing parts for the build, Wes managed to fix two 'broken' NES consoles he bought off eBay. ◻

Ted the Talking Toaster

By 8 Bits and a Byte · hsmag.cc/FTyXZ8

This is Ted the Talking Toaster, made by Dane and Nicole, creators of the wonderfully surreal 8 Bits and a Byte YouTube channel. The project started life as an ordinary toaster, but thanks to a Raspberry Pi 3 Model B and a Google AIY Voice Kit, it can now talk at least as much sense as 70% of the humans we know.

Logic is provided by Google Dialogflow running on Google Cloud Platform, and Raspbian on Raspberry Pi.

For a little extra whimsy, the toaster has been given eyebrows – these are controlled by two servos glued into holes drilled through the aluminium wall of the toaster. According to the angle and motion of the service, these can convey a range of emotions, including the has-to-be-seen-to-be-believed 'frisky'. Now would anybody like any toast? ◻

Right ◈
Ted's human servants have attached him to the internet, so you can converse with the machine over a video chat yourself:
hsmag.cc/nLQFpi
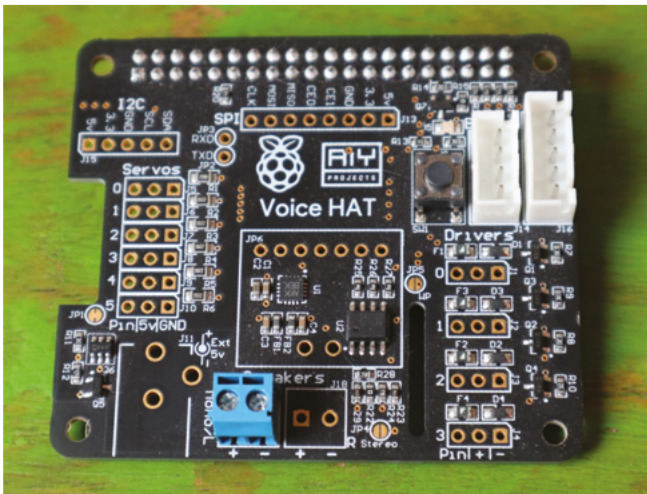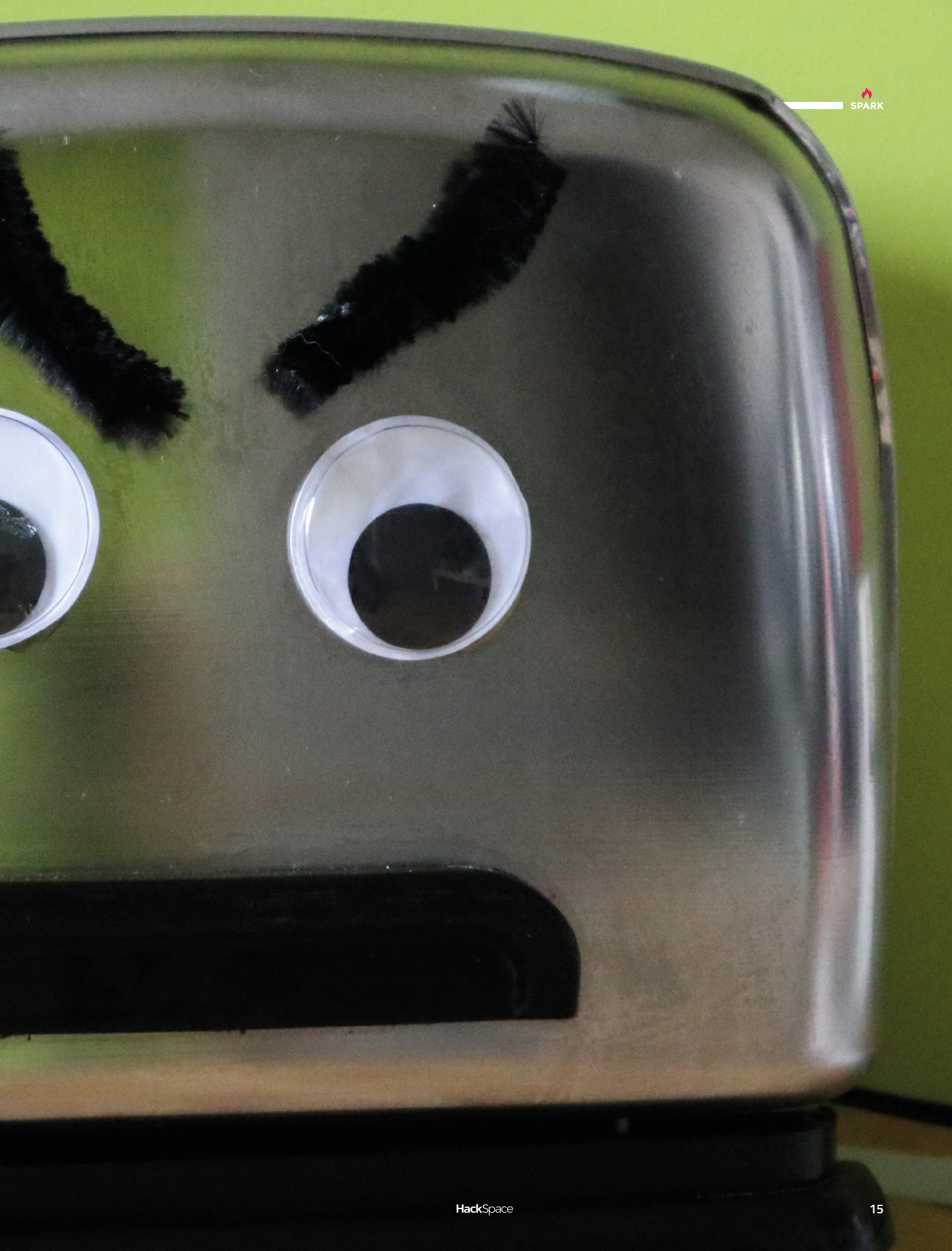
Objet 3d'art

3D-printed artwork to bring more beauty into your life

Sadly for carnivores, there's increasing evidence that meat production is bad for the planet. We like to think of ruddy-faced farmers in bucolic countryside scenes, but increasingly cows are reared in vast factory farms, producing greenhouse gases, and industrial quantities of effluent.

One person who wants to give us an alternative is Giuseppe Scionti, CEO of Novameat. He's developed a printer that uses a mixture of pea and rice protein to create entirely plant-based steaks.

3D printing isn't incidental to this manufacturing process – it's what gives it its defining characteristic. Pea and rice mushed together doesn't sound too meaty, but the lines deposited by the printer mimic muscle fibres in animal meat, giving a steak-like appearance and texture. ▫

> 3D printing isn't incidental to this manufacturing process – **it's what gives it its defining characteristic**

Meet The Maker:
Kim Freeburn

From a land down under to the Silicon Fen – meet the maker
determined to put robotics in the hands of the people

"**W**e [Kim Freeburn and Arron Churchill] are PiBorg, and we make educational robots and add-on boards for Raspberry Pi, including the MonsterBorg, DiddyBorg, and ThunderBorg.

"Just under ten years ago, we started a company to be able to take on contract work in whatever form that suited us at the time. We moved to Cambridge, and started off in a very small garage on the side of a house. We went to trade shows looking for contract

> ❞ Just after the first version of Raspberry Pi was released, **we got our hands on one and immediately saw the potential** ❞

work, and got occasional work through that. Between contracts, we wanted to create our own products and grow our own brand. Just after the first version of Raspberry Pi was released, we got our hands on one and immediately saw the potential. We decided to make robotics to go with it. We thought cyborgs and Raspberry Pi – hence the name PiBorg!"

PiBorg was one of the first companies to realise the potential of third-party add-ons for Raspberry Pi. In the case of PiBorg, the key product was a motor controller for Raspberry Pi that took the hard work out of building a robot on the Raspberry Pi platform.

"Getting to where we are was a long journey. The first iteration of our control board was far too →

Right ◈
Kim says the best bit of running PiBorg is working on new projects – especially the ideas phase

KIWI

electronics

SCHOOL
FAB LAB

ROLLS
RR
ROYCE

Rol
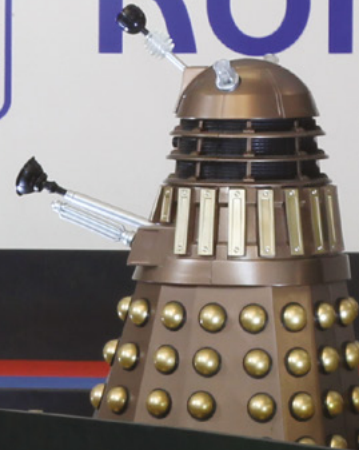
Above ⬈
Kim's pictured here with Arron Churchill, the resident software person, who codes everything from the PiBorg racers to the on-board logic

complex to be useful for education. We started to make very simple boards such as RGB LEDs, accelerometers, and so on. We slowly added more boards and robot kits. We did three Kickstarters – one for a motor control board, one for our Formula Pi, and another for a more advanced robot kit [the MonsterBorg] to go with it."

Today the PiBorg range comprises robot kits, motor controllers, Raspberry Pi add-ons, plus parts such as motors and wheels. And as well as the robotics, Kim's also involved in Formula Pi: is a self-driving robot racing series, in which competitors send in their code to control autonomous racing robots, competing against each other in challenges and races. Competitors have come from over 20 countries. The robots are all maintained by Formula Pi: everyone uses the same kit – the only difference is your skill as a programmer.

SMALL BEGINNINGS

"We were both working as software developers for a large company in Portsmouth [before PiBorg]. I was writing embedded software to control the servo position, Arron was writing host software for it. We met through that company and found that we had skills and work practices that complement each other. We decided to have a go at setting up a company, which is where Freeburn Robotics started."

Like many of the makers we speak to who have gone on to form businesses, PiBorg's growth hasn't been completely linear:

"What do we know now that we wish we did when we started? That you should start small and stay small for quite a while. We went from zero to five employees very quickly, and it was a mistake that was unsustainable." □
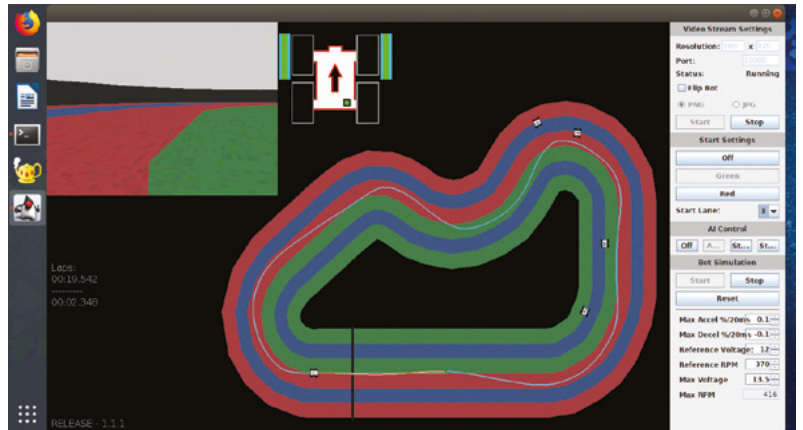
Above
Competitors in Formula Pi can download sample code to get them started – then the competition begins

Left
The ThunderBorg controller can handle up to 5A per connection, making it ideal for bigger robots

Get ready to rumble

Behind the scenes of a robot fight

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**

Backstage, or 'the pits', of a heavyweight robot fighting event is interesting. There are a couple of heavyweight robot fighting tours that go around the UK. The organisers erect an arena inside a leisure centre, and roboteers bring their creations along and compete in competitions – usually judged on damage, aggression, and control. All this becomes a show that is open to the public, where VIP guests also get a tour around the pits.

I have recently joined one of the robot teams, and have started touring with them. Helping to build a robot in the comfort of a workshop, with all the tools at hand, good lighting, and no time pressure is one thing. And, a thing I really enjoy. Trying to fix a battered robot, when all you have are the tools that you remembered to put in the car, a collapsible table, and one floodlight, and you need to get it working within 30 minutes ready for your next fight, is quite another!

And, it's noisy. The music and commentator for the show adds to the crash and bang of 110 kg robots hurling themselves at each other, and ricocheting off the arena walls. Which means that ear defenders are required.

> You can't plan for everything – you never know quite how the robot will break!

Communication with your teammates gets tricky. As does listening for the continuity test beep from a multimeter.

I am used to making things outside of my own workshop. For example, I once made a Windsor rocking chair – in a wood. But there, everything was set up for it – tarps covered the pole lathes, the open fire fed the steamer for steam bending, the shave horses were already on site etc. The weather, the smell, the birds, the composting toilet, all added to the experience, and made it a very memorable make.

Another kind of memorable make was soldering NeoPixels to cables – on a crowded table, while on my knees (I didn't have a chair), while being sensory deprived – it was cold and dark and I was wearing ear defenders. There was also the constant knowledge that I had to be quick, as we still needed to put the armour back on the robot. And yes, I am sure with a little more forethought, many of these things could be alleviated. But you can't plan for everything – you never know quite how the robot will break!

However, the buzz of pride when our robot does the things it is designed to do, and does them well, in a competition setting, outweighs the uncomfortable conditions. I can put up with a lot for that. □

The rise of the FPGA

Reconfigure your chips to suit your project

Drew Fustini

🐦 @pdp7

Drew Fustini is a hardware designer and embedded Linux developer. He is the Vice President of the Open Source Hardware Association, and a board member of the BeagleBoard.org Foundation. Drew designs circuit boards for OSH Park, a PCB manufacturing service, and maintains the Adafruit BeagleBone Python library.

F PGAs have been the talk of the town at many of this year's hacker conferences. But what exactly is an FPGA, and why are they so hot right now?

FPGA stands for Field Programmable Gate Array, a digital logic chip that can be programmed to reconfigure the internal hardware. An FPGA does not run software — it physically changes the configuration of its gate arrays to adapt to the task at hand, making an FPGA an incredibly versatile tool. Need 25 PWM pins for a project? No problem. Want to replicate the functionality of a vintage CPU? Your FPGA has you covered. Not only is an FPGA versatile, but it is also better at handling timing-critical tasks than a microcontroller. You can filter high-speed sensor data before it's read by your processor, or offload repetitive tasks like debouncing buttons, reducing the burden on your microcontroller.

FPGAs are hot right now, but they're not a new technology — they've been used in industry for decades. However, pricey FPGA development boards, and the massive proprietary software suites needed to develop FPGA designs, means that there has historically been limited adoption in the maker community.

When you write an Arduino program, you're using a language called Wiring, a variation of C++. When you press compile, this high-level code is used to generate a binary file that the processor on the Arduino board executes to achieve the purpose of your program. Similarly, we describe the circuits we want in an FPGA using a high-level, text-based format known as a hardware description language (HDL), such as Verilog. The HDL design is then transformed by a synthesis tool into the basic building blocks that exist in the FPGA.

This process is normally done with proprietary software from the FPGA vendor, but these tools can take ages to download and devour disk space. That is, until Project IceStorm, led by Clifford Wolf, enabled a complete open-source workflow for the Lattice iCE40 FPGA.

> **FPGAs are hot right now but they're not a new technology – they've been used in industry for decades**

This opened the door for low-cost, open hardware boards such as myStorm BlackIce, TinyFPGA, iCEBreaker, and Fomu, which are great tools for teaching workshops and building projects.

The Lattice ECP5 FPGA is capable of more advanced features than the iCE40, and it now has an open-source workflow too, thanks to Project Trellis led by David Shah. This enabled the ECP5-powered Supercon badge to have cool features like HDMI video, while still being open for anyone to hack on without requiring proprietary tools.

FPGAs are a fascinating technology with lots of awesome applications. If you want to find out more, start off by reading Luke Valenty's *The Hobbyists Guide to FPGAs* on **Hackaday.io**, (**hsmag.cc/G0AQwR**), and watch Tim Ansell's Supercon talk to learn about the exciting future of open-source FPGA tools (**hsmag.cc/kyStPD**). ▢

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**

KEEPING IT REAL

I'm not in the market for a US$145 Nixie tube. But there is something extremely comforting about the thought of a chap making them by hand, completely by hand, in the Czech Republic. It's even better that his factory is in a castle. It's a bit like when Kate Bush did her live run at Hammersmith Apollo – I was never going to go, but I'm glad she's out there doing her thing.

Kevin Campbell
Glasgow

Ben says: Quite so. Only, if you're developing a high-end product that uses Nixie tubes and charging top dollar for it, you probably don't want to use Nixies that have been sitting in some warehouse in Ukraine for the last 30 years. Dalibor's a perfectionist – his stuff is the best that money can buy.

SWAG

I normally hate thinking about Christmas stuff in November (about 23 December is early enough for me), but cheers for reminding me about Boldport kits. That's two stockings filled, and an extra one bought for me.

Janet
Ashford

Ben says: Just remember: the gift of soldering is for life, not just for Christmas. Treat your Boldport purchases with love, and they'll give it back a thousandfold.
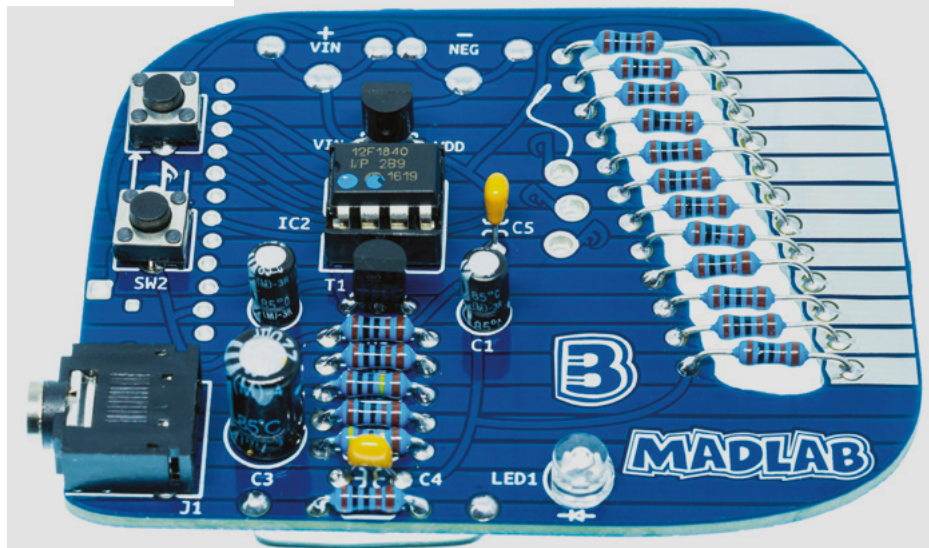
THE FINAL FRONTIER

The makers in the Netherlands messing around with leaf blowers and bog iron are geniuses! There's something really special about making right from the bottom-up, buying in as little as possible, and doing it all yourself. Smelting your own iron ore is pretty hardcore, but I wonder if they made the charcoal themselves to do it?

James Caraway
Minnesota, US

Ben says: Making your own charcoal? Pah! A real completist would terraform their own planet, genetically engineer their own trees, and make charcoal that way. Seriously, there's nothing wrong with building on the work of others if that's what you want to do – someone said something once about standing on the shoulders of giants, which may be applicable here.

SPACE

Interesting to read about Hackerspace. gr – I'd always assumed that the Libre Space Foundation came out of an EU-funded programme, like CERN. The idea that makers are putting things into space from what is basically a shed is mind-blowing.

David Worthington
Cumbria

Ben says: The most valuable bit of any scientific project is the people running it – the brains put things into space, not the building.

ERRATA

We embraced the Silicon Valley culture of failure last issue, perhaps a little bit too much so. In the Bela and BeagleBone feature on page 74, we failed to mention that it was written by Helen Leigh. Apologies, Helen, and to writers everywhere. Also in issue 25, a bad link in the PDF version led anyone hoping to read about the RC2014 micro kit to somewhere completely different. Sorry for the inconvenience.

CROWDFUNDING
NOW
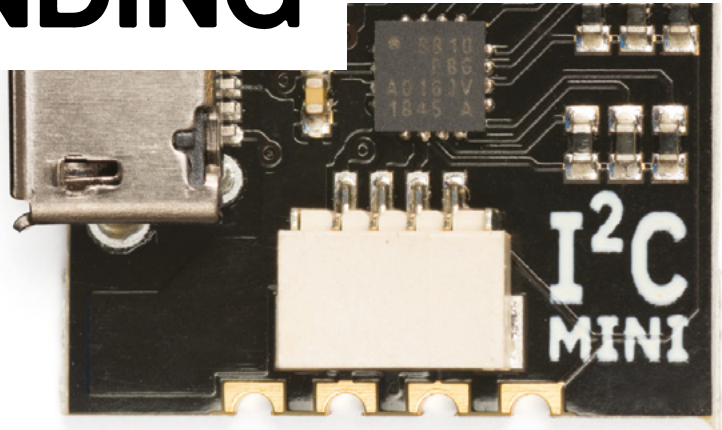
I²CMini

Linking hardware to your desktop

From $17 | **crowdsupply.com** | **Delivery:** Jan 2020

²C is a very common protocol for connecting sensors, actuators, and other devices to microcontrollers. It requires two wires and is usually quite easy to use. However, what if you need to connect an I²C device to a laptop or desktop?

You'll need some extra hardware. The I²CMini provides a little bridge between your computer's USB port and an I²C bus.

Of course, this is just half of what you need – as well as the hardware, you'll need some software to get everything talking to each other properly. There's a GUI, command line interface, and APIs for C, C++, and Python, so you should be able to use whatever tools you're most comfortable with.

You could use this device for quickly setting up sensor readings, controlling builds from your PC, testing out hardware, and loads of other things. It comes from the same Excamera Labs that make I²CDriver, a more heavyweight I²C-to-computer interface, so there's some proven track record for the people behind this product.

> **"** You could use this device for quickly **setting up sensor readings "**

A USB-to-I²C bridge isn't the most exciting hardware. It's not the fastest thing or the most powerful doodad. However, we makers are in many ways constrained by the tools we have available, and having better tools means we can make better things. If this tool can help people get better control of their I²C buses, then it'll be a great addition to the maker's arsenal. ◻
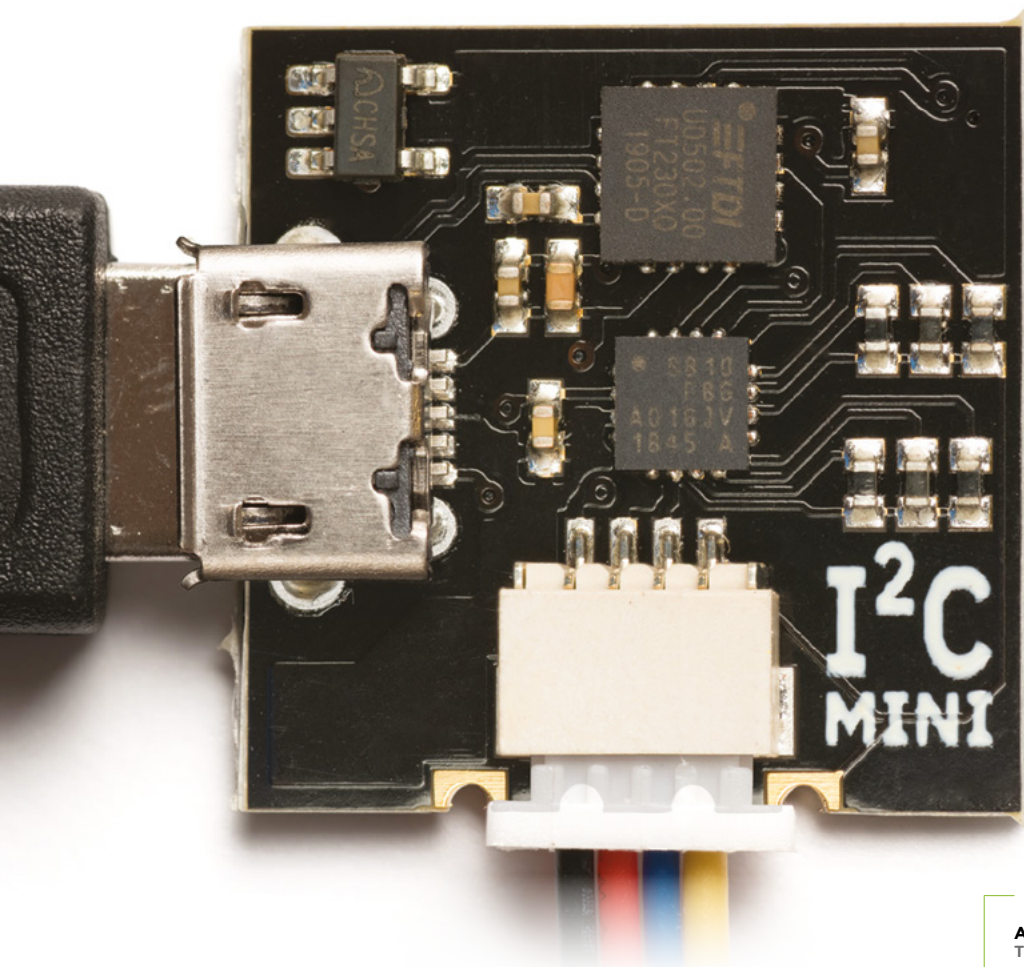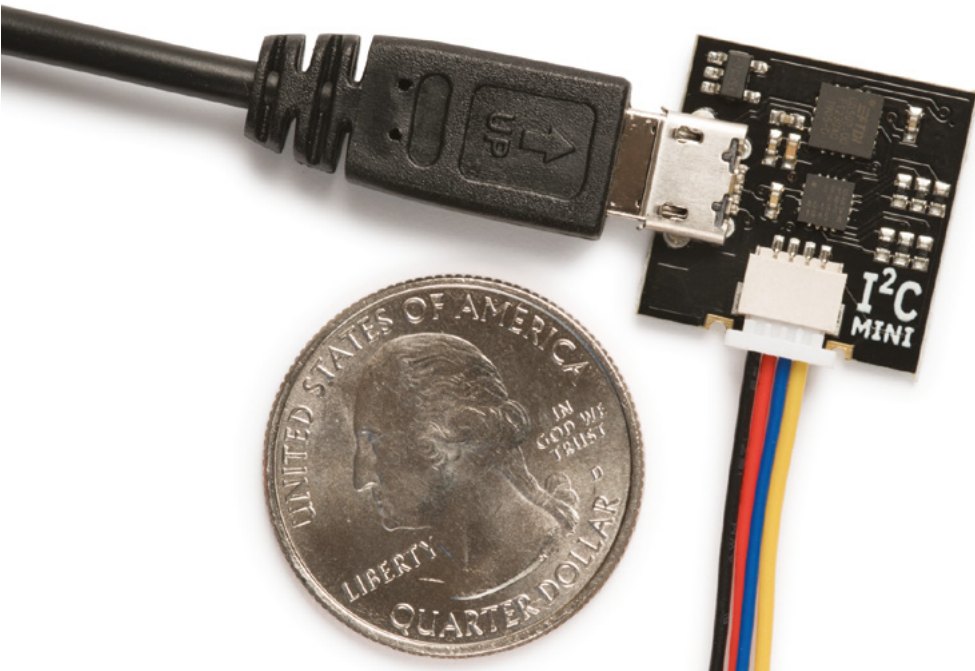
Above ◩
The I²CMini is, as the name suggests, quite small – just 61×49 mm

Space of the month:
Geekspace Gwinnett

We got started from a few in our core group that got together using Meetup.com to host meetings in a pizza restaurant [says Steve Smith, President of Geekspace Gwinnett]. Geekspace first opened in 2013 in its first smaller location [in Gwinnett County, Georgia, USA]. The rent got beyond what we could sustain, so we put the stuff in storage and waited for a good space. We found a local town called Suwanee that had an initiative to provide education-focused items to the area, and they offered assistance with the current rent. It helps keep the cost down so we can focus on the fun.

We run a load of differing groups at Geekspace Gwinnett, so our calendar is always packed. We have a FIRST [For Inspiration and Recognition of Science and Technology] robotics team (high school), a FIRST Lego League team (much younger kids), and a LAN party group, as well as our normal makers.

We also have monthly meetings to keep up the collaboration. I find that if we all keep in touch, awesome things happen. Many high schools in the area are very tech-focused. We'll even be working towards our second Girls Who Code workshop this summer.

Any given day, someone is lasering/CNC-ing/crafting/teaching something. Our members are very creative and always bring the best out. Like any initiative, it starts slow, and if you keep it up, people notice and things snowball to bigger, better, cooler stuff. ◻
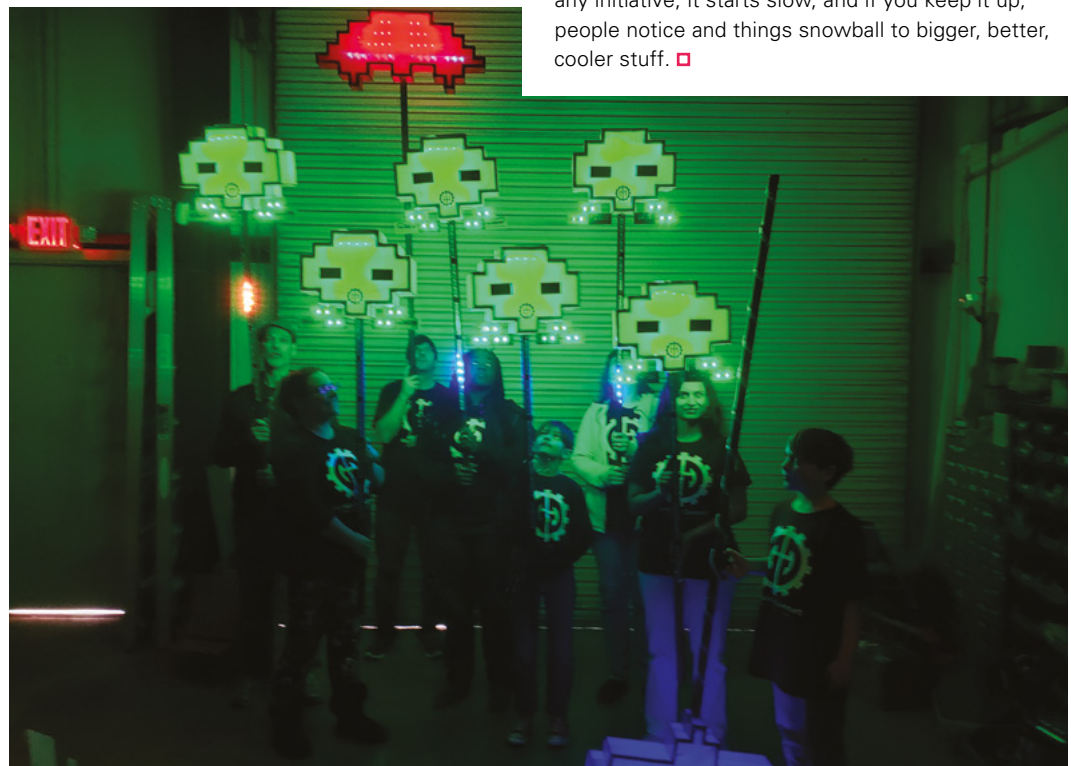
Right ◆
Geekspace Gwinnett participates in Maker Faire Atlanta, Dragon Con, local school STEM nights, high school robotics, Odyssey of the Mind, and many other events

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.

Above

Come to Geekspace Gwinnett for skill-building workshops on many topics, including woodworking, electronics, CNC routing, welding, laser cutting, 3D printing, sewing, and costume design

Build a Makerspace for Young People

Join our **free online training course on makerspace design** to get expert advice for setting up a makerspace in your school or community.

Sign up today: rpf.io/makerspace

Raspberry Pi

#RPiLearn

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

WHAT 3D PRINTER?

Find the ultimate replicator for 2020

3

D printing may no longer be the new kid on the making block, but that's a good thing. Over the past decade, we've seen the rise of homemade printers, the gradual introduction of affordable commercial models, and then these commercial models become more and more reliable. There are now literally hundreds of different models of 3D printer out there; some are very similar to each other and based on open-source designs, while others are vastly different and unique. We wanted to know how they work – not in the test lab, but out there in the real world, where real makers are making real things.

Throughout October and most of November 2019, we ran a survey asking people who used 3D printers to let us know about their experiences with different models – the good, the bad, and the ugly. This way, we can get a good idea of what printers are worth buying and which will struggle. We extend our thanks to the hundreds of makers who let us know their experiences.

Let's take a dive into the results, and see if we can find out what are the best 3D printers around. →

SATISFACTION

Do people like their 3D printers?

We got hundreds of responses to our survey, and here we're going to focus on the twelve most **popular printers.** There were many more that we didn't get enough different reviews of to build up an accurate picture.

We asked a broad question 'How satisfied are you with your 3D printer?' The average across all printers was 4.3/5. In other words, most people are pleased with their 3D printer choices.

The 3D printer with the most satisfied users is the Prusa MK3S, with an average rating of 4.88. What makes this even more impressive is the fact that the second most-liked printer is the earlier version of this printer – the Prusa MK3. Third place goes to the Anycubic i3 Mega.

At the other end of the scale, the Anet A8 and Monoprice both scored 4.0/5 – something that may be partly explained by them also being the cheapest printers on test. Despite this being the lowest score on test, they still show that most people who have these printers like them.
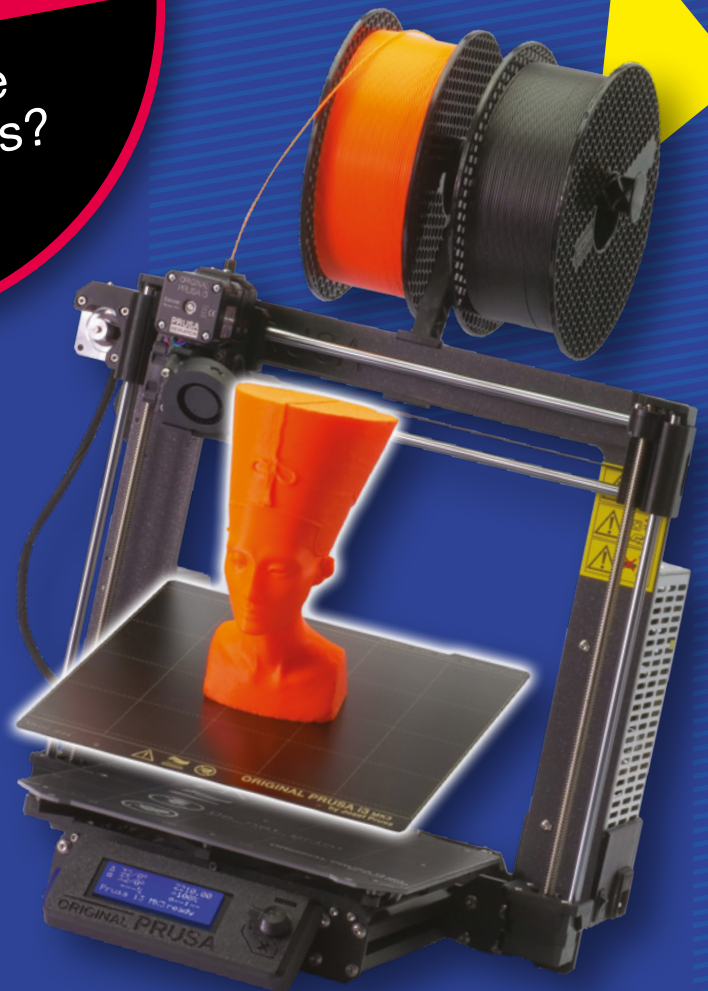
We'd also like to give honourable mentions to the Creality Ender-5 and the Flashforge Creator Pro. Although we didn't get enough responses on either of these to give a verdict with confidence, the responses we did have were highly positive, so they're definitely printers to watch.

WOULD YOU RECOMMEND THIS PRINTER TO A FRIEND?

On the whole, most 3D printer users would recommend their current printer to a friend. Across all printers, 89% would recommend their printer to a friend. However, across the most popular 12 printers that we're looking at in more detail, 96% would recommend them.

Only one printer, the Anet A8, had more than one person not recommend it, and that had 13% of responders say they wouldn't want a friend buying this printer.

RANKING
out of 5

Prusa MK3S	**4.86**
Prusa MK3	**4.78**
Anycubic i3 Mega	**4.75**
Creality Ender-3 pro	**4.72**
Flashforge Finder	**4.71**
Ultimaker 2+	**4.67**
Prusa MK2S	**4.55**
Creality Ender-3	**4.42**
Anet A6	**4.41**
Creality CR-10	**4.14**
Anet A8	**4.07**
Monoprice Select Mini V2	**4.03**

DOES IT PRINT?

The fundamental job of a 3D printer is to produce 3D prints (as opposed to vast amounts of plastic noodles).

We asked 'On average, approximately how many prints fail?' with a set of options ranging from 'one in five' to 'fewer than one in one hundred'. There's more than one way of looking at this data. We've calculated the average failure rate for each printer – the total average across all printers is one print in 36.7 failing. Bear in mind that this is based on people who have been using their printers for a while – you might find that you get significantly worse than this initially with a 3D printer until you've got your settings dialled in. You might also have a lower success rate if you're printing particularly large/challenging models. The winner here is the Anycubic i3 Mega, with the Prusa MK3 and MK3S coming in second and third. The lowest scores are the Monoprice Select Mini V2 and the Anet A8, at 17.5 and 20.8 respectively.

Another way of slicing the data is by looking at the most common response to the question. The basic order of the printers doesn't really change with this, but it does perhaps give a more realistic expectation for how often prints fail. Across all printers, the most common response was one in 20, but this ranged from one in ten for the Monoprice Select Mini V2 and Anet A8, to one in 50 for the Prusa MK3S and Anycubic i3 Mega.
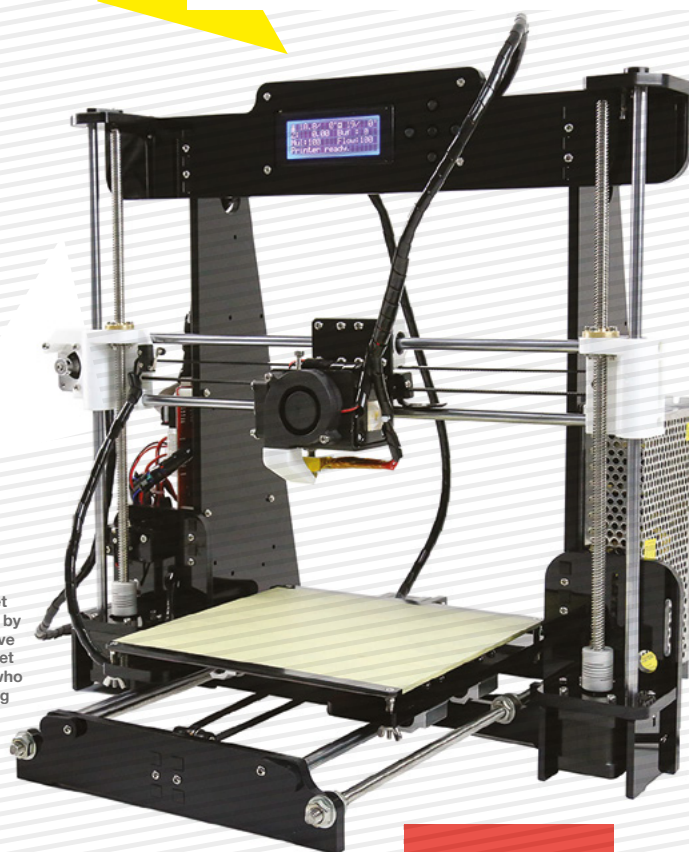
Right ◈
While the Anet A8 isn't loved by all, it does have a dedicated set of followers who enjoy tinkering

UPGRADES

Do people feel the need to increase their printer's power?

The least upgraded printer is the Prusa MK3 (0.1 upgrades per printer) followed by Flashforge Finder (though several Finder owners listed 'lack of upgradeability' as the worst thing about the printer), and jointly the Ultimaker 2+ and Anycubic i3 Mega (0.5 upgrades per printer). The most upgraded printers were the Anet A8, the CR-10, and the Ender-3 Pro, all with 1.7 upgrades per printer.

The takeaway from this is that if you're planning on buying a printer, make sure you budget for the parts you'll want to upgrade. The most common upgrade is the print bed (with 33% of printers having an upgrade here – that number goes up to 54% for CR-10s, 50% for Select Mini V2s, and 50% for the Ender-3). For budget printers, this can quickly add up to a sizeable proportion of the total printer cost, but it does mean that you can get the exact set of features you want. →
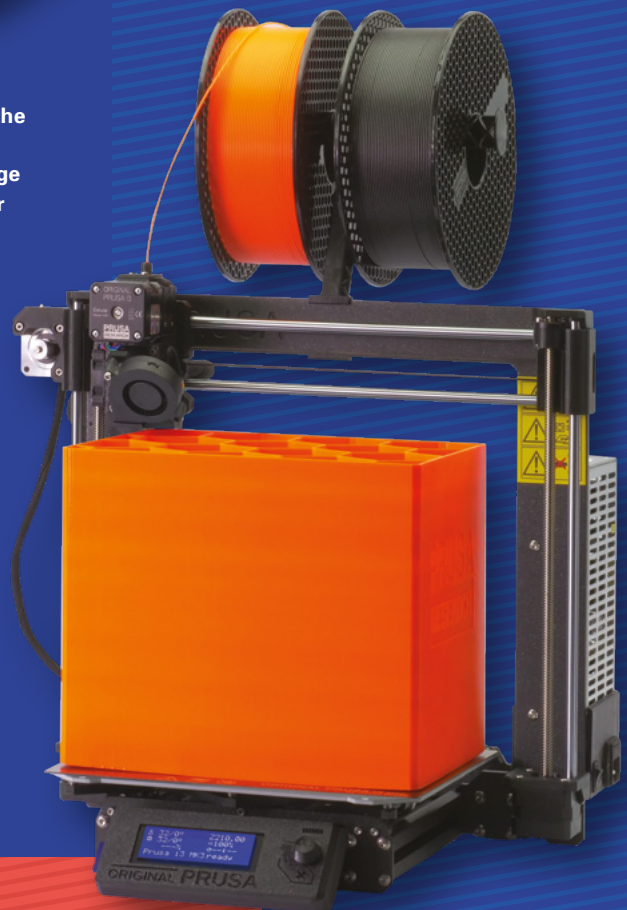
Right →
Yes, you really can use the full print volume

CHANGES

What have you upgraded?

Legend:
- ■ Other
- ■ Changed the extruder
- ■ Changed the drive system
- ■ Changed the frame
- ■ Changed the control board
- ■ Added auto levelling
- ■ Added temperature sensing
- ■ Changed the motors
- ■ Changed the print bed

SOFTWARE WOES

When we asked for the worst aspect of your software setup, as well as the selectable options, we gave you the ability to enter anything you wanted. Lots of people used this to vent their frustrations with the software, and three themes kept coming up:

1) The split between design and slicing – people want the ability to slice (and possibly send to the printer) directly from the 3D design tool they're using.

2) Better control over the supports – whether it's control of placement, or the design of the supports themselves (such as tree supports), many users feel that their needs aren't being met by the current slicer options.

3) Lack of support for multi-material prints in 3D design tools – while some slicers can now help you print with more than one filament (either through support in the printer itself or by pausing to allow a manual change), this hasn't yet filtered down to design software.

% of printers upgraded (can be over 100 because of multiple upgrades per printer)

Chart categories: A6, A8, CR-10, Ender-3, Ender-3 Pro, Finder, i3 Mega, i3 MK2S, i3 MK3S, i3 MK3, Select Mini V2, UM2+, **Total Result**

Y-axis: 0, 50, 100, 150, 200, 250
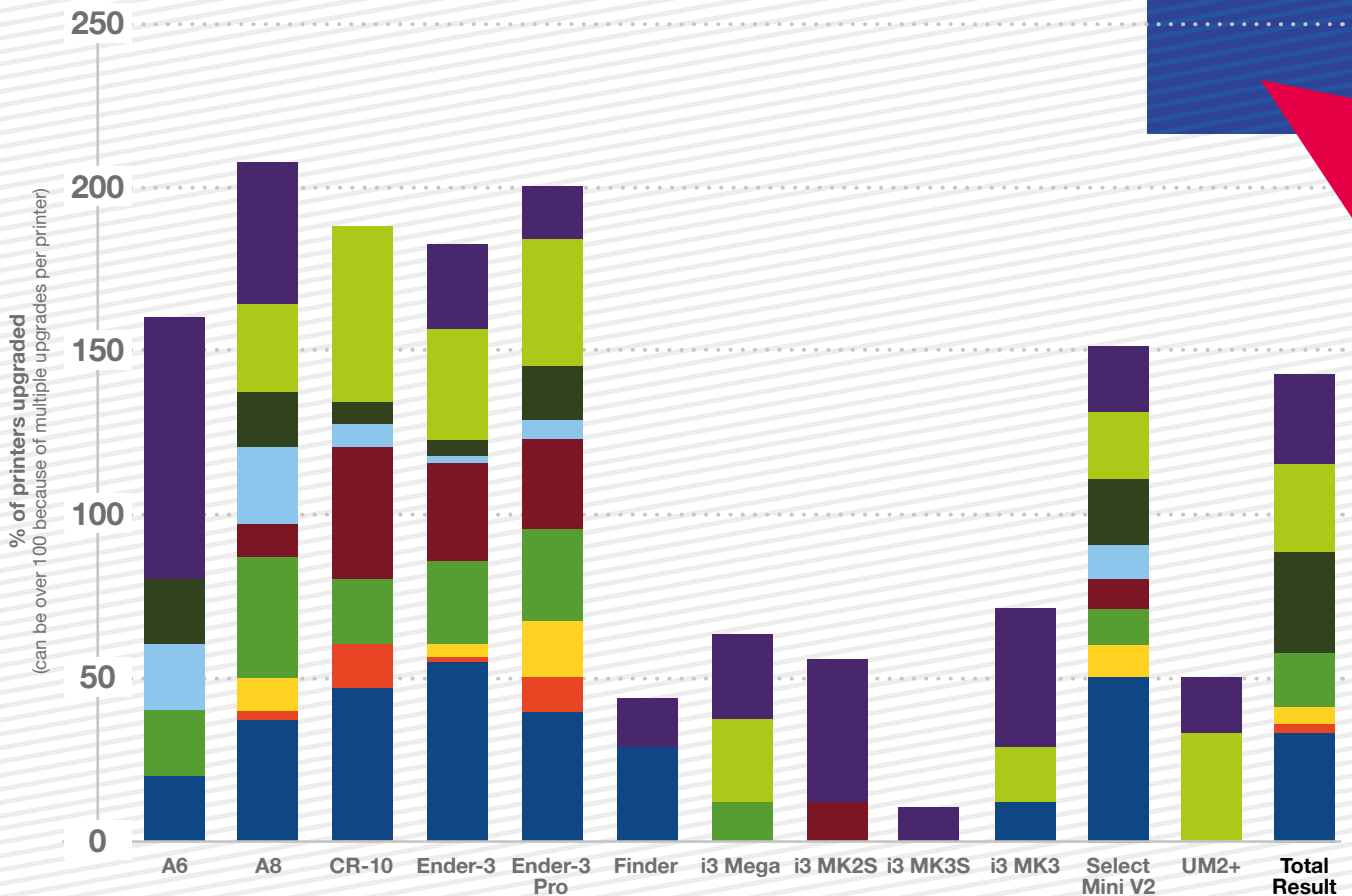
THE AVERAGE 3D PRINTER

On average, people using 3D printers…

Would like a bigger print area

Only use PLA

Give their printer five out of five stars

Slice using Cura

Design using Fusion 360

Have one in ten prints fail

Print one to five times a week

Have upgraded at least one aspect of their printer

Download models from Thingiverse
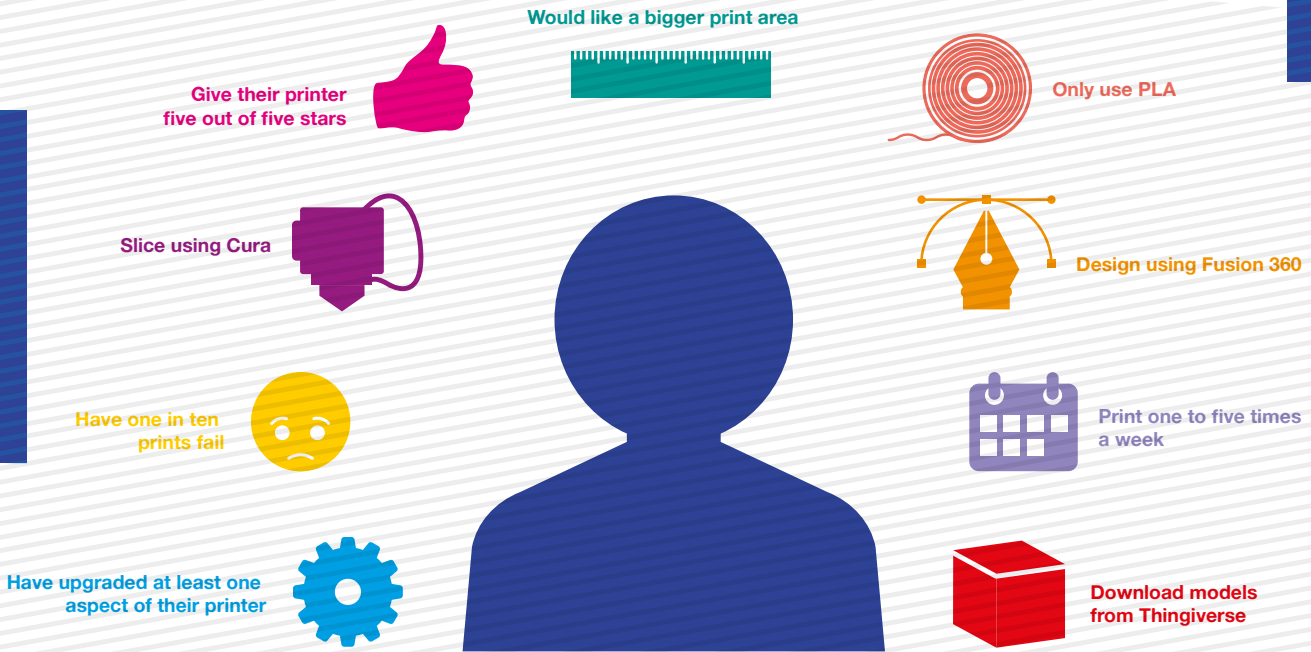
3D PRINTING SOFTWARE

Total exceeds 100% because users could select multiple options

What's the most popular slicer?

56% Cura

16% PrusaSlicer

9% Slic3r

What's the most popular design software?

14.1% use Blender

48% use Fusion 360

37% use Tinkercad

23.4% use OpenSCAD

Is 3D design easy?

42% said the best thing about their software was the ease of use

25% said the worst thing was the difficulty of use

6% don't design models
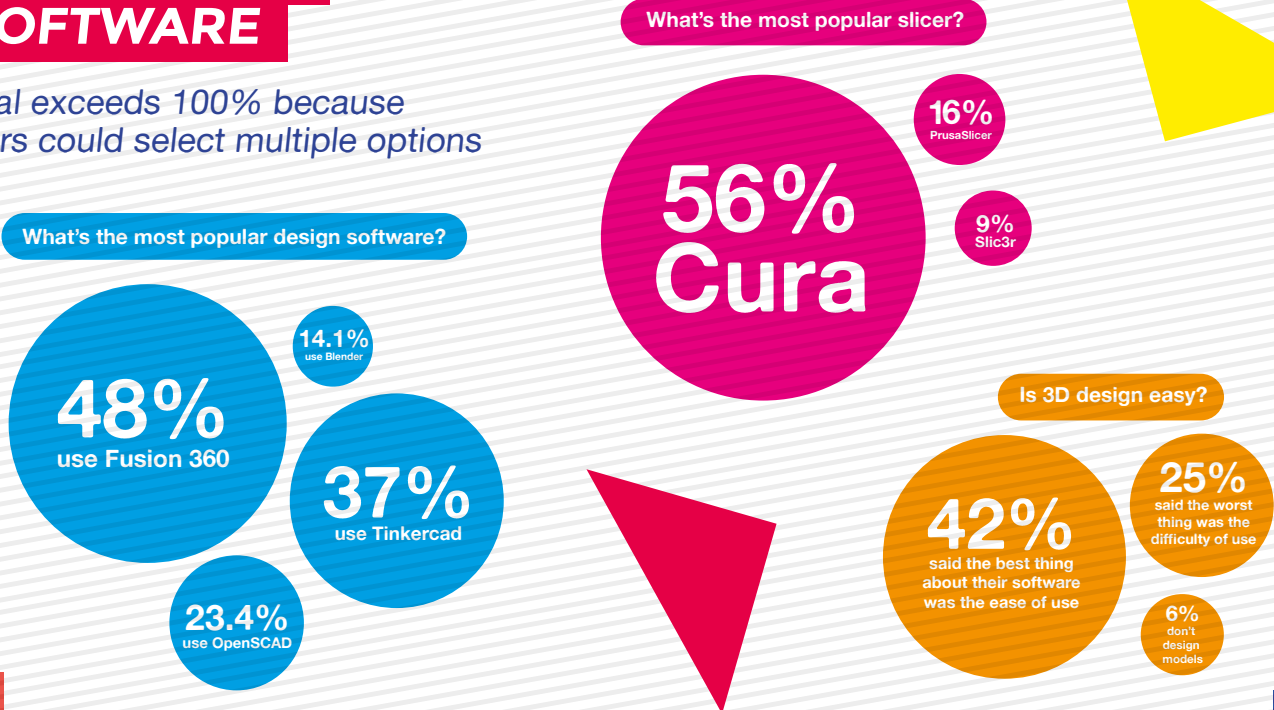
WHAT TYPES OF FILAMENT DO YOU USE?

Total exceeds 100% because users could select multiple options
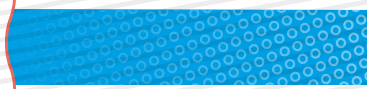
97%
PLA
Polylactic Acid (PLA) is the first filament most people use. It's a bioplastic often made from sugar cane or corn starch. PLA is probably the easiest filament to print with and works well for general use, but it is brittle and not as strong as some other plastics.

43%
PETG
Polyethylene terephthalate Glycol-modified (PETG) doesn't print quite as easily as PLA and can be more expensive, but it is stronger, more durable, and more heat resistant.

31%
ABS
Acrylonitrile Butadiene Styrene (ABS) was, for a long time, the go-to plastic for strong prints, and while it is a little stronger and more heat-resistant than PETG, it is also harder to print. You need to be able to control the temperature well, and some ABS filaments give off unpleasant odours or dangerous gasses when printing.

29%
FLEXIBLE
There is a wide range of different filaments that fall under the umbrella 'flexible', and it's important to pick the right one. Keep an eye out for the elastic range and the Shore hardness of any filament you're considering as these will tell you how much they can stretch and how easily they deform.

15%
TEXTURED
Textured filaments are typically PLA with additional materials to give the appearance of wood or metal. Results vary, but with the right filament in the right model, you can achieve a really striking visual effect. Bear in mind that some textured filaments can wear out your 3D printer's nozzle (toughened nozzles are available).

10%
NYLON
This tough, semi-flexible plastic is great for making durable parts, but has high heat demands for the 3D printer, and is very prone to absorbing moisture from the air so doesn't store well in humid environments.

6%
REINFORCED
Along with a polymer, reinforced include a fibre (such as carbon) that can give the final print far more strength than a polymer alone. They're particularly useful when lightweight is important (so you can't use high infills).

5%
DISSOLVABLE
In printers which can use more than one material, dissolvable filament can be used for support structures so that, once your print is finished, you just need to give it a bath and you'll have the final product. No more snipping off supports and dealing with the rough surface left behind.

WHAT ARE THE MOST POPULAR 3D PRINTER UPGRADES?

How to improve your printer

17%
UPGRADING CONTROL BOARD
most popular on Creality CR-10 (40%) and Creality Ender-3 (30%)

27%
EXTRUDER
Most common on Creality CR-10 (53%) and Creality Ender-3 Pro (39%)

17%
ADDING AUTO-LEVELLING
Most popular on Anet A8 (37%) and Creality Ender-3 Pro (27%)

33%
PRINT BED
Most common on Creality Ender-3 (54%) and Monoprice Select Mini V2 (50%)

VERDICT

Which is best?
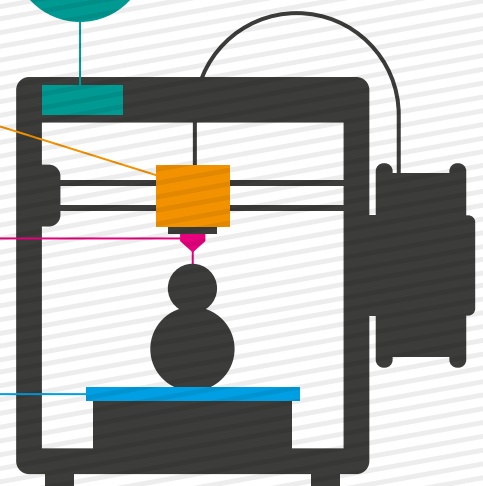
In the past six pages, we've looked at loads of data about some printers, so let's filter it all through and see what it tells us about what printers our community like the most.

Since we are looking at machines over a huge range of prices, it's only fair to split them up into price brackets.

BUDGET (under £200)

Creality Ender-3 has brought a new level of performance to the budget 3D printer market. It can print well with the out-of-the-box hardware, but there's a great set of upgrades that can add performance and reliability. There's no hiding the fact that this printer requires a little love and care to keep it printing well, but the flip side of this is that you'll learn more about how 3D printing works. And for the price, it makes 3D printing affordable to lots of people who wouldn't get a more expensive printer.

If you're after a slight upgrade, the Pro version doesn't bring a lot on paper – a slightly stronger frame and a few tweaks, but our survey shows that it's enough to bump the median print failure rate from one in ten to one in twenty, and the average satisfaction from 4.42 to 4.72.

MIDRANGE (£200 – £500)

Perhaps the dark horse in our survey, the Anycubic i3 Mega isn't as well-known as either the Ender-3 or the Prusa i3 MK3S, but its owners believe that it works well and hold it in high regard. This is a printer that doesn't dance about with the latest, flashiest features, but it works reliably time after time. It's based on the tried and trusted i3 design, which is more solid than the extruded aluminium design that many cheaper printers use.

Manufacturer	Model	Price (*)	Satisfaction	Prints per failure	Breakdowns per printer	Upgrades per printer
Anet	A6	£140	4.4	39	0.20	1.4
Anet	A8	£100	4.0	21	0.37	1.7
Anycubic	i3 Mega	£260	4.75	73	0.22	0.5
Creality	CR-10	£300	4.13	32	0.43	1.7
Creality	Ender-3	£160	4.42	29	0.21	1.6
Creality	Ender-3 Pro	£200	4.72	29	0.33	1.7
Flashforge	Finder	£260	4.71	24	0.14	0.4
Monoprice	Select Mini V2	£115	4.0	18	0.6	1.4
Prusa	MK2S	n/a	4.56	48	0.67	0.5
Prusa	MK3	n/a	4.78	61	0.11	0.1
Prusa	MK3S	£699/£899	4.86	52	0.21	0.7
Ultimaker	2+	£2150	4.67	36	0.33	0.5

(*Prices are based on last-available sales data and are subject to change).

HIGH END (£500+)

If you buy the pre-built version (which comes with a hefty £150 premium), then getting the Prusa MK3S printing really takes little more than unpacking the printer and plugging it in, though building from a kit isn't too onerous either. The auto levelling bed, accurate temperature control, excellent software support, and high-quality components make this a great choice for anyone who wants to 3D-print, and not tinker with a 3D printer.

RESIN

We targeted our survey at people with hot plastic 3D printers, rather than those with resin printers (this was a little unclear, and we apologise for any confusion caused by the lack of explicit wording around this). However, we are interested to find out what people thought of resin printers.

10% of survey responders who own a hot plastic printer also own a resin printer. Even more strikingly, 28% of responders are considering buying a resin printer. That potentially means that the next few years could see a dramatic rise in this technology.

HackSpace
Best 3D printer 2019
Prusa MK3S

OVERALL

Picking an overall winner is a tough choice – all of these three printers are great choices for some situations. However, the community have spoken, and owners of the Prusa MK3S were more satisfied than any other 3D printer users, and we're not going to argue with that. If you want a 3D printer because you want to 3D-print things, then the Prusa MK3S is the printer for you. If you want a 3D printer because you want to tinker with it and understand the tech behind it, then one of the other printers will – given a bit of tweaking – get similar results, and give you more options for upgrades and experiments. Take a look on the next page for our full review. →

REVIEW
PRUSA MK3S

The community's favourite printer in our test lab

Below ⬊
You can get the plastic parts in black if the classic Prusa orange is too much for you
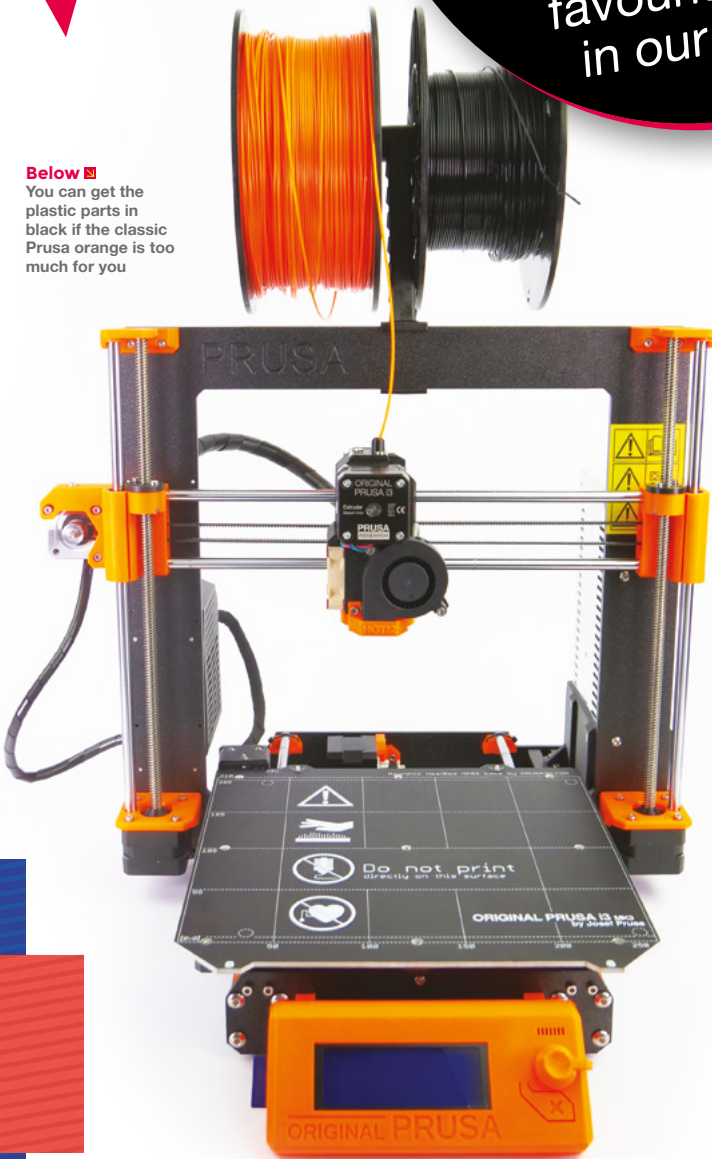
Our community recommended the Prusa MK3S, so we got one into the HackSpace magazine test lab to make sure that it was really as good as the community claimed.

We spent more time eating the pack of gummy sweets that came with the printer than we did setting it up. There are some pre-sliced models on the included SD card, so once you're set up, just plug it in, add some filament (1 kg is included) and you're good to go. If there are any problems, you at least know that the problem isn't in the G-code.

The printer's frame is made from aluminium with the Y axis running on extruded aluminium and the hight of the frame is made of dural (an aluminium – copper alloy). Twin lead-screws control the Z axis, while the X and Y are driven by belts. The result is a printer capable of high- quality printing with a layer height between 0.05 mm and 0.3 mm (at least with the supplied 0.4 mm nozzle – other nozzles give different layer heights).

We're particularly fond of the sprung steel print bed. This allows you to easily pop models off, then reattach it for the next print. This surface does need to be kept clean of grease in order to work properly, and we found that the first layer quickly stopped sticking properly if we handled the print bed. The solution to this (and the recommended way of using this print surface) is to wipe down the surface with isopropyl alcohol (91 % or more is recommended) between uses. One alcohol wipe is included with the printer, and after this, you'll need to source your own. We used some high-strength alcohol that we had left over from a previous project, but it's available online from a variety of sources. It sometimes comes in spray bottles, but we'd discourage this, given that one particular risk with IPA is getting it in your eyes.

After printing a lot of models with a lot of different print settings, the only problem we've had is the first layer not sticking if we haven't wiped down the print bed first. While you don't have to wipe it down every time – especially if you're careful not to get your fingers on the print bed too much – our experience is that you do need to do it regularly.

LEVEL-HEADED

Perhaps the one standout feature that few other printers have out-of-the-box is mesh auto levelling. The name is a bit of a misnomer because it doesn't actually level the bed. Instead, it takes a set of readings to find out how level the bed is and adjusts the print to take the slight unlevelness of the bed into account when printing. The result, though, is the same – you don't need to level the bed. Before each print, the print head checks the height at either 9 or 49 points around the bed, then starts printing. We found this worked flawlessly and we've been printing for several weeks, and have moved the printer several times, yet have never adjusted the bed levelness in any way.

Although the printer works out-of-the-box, the MK3S is built with an eye to upgradeability, and there's a range of upgrades from both Prusa and third parties. Perhaps the biggest additional feature is the Multi Material Upgrade V2 (MMU2) from Prusa that lets you print up to five different filaments in a single model. These can be different colours, or even different materials (with some limitations).

On the software side of things, the MK3S will accept G-code created by most slicers, but Prusa has created its own unimaginatively named 'PrusaSlicer'

which, as well as being a well-featured slicer, contains well-tested settings for different print speeds, and a wide range of filament types (not only those made by Prusa).

This isn't a cheap printer, and if you're looking for a 'killer feature' of the MK3S that makes it the one printer you must have, then you won't find it. That's not the sort of printer this is. In many ways, the features list isn't that different from many other printers (other than auto levelling, which is an optional upgrade on many others). You can get printers that print the same things that the MK3S prints for less money. However, the MK3S is a very polished machine, and has a habit of just working. The hardware's well made and tested, and so is the software. There's no guessing, or tuning, or tweaking settings (well, you can if you want, but you can get great prints without doing this). It's this polish that makes the printer worth the money in our opinion. You can get under the hood and tinker with this or change that, but you don't have to. And if you do, you'll find the extensive documentation that Prusa provides both in print and online useful for keeping everything running smoothly.

Above
The traditional first print on a newly set up MK3S

Below
The control board has headers for a Raspberry Pi Zero if you want to add networking to your printer

VERDICT

A printer that 'just works' for beginners, yet is still powerful enough for advanced use.

10/10

OTHER MODELS TO LOOK OUT FOR

Newer and more specialist printers

There are literally hundreds of 3D printers on the market. Some are great, and we've only been able to look at the most popular ones around at the moment. There are some other interesting printers that haven't reached the critical mass to get enough reviews to make it through to our short list. This might be because they're a bit specialist, or it might be because they're too new. Let's take a look at some of the models to look out for. ▫

PRUSA MINI

The Prusa MINI isn't just a scaled-down version of the MK3S, it's a whole new printer based on a new design. While the name may say 'mini', the print volume is only slightly smaller at 18×18×18 cm. If this can bring Prusa reliability to this form factor and price point ($349), then it will be a great option. However, that's a big if, and we'll bring you an update when we get one in the HackSpace magazine test lab.

BCN3D SIGMA R19

Most 3D printers have a single nozzle that spits out hot plastic to build up your model. However, there's a growing number of printers available that have two nozzles, and where these can move independently, they're known as independent dual extruder (IDEX) printers. These have a few advantages – the two heads can print separate objects at the same time, meaning you can print multiple objects twice as fast. Alternatively, the two nozzles can have different materials and print the same object. The results can be both useful and visually impressive. The Sigma R19 from BCN3D is a reasonably priced IDEX printer that's well worth investigating if you're looking to print in more than one material.

CREALITY ENDER-5

The Ender-5 improves on the Ender-3 in a number of ways – it's got a more solid four-legged frame (with more space on the z axis), a better hot end, and an improved print bed. The geometry is also slightly different, with the print bed moving up and down. We got some respondents to the survey with this who rated it highly, and it looks like a more solid printer than the Ender-3. Of course, all these changes come with a higher price tag, and this is retailing for around £250/£340 for the regular/pro versions.

CREALITY CR-10 V2

We looked at the CR-10, and it's a popular printer, but didn't do particularly well. This is, however, a bit unfair. It's an old printer that was important in its day, but has since been superseded by newer and better models. However, it's back with a re-worked version 2. This printer brings large format printing to the masses. The 30×30×40 cm build volume gives space for some much more ambitious prints (most hobbyist printers are around 20×20×20 cm). We hope to get one in the HackSpace magazine test lab shortly, and will bring you results when we have them.

MARKFORGED METAL X

We're looking at hot plastic printers in this feature, and this printer stretched the definition of this slightly. The Metal X prints in metal powder that's bound together with plastic to make a filament. They're then washed and sintered, which removes the plastic and binds the metal together. The result is completely metal (and slightly smaller than the originally printed object).

SUBSCRIBE TODAY

Subscribe for 12 months | Rolling monthly subscription

£55 (UK) **£90** (USA)

£80 (EU) **£95** (Rest of World)

Free Circuit Playground Express with 12-month upfront subscription only (no Circuit Playground Express with rolling monthly subscription)

▸ **Low monthly cost** (from £5)

▸ **Cancel at any time**

▸ **Free delivery to your door**

▸ **Available worldwide**

WORTH
£25
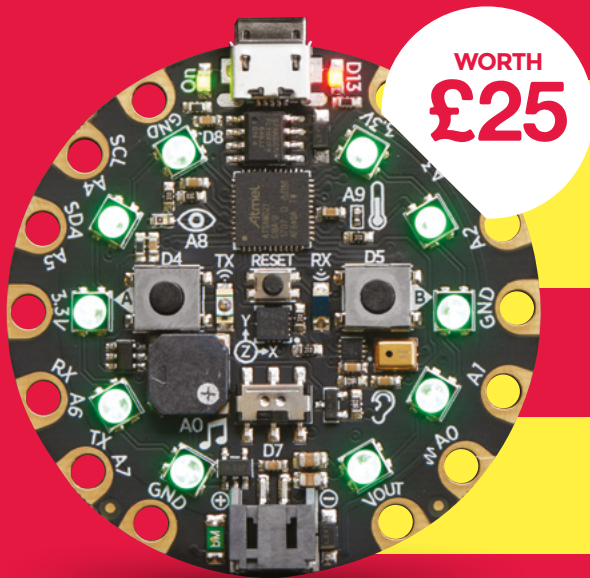
FREE!

Adafruit Circuit Playground Express

With your 12-month print subscription

This is a limited offer. Offer subject to change or withdrawal at any time.

SUBSCRIBE
on app stores

From **£2.29**

Available on the **App Store**

GET IT ON **Google Play**

🔺 **Buy now:** hsmag.cc/subscribe

How I Made

A GLASS KILN

Learning to control the heating and cooling of glass

By **Jon Rusho**

G lass is an interesting material to work with. Heat it up; it softens and melts. Cool it down, and it becomes a solid. If you do either too quickly, though, bad things can happen.

Heating or cooling too rapidly can cause the glass to crack. Sometimes so dramatically that glass can 'explode' within a kiln and stick to the sides – or worse, to the heating elements and damage the kiln. Cooling too quickly can also result in insufficient annealing, so the glass has internal stress and can be prone to cracking later. The answer is to control how the glass is heated and cooled in a kiln. Additionally, all of the desired changes in glass happen within a 200°F/111°C temperature window, between about 1350°F/732°C and 1550°F/843°C, making temperature control and accuracy a critical part of the process.

Kilns are fairly simple. Usually, the only moving parts might be a lid or door. Beyond that, you need a heat source (electric heating elements or gas) and insulation (firebrick). Broadly speaking, there are two types of kilns: those with programmable controllers, and those with manual controls. Kilns with programmable controllers tend to be fairly expensive, often twice (or more) the cost of kilns with manual controllers. Kilns with manual controllers are fairly simple: a thermocouple tied to a meter showing the current temperature, and a control knob that controls how often the kiln cycles on and off. Kilns with manual controls often show up on used/surplus websites at low prices.

Let's look at a typical firing schedule, and see why a programmable controller makes life better.

Above ◆
A manual kiln ready for hacking

WARNING!

MAINS ELECTRICITY

This project uses mains power. Be careful if you plan to recreate it.

WARNING!

Hot glass looks the same as cold glass. Don't touch glass in a hot kiln. If the kiln is still hot, the glass will be, too.

Here's a typical firing schedule to fuse glass with other glass:

- 300°F/hr (167°C/hr) up to 500°F (260°C). Hold for 15 minutes.
- 250°F/hr (139°C/hr) up to 1550°F (843°C). Hold for eight minutes.
- Without opening the kiln, let the temperature drop as fast as possible to 1060°F (571°C). Hold for 20 minutes.
- Dropping at 50°F/hr (28°C/hr) down to 940°F (504°C)
- Dropping at 100°F/hr (56°C/hr) down to 740°F (393°C).
- Then turn off the kiln, and let it return to room temperature on its own.

What's happening? First, we heat up the kiln and let everything inside reach a stable temperature at 500°F/260°C. The next step is where the firing profile really changes. Between 1350°F/732°C and 1550°F/843°C, exciting things happen: the glass softens

and then melts. Depending on how much glass, how thick it is, and the size of the kiln, the hold times might need to be adjusted.

After hitting 1550°F (843°C), we don't want to stay there too long or cool off too slowly, because glass starts to form a rough surface when overcooked (usually called devitrification). Once it's melted, let the temperature drop fairly quickly to 1060°F/571°C. Now we start the annealing part of the schedule. This is where any internal stress within the glass is released. Then, a slow cooling through 940°F/504°C and then 740°F/393°C, with a slow cool to room temperature. Again, not too fast because we don't want it to crack.

That's a lot of steps. Imagine sitting in front of the kiln watching the clock and the temperature gauge to make sure you're not heating too fast or cooling too quickly, making sure you stay at a certain temperature for a given amount of time and

not falling asleep at the switch. The firing schedule above takes more than eight hours, and you can't really see what's going on. Time for some physical computing. My background is in electronics, radio, and computers. I first started with physical computing many years ago. By day I have a not-so-traditional network engineering job, but the rest of my time is as chief hardware hacker for Glass With A Past. Recycling glass and working with old glass is what we do.

I began by finding a used kiln that I could play with. It had a thermocouple with an analogue meter showing the temperature, and a manual control that just cycled the heating elements on and off periodically. The heating control, while next to the meter, didn't have a way to use the temperature data, so the only way to maintain a firing →

Broadly speaking, there are two types of kilns: those with programmable controllers and those with manual controls

Right ◈
An off-the-shelf digital kiln controller

Glass compatibility

When fusing different glasses, often the most common metric of compatibility is coefficient of expansion (CoE). In a nutshell, if one type of glass expands more with a given temperature change than another, fusing them together is a recipe for cracked glass later. Test, test, and test again.

More info: **hsmag.cc/ApeNkE**
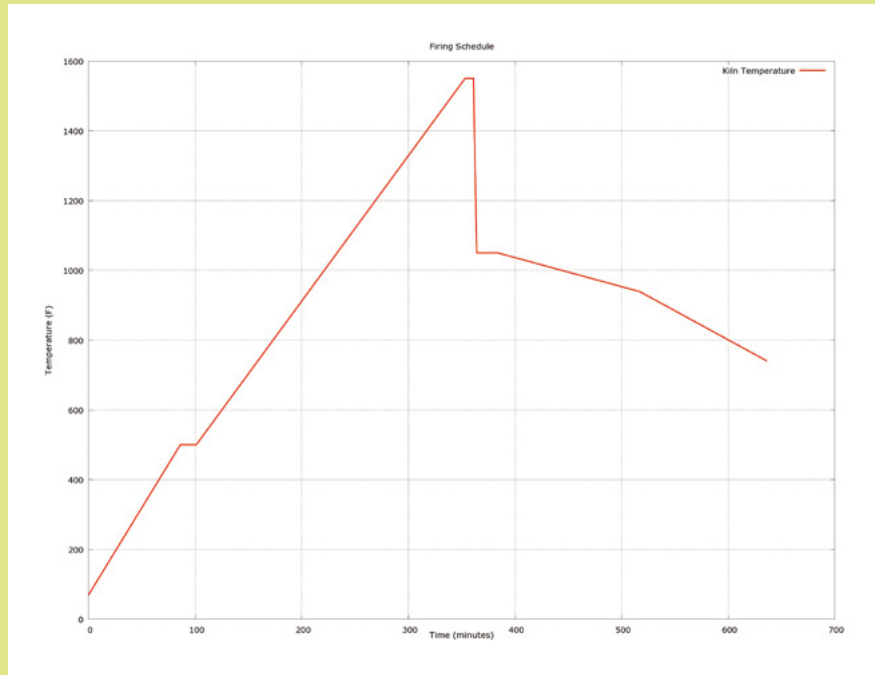
Firing Schedule

Much like any heating or cooling system, you have to watch out for hysteresis – cycling on and off too quickly

elements in the lid. The thermocouple sensing the temperature is near the elements, so it tends to read a bit higher than the centre of the kiln.

SECOND STEP
Build and experiment
The kiln control ranged from OFF to HI, with the settings in between changing the on/off cycle time. The HI setting kept the elements on without cycling. Knowing that, I opted for a simple controller that would just turn mains power off and on to the kiln, rather than ripping apart the existing system.
Here's what I would need:
- Raspberry Pi (almost any model)
- Thermocouple interface (I used an Adafruit MAX31855 breakout)
- Temperature display (Adafruit 4-digit LED)
- Relay or solid-state switch to turn the kiln on/off
- A power cord to feed the system
- An outlet to plug the kiln's power cord into
- A network connection so I could monitor and control it remotely

I found a solid-state switch that could handle the kiln load (roughly 10 A at 120 VAC, peak). I thought I might need an additional transistor switch circuit to drive it, but I found one that could switch based on a 3–32 VDC input, just above the GPIO pin output from a Raspberry Pi. Direct control from a GPIO pin to turn the power on and off would probably work.

Using the Adafruit code for the four-digit display on the I²C interface, and the thermocouple amplifier on the SPI interface, made it fairly easy to read temperatures and display them. The standard Python GPIO simplified turning the kiln heating elements on and off. I also learned that the thermocouple and thermocouple amplifier need to be connected the right way,

Shelf paper, kiln wash, and fibre paper

Kiln wash (or shelf primer) is a dry powder suspended in water and painted on firebrick or other kiln shelves. Probably the lowest cost, it typically requires multiple coats and later removal.

Shelf paper or kiln paper, while a bit more expensive than kiln wash, can be cut to fit your kiln or kiln shelf.

Fibre paper is thicker than shelf/kiln paper, and doesn't provide as smooth a surface to fire on. I used fibre paper to help control heat leaks in the kiln.

schedule would be to sit with the kiln and a clock and make sure it wasn't heating too fast, too slow, or just staying at a stable temperature.

FIRST STEP
Don't reinvent the wheel
At Glass With A Past, we have kilns with digital controllers. I parked myself in front of one during a firing, and took notes. Two key things I found: the on/off cycle time varied and, after the elements were turned off, the system waited for the heat to spread through the kiln.

Much like any heating or cooling system, you have to watch out for hysteresis – cycling on and off too quickly – because temperature throughout the kiln isn't uniform. The heating elements in this kiln are on the sides; many glass kilns have

110VAC input   Power to Kiln
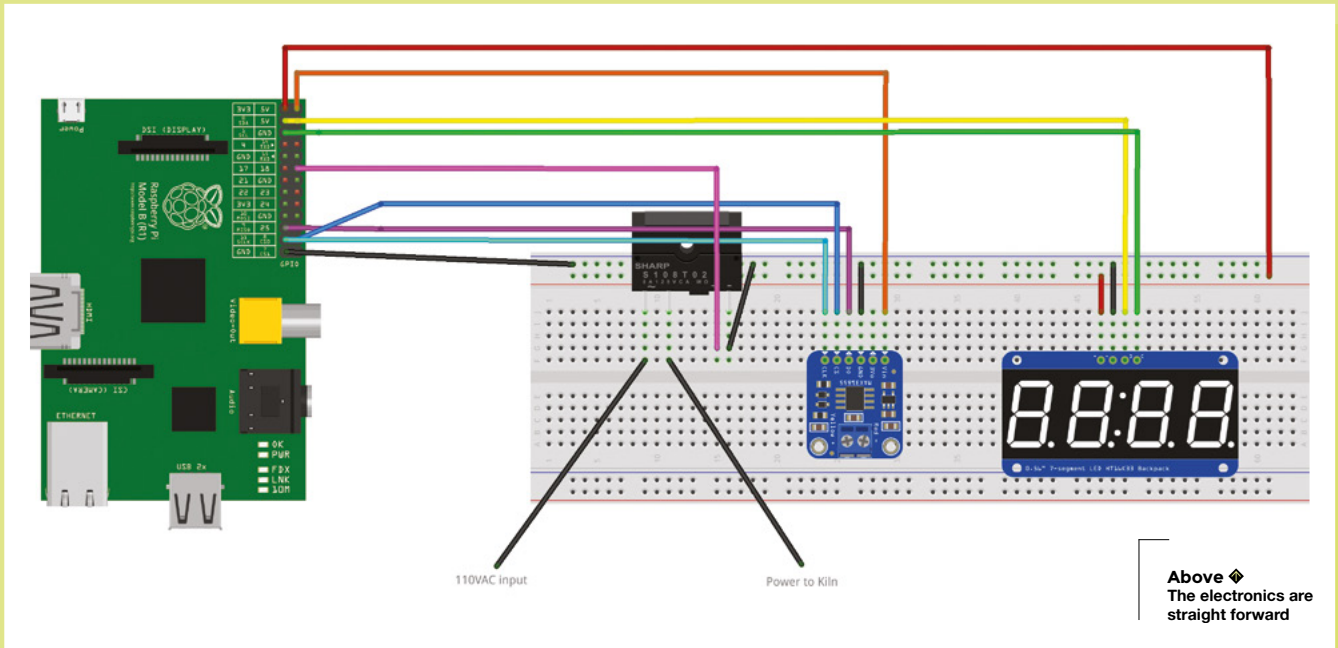
otherwise as you heat, the results indicate the temperature going down rather than up.

The experiment-side was an eye-opener. I started out with a simple on/off cycle with set times. At low temperatures, I could match the rate of heating to meet the firing schedule. I learned some practical thermodynamics the hard way. As the kiln heated, its rate of heat loss increased. My fixed timing wasn't enough to get to the temperatures needed to soften/fuse glass. The solution had two parts – adjust the firing algorithm, and better insulate the kiln.

The kiln was a fairly simple setup: a firebrick base, the central part with heating elements and control, and then a firebrick top. I had two problems. There were gaps where the base and top connected to the central sections, so I was losing heat. Adding some fibre paper where they connected helped plug the leaks. The heating algorithm was a different matter and would require some testing.

Going back to my notes observing other programmable controllers, I had noted that the cycle times varied. Now, I had to figure out how to do the same. At this point, I needed a better way to check the desired firing schedule against reality. I could already log the two temperatures. I opted to use gnuplot to plot the two, to see where things weren't working.

Following my observations, I added an adaptive step to how the kiln cycled. It would always cycle on for five seconds at a time, but rather than having a fixed time off, it could adjust. If the kiln temperature and heating rate were above the desired temperature and heating rate, the time off would increase. If it wasn't heating fast enough, or was below the projected temperature, the off-time would decrease.

Occasionally, if I stopped an existing firing (**CTRL+C** or `kill`), my code left the GPIO pin high and the solid-state switch on. I added a block of code that runs on an interrupt signal; it makes sure to set the GPIO pin to low when exiting.

FINALLY – THE GLASS TESTS

Now that the temperature profile matched the planned firing schedule, it was time for glass tests. I knew how glass behaved in our other kilns with programmable controllers, and I had read how glass was projected to work in various glass fusing books. The big 'gotcha' comes down to how accurate my thermocouple is compared to our other kiln controllers. 'Results may vary' is a nice way to put it. A degree here or there might not be a big deal, but anything much bigger could be a problem. Since the end result I was looking for was fused glass, the numerical temperature was somewhat secondary. While the numbers didn't need to match, the results did. I'd just adjust my small kiln's firing schedule to match the results of the larger kilns. →
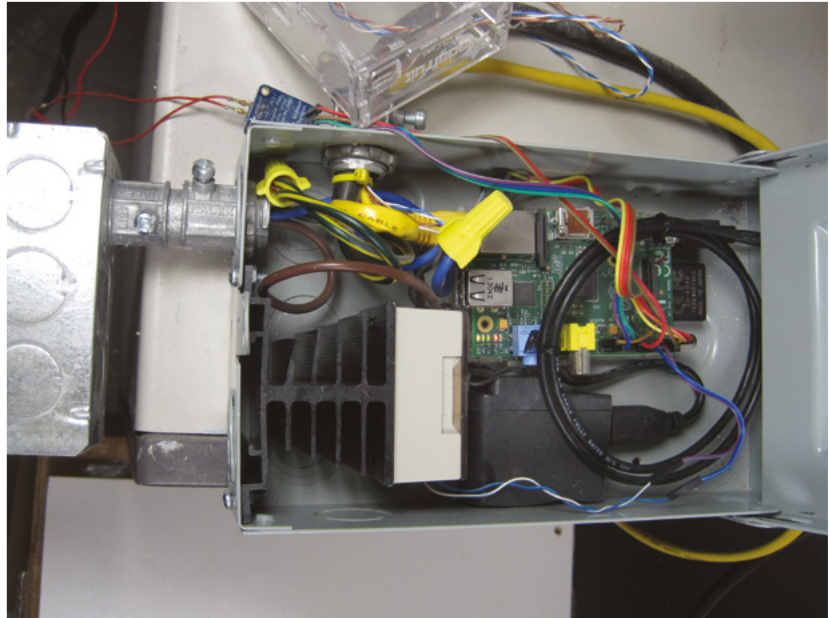
Right ↗
The innards of my digital kiln

Below ⬇
Only work with mains electricity if you're confident you can do it safely

When I started the glass tests, I noticed something odd. The kiln had a red light that would illuminate when the kiln was firing. I noticed that it would flicker on when I turned the original manual control from OFF to HI, even if the solid-state switch was off. It turns out the solid-state switch I was using leaked some current. Not enough for the kiln to fire, but enough for the light to come on. I plugged in an outlet tester (it checks if the outlet is wired properly) to check for voltage. Sure enough, just enough leakage to turn on the lights, but not the kiln itself. A mechanical relay wouldn't have this problem, but the operating current for a relay would be higher and wouldn't be directly controllable from a GPIO pin; a transistor switch would probably be needed to drive the mechanical relay. Keeping the manual control in place created an additional safety feature – I could just turn it to OFF, and the kiln couldn't come on even if my controller told it to.

Before testing with glass, I needed a way to make sure the glass wouldn't stick to the firebrick or kiln shelf. The two common ways of doing this are to use kiln wash or shelf paper. Kiln wash has to be painted on the shelf and allowed to dry. Multiple coats are usually needed and it has to be scraped off later. Shelf paper does the same thing, preventing the glass from sticking, but it tends to cost more. On its first firing, though, an organic binder (sugar/starch) burns out and

tends to smell bad. Shelf paper can last through several firings, so, other than the smell on the first firing, I think it's a pretty good investment.

The first test was just to stack two pieces of glass and fuse them. Using the kiln as designed, you can only put stuff in, start it, and then wait. You don't get to watch things happen. Start it. Watch the temperature display and log files from the code. Check the changing temperature graph. Be patient. [I'm not so patient. I used some polished quartz plates and firebrick so I could see inside the kiln while it was firing. It skewed the firing schedules, but I could see what was going on.]

After several test firings, and a few adjustments (including code and insulating the kiln), the end results match fairly well, compared to our larger kilns with their programmable controllers.

Melting or slumping a glass bottle is fairly easy. Labels, though, need to be removed or they'll burn during firing. Fusing window glass can be interesting, too. Low-E window glass tends to yellow, for example.

After a lot of trial and error, I finally have a programmable kiln controller for a small kiln, but it could probably be extended for a larger kiln, too. My next step is a web interface to select or create a firing schedule, and to monitor/control the kiln while it's firing. ▫

IN THE WORKSHOP: Making a cheese knife

By **Ben Everard**

The easy way to start tool-making

Knife-making has always interested me as a hobby. I think it's because I like the idea of tool-making and knives are probably the tool I use most, and they're among the oldest tools around. However, there are a few bits that have put me off going all the way into knife-making – there's quite a bit of kit you need (including a way of heat-treating blades), and there's also a fundamental limit on the number of knives I need or want. It seems like a big investment in time, equipment, and material to make the four or five knives I need. I'd been looking for an easier way of getting into the hobby, when one night I was struggling to cut up a truckle of Cheddar with a tiny cheese knife. Why don't they make decent-sized cheese knives? No matter, I'll do it myself.

The big advantage of cheese knives is that they don't need to hold a sharp edge. This means two things. Firstly, you don't need to heat-treat the steel, and secondly, you can use cheaper blends of stainless steel that are easy to work with and don't go rusty.

For my first cheese knife, I created a cleaver design with a fork for picking up the cheese, and I've been

Stock removal tools

There's no right set of tools that you need for stock-removal knife-making. I used an angle grinder and a rotary multitool, but you could use a metal saw and a file – anything that's capable of removing stainless steel should work. For finishing the blade and handle properly, a good set of sandpaper will give you a better finish than we've achieved, but these are just prototypes. We're very slowly designing our perfect set of cheese tools, and we'll spend the time necessary to make a good-looking set once we've finalised the basic design.

Left
The handle clamped in place while the epoxy set

using this for about a year with great success. This month, I decided to create a second in a chopper design. This doesn't have a handle in the traditional sense, but has one built into a squarish blade. Hopefully, this should make it easier to accurately guide the knife through hard cheeses, and should also be a stronger design using unhardened steel.

There are two fundamental methods for making a knife – forging, where you start with a lump of steel (often as a round bar), and heat-hammer-repeat until you've got a knife-shaped piece of metal. The second method is to start with a bit of metal larger than the knife you want to make, and cut it up until the bit remaining is knife-shaped. This 'stock-removal' method is much less efficient than forging, but it's also far easier, and the only option if you don't have a forge and anvil.

CUT IT OUT

I started with a 30 cm square of 2 mm-thick food-grade stainless steel. This was far bigger than I needed, but it was the only size available, and I'll be able to use the offcuts in other projects. The previous cheese knife was made from 0.9 mm-thick stainless steel, and it was a little wobbly, so I opted for a thicker size this time. I sketched out the shape of the blade with a pencil, then cut this out with an angle grinder. Angle grinders can't cut curves (as it puts stresses on the side of the disc), so you have to cut out a rough shape around the desired shape, then use a grinding disc to work the straight edges down to curves.

This was a bit of an experiment, since I had no idea if this shape of blade would be useful, so for the sake of speed, I didn't pin the handle in place – it's only held in place by a generous dollop of epoxy resin. Perhaps a little too generous as it overflowed onto the blade a little. I left the 15-minute epoxy to harden for two hours, and it was still gloopy!

The British winter had evidently come a little sooner than I thought – it was below the curing temperature, so the proto-knife had to come inside to warm up before the glue would set.

Speaking of the handle, it's just two pieces of white pine offcuts. You can get some really good-looking pieces of wood for knife handles, but since this knife is just a test, I decided to go with the cheapest option and then spend money once we were more confident of the design. That said, while it doesn't have the luxury feel of some wood, white pine with a bit of oil (we used butcher's block oil as it's food-safe, and we had some left over from a previous project) does look nice.

Why don't they make decent-sized cheese knives?

The final process was just sanding down the wood to a comfortable handle shape – a sanding attachment on a rotary tool is a good way to very quickly shape wood. I left it quite square with a little rounding to take the edge off the corners. The blade got a similar treatment, but with a grinding rotary tool attachment to smooth the cut edges and give it a blunt blade.

That's all there is to it. Bar the unexpected wait time on the epoxy; the knife only took a couple of hours to make. But, of course, the ultimate question for any knife is: how does it cut? I'm fairly pleased with how this came out. It feels really solid in the hand, and I like the cutting motion of pressing down. However, the sheet steel used might be a little too thick. For my next iteration of attempting to design the perfect cheese knife, I'll try some slightly thinner sheet steel. →

Repairing with gold

S **tuff breaks – that's an inalienable part of using things.** Knowing this basic fact doesn't make it any less painful when something with sentimental value breaks. However, the only alternative would be not using it, and that kind of removes the point of having it in the first place. Sometimes, we can repair things after they break, but they often end up looking a bit worse for it. Your pristine ceramics don't look quite the same with glue lines (however slight) running through them.

However, we don't have to accept this. Rather than trying to hide the fact that something has been broken, we can celebrate it. Rather than attempting to make the break as invisible as possible, we can make it stand out as something beautiful. This way the object isn't returned to its former state, it's elevated above it. The breaks become a testament to the fact that the object has been used and loved rather than shut away. This is the philosophy behind the Japanese art of Kintsugi. Traditionally, this technique uses a tree sap lacquer and powdered gold. We used the Kintsugi kit by Humade in order

to create a similar effect with epoxy resin and gold-coloured powder.

The process is fairly straightforward. Epoxy resin comes as two parts – the resin and a hardener. Mix the two together, add a small amount of gold powder, and you have a few minutes to apply it before it becomes rock-hard. You can add a generous amount so that it oozes slightly to really highlight the former break.

Once it's on the pottery (but before it's fully hard), use the supplied paintbrush to dust with more gold powder to give it the full lustre.

Of course, anyone who's attempted to repair pottery will know that there are often small bits that get lost. Perhaps they shatter into tiny shards that are never seen again, or perhaps a small piece just shoots under a sofa and isn't found. For these cases, the kit comes with a putty. Cut off a small amount and massage it until it's an even colour, then poke it into the hole. The gold powder won't stick to this, so you have to give it a coat of resin.

We were perhaps a little generous with the epoxy – a smaller coat would have created a smaller, and

Right ➡
The fixed mug, ready for a cup of tea

We can make it stand out as something beautiful

Other repairs

This kit is for repairing ceramic, but there's all manner of ways you can take this basic idea and apply it to other broken things. Here are a few ideas, but really, it's just a case of thinking about aesthetic options whenever you have to repair something.

Fabric
Gold or brightly coloured thread and material is available and can be used for repairs. This is perhaps similar to the old tradition of patching using contrasting materials.

Glass
Epoxy resin should stick to glass just as it does to ceramic, so you should be able to use this technique on glass without any problems.

Plastic
Epoxy should stick to plastic, but in some cases you can also weld plastic using different coloured fillers, or build a repair using a 3D pen.

Wood
There are many options here. A poured resin can be a fantastic way of fixing wood while retaining the character of the break. Similarly, if you have to replace any wood, using a contrasting colour can celebrate the break as well.

more subtle, gold bead. That said, we're still happy with the result. The kit came with plenty of material, and it should all store until we end up breaking the next thing (which, to be honest, probably won't be that long).

This is such a fantastic addition to our toolbox, and the sentiment of repairing something while celebrating the break is one that really resonates with us. Things tarnish and age – we tarnish and age – there's nothing wrong with that, and we should

cherish the use that things get, rather than trying to hide it.

You may be able to get the same effect using any epoxy and gold-coloured powder, but given that you're very likely to be applying this to objects that you care about, it's probably best to use a formulation that's been tried and tested, such as the Humade kit. As we're writing this, the mug has been used and sent through the dishwasher several times, and it still has its original lustre. ☐

HackSpace magazine meets...

Chan'nel Vestergaard

Turn waste products into new materials

S **cience is magic.** It's complicated, it's hard, and should only be attempted by those with the adequate training and a piece of paper from a university to prove that they have permission to access the secret knowledge. At least, that's how it can feel.

One person bucking that stereotype is Chan'nel Vestergaard. She's a self-taught programmer, maker, biologist, geneticist, and a visiting fellow at the Massachusetts Institute of Technology (MIT), despite not having passed any school qualifications.

We spoke to her from her new lab in Copenhagen, where she makes rubber from banana skins, silk from milk protein, and tries to teach people that science is for everyone, not just PhD candidates on university research programmes. →

HS When we first met, you had some samples of fabric made out of various plant derivatives. For the benefit of anyone who hasn't seen fabric made out of milk, what is it that you do?

CV The easiest thing I tell people is that I facilitate the engagement of new technologies or new materials via citizen science. What I do is I sit there and think 'I can make this from this.' I try it, and it works, and I put the protocol together, and then I teach people how to do it. That's me verifying that my protocol works, that everything I'm doing works, and from there I develop the recipe and move further and further forwards. Technically speaking, most people call me a futuristic material designer – maybe that's what I am?

If something interests me, I'll dive into it. For example, I'm really lucky to be working with a company based in Vancouver who are making the world's first fully plant-based shoe. I'm working with them to develop a PETA-verified textile dye to dye these shoes. Everything has to be 100% plant, but because of that, there are a lot of problems. I'm working on a dye at the moment taken from E. coli; when it grows in the soil, E. coli is black, but when you extract it into its raw form, it's a bright, almost neon purple. You can use it to make dyes that are almost indigo, like dark denim jeans. That's what I'm working on right now.

HS This may be a silly question, but is that a safe thing to do?

CV This is why I do what I do. We all think of bacteria and fungus or mould as these terrible horrible things; they shouldn't invade our homes, they shouldn't invade our bodies. But there are different strains. This is where it comes down to education. As soon as we see a black mould, we're told that it's a very bad thing. But it's not: some black mould is used to make cheese. Some types of bacteria are used to culture yogurt. Some

fungus is used to make antibiotics, penicillin, for example. These can be used for good rather than harm. It's the same thing with GMOs (genetically modified organisms). They're not necessarily bad, it's just the narrative we've been fed that they are bad and we shouldn't use them.

How do we change that narrative? Well, it's by doing projects like I do, teaching people that they're very safe, they're harmless, as long as you work in a safe and sterile manner.

The bacteria that I use have less implications than using traditional dyes that we use now in the textile industry. When a pair of jeans is dyed using current techniques, all these chemicals and pollutants go into rivers and

> "If we can use milk to make plastic, what else can we make with milk? **I made pens a few weeks ago out of milk**"

eventually estuaries and kill fish, and get into the soil further downstream.

With this E. coli dye, it doesn't. You use boiling hot steam pressure to kill the bacteria, and then the water's completely natural again. Nature can do so much for us. How do we educate people to the fact that nature can save us, when people don't want to educate themselves?

HS What else?

CV Rubber from bananas. The exterior of a banana skin contains the same enzyme as that found in *Ficus lyrata*, the Fancy Fiddle Leaf Fig. If you've ever taken a leaf off one of those, the white sap that comes out, that's natural rubber. If you have enough of these trees in a large concentrated space, you can make rubber from it, the same as banana skins. When you play with it, when you laser-

cut it, it cuts just like rubber. It has the same properties.

I think the best thing I've made out of milk so far is a kind of plastic. Milk contains a protein called casein, which can be used in a lot of things. If you think of silk, silk is a protein. You can do the same isolation process, and you can turn casein into fibre and weave a dress out of milk. But you can also use it to make plastic. Now if we go back to the 1920s, in the very early era of Prohibition, people were playing board games, gambling, card games and so on. What they didn't have was dice. What happened was somebody developed a recipe using milk and an acid that they'd found in their kitchen. They'd mix it together, heat it, and formed this hard plastic. They shaped it into a square, and used cochineal to paint the numbers on the sides. That was the very first form of plastic that we, as scientists, know of.

That opened my eyes. If we can use milk to make plastic, what else can we make with milk? I made pens a few weeks ago out of milk, and I've CNC'ed bricks of milk plastic to make moulds and laser-cut shapes out of said milk. This is 100% biodegradable and biocompostable – you can put this into a compost heap, and in six weeks it will completely disappear. This is hard plastic, like a very tough acrylic. The mind-blowing thing for me is that we have so many great materials right in front of us, but we're still choosing to drill for oil and extracting from the planet. Again, it comes down to education.

HS Speaking of education, am I right in thinking that you're completely self-taught?

CV As a teenager, I was quite sick. I spent a lot of time in and out of hospitals and, because of that, I didn't get an education. If you're in hospital for one month, then you're out of hospital for one month, you can't exactly go to school if your immune system's compromised, and you can't sit →

in a classroom and develop friendships with people because you're at risk of getting very sick, very quickly. But also, you take a lot of time out of the school; you miss the classes, you can't catch up. I did some school, but that was just how it was. I tried to catch up at night school but it was never enough to get the GCSE grades that you need to get somewhere in life. More than that – I've missed out on some really basic things: my mathematics is horrendous, to this day.

I spent the majority of my teenage years and early twenties working with my mother, who is a seamstress and tailor based in Liverpool. That's where I learned the majority of my textile knowledge, what materials are, the compounds of how things are made, how to weave, how to make these *Big Fat Gypsy Wedding*-looking things that people would ask for.

The thing for me that got me into science was asking about the materials. Where are they coming from? Who makes them? Are they paid a fair wage? Why does it have this smell? – I think we can all relate to that: when we buy something from China, it has this very strange factory smell. I kind of think science found me, not I found science. Science is all around us, but it's how we figure out science that makes us get into it.

HS So you went from textiles to plant bioscience?

CV Textiles to wearables. Working out how I could laser-cut Lycra to make carnival costumes. Figuring out how to make things easier and quicker to do – you can spend ages cutting out a pattern, but what if you can put the pattern onto a computer and the computer can control a laser cutter, and you can sit there doing something better, while 50 of the same T-shirt design get cut? That kind of thinking is what got me into the technology, which is when I found the Fab Lab. I never knew a Fab Lab

existed before then; I just walked in, told them my idea, and I had no idea how to use any of the kit. Vinyl plotters, laser cutters, CNC machines – I had zero idea. I spent time [searching for] how to use this thing, and finding people on the internet who would help me.

That's where I found the open-source community. I started to use Inkscape, and realised that if learned it properly, I could use all of the machinery in front of me, and doors would really open. That's how I started as a maker. I think everyone has a maker in them; they just need a push to get them into the right place.

When I first started programming, I taught myself Arduino; that was the

> The thing for me that got me into science was asking about the materials. **Where are they coming from?**

gateway. The very first project I did, I used an Adafruit Gemma V2, and I had these little tiny LEDs and so on; so I taught myself how to do electronics via that medium. How to use conductive thread, how to program the thing so it did twinkly lights. So, that was my first gateway into programming. From Arduino, I went to Raspberry Pi. From Raspberry Pi I did Python, and right now I'm still working with Python, as well as JavaScript and all that web stuff. My husband helps me when I get stuck, but I hate Python with a passion.

HS I guess it just does what you tell it to do?

CV For some reason, I tell it what I want it to do, but then it doesn't do that. Then I have to back-track, and I'm like 'I don't have time for this'. I always revert to Arduino, because it's such an easy

programming language that people can learn very quickly, but also thanks to things like Scratch and micro:bits, children can learn this language very fast and easily too, and that's something that's really overlooked – once you know one way of doing something, you also know the next.

HS You've mentioned makerspaces a couple of times. Do you work out of a makerspace?

CV It's part of the wave of what's called DIY community biology labs, or science labs, across the world. We're one of 270 (as it stands right now). You can just walk in, and we'll teach you everything from plating mushrooms and turning them into materials, down to testing your DNA and telling you what your genomic sequence is. We do all of that, and we do it for free.

Grant application season is always on our mind – who can we apply to, how much can we apply for, how many events can we do to pull in money to keep the space alive? That's how we live. Everything I do is non-profit, so I'm always in a challenging boat.

HS You do genetic sequencing as well, don't you?

CV Oh my God, me and genetics are a love-hate affair these days. In our lab, we have a lot of equipment. One of the things we don't have is a genetic sequencer. There are two places you can buy from: there's Illumina, then there's Oxford Nanopore Technologies.

Now, they're super-expensive, and we don't have all the money in the world to afford fancy machines. So, we get the kits and send them off to a third-party company to do, and they send them back to us. But you have to do all the base work. You have to PCR everything [polymerase chain reaction – a technique used to duplicate a given section of DNA], you have to get all the DNA down to, I →

mean, micro fragments to send to the company. And if you ship it in the wrong conditions, it's denatured by the time it gets there, and it's unusable; you have to do it all over again. So it's a kind of love-hate affair with genetics. Because of my illness, I'm waiting for someone to find a cure. But again, it's not that easy, and it's not easy to find the cause of something that can change someone's genomic structure.

There's a really simple DNA workshop you can do at home using dish soap, salt, vodka, and water. That's all you need to see your DNA in a shot glass. You forget that we're all made up of the same 99.9% DNA. It's a great community-building workshop to do, especially if you live in an unprivileged, minority community. You can see that we're all made of the same thing, with 0.1% that fractionates us.

Then all of these new technologies that are coming out like CRISPR – that's what I did when I was at MIT, I did this CRISPR Cas9 and Cas13, where you cut into the DNA, insert a new bit, and heal it and let it grow to see what happens. People call it the designer babies technology, because that's what it's intended for: making the genomic sequence of embryos different to eradicate diseases that these children may have.

HS Hang on, MIT?

CV Yeah, it's been super-strange. I got offered a fellowship at MIT. It's called the global community biotechnology initiative. In general, it's about creating this cohort of people from across the world who engage and create science with people. It doesn't matter about their backgrounds, their economic status, it doesn't matter if they have a PhD or papers published. MIT have been around the world and found people who have come to their attention, and I happen to be one of them, which is really strange. I know people who are so much better than me, doing so much more amazing work than me, and yet here I am standing on stage at MIT with all these other people. It was really surreal.

HS Surely every psychologically balanced person thinks like that? There has to be something missing from your mind to lack self-doubt?

CV Everyone's level of amazement or wonder, it's so personal. Everyone thinks you're doing amazing work, but then when we come back and talk about it, it's just work for me. It's just everyday life. Other people go 'Wow! You can turn tea into leather!' And I'm like, 'But you can too!' The point is to get more people to be makers and scientists. Because science is hard, but people are harder. That's kind of our quote right now. It's true. Science is really tough, and it can be a strange environment to be in. But people are much harder. If people don't want to understand science, there's no way they're going to understand climate change, or why the dinosaurs became extinct, or why the Earth is round and not flat...

HS ...or vaccines?

CV It's so interesting to me, but at the same time, it's such a weird world to be in – I'm not from academia whatsoever. Everything that I do, I'm completely self-taught. But because academia makes it seem so hard and fancy, people don't want to do it. But we need to break that barrier down. Simple DIY science is what we need in order to understand what we have. ◻
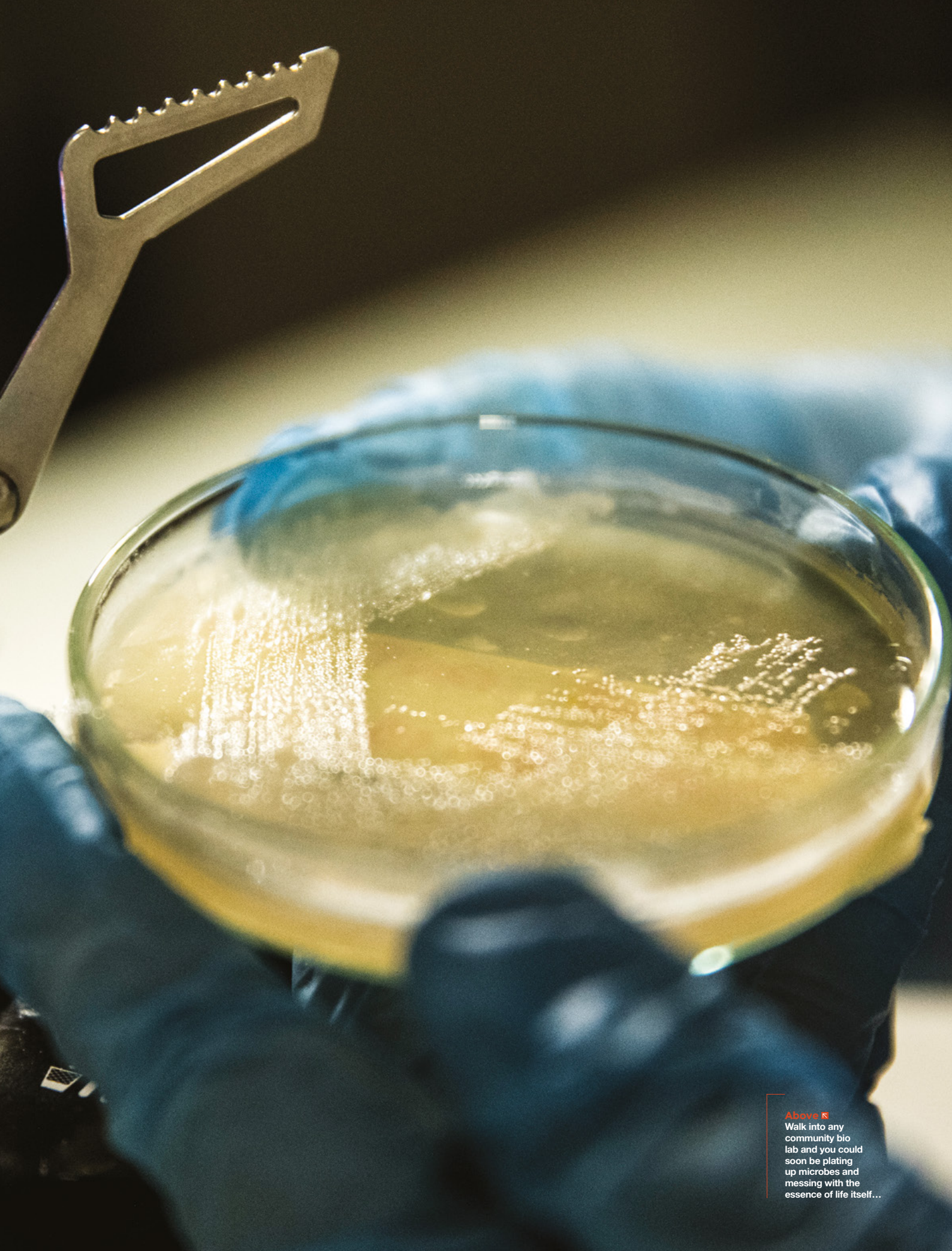
HAIR-DRYER

Give your builds a breath of fresh hair

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.

The hair-dryer is one of the quintessential electronics in a modern bathroom. French hairdresser Alexandre Godefroy is often credited as its inventor, since he filed a patent for a device for drying hair using heat back in the late 1880s. His device, however, lacked the other crucial element in a modern-day hair-dryer: airflow. It wasn't until a couple of decades later that American Gabriel Kazanjian filed a patent for a hand-held device that used heated airflow.

The first mass-produced hair-dryers in the early 20th century were overwhelmingly underpowered. They produced only about 100 watts of heat, as compared to about 2000 watts for a modern-day equivalent. Not only did they not dry hair any quicker than the sun, but they were also quite bulky. The earliest versions of the gadget were made with steel and other heavy metals, and weighed over 2 lbs.

Over the years, the mechanism of the hair-dryer hasn't had any major change. A typical hair-dryer still contains an electric motor that drives a fan, which blows air across nickel-chromium heating wires to produce a warm flow of air. However, they have gone through a significant upgrade in terms of their build and their heating wattage. On the one hand, the electric motors have become smaller and more efficient and, on the other, the introduction of light-weight materials like plastic have made the appliance easier to handle.

Another significant milestone in the evolution of the hair-dryer has been the upgradation of the

safety regulations. Hair-dryers were one of the major causes of electrocutions until the 1970s, when strong guidelines and regulations were developed and enforced. A ground fault circuit interrupter (GFCI) is used in modern hair-dryers to safeguard against electric shocks. That said, a hair-dryer is an electrical appliance, and should be used with caution around water.

> **"A TYPICAL HAIR-DRYER STILL CONTAINS AN ELECTRIC MOTOR THAT DRIVES A FAN, WHICH BLOWS AIR ACROSS NICKEL-CHROMIUM HEATING WIRES TO PRODUCE A WARM FLOW OF AIR"**

A hair-dryer works by blowing hot air over damp hair to help water evaporate faster. Additionally, it also accelerates the formation of temporary hydrogen bonds inside strands of hair, which sets and styles the hair in place. This is why most hair-dryers come with different attachments to control the spread of air and help you style your hair.

Let's take a look at some creative uses for these appliances, but if you want to have a go, remember that these are mains-powered devices, and you should only alter them if you have the skill and experience to do it safely.

VACUUM CLEANER

Like all good makers, Chialiang wanted a vacuum cleaner to keep his desk clean, but would rather create one using discarded pieces of junk, instead of buying it from a store.

He has shared the process of how he modified a hair-dryer from blowing air into sucking air. After unscrewing it, he removed all components besides the electric motor and the fan. He then reversed the motor and sawed off any part that blocked the reversed fan. Next up, he created the dust chamber using an old container that had been discarded because its lid had some holes. He glued the lid to a piece of wood and then used a cylindrical HEPA air filter, and attached it to the inside of the lid. He's detailed the process in his Instructables page, which basically involves cutting the lid to fit the filter, and then drilling five holes into the piece of wood; one to attach the filter and the other four for airflow. He then drilled a larger 32 mm hole to the side of the container and passed a hose through it. Finally, he connected the hair-dryer to the other side of the piece of wood and also wired a control switch to the contraption. Flick through his images to understand the procedure for these steps. □

Project Maker
CHIALIANG CHANG

Project Link
hsmag.cc/9e12wu

"HE'S DETAILED THE PROCESS IN HIS INSTRUCTABLES PAGE, WHICH BASICALLY INVOLVES CUTTING THE LID TO FIT THE FILTER AND THEN DRILLING FIVE HOLES INTO THE PIECE OF WOOD"

SOLDER FUME EXTRACTOR

Andrew Terranova is a robotics hobbyist and, like many makers, is tired of inhaling solder fumes. He too decided to reverse the airflow of a hair-dryer to extract the harmful fumes away from his face. Andrew's process is pretty much the same as Chialiang's; unscrew the hair-dryer, strip away all the components besides the electric motor and the fan, which he then reverses. Andrew suggests hooking up a battery to the motor to figure out which side has to be plus and minus in order to

Project Maker
ANDREW TERRANOVA

Project Link
hsmag.cc/N1G0BL

—

reverse the motor's airflow. He then inserts a piece of carbon filter pad that he's cut to fit inside the hair-dryer. He uses the original heat shield to hold the filter in place, but suggests that you can instead use a roll of cardboard, or something else to keep the filter to the back of the hair-dryer. That's really all there is to it. He then reassembles the hair-dryer and connects it to a 7.2 V 3800 mAh NiMH rechargeable battery pack using some electric tape. He had used the switch in his hair-dryer for another project, so he used a connector switch. Andrew's hair-dyer had diodes around the motor and the switch and, if yours have them too, he recommends you unsolder them. □

> **"ANDREW SUGGESTS HOOKING UP A BATTERY TO THE MOTOR TO FIGURE OUT WHICH SIDE HAS TO BE PLUS AND MINUS IN ORDER TO REVERSE THE MOTOR'S AIRFLOW"**

Left ◆
If you are interested in robotics, check out Andrew's *LetsTalkRobots* **show on YouTube:**
hsmag.cc/dW2Qdv

VINTAGE LAMP

Project Maker
Urs Grädel

Project Link
hsmag.cc/awWCQ1

Urs is a teacher who spent some time as a theatrical props maker. Touring through thrift stores close to his home, he found a vintage hood dryer that he decided to repurpose as a lamp. He began by taking it apart and removing the motor and the heating wires. He then hacked the switch, which allowed him four possible combinations. To make use of them all, he grabbed three bulbs with different wattages and then carefully wired them inside the hair-dryer, making use of the existing fixture. After a little trial and error, he got the wiring right such that each position on the switch triggered a bulb with a higher wattage in ascending order. While it wouldn't be much help unless you find the exact same hair-dryer, Urs's Instructable will give you some ideas about the process and what's involved. □

Right ⬥
Urs uses the lamp for reading while he's perched on the couch

ELECTRIC SCREWDRIVER

Project Maker
DIYFunTech

Project Link
hsmag.cc/dCjq1F

Below ⬥
The only limitation of the build is that you can't control the speed of the screwdriver, which he hopes to correct in the next version

If you need an electric screwdriver but don't want to buy one, the 15-year-old who prefers to be known by his screen name, DIYFunTech, shows you how to make one yourself. He takes a screwdriver kit and removes the bit end from the handle of the screwdriver, and then attaches it to the shaft of a 30 rpm motor using some epoxy. He then unscrews a hair-dryer, strips it bare, and attaches the epoxy'ed motor to the front of the hair-dryer. Next, he drills holes in the handle to place a 9 V battery jack, a female DC power connector, a momentary switch, and a DPDT switch and wires them according to the wiring diagram in his Instructables. He's mounted the switches to the side because he wanted the hair-dryer to fold like the original design. The DPDT switch helps reverse the motor so that you can use your electric screwdriver to both screw and unscrew. □

FORGE

HACK | MAKE | BUILD | CREATE

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

Make music with CircuitPython

Use maths and a bit of code to create your own waveforms
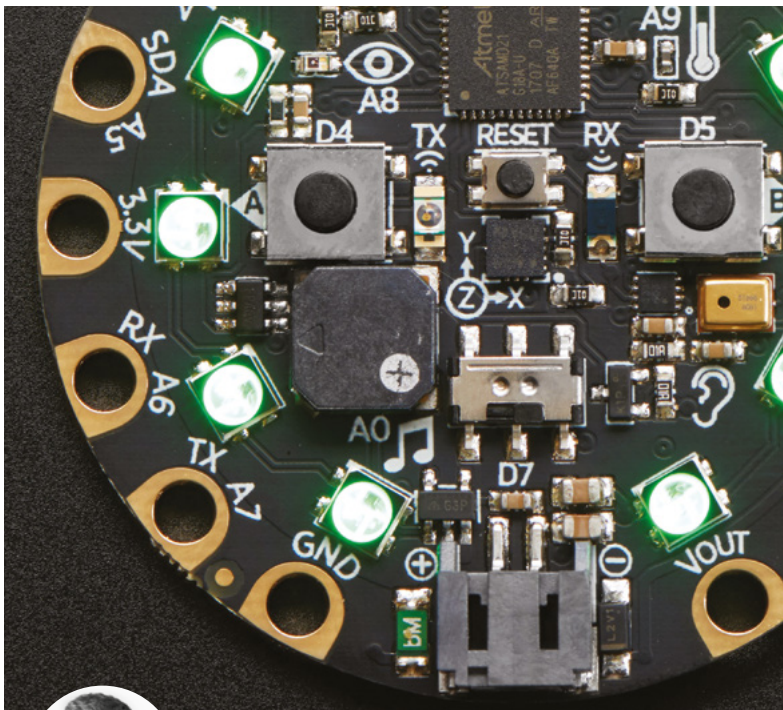
Ben Everard

🐦 @ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

Many CircuitPython boards (including all the M0+ category) have some way of outputting analogue values, which means we can use them to make music! We tested this out on a Circuit Playground Express (the sort that you get included with a 12-month HackSpace magazine subscription), but it should work on any CircuitPython board, provided it supports either AudioOut or PWMAudioOut.

Rather than playing pre-recorded music, we're going to create our own. Before we get to that though, we need to import the modules we need. These are:

```
import time
import array
import math
import board
import digitalio
from math import floor
```

```
try:
    from audiocore import RawSample
except ImportError:
    from audioio import RawSample
try:
  from audioio import AudioOut
except ImportError:
  try:
    from audiopwmio import PWMAudioOut as AudioOut
  except ImportError:
    pass
```

If we're going to create music, the first thing we need to know is the sample rate. This is the number of samples per second that we store (each sample is a number). The more samples we have, the more accurately we can reproduce sounds. CD-quality audio is 441,000 samples per second, but this can be a bit much for some boards, so we're going to use a rather more sedate 8000 samples per second.

The next thing we need to do is to create the values for the samples. Let's start with a really easy first waveform – a square wave. This is either zero or max, so if you viewed it in a waveform viewer, it would look like a series of squares placed on a flat surface, or the battlements of a castle.

We create this with:

```
SAMPLERATE = 8000

def generate_square(frequency):
    length = SAMPLERATE // frequency
    square_wave = array.array("H", [0] * length)
    for i in range(length):
        if i < length / 2:
            square_wave[i] = 0
        else:
            square_wave[i] = 20000
    return RawSample(square_wave, sample_
rate=SAMPLERATE)
```

Here, we've got a function that creates the sample we want to play. The first thing it calculates is the length of the necessary array. We'll be creating 8000 samples per second, but we don't need 8000 individual samples. We only need enough to hold one complete waveform, and we'll keep looping that one array.

```
21
22 def generate_sine(frequency):
23     length = SAMPLERATE // frequency
24     sine_wave = array.array("H", [0] * length)
25     for i in range(length):
26         sine_wave[i] = int(math.sin(math.pi * 2 * i / 18) * (2 ** 15) + 2 ** 15)
27     return RawSample(sine_wave, sample_rate=SAMPLERATE)
28
29 def generate_square(frequency):
30     length = SAMPLERATE // frequency
31     square_wave = array.array("H", [0] * length)
32     for i in range(length):
33         if i < length / 2:
34             square_wave[i] = 0
35         else:
36             square_wave[i] = 20000
37     return RawSample(square_wave, sample_rate=SAMPLERATE)
```
Adafruit CircuitPython REPL

So, we need to know the length of one complete waveform in samples, and that's what we store in the length variable.

We then create an array to store our samples in. In order to be as efficient as possible with RAM, we define the type of the array as 'H', which is unsigned short. This stores each value as a two-byte positive number.

We need to set the first half of the array to 0, and the second half to the maximum value (we've used 20,000 here because that roughly matches the audio level of the other waveforms we'll look at later).

The `RawSample` function converts our array into the format used by the CircuitPython audio playback function which we'll get to shortly.

The great thing about creating waveforms in code is that we can create whatever we want. We've just looked at what it takes to create a square wave, but what about a sine wave or a triangular wave? We can do these with the following:

```
def generate_sine(frequency):
    length = SAMPLERATE // frequency
    sine_wave = array.array("H", [0] * length)
    for i in range(length):
        sine_wave[i] = int(math.sin(math.pi * 2 * i / 18) * (2 ** 15) + 2 ** 15)
    return RawSample(sine_wave, sample_rate=SAMPLERATE)


def generate_triangle(frequency):
    length = SAMPLERATE // frequency
    triangle_wave = array.array("H", [0] * length)
    max_val = 65000
    for i in range(length):
        if i < length / 2:
            triangle_wave[i] = math.floor((i/(length/2))*max_val)
        else:
            triangle_wave[i] = max_val - math.floor(((i-(length/2))/(length/2))*max_val)
    print(triangle_wave)
    return RawSample(triangle_wave, sample_rate=SAMPLERATE)
```

We can now create sounds using these functions. To make them at all tuneful, it's probably best to stick with established note frequencies (though if you're feeling experimental, you can do whatever you like). We created all three wave forms in middle A and middle B:

```
sine_A = generate_sine(440)
sine_B = generate_sine(494)
```

```
square_A = generate_square(440)
square_B = generate_square(494)
triangle_A = generate_triangle(440)
triangle_B = generate_triangle(494)
```

The last piece of the puzzle is enabling the speaker and creating a function to play a note for one second. This is done with:

```
def one_sec(note):
    audio.play(note, loop=True)
    time.sleep(1)
    audio.stop()
speaker_enable = digitalio.DigitalInOut(board.SPEAKER_ENABLE)
speaker_enable.direction = digitalio.Direction.OUTPUT
speaker_enable.value = True
audio = AudioOut(board.SPEAKER)
```

The `speaker_enable` code is for boards such as the Circuit Playground Express, which have an on-board speaker, and won't be needed for other boards (but you'll need to hook up headphones or a speaker in some way – see your board's documentation for details).

That's all the setup for our little music maker. You can now play one second clips of music with the `one_sec` function. A little test that shows off the different wave forms is:

```
one_sec(sine_A)
one_sec(square_A)
one_sec(triangle_A)
one_sec(sine_B)
one_sec(square_B)
one_sec(triangle_B)
```

The full code for this project can be found at **hsmag.cc/H3k4iU**. You can listen to our three jangly tones for hours on end, or (and we would recommend this), you use this as the starting point for your own projects.

You can create fantastic sounds, even with the limited hardware and processing power of embedded microcontrollers. So, get coding – your number one single is out there waiting for you. ◻

Above ◈
A few lines of code can create waveforms for you to play directly on a Circuit Playground Express

Using CNC machines

Having explored CAD and CAM using FreeCAD in last month's issue, let's look at getting started with CNC routing

Jo Hinchliffe

🐦 @concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open source. He also has a shed full of lathes and milling machines and CNC kit!

We looked at manual milling machines and their use in issue 14, and a CNC router is a similar machine, consisting of a cutting bed and a spindle that can move a cutting tool in to cut a workpiece. Laid out a little differently from a manual mill, CNC routers have a static work bed and the X, Y, and Z movements are made by a gantry system. Stepper motors are used to move these axes via either a belt system or a lead screw system. Each stepper motor has a controller/driver board, and the driver board is connected to a computer which sends the information and instructions about where to move and how quickly. Similar to milling machines, we can use different tooling in CNC routers which can be used for different types of cuts, allowing a range of finishes.

Last month we used FreeCAD to make a model, and to create some tool paths and tool controllers which we compiled into a G-code file. In this tutorial, we will send this file to the CNC from a computer via a piece of software referred to as a G-code sender. The controller on the CNC router that we have access to (**Figure 1**) uses the Grbl system to control the machine and we tailored our G-code file last issue to work with this system. Grbl runs on any Arduino with an ATmega328P processor and acts as an interface between a computer and the control hardware of the machine, and many CNCs use this controller. The G-code sender that we are using is the standard Grbl universal sender application. In fact, we are using a slightly older version than the currently available one, as we set it up when the machine was built. It's common for those building CNC machines to find a solution that works and then stick to it!

YOU'LL NEED

- Access to a CNC machine
- A 3 mm 2-flute end mill
- Some 3 mm plywood

Figure 1
The CNC router we are using in this tutorial
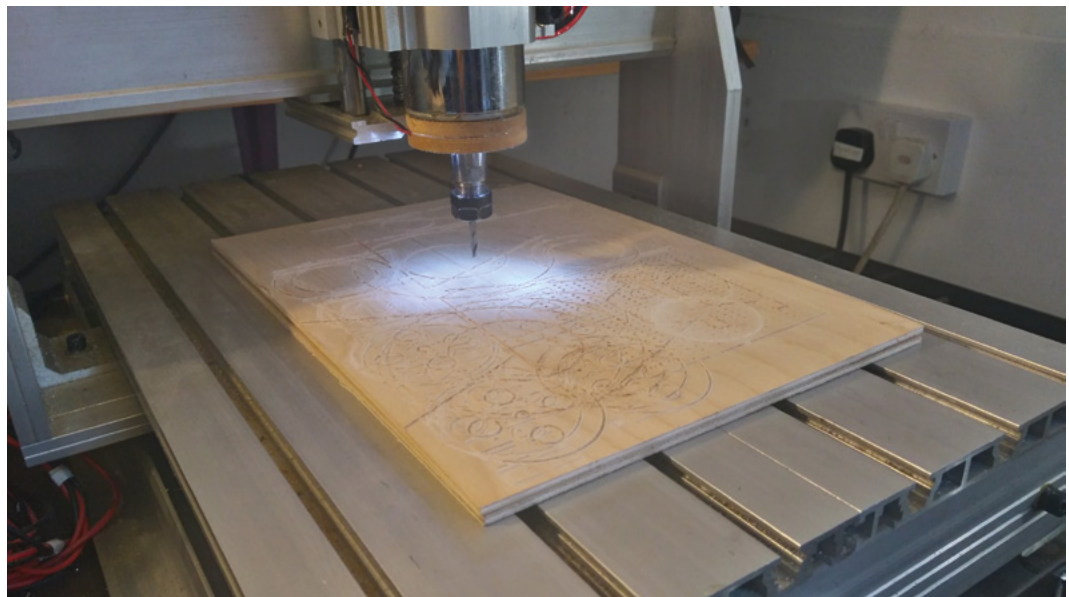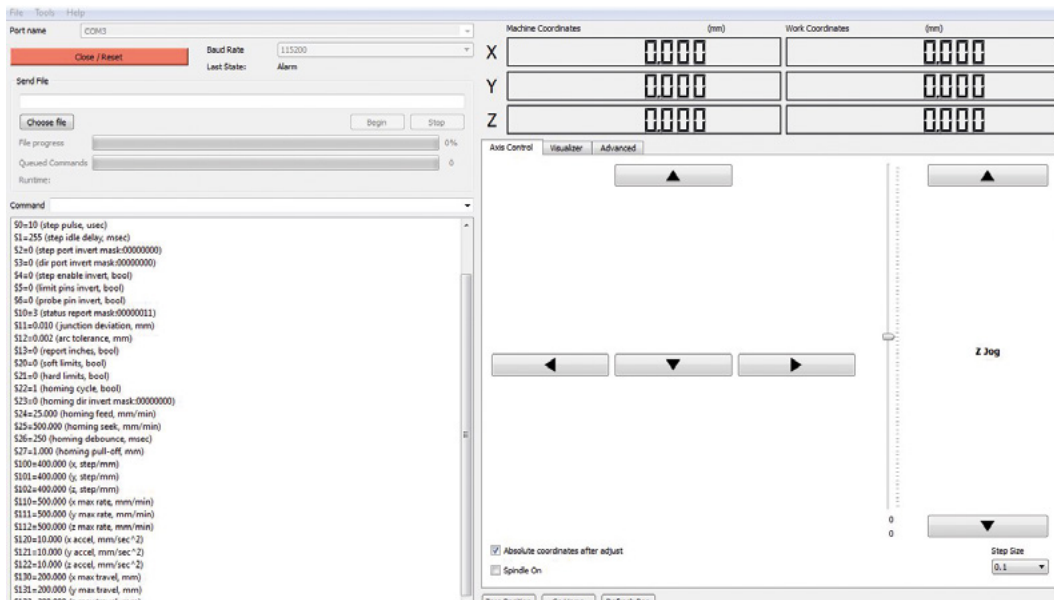
Powering up our CNC router and launching the Grbl universal sender program gives us a screen that looks like **Figure 2**. We need to select the USB port that the CNC is attached to and click the Connect button. We should then see the left-hand side of the screen populate with the current machine settings. CNC routers using Grbl need setting up once with some configuration that tells Grbl how many pulses each stepper motor needs to move the axis a known amount, and some other basic settings for the machine. Having connected to the machine, it remains in a locked state and cannot be moved, so first click the Advanced tab and then click Unlock Grbl (**Figure 3**).

> ❝ Some CNC machines have a replaceable sacrificial wooden **base fitted, but ours has an aluminium bed** ❞

LET'S GO JOGGING

Clicking back to the Axis Control tab, we can try to click an arrow to move or 'jog' an axis. At the moment, this will result in no movement and an error message stating that we have an 'undefined feedrate'.
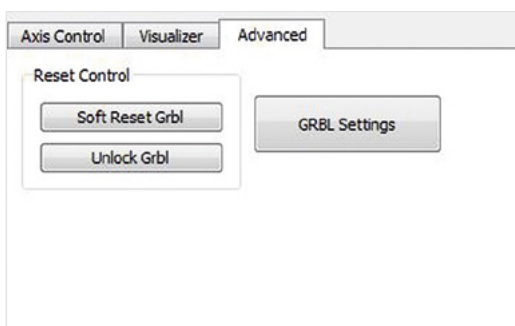
When we are actually sending G-code as part of a job, the feedrate is defined as part of the tool path information it is currently cutting. However, we need to define a feedrate to be able to move the machine before sending the job file. On this machine, we need to manually send a small G-code command to define a feedrate in mm per minute. Type into the Command box 'G01 F400' and hit **RETURN**. The machine is now unlocked and has a feed rate of 400 mm per minute defined. Before we start jogging the machine head (the spindle assembly and axis), it's worth double-checking a couple of things. Firstly, look at the machine and check where the head is located and whether there are any obstacles on the machine you could crash into, such as the endpoints, previous jobs, or clamps left on the bed. Next, check the Step Size amount in the lower right corner of the screen (**Figure 2**). In our version of the universal sender program, it gives us a range of options between 0.01 mm and 100 mm. Our Z axis total length is only just over 100 mm, so if we have it set on that and raise or lower the Z axis, there is an increased chance of a crash! Setting the step size to our required size, we can now use the axis control buttons to jog the machine head around the work area.

CLAMPING UP

Next, we are going to mount the stock onto the bed, ready for cutting. Some CNC machines have a replaceable sacrificial wooden base fitted, but ours has an aluminium bed and we are going to add a →

QUICK TIP

CNC stands for 'computer numerical control' and can be applied to many computer-controlled machines, from sewing machines through to lathes and everything in between!
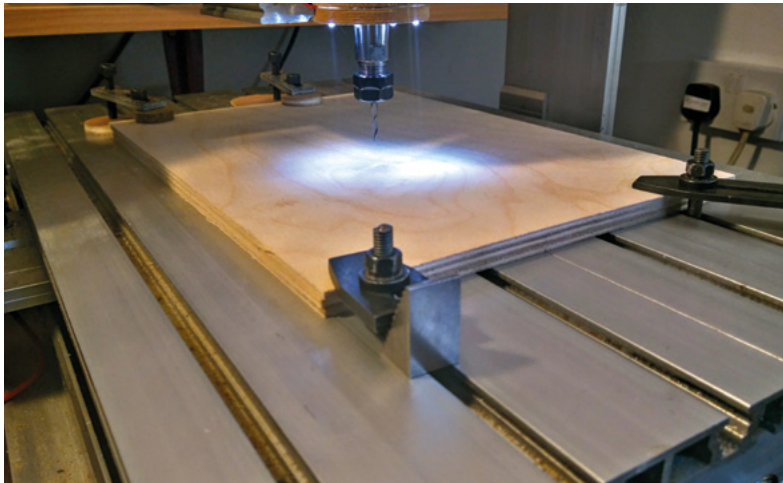
Figure 4
The stock plywood and some scrap or sacrificial material clamped firmly to the machine bed, ready for routing

Figure 5
Once the file is loaded, the tool paths are displayed in the Visualizer tab

scrap piece of plywood under our piece of stock so we don't damage the machine. Clamp your 3 mm plywood stock and sacrificial piece of wood to firmly attach it to the work bed (**Figure 4).**

Having our material in place and the tool set in the spindle, the next task is to choose a point at which to zero the machine. This will act as the origin or datum point from which our job will start. Choosing a zero point in the X and Y axes for a job like this one, when cutting out a flat item from a sheet, is relatively straightforward. We need to move the axes to a position where there is enough material to cut the object out of, and also where the machine head and spindle cannot foul any clamps. It can be hard to visualise where and how big a job might be, so let's load the file for the job we created last month so we can see the tool paths in the Visualizer window. Click the 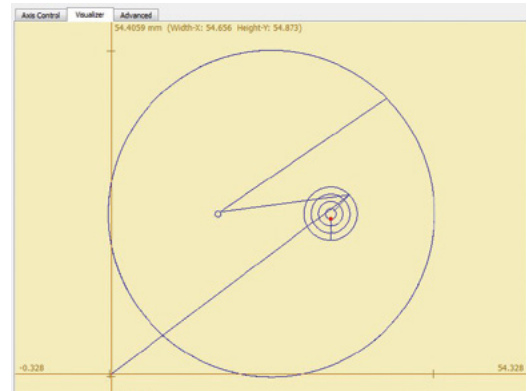Choose File button and navigate to the NC file we made in FreeCAD last month; select and open it. Looking in the Visualizer window, it tells us the maximum dimensions required to run the job on the CNC, including the diameter of the tool on each side (**Figure 5**). In our file, the outer profile cut actually goes past the zero point a couple of times due to

> We need to move the X and Y axes to a position where there **is enough material to cut the object out of**

the tool diameter, so we need to consider this when setting the X and Y zero position.

Having jogged the X and Y axes into a suitable position, we need to lower the Z axis so that it is touching the surface of the stock. There are numerous approaches (including accessories that automate the process). However, a simple and accurate technique is to use a feeler gauge. Feeler gauges are thin pieces of steel of known and accurate thicknesses. Using a feeler gauge that is 0.2 mm thick, we placed this below the tool on the surface of the wood stock. We lowered the Z axis with the units set to 1 mm to get it close, and then changed the units to 0.1 mm and brought the Z axis down until we felt some friction between the feeler gauge and the tool (**Figure 6**). We then know that the tool is 0.2 mm from the work stock surface. We can then choose to use one of a couple of different methods – for one, we can remove the feeler gauge

---

END MILLS AND SLOT CUTTERS

We set up our tool paths last month to be cut by a 3 mm end mill. We have used a 2-flute slot cutter, as the flute arrangement on slot cutters means that they can be plunged vertically into work, rather like a drill bit. Most 4-flute end mills have a small gap in the centre of the face where the flutes don't cut, meaning they cannot be plunged vertically. There are ways to still use these tools on a CNC router which involve ramping the tool into the work at a shallow angle rather than vertically, allowing them to 'cut their way' into the workpiece. Our machine uses ER11 collets in the spindle, so we inserted the 3 mm 2-flute slot cutter into the correct-sized collet, and then tightened it into position.
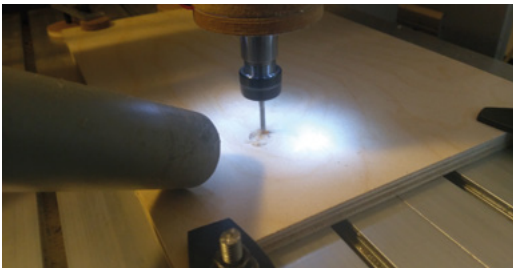
LET'S SPIN IT UP

Having left the tool 0.2 mm clear of the cutting surface, it also means that we can start the spindle before setting the CNC in motion. On this CNC, the spindle is controlled manually with a switch to turn it on and off, with a speed controller dial for the spindle speed. Other machines may have the spindle speed under CNC control as well, and Grbl can indeed control spindles on, off, and rpm. We find that for plywood cutting with a small diameter cutter like this one, around 9000–10,000 rpm gives a decent cut. Again, different machines may vary.

Having set the spindle speed and started it, we click the Begin button. Grbl universal sender starts to stream the G-code commands to the CNC. The spindle lifts to its clearance height set at 5 mm and moves to above the pocket cut. It lowers and begins to clear material, following the tool paths we made last month. Whilst the job is running, we started our small shop vacuum cleaner and held the nozzle near the job to extract the woodchips and dust created (**Figure 7**).

The job continues, and the last action it takes is to do the outside profile cut, which includes the tags that mean the workpiece stays on the machine (**Figure 8**). Once finished, we turned off and jogged the spindle out of the way, and gave the surface of the workpiece a light sanding to remove the straggly burrs left on the edges, as it's easier to do this whilst the work is held flat. We then removed the board and trimmed the completed workpiece out using a craft knife.

The job was created perfectly, and we love that the entire process of CAD and CAM, through to sending the code to the machine, used free and open-source software. If you go down this path (pun intended!), and this article has helped you understand CNC routing a little more, do tweet us and show us your work **@HackSpaceMag**. □

Figure 6
Using feeler gauges to accurately position the Z axis

Figure 7
We are off! Having set the job in motion, we occasionally move our shop vacuum close to the job to clear the sawdust

Figure 8
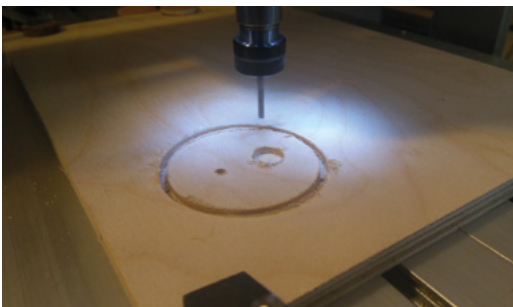The final cuts are the outer profile cuts, which include the movements to leave the tags of wood that keep the workpiece connected to the stock

and lower the Z axis a further 0.2 mm and then we can click the 'zero position button'. This is a good approach, but sometimes we may be working with a delicate material where we don't want the tool to contact the workpiece before we start the actual job. The second approach is to send another manual command telling the machine its current coordinates. If we type 'G92 X0 Y0 Z0.2' into the command line, this sets those coordinates at the current spindle position. Having used either of the previous methods, click the Refresh Coordinates button. The Work Coordinates will either zero fully or set to X0, Y0, and Z0.2, depending on the method used.

We are nearing the point where we set the machine in motion and begin cutting. Make sure that before you do so, you follow your safety practices! Make sure you have eye protection on, and any loose clothing or long hair is safely stowed away. Make sure that you have checked that any emergency stop buttons are working properly, and make a final check that nothing is impeding the free movement of the machine.
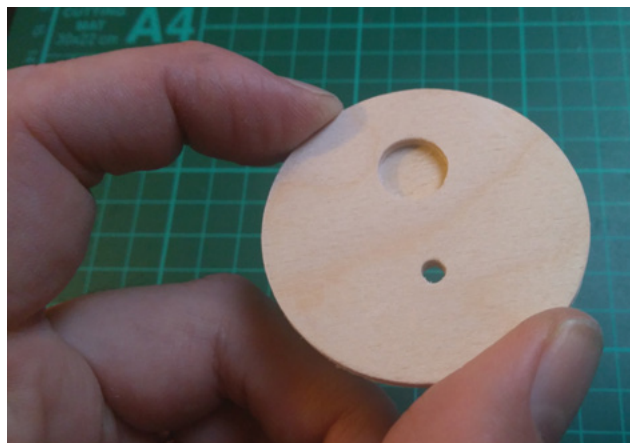
Left
The finished job in hand, having trimmed the tags and given the surface a light sanding to clean it up

Maker's toolbox:
Impact drivers and wrenches

Screwing screws and bolting bolts the easy way

TOOLS
Provided by
SCREWFIX

Ben Everard

🐦 @ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

If you've done much woodwork or DIY, there's a pretty good chance you've used an electric drill as a screwdriver, and for basic use, they work quite well. After all, they're designed to spin things around, and that's what a screwdriver does as well. However, while it's fine for light use, a drill is really designed for drilling, not driving in screws or attaching bolts. There are specialised tools for this: impact drivers and wrenches.

While impact tools still spin around like a drill, they also use a series of 'impacts' which are sudden, short rotations – perhaps think of it a bit like putting a wrench on a bolt and then tapping the handle of the wrench with a hammer – and it's these impacts that let you screw things in or out with more power than the clean rotation of a drill. This is different to the hammering action of a hammer drill, which goes in and out rather than in a circular motion.

Using an impact tool is really simple. Impact drivers typically have a hex attachment which you can slot screwdriver bits into, while impact wrenches generally have square attachments for sockets. One slight trap

Right ↗
This impact wrench may look like a drill, but the rotation works in a very different way

is that there are several different sizes of square drive typically measured in inches (¼, ⅜, and ½ inch), so you need to make sure this fits with the sockets you have (or intend to purchase). While these sizes are the same-sized bits as commonly used on non-impact screwdrivers and wrenches, not all attachments are created equal. The impacts these tools use can damage some driver bits and sockets, so you should make sure to use impact-rated accessories.

Once you're set up with the appropriate bits and sockets, you can use the tool more or less as you would a drill. Point it in the right place and squeeze the trigger until what you want is either screwed or unscrewed. Don't be alarmed if it starts clicking – this may sound a little like a drill clutch slipping, but it's the impact mechanism doing its job. These little hammering motions will firmly attach your screw or nut.

There are a few big advantages to impact tools over drills. Their power-to-weight ratio for screwing is significantly better, so a lighter tool will often be able to drive with more force. The impact action puts less strain on your arms and wrists than a pure twist, so is less tiring on your muscles (but they do vibrate more). With impact drivers, the motion also grips screw heads better than a plain twist, so they're less prone to pop out and strip the screw head.

DRIVING FORWARD

We've been talking a bit about drivers and wrenches, but what is the tool for you? Impact drivers generally take screwdriver bits and are most commonly used for screwing long wood screws into (and out of) wood. However, you can get impact-rated sockets that fit into the hex socket. Impact wrenches, on the other hand, are most commonly used for attaching and removing nuts and bolts and are often used by vehicle mechanics. You can get sockets that will let you use a screwdriver bit with an impact wrench, but you have to be careful as wrenches are often more powerful than drivers and may rip the head off the screw rather than driving it.

When picking an impact wrench, it's probably most important to think about the power you need. While it might be tempting to just go for the most powerful you can afford, this isn't necessarily the best plan. After all, if you don't need the power, it's wasted, and more power comes with more size and weight, which means a tool that's more tiring to use. Pick the tool that's best suited to your needs – even if you need to use a breaker bar occasionally for the most stubborn of bolts, you'll have a more comfortable experience the rest of the time.

With an impact driver, you'll need to make sure it's got enough torque for your needs, but there is less variation than with wrenches (and over 150nm should drive most screws in or out, and you might be able to get away with less). Picking a tool that's comfortable – particularly if it fits with the batteries and chargers you all ready have – is more important.

We tested out a DeWalt DCF813D2 impact wrench with a 10.8V 2Ah battery. By impact wrench standards, it's at the smaller, low-power end of the spectrum, but still a capable tool for all but the most

> **"** Their power-to-weight ratio for screwing is significantly better, **so a lighter tool will often be able to drive with more force "**

stubborn bolts on a small vehicle. It's light, and easy to use time after time, and comfortable in the hand. For general wrenching work, it's a good option, particularly at the DIY/hobbyist end of the spectrum. We also tested it out for driving screws, and while it worked well, a proper driver would perform the job better.

Do you need an impact driver or wrench? If your current setup – be it hand-powered screwdriver, electric drill, or traditional wrench – is working well for you, then there's no reason to switch. However, if you're finding yourself struggling with tough screws and nuts – particularly if you have to do a lot of them in quick succession – then you might want to consider switching. ◻

Below ◈
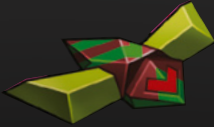Make sure you get the tool with the type of attachment you want such as ⅜ square (shown here) or hex

CODE
THE
CLASSICS

VOLUME 1

CODE
THE
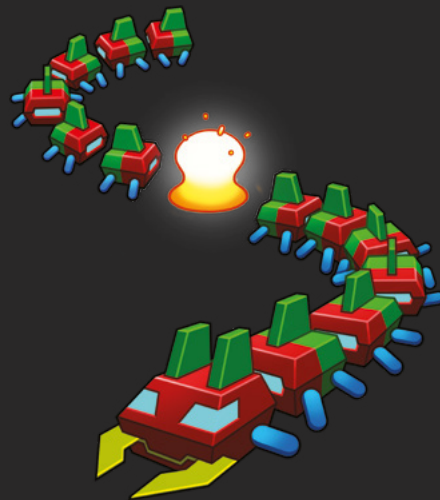CLASSICS

VOLUME 1

Brimble
Crookes
Gillett
Malone
Tracey
Upton²

Sew your own clothes

How to use a sewing machine and patterns

Poppy Mosbacher

🐦 @PoppyMosbacher

Poppy loves getting tech into the hands of people who do traditional crafts. She is helping set up a makerspace in Devon and was a director of Build Brighton MakerSpace.

The fashion industry is one of the most polluting industries in the world, so making your own clothes might be a good way to reduce your carbon footprint. For sewing to be part of a modern lifestyle, it needs to be easy to learn with quick, reliable results. And that's how new tools and digital technology are transforming sewing. It's early days, but it's already possible to create bespoke, quality garments without years of experience, and see them on you in virtual reality before you begin.

This month's tutorial is an introduction to sewing with a modern, electric sewing machine. In the next issue, we'll cover how to make a jacket, using **freesewing.org**. It's open-source software that automatically creates sewing patterns based on your body measurements, so you can make clothes that fit well without being skilled at pattern design or alterations.

Sewing machines are available with a variety of different features, but the way they use two threads to construct stitches is the same in most machines. The upper thread goes through a needle, which joins the lower thread coming from a bobbin.

For this tutorial, we're using a Brother Innov-is 15. As each make and model has slight differences, you will need to refer to your machine's manual for specific setup instructions.

WIND THE BOBBIN UP

The first step in machine sewing is to fill a bobbin with the thread you'll be using. There's a handy bobbin winder on the machine, shown in **Figure 1**. Place an empty bobbin on the bobbin winder shaft and a reel of thread on the spool pin. Follow the threading instructions on the top of the machine to guide the thread to the bobbin winder. Push the bobbin shaft to the right to engage the winding feature. Push down on the foot pedal, and the

YOU'LL NEED

- **Sewing machine**
- **Sewing machine manual**
- **Reel of cotton or polyester thread**
- **Fabric**
- **Bobbin**
- **Rotary cutter**
- **Cutting mat**
- **Small weights**
- **Sewing pattern**
- **Stitch unpicker**
- **Fabric clips**

Needle position button

Reverse stitch button

Thread take-up lever

Thread tension dial

Spool pin

Bobbin shaft

Bobbin winder

Stitch length and width

Handwheel

Selected stitch

Stitch selection dial

Cutting blade

Speed control

Presser foot lever

Baseplate

Needle bar

Needle

Presser foot

Feed dog

Bobbin holder
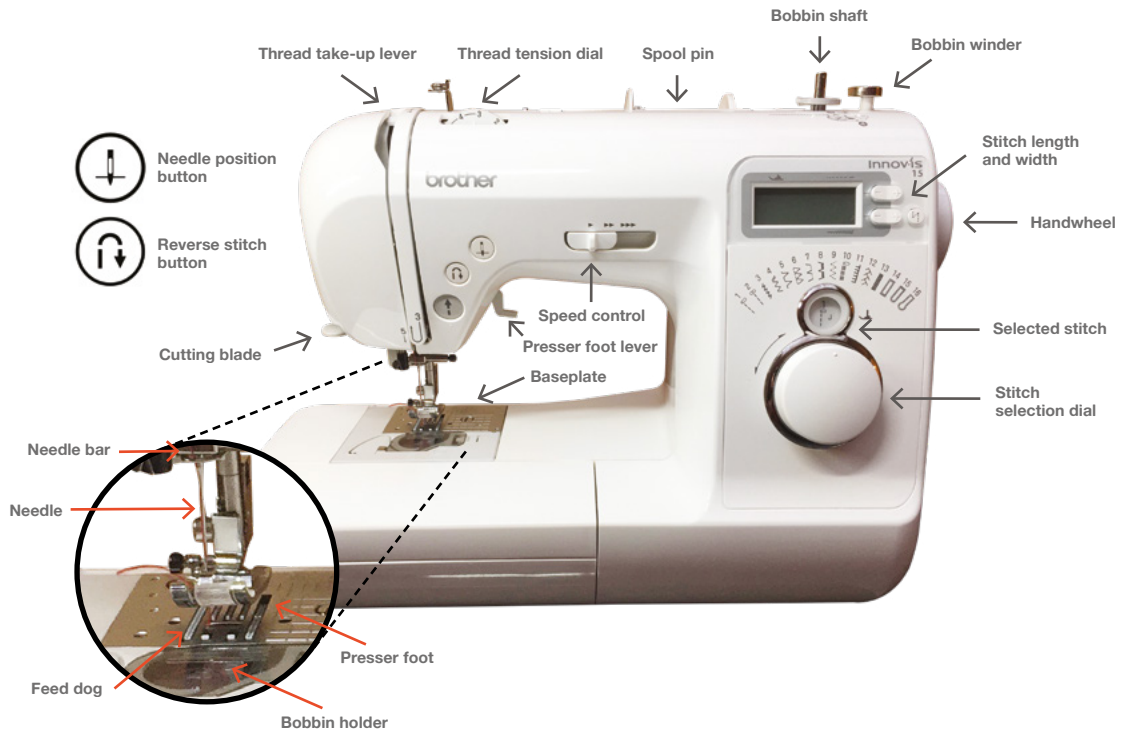
SEWING IN HACKSPACES

Some hackspaces have dedicated sewing areas with specialist sewing machines, such as automatic embroidery machines and overlockers/sergers. A few even have second-hand industrial machines, which are great for doing big or bulky projects such as coats, curtains, and bed covers. They may also have regular textiles groups where you can work on projects in a social setting and share tips and tricks.

bobbin will start to spin. When the bobbin is full of thread, release the foot pedal. Push the bobbin shaft to the left, remove the bobbin, and cut the thread to separate it from the original reel.

Place the filled bobbin inside the bobbin holder (see **Figure 2** overleaf). Look for symbols on the machine that show which direction the thread needs to go. Make sure you put it in the right way, otherwise the machine could get jammed. If your sewing machine doesn't automatically thread the bobbin through the hole in the baseplate when you start sewing, refer to your manual for how to do this.

> It's like the accelerator on a car: the more you press **down, the faster the machine will sew**

EYE OF THE NEEDLE

Before threading the needle, raise the presser foot using the lever at the back of the machine. This will release the tension disks, making it easier to move the thread. If the needle is touching the baseplate, press the needle position button to raise it. Then turn the machine off, to avoid accidents whilst threading the needle.

With the reel of thread on the spool pin, take the thread through the metal thread guide on the top of the machine. It's usually marked with the number 1. Pass the thread through the thread guide cover, marked number 2. Follow the numbered path going through the take-up lever from right to left (see **Figure 3** overleaf) and behind the needle bar. Some machines have automatic threading levers, which will pass the thread through the eye of the needle, and others need to be threaded by hand. Once the thread is through, guide it into the slit in the presser foot and pull approximately 5 cm out towards the back of the machine.

READY, SET, SEW

Always sew a test piece before sewing a garment, to check you have the appropriate stitch settings. To

do this, cut two rectangles from the fabric you'll be using and place them on top of each other, under the presser foot. Lower the foot using the presser foot lever to hold the fabric firmly in place.

Next, choose a stitch by turning the stitch selection dial, or equivalent on your machine. For now, choose a straight stitch, which is likely to be the first option. Use the + and - buttons to choose a stitch length, such as 3.0.

A letter will be displayed next to the selected stitch to indicate which presser foot is needed. For most stitches, including the straight stitch we're using, it needs the standard presser foot, called a zig-zag foot.

Push the foot pedal to start sewing. It's like the accelerator on a car: the more you press down, the faster the machine will sew. Gently guide the fabric with your hands to keep it on course, making sure your fingers are away from the needle. Remember, needles are sharp and the machine →

Above ◈
A zig-zag stitch is useful for attaching ribbons

QUICK TIP

It doesn't matter if you make a mistake when sewing. You can use an unpicker to cut out stitches without ripping the fabric.

Figure 1 ◈
The bobbin winder fills a bobbin in less than a minute. You can also buy colourful sets of pre-filled bobbins

Figure 2
This machine has a top-loading bobbin holder, but some use a removable bobbin cage that is inserted inside the machine

doesn't know if it's sewing fabric or flesh. If things feel like they're going wrong, take your foot off the pedal and re-orientate yourself.

It's common practice to stop sewing just before you reach the end of the fabric, rather than going off the edge. If the needle is in the down position when you stop, press the needle position button to raise it.

make sure the needle is in the up position. Then, turn off the machine. Raise the presser foot lever and press the button behind the presser foot holder to release it. Hold the new presser foot under the metal bar of the presser foot holder, and slowly lower the presser foot lever so it snaps in place. Some machines have a screw instead of a snap fitting.

CUTTING FABRIC

A popular alternative to scissors and pins is to use a rotary cutter on a self-healing mat, but beware of the blades on rotary cutters: they can be very sharp. Use small weights to hold the fabric in place when cutting, and clips to join fabric before sewing.

If the pattern asks you to cut on the fold, fold the fabric over, and place the paper pattern on top of both layers of fabric, with the straight edge of the pattern in line with the folded edge of the fabric.

> **"** A popular alternative to scissors and pins is to use a **rotary cutter on a self-healing mat** **"**

Lift the presser foot lever and remove the fabric by pulling it a few centimetres away, and cut the threads. On the left side of many machines, there is a thread-cutting blade.

Check the test piece to make sure the stitches look even on both sides. The tension usually works best with the dial set at 4, but if any loops are showing, make it a bit tighter and if the stitches are puckering, reduce the tension.

When sewing a test piece, it doesn't matter about loose threads at both ends, but normally you need to secure them to stop the seam unravelling. Modern machines have a locking stitch button for this which you press at the beginning and end of each seam to sew a few stitches on top of each other.

Sewing machines come with a variety of presser feet, with special ones for attaching zippers and making buttonholes. To change the presser foot,

Right
A zipper foot allows you to sew close to the teeth of the zipper

Above ⬆
The feed dog keeps the fabric moving by feeding it through the machine, so you don't need to pull or push the fabric

There will sometimes be arrows on the paper pointing to the fold line. Notches are little triangles on the pattern to show where to match up with another section. When you're cutting out a pattern, you need to mark the notches on each layer of fabric, either by a tiny cut or with a water/air erasable pen.

UNDERSTANDING SEWING PATTERNS

Sewing patterns refer to fabric as having a right and wrong side. The right side is visible on the garment when it's finished, and the wrong side is inside. When fabric looks the same on both sides, it's worth sticking a small piece of masking tape on the wrong sides after cutting the pattern pieces, so you can easily identify them. It might not seem important, but if you've watched *The Great British Sewing Bee* when a sleeve is sewn on inside-out, you'll have seen how it hangs backwards at a strange angle.

Sewing patterns usually specify a seam allowance, such as 1 cm or ⅝ inch. This is the distance between the line of stitches and the edge of the fabric. The baseplate has measurements on it, so you can line up the fabric as you guide it through the machine.

Beginners often worry about making sure their seams are dead straight, but small mistakes aren't usually seen. Most people don't look that closely at clothes. If you inspect a few items you've bought, you may even find top-stitching in a contrasting colour that you've never noticed, such as blue jeans sewn with bright orange thread.

A scarf, sleeveless top, wash bag, or laptop case are easy projects to put into practice the skills covered in this tutorial. ▢

Figure 3 ◈
Make sure the thread take-up lever is at the highest position when threading

QUICK TIP

If the seam allowance isn't marked on the baseplate, measure from the needle and stick a piece of tape at the required distance as a guide.

NEW TOOLS FOR TEXTILES

- 3D body scanner to import accurate models into design software
- CIO3D to design patterns directly onto a digital version of yourself, and see how the clothes will look from every angle
- Pen plotter to draw patterns
- Laser cutters to cut some fabrics, such as felt and engrave on denim
- Overlocker/serger to sew stretch fabric, e.g. to make T-shirts quickly
- 3D printer to make belt buckles, buttons, and to print snap-fasteners directly onto fabric
- Digital cutting machines such as Silhouette or Cricut

Please note: fabric can catch fire in laser cutters if proper precautions are not taken. Make sure you understand, and properly mitigate, any risks before using fabric in the laser cutter.

Get the most out of your paints

Tips and tricks for using and maintaining paints and brushes

Gareth Branwyn

🐦 @garethb2

Gareth has been a lifelong practitioner (and chronicler) of DIY tech, media, and culture. He is the author of ten books, including *Tips and Tales from the Workshop*, and is a former editor for Boing Boing, Wired, and Make:.

Above ↗
Stirring at speed (but make sure you protect your eyes)

QUICK TIP

Did you know that you can get different nozzles for spray can paints? There are tips that put out a much fatter spray, fine spray, and different shaped sprays. Search on 'spray can caps' at your favourite online store, or at a local bricks and mortar store that sells art supplies.
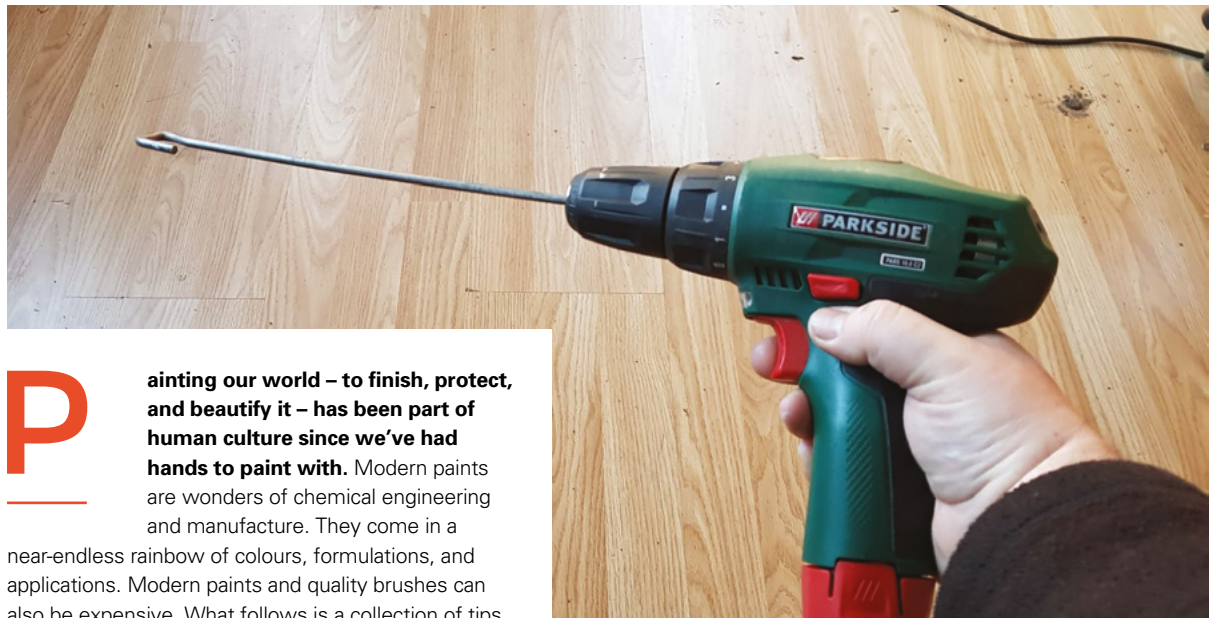
Painting our world – to finish, protect, and beautify it – has been part of human culture since we've had hands to paint with. Modern paints are wonders of chemical engineering and manufacture. They come in a near-endless rainbow of colours, formulations, and applications. Modern paints and quality brushes can also be expensive. What follows is a collection of tips for working with, storing, organising, and extending the life of your paints and brushes.

Tent peg (or whisk) paint stirrers – if you have to stir a can of house paint or other large quantity of paint, chuck a metal tent stake, or even a kitchen whisk, into a power drill and you will spare your wrists as you stir.

Heat old spray cans – If you have old cans of spray paint, or you are trying to spray in a cold environment, before shaking, place the can in a pot of warm (not boiling!) water. This will not only work to thin the paint, getting it ready for a good shaking/remixing, it will also help to re-pressurize the propellant. Make sure to shake the can especially well and test the output before spraying your object.

Turn the can upside down – When finished with spray paint, turn it upside down, and give the nozzle one last toot to clear the throat of any residual paint.

You can spray Styrofoam – There's a common belief that the solvent in spray paint will melt Styrofoam. This is not universally true. Some paint will melt some Styro. First off, you can help prevent this by sealing your Styro-based object with Mod Podge or similar. Then, you want to test your paint on a scrap piece of Styro. With sealant, the right paint, the right distance, and testing, you can spray paint Styrofoam.

Mark your paint cans – When you've finished using a can of paint, before you close the lid, use your brush to mark the level of paint inside. This will not only show you how much paint is inside, it will also show you the colour. You can also mark a large dot of the colour on the lid for easy identification.

Identify which rooms used which paint – On the bottoms of your house paint cans, write the rooms in which that paint was used in.

Use poster putty/cheap clay as a mask – If you need a quick mask for spray painting an area that's hard to tape-off (such as a curved piece of moulding), consider using poster putty/mounting putty, or cheap dollar store/pound shop clay. It can be quickly moulded over anything you wish to cover.

Use agitator balls – If you use small pots, squeeze bottles, or dropper bottles of craft/hobby paints, consider adding stainless steel agitator balls (or 3 mm haematite beads) to them. Shaking with these agitators inside will greatly assist in mixing your paints.

Use screws to raise items for painting – If you're painting or finishing an object, say a piece of furniture, you can raise it up from your work surface by sinking some screws or nails into the bottom of it. When done painting, remove the screws.

Improving cheap brushes – Cheap craft shop brushes are great for things like painting with kids, applying glues or other substances that are tough on brushes, applying faux finishes, and the like. But since these inexpensive brushes are cheaply made, the bristles often come out into your work. To prevent this, before you first use them, place a bead of superglue along the base of the bristles, above the ferrule.

Clean brushes with Murphy's soap – Murphy's Oil Soap is great for soaking and releasing dried oil paint from brushes. It can restore them. Soak for 24–48 hours at near full strength.

WET PALETTE

If you do any sort of small-scale acrylic painting, such as scale modelling, dioramas, crafting, or painting gaming miniatures, consider making a wet palette to paint from. It will keep your paints moist, even overnight, and it allows you to mix gradations of colours.

To make a wet palette, all you need is an airtight plastic container of the desired size, a piece of sponge cloth cut to fit inside the container, and a piece of baking parchment paper.

To use, you soak the sponge and wet the parchment paper. You can then place paint on the paper (the water in the sponge even helps thin the paint) and mix colours as desired. When done with a painting session, snap on the lid. The paints will sometimes remain viable for days.

Above ◆
Keep mixed paint in a wet palette

Let brushes dry flat – After you clean your brushes, don't place them standing up with the handle-side down. Water from the cleaning will leak down into the metal ferrule, and eventually degrade the glue. Let them dry flat or hang them brush-side down.

Special brushes for metallics – It's a good idea to use brushes that are dedicated to metallics (and use cheap brushes, if you can get away with it). The metallic flecks in the paint can damage brushes, and residual flecks can make their way into other paints if you use the same brush for your metallic and non-metallic paints.

Acrylic storage shelves and racks – If you have a lot of small paint pots or bottles to organise, there's a dizzying array of shelves and racks you can purchase. Many of them involve assembly and are fairly expensive. Consider using nail polish shelves and racks (made for beauty salons). They are cheap and they screw together in seconds. You can find them at online stores like Amazon.

Handy touch-up bottles – If you do a search at your favourite shop for 'paint touch-up bottles', you will find small bottles with brushes in their lids. When you paint a room in your house, pour some of the paint into one of these bottles and list the room on the label. Then, if you get a ding on your wall or remove hanging hardware, you can use the pot of paint and brush to mend the wall. □

Below ◪
Keep small amounts of paint in handy touch-up bottles

QUICK TIP

Don't be afraid to thin down cheap acrylic craft paints that come in tubes for use in an airbrush. You have to experiment with brands and paint-to-thinner ratios, but you can save a lot of money if you have a large area you need to spray paint on the cheap.

THANKS

Special thanks to Andrew Lewis, Christie Uhler, Susan Jamison.

RaspBaby Pi WiFi Cam

Build a wireless camera that monitors temperature, pressure, and humidity

Dr Andrew Lewis

🐦 @monkeysailor

Dr Andrew Lewis is the owner of **Shedlandia.com**, a restorer of old tools, a fabricator for hire, a research scientist, and a founder member of the Guild of Makers.

Y

ou can get wireless baby monitors that have a whole range of great features for making sure your little ones are safe, sound, and sleeping happily, but they come with a hefty price tag. In this article, you'll find out how to make a Raspberry Pi-powered streaming camera, and combine it with a built-in I²C sensor pack that monitors temperature, pressure, and humidity. You'll also see how you can use the GPIO pins on Raspberry Pi to turn an LED night light on and off using a web interface.

The hardware for this project is quite simple, and involves minimal soldering, but the first thing you need to do is to install Raspbian onto a microSD card for your Raspberry Pi. If you're planning on doing a headless install, you'll also need to enable
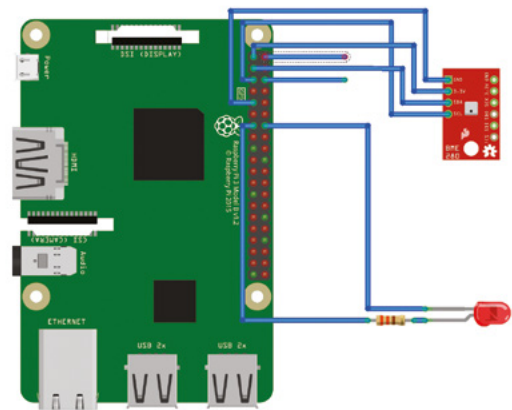
SSH by creating an empty file called **SSH** on the root of the Raspbian install, and a file with your WiFi details called **wpa_supplicant.conf**, as described in **hsmag.cc/SbWpQ7**.

You can download the code for this as well as the 3D printable files from **hsmag.cc/issue26**. You'll need to transfer the code to the Raspberry Pi.

Next, connect the camera, the BME280 board, and the LEDs to the Raspberry Pi, as shown in the circuit

YOU'LL NEED

- **Raspberry Pi 3**
- **ZeroCam night vision camera**
- **50 cm Raspberry Pi 3 camera ribbon cable**
- **BME280 I²C temperature/ pressure/ humidity sensor**
- **Braided USB 3 cable**
- **2 × 5 mm white LED**
- **330 Ω resistor**
- **Tin can, approximately 72 mm diameter**
- **USB microphone (optional)**
- **60 cm of 2 mm aluminium armature wire**
- **30 cm of 40 mm heat-shrink tubing**
- **DuPont connectors**
- **A weight – approximately 150 g, although this isn't critical**
- **3D-printed parts**
- **Spray paint PlastiKote stone effect and blackboard paint**

Above ◈
Finishing a 3D print by applying a thin layer of car body filler and sanding back will give a much smoother surface. This isn't always necessary, but if your filament is damp or your nozzle is worn, it can make a model look much better when it's painted

diagram. Insert the microSD card into the Raspberry Pi and boot up. If everything is working OK, you should be able to see the IP address for your device listed on your hub or router, and you should be able to connect to it via SSH. If you don't see the Raspberry Pi listed, check your wireless connection details and make sure your adapter is supplying enough power. It's worth taking the time to assign your Raspberry Pi with a static IP address on your network, so it won't change IP unexpectedly.

babycam script are the RPi.bme280 module, Flask, PyAudio, picamera, and NumPy. Chances are that these are already installed on your system by default, with the exception of RPi.bme280, which can be installed by typing `sudo pip3 install RPi.bme280` from the terminal. Once all of the dependencies are satisfied, load up the script and give it a test run, and point your web browser →

> **It's worth taking the time to assign your Raspberry Pi**
> with a static IP address on your network

Below ◈
If you want to add more than a couple of LEDs (or want to use brighter LEDs), you should connect your LEDs to the power input, and use a transistor on the GPIO to trigger them

SMILE FOR PICAMERA

Use the raspi-config application to enable the camera interface and the I²C interface. If you're planning on modifying the code yourself, we recommend enabling VNC access as well, because it will make editing code and debugging much easier once the device is put together. All that remains on the software side is to update APT, download the **babycam.py** script, install any dependencies with PIP, and set the script to run automatically. The main dependencies for the

Right ➡
You need to mount the BME280 so that the sensor is exposed to the air in the room. Do this by drilling a small hole in the 3D-printed top piece and hot gluing the sensor into position. If you're going to use the optional microphone, you can add an extra hole and glue the mic into place in the same way. A short USB port extender will give you enough cable to plug the USB microphone into the socket on your Raspberry Pi

QUICK TIP

For this project, the power for its Raspberry Pi is being provided through the GPIO socket on pins 4 and 6, and not through the USB socket, because there isn't enough space inside the case to use a USB plug.

Left ◆
If you're having trouble getting the armature wire through the 3D-printed parts, try using a drill to help twist the wire through

at port 8000 on the Raspberry Pi. You should see a webpage with a camera image, controls for the LED lights, and a read-out of the temperature, pressure, and humidity of the room.

The easiest way to get the **babycam.py** script to run on boot is to add a line to the **rc.local** file. Assuming that the **babycam.py** file is located in your home directory, you should add the line `pyth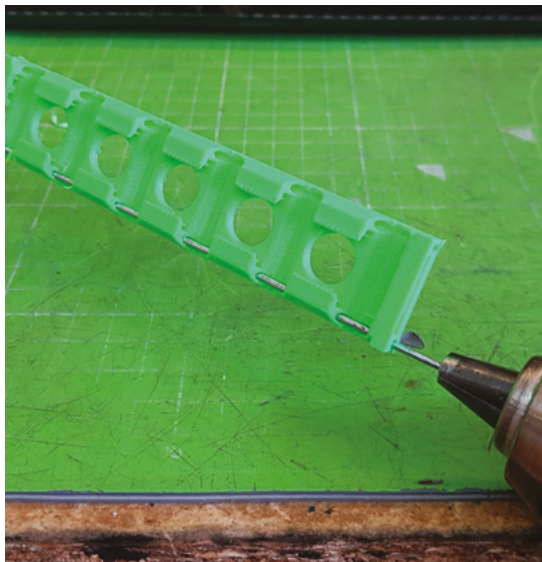on3 /home/pi/babycam.py &` to the **rc.local** file, just before the line that reads `exit 0`. It's very important that you include the ampersand at the end of the line, otherwise, the Python script will not be run in a separate process, the **rc.local** file will never complete, and your Raspberry Pi will never boot.

TINNED RASPBERRY PI

With the software and hardware working, you can start putting the case together. You might need to scale the 3D models to suit the tin can you have before you print them out, so measure your tin before you click Print. You'll also want to remove any inner lip from the top of the can using a can-opener, and make a small hole in the side of the can near to the bottom, to put the USB power cable through.

ACCURATE TEMPERATURES

The temperature reading from the BME280 can be inaccurate, due to self-heating of the sensor. It's worth considering this fact when you're setting up your camera. You can either add a transistor to power the device only when it's needed, or apply an offset to the reading from the sensor in the code. We opted for the latter, but might change that in later versions of the camera if we want more accuracy.

Next, make a hole in the bottom of the can for the LED cables to pass through.

If you haven't already done so, solder appropriate leads to your LEDs, and don't forget to put a 330 Ω resistor in-line on the positive side. The neck of the camera is supported by two lengths of aluminium armature wire. Push the wire up through each of the printed neck pieces, and use a clean soldering iron to weld the pieces together in the middle. Push the

> **The neck of the camera is supported by** two lengths of aluminium armature wire

neck into the printed top section, and weld into place with a soldering iron from underneath. Be careful not to block the narrow slot with plastic, as this is where the camera cable passes up through the neck and into the camera.

Paint the tin can and the 3D-printed parts. We found that spray blackboard paint gives a good effect on 3D-printed parts, and PlastiKote stone effect paint made the tin can look a little more tactile than a flat colour. →
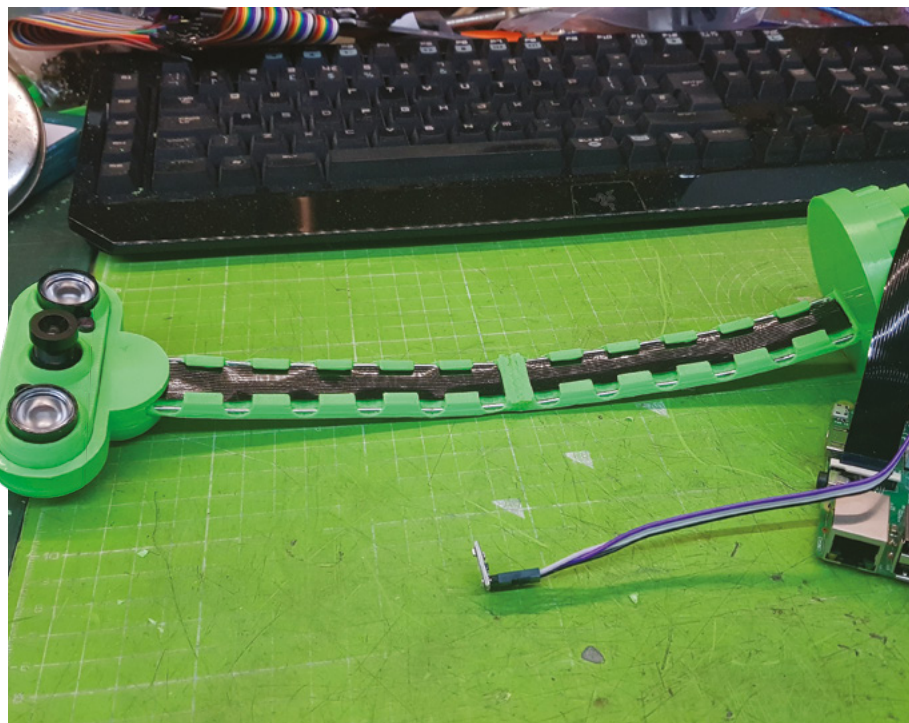
Above ◈
If you're having trouble getting the armature wire through the 3D printed parts, using a drill to help twist the wire through can be help.

Once the paint is dry, pass the camera cable up through the slot in the neck, and then apply the heat-shrink tubing to cover the neck with a small gap at the top and bottom. Connect the camera to the top of the cable, and push the front piece on to hold it into place. Glue shouldn't be necessary, but a little hot glue might help if the front parts don't hold together well.

Push the power cable through the hole in the case, and secure it with a knot and some hot glue. Leave enough cable free to easily remove the top section from the can in future without stressing the wires.

THIS IS GETTING HEAVY

Glue the bottom section onto the can with hot glue, and hot-glue the LEDs into place on the bottom, feeding the cable up through the hole and into the GPIO header. This is a good time to hot-glue a weight into the bottom of the can to improve its stability. I used an old weight from some kitchen scales, but any small weight should be fine. Finally, fix the Raspberry Pi into place on the top piece by either drilling or gluing, then reconnect the rest of the cables, and push the 3D-printed top section into the tin can. If the top section is too loose, you can add a little bit of hot glue to hold things together once you know everything is working.

That should be all of the steps complete. Plug in the USB and check the camera from a web browser. The **babycam.py** script includes video, sensors, and light control. If you are using the optional USB microphone, you can expand the functionality of the app to include audio streaming, use cry detection to activate the LEDs (don't make the LEDs too stimulating or you'll never get a night's sleep again), or maybe even add a Bluetooth speaker and integrate a home assistant. □

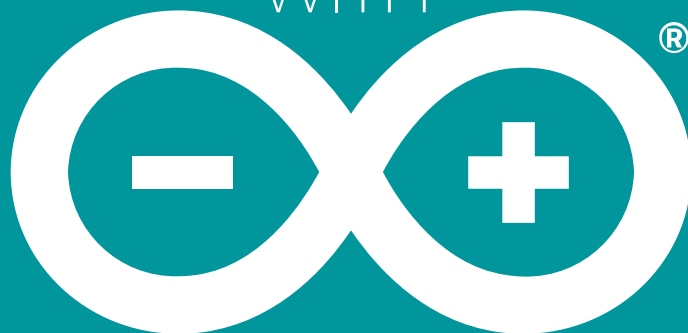> **Push the power cable through the hole in the case,** and secure it with a knot and some hot glue

Below ◈
A bit of heat-shrink tubing provides the finishing touch

How to set up OctoPrint

Retrofit your 3D printer with a Raspberry Pi
for awesome WiFi functionality

Glenn Horan

🐦 @BatGlenn13

Glenn is a software developer by day and a 3D printing enthusiast by night. Rumour has it that building and troubleshooting cheap Prusa clones is what caused him to go bald before his years.

YOU'LL NEED

- **Raspberry Pi 3** (or better)

- **Raspberry Pi appropriate microSD card** (hsmag.cc/JyfGFv)

- **Raspberry Pi power adapter**

- **USB cable** (the connector type will depend on your printer)

- **Webcam/ Raspberry Pi Camera** (optional)

- **3D-printed camera mount** (optional)

f you own a 3D printer, you'll likely have at least heard of OctoPrint from the ever-benevolent 3D printing online community. It has the potential to transform your 3D printing workflow for the better, and it's very easy to set up. This guide will take you through the process step by step, and give you some handy tips along the way.

Before we start finding out how to install OctoPrint, let's look at why you might want to. OctoPrint is a piece of open-source software that allows us to add WiFi functionality to any 3D printer with a USB port (which is pretty much all of them). More specifically, you'll be able to drop files from your computer onto your printer, start/ stop prints, monitor your printer via a live video feed, control the motors, control the temperature, and more, all from your web browser. Of course, with great power comes great responsibility – 3D printers have parts that are hot enough to cause fires, so make sure you have a safe setup, which may include not letting it run unsupervised.

Before we get started, it is not recommended that anything less than a Raspberry Pi 3 is used for this project. There have been reports of limited success using a Raspberry Pi Zero W, but only if you have no intention of using a camera to monitor your prints. If you want to try this with a Pi Zero or an older Raspberry Pi, you may experience unexpected print failures.

Firstly, you will need to download the latest version of OctoPi from the OctoPrint website – **octoprint.org/download**. OctoPi is a Raspbian distribution that comes with OctoPrint, video streaming software, and CuraEngine for slicing models on your Raspberry Pi. When this has finished downloading, unzip the file and put the resulting IMG file somewhere handy.

Next, we need to flash this image onto our microSD card. We recommend using Etcher due to its minimal UI and ease of use, plus it's also available to use on both Windows and Mac, get it here – **balena.io/etcher**. When Etcher is installed and running, you'll see the UI displayed (**Figure 1**). Simply click the Select Image button and find the IMG file we unzipped earlier. Next, put your microSD card into your computer and select it in the middle column of the Etcher interface. Finally, click Flash!, and while the image is being burned onto the card, get your WiFi router details, as you'll need them for the next step. Now that we have our operating system, you'll want to add your WiFi details so that the Raspberry Pi can automatically connect to your network after it's booted. To do this, remove the microSD card from your computer (Etcher will have 'ejected' the card after it's finished burning the image onto it) and then plug it back in again. Navigate to the microSD card on your computer – it should now be called '**boot**' –

and open the file called **octopi-wpa-supplicant.txt**. Editing this file using WordPad or TextEdit can cause formatting issues; we recommend using Notepad++ to update this file (**hsmag.cc/SjuXIP**), but there are instructions within the file itself to mitigate formatting issues if you do choose to use another text editor. Find the section that begins `## WPA/WPA2 secured` and remove the hash signs from the four lines below this one to uncomment them. Finally, replace the SSID value and the PSK value with the name and password for your WiFi network respectively (keeping the quotation marks). See the example below for how this should look.

```
## WPA/WPA2 secured
network={
  ssid="MyNetworkName"
  psk="myPassword123"
}
```
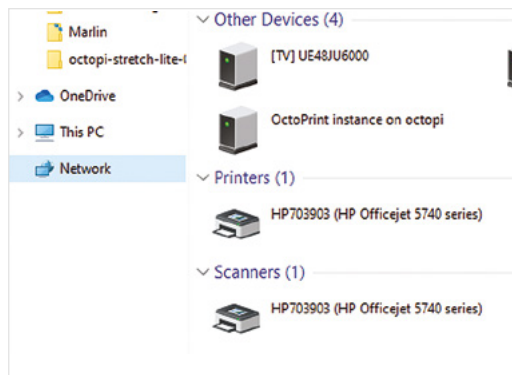
Further down the file, there is a section for what country you are in. If you are using OctoPrint in the UK, leave this as is (by default, the UK is selected). However, if you wish to change this, simply comment the UK line again by adding a # before it, and uncommenting whichever country you are setting up OctoPrint in. The example below shows how the file will look if you are setting this up for use in the US.

```
# Uncomment the country your Pi is in to activate
Wifi in RaspberryPi 3 B+ and above
# For full list see: https://en.wikipedia.org/
wiki/ISO_3166-1_alpha-2
#country=GB # United Kingdom
#country=CA # Canada                              →
```

TROUBLESHOOTING

Not seeing that OctoPrint splash screen? Fear not, you are not the first. While a full list of issues is beyond the scope of this article, common issues include: double-checking your WiFi details are entered correctly in the **octopi-wpa-supplicant.txt** file, ensuring your Raspberry Pi is working correctly (plug the Raspberry Pi into a monitor and watch what happens during boot), or your Raspberry Pi may be out of range of your WiFi router. There's a detailed list of troubleshooting suggestions on the OctoPrint website here: **hsmag.cc/4nPSfL**.

QUICK TIP

You can actually access OctoPrint from outside your network using a choice of plug-ins, which you can find here: **hsmag.cc/XVISFS**
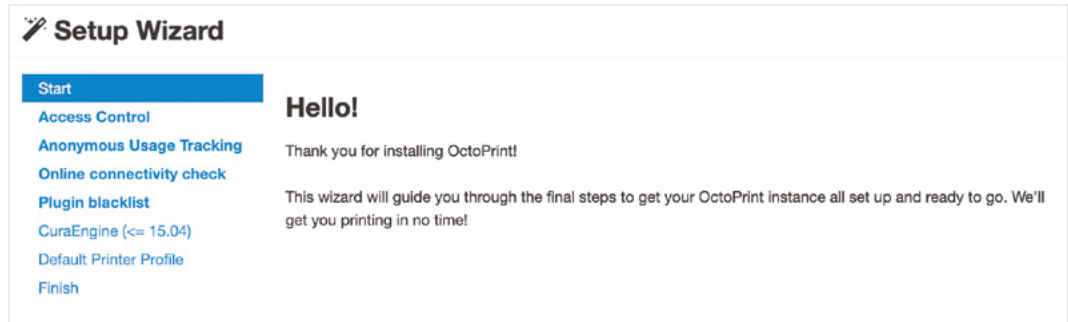
```
#country=DE # Germany
#country=FR # France
country=US # United States
```

When the changes have been made, save the file and then eject/unmount and remove the microSD card from your computer and put it into your Raspberry Pi. Plug the power supply in, and go and make a cup of tea while it boots up for the first time (this may take around ten minutes). Make sure the Raspberry Pi is running as expected (the green status LED is flashing intermittently). If using a Mac, visit **octopi.local** in your browser of choice. If you're using Windows, you can find OctoPrint by clicking on the Network tab in the sidebar (see **Figure 2**, previous page). It should
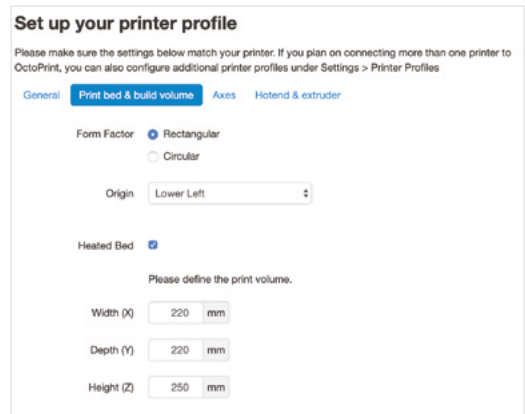
Figure 3
The 'Welcome to OctoPrint' page; the hard part is over

Figure 4
Printer profile for the Creality Ender-3

> **Plug the power supply in,**
> and go and make a cup of tea while it boots up

be called **OctoPrint instance on octopi** – double-clicking on this will open the OctoPrint dashboard in your browser.

If you see the screen shown in **Figure 3**, then congratulations! You have set up OctoPrint. If not, check out our Troubleshooting box, which will hopefully put you on the right track to getting up and running. We now have the opportunity to set up OctoPrint for our printer using the handy wizard. Most of this is very straightforward – setting up a password, signing up to send anonymous usage stats, etc., but there are a few sections which require a little more thought. We recommend enabling the connectivity check and the plug-ins blacklist to help keep things nice and stable. If you plan on using

OctoPrint as your slicer as well as a monitoring tool, then you can use this step to import a Cura profile. However, we recommend skipping this step as it's much quicker (and you can use a slicer of your choice) to slice the model on your computer, and then send the finished G-code over.

Finally, we need to put in our printer details. We've included some of the specs of the Creality Ender-3 (**Figure 4**) as an example. If you can't find the exact details of your printer, a quick search on Google should throw up what you need for this section. The 'General' tab can have anything in it, it's just an identifier for your own use. 'Print bed and build volume' should be easy to find out – if not, you can measure your print bed and find out the position

QUICK TIP

Make sure not to accidentally add whitespace on to the start or end of your WiFi network name or password.

OUR FAVOURITE OCTOPRINT PLUG-IN

OctoPrint has scores of community-created plug-ins (**hsmag.cc/XVISFS**), but our favourite, Octolapse, makes beautiful hypnotic time-lapses. What makes them so special is that the plug-in alters the G-code of whatever object you are printing so that once each layer has finished, the extruder moves away from the print to let the camera take an unobstructed shot of the model. The result is an object that seems to grow out of the build plate as if by magic. You'll not find a finer example of it than here: **hsmag.cc/BW6faQ**.

TAKING **PICTURES**

Undoubtedly the easiest way to set up video monitoring of your prints is to use the official Raspberry Pi Camera Module. There are dozens of awesome mounts on Thingiverse for a Raspberry Pi Camera Module, to allow you to get the best angle of your models as they print. There are also some awesome OctoPrint-themed Raspberry Pi cases to house your new printer brains. While it isn't officially supported by OctoPrint, you can use a USB webcam instead if you have one handy, or just want some very high-quality video streams. The OctoPrint wiki has a crowdsourced list of webcams known to work, as well as a link for the extra steps needed to get the webcam working correctly (**hsmag.cc/aREXbo**).
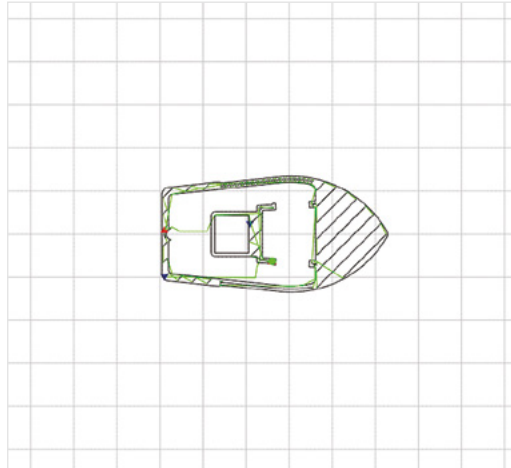
Figure 6
Use the G-code Viewer to see a preview of each layer of your model before you print it

of the origin by looking at your Cura printer profile. Leave 'Axes' as default; for the 'Hotend and extruder' section, defaults are almost certainly fine here (unless you've specifically changed your nozzle; 0.4 is usually the default diameter for most consumer printers).

OCTOPRINT IS BETTER WITH A CAMERA!

Now that you're set up with OctoPrint, you're ready to start printing. Turn off your Raspberry Pi, then plug it into your 3D printer. After it has booted up, open OctoPrint again in your browser and take your

newly WiFi-enabled printer for a spin by clicking the Connect button. After it has connected, you'll be able to set the hot end and bed temperature, then watch as the real-time readings are updated (**Figure 5**). In the Control tab, we can see the camera stream (if you're using one) and the motor controls, as well as commands to home the axes. There's a G-code file viewer to look through a cross-section of the currently loaded model, and a terminal to send custom G-code commands to your printer. The last tab is for making time-lapses; however, there is a plug-in available to help with this process (see Our Favourite OctoPrint Plug-in box).

As mentioned earlier, our recommended way of printing a model using OctoPrint is to first use your slicer as you would if you were creating a file to save to a microSD card. Once you have the file, save it somewhere handy on your computer, and open the OctoPrint interface. On the bottom left of the screen you will see the Upload File button – click this and upload the G-code you wish to print. You'll see the file/print details appear, including information on how long it'll take for the object to print. Before you kick things off, check out the G-code Viewer tab on the right. You can not only scroll through the layers of the object, but, using the slider at the bottom, you can see the exact pattern the 3D printer will use to 'draw' each layer (**Figure 6**). Now click Print and watch your printer jump into action! □
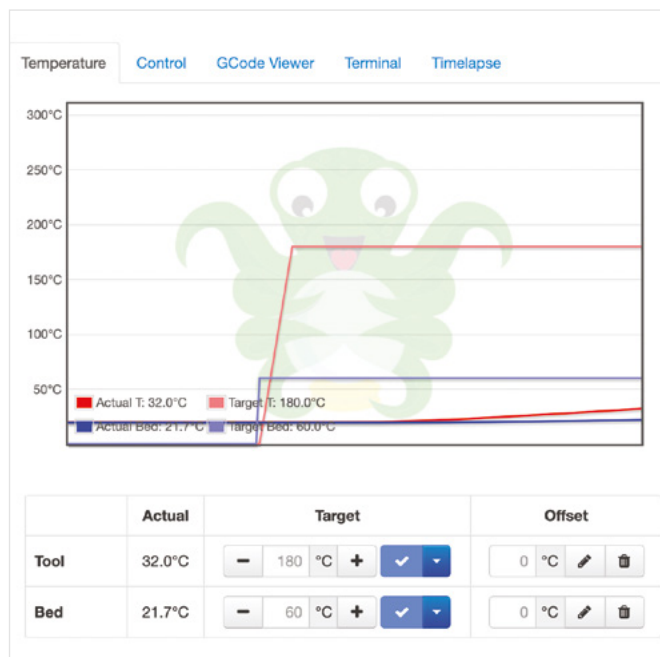
Figure 5
Visualising the temperature on a graph like this can highlight the need to calibrate your PID values, as well as other heater issues

Logging data with Arduino

Save information to an SD card

Demetrio Pinna

System Administrator by profession and hobbyist maker

YOU'LL NEED

- **Arduino Uno R3**
- **Breadboard**
- **Linear Rotation Potentiometer 250 kΩ**
- **Liquid-crystal display (1602A) 16 × 2 (blue)**
- **Resistor 220 Ω**
- **Red LED diode 5 mm**
- **Passive infrared sensor HC-SR501**
- **Module DS3231 RTC (real-time clock) chip + AT24C32 EEPROM chip – I²C**
- **Module SDHC microSD card adapter LVC125A – 6-pin SPI**
- **2GB microSD**

Above
The final project, logging movements to the microSD card
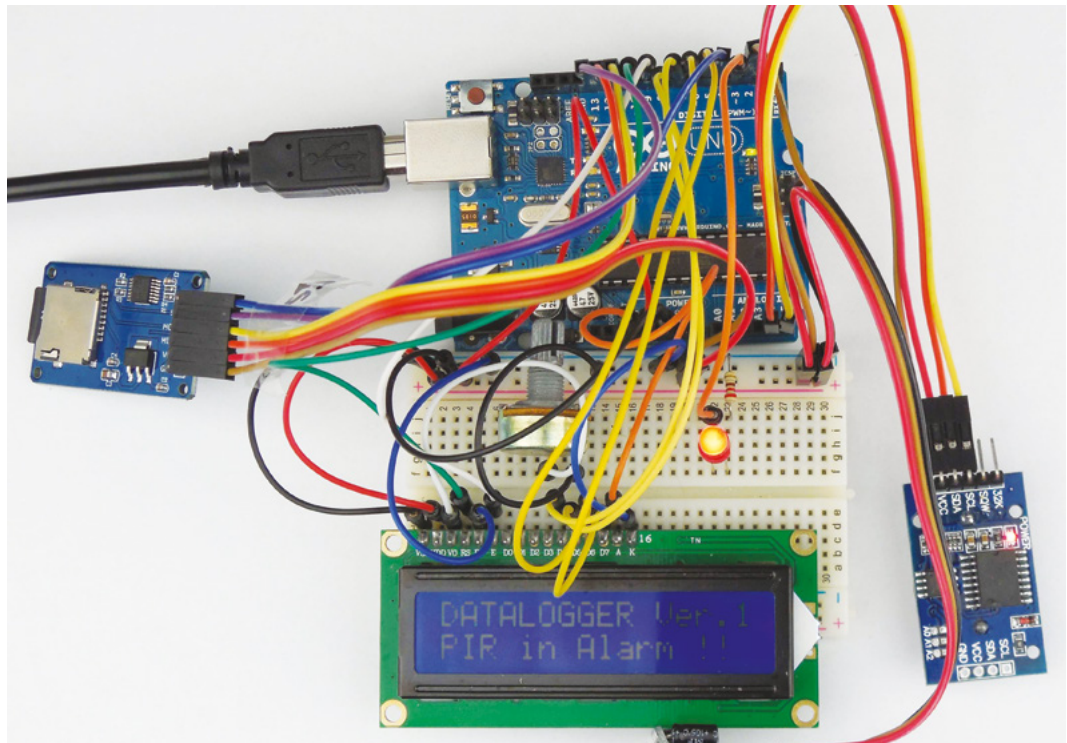
Through the use of Arduino, we can work with several useful sensors to constantly obtain lots of information about the environment in which we find ourselves: brightness, temperature, degree of humidity, position, magnetic field, presence detectors, gas detectors, and many others.

The world around us is an immense sea of information which, thanks to current technology, we can capture, monitor, and share in real-time. But there is a problem: we can't always stay for hours or days in front of a display or an electronic device to constantly see all the data taken. It is often essential that the data is stored automatically, and then analysed more carefully at a later time.

This solution is made possible by the data logger, a device which stores one or more files of the data detected by the sensors. These files are known as log files.

In our case, we will experiment with the use of the data logger by connecting a passive infrared (PIR) sensor, an LED, an LCD, a real-time clock (RTC), and finally a microSD card adapter, with an Arduino Uno R3 board. The operation is very simple: every time the PIR sensor detects the passage of an individual within its range of action, the Arduino provides a visual signal by turning on a red LED and simultaneously storing the date and time of the event in the log file (**ArduLog.txt**), for example: 'PIR in ALARM !! -> [17:30:05 17/10/2019]'. Date and time are taken from the RTC module, while the log file is stored on a microSD card present within the adapter. In addition, the various messages concerning the

operation of the PIR sensor can be displayed on the LCD and in the Serial Monitor. By consulting the log file, we can know at what times of the day our sensor has detected presence. To view this information, simply connect the microSD card to a PC and then access the text file stored on it. Or, alternatively, the log file is available directly in the Serial Monitor. It is important to remember that the microSD must be formatted using the FAT16/FAT32 format.

> **"** PIR sensors respond to heat, which means they **can also be triggered by animals "**

Let's take a look at the hardware we'll be using. The display used in this circuit is the LCD 1602A – a blue display consisting of 16 columns and two rows.

A linear-rotation potentiometer is also connected to this LCD which is used as a brightness regulator. The connections for this are given in the table below.

LCD	Wire	Arduino UNO R3
(01) VSS	Black	GND
(02) VDD	Red	5 V
(03) V0	White	RP (central pin)
(04) RS	Green	10
(05) RW	Blue	GND
(06) E	White	9
(07) D0	-------	-------
(08) D1	-------	-------
(09) D2	-------	-------
(10) D3	-------	-------
(11) D4	Yellow	8
(12) D5	Yellow	7
(13) D6	Yellow	6
(14) D7	Yellow	5
(15) A	Orange	3.3 V
(16) K	Blue	GND

The passive infrared motion sensor used is an HC-SR501. PIR sensors respond to heat, which means they can also be triggered by animals such as cats and dogs, as well as people and other heat sources.
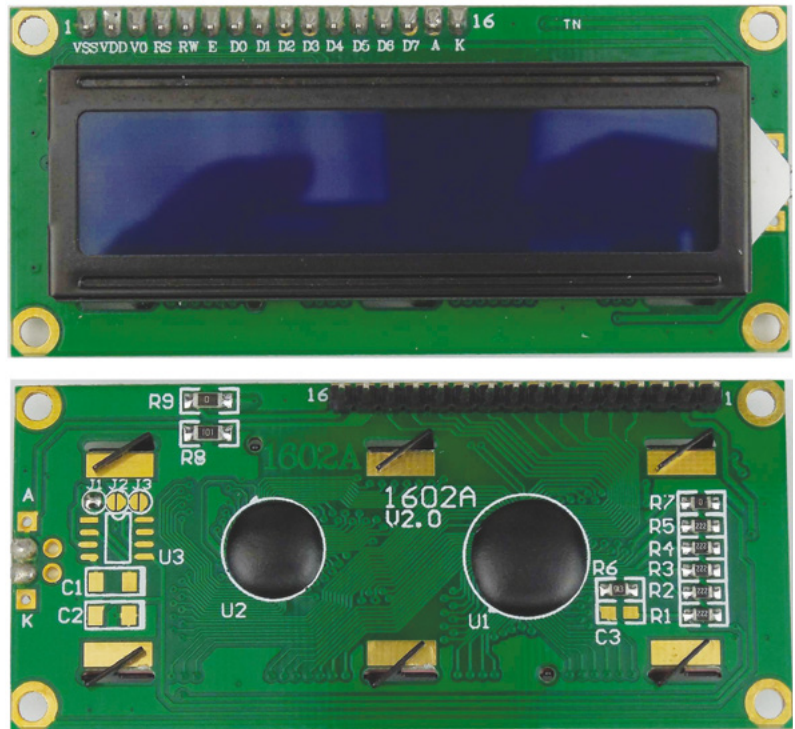
This is connected with VCC going to 5 V, Out going to pin 2 on the Arduino, and (fairly predictably) ground going to ground.

The red LED used in the circuit will light up every time the PIR sensor is activated. To connect it correctly, let's remember that the longest terminal is the anode (positive), while the shorter is the cathode (negative). To avoid inappropriate currents which can burn it, we will connect it with a 220 Ω resistor.

The RTC module used contains the DS3231 and AT24C32 chips. With the AT24C32 EEPROM chip, it's possible to keep the date and time settings →

Above ◆
You can get two lines of text on this LCD – that's plenty for this project

Below ◪
Real-time clocks – like a watch but for your microcontroller
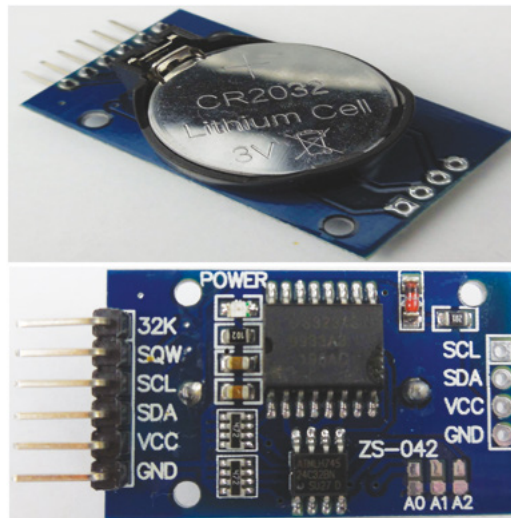
(maximum 32 bytes) even when the Arduino is switched off. The DS3231 also keeps the calendar up to date and counts the time. In trade, you often find this module with the DS1307 chip. Out of the two, the best is the DS3231 because it's much more precise and can manage leap compensation until the year 2100. Furthermore, the DS3231 also incorporates a precise digital temperature sensor. Everything is manageable via the I²C interface. SCL is connected to A5 on the Arduino, and SDA goes to A4. 5 V and GND are also connected to their respective pins on the Arduino.

Using the SPI bus, it is possible to store the log file with the PIR sensor activity on the microSD card.

Above ◩
You'll need to connect your circuit like this before programming your Arduino

Before proceeding with the assembly, make sure that the Arduino is disconnected from the computer and therefore not powered. It will be powered via USB cable only after all components are connected properly.

ENTERING THE CODE

Now we've got the circuit setup, it's time to get coding. The microSD card can be controlled via the SD library which lets us easily append data to a file with the following code:

```
void writeSD() {
    ArduLog = SD.open("ARDULOG.txt", FILE_WRITE);
    ArduLog.print(message);
    ArduLog.println(event);
    ArduLog.close();
}
```

We use this to push two bits of data to the file: those stored in **message** and those stored in **event**. These are both written to the same line because the first line uses **print** rather than **println**, which doesn't add a new line at the end of the text.

We also use the LiquidCrystal library that prints data to the LCD in a similar manner to how we do with the microSD card.

Real-time clocks are amazing things. Once they're set up (see main code listing), you can get the actual time using the following code:

```
DateTime now = rtc.now();
```

If you keep the battery in place, it will remember the time between power-offs, and the clocks are so efficient that they can run on a single battery for years.

PIR sensors are great for detecting people since they're so easy to use. There's one pin, and it goes high when the sensor detects the movement of a hot object. That way, you can use it almost like a button, but it's triggered by movement.

To program this sketch, open the Arduino editor and enter the following code, then connect the board and upload as you would any other sketch.

```
#include <SPI.h>
#include <SD.h>

#include <Wire.h>
#include "RTClib.h"
RTC_DS3231 rtc;

#include <LiquidCrystal.h>
LiquidCrystal lcd(10, 9, 8, 7, 6, 5);
const int sdCard = 4;
const int ledPin = 3;
```

```
const int pirPin = 2;
char event[100];
String message;
File ArduLog;

void setup() {
    Serial.begin(9600);
    SD.begin(sdCard);
    rtc.begin();
    rtc.adjust(DateTime(F(__DATE__),F(__
TIME__)));
    Wire.begin();
    pinMode(ledPin, OUTPUT);
    pinMode(pirPin, INPUT);
    lcd.begin(16, 2);

    message = "DATALOGGER Ver.1";
    lcd.setCursor(0, 0);
    lcd.print(message);
    Serial.println(message);
    digitalWrite(pirPin, LOW);
    delay(4000);

    Serial.println();
    message = "PIR is Scanning";
    printMessage();
}

void loop() {
    int val = digitalRead(pirPin);
    if (val == HIGH)
     {
        digitalWrite(ledPin, HIGH);
        message = "PIR in ALARM !!";
        printMessage();
        writeSD();
        digitalWrite(pirPin, LOW);
        delay(3000);

        digitalWrite(ledPin, LOW);
        message = "PIR is Scanning";
        printMessage();
     }
}

void printMessage() {
    DateTime now = rtc.now();
    sprintf(event, "%02d:%02d:%02d
%02d/%02d/%02d",
            now.hour(), now.minute(), now.
second(),
            now.day(), now.month(), now.
year());
```

```
    lcd.setCursor(0, 1);
    lcd.print(message);
    Serial.print(message);
    Serial.println(event);
}

void writeSD() {
    ArduLog = SD.open("ARDULOG.txt", FILE_WRITE);
    ArduLog.print(message);
    ArduLog.println(event);
    ArduLog.close();
}
```

Once it's assembled, open the IDE, load the sketch into the IDE, connect the Arduino to your computer via the USB cable, and finally load the sketch via the Upload button. If the assembly was done correctly, the circuit will start to work immediately. In addition to the LCD, remember to take a look at the Serial Monitor where you'll see the data feed through.

After letting it run for a while, you can power down the Arduino, then take the microSD card out. Pop it into a computer and you can read the data in the **ArduLog.txt** file. In this example, it just show the times when the PIR sensor was triggered, but you could use the same concept for storing almost any information, from sensor data that you want to investigate later, to diagnostics that will help you understand what's happened if something goes wrong. ▢

OPENLOG

If you're looking for a more drop-in solution, the OpenLog by SparkFun has a serial connection and simply logs any data sent over serial. This works with microcontrollers such as Arduino and lets you use `serial.write()` to send data. You can set various parameters by creating a **config.txt** file (for details see documentation).

Below ◈
A single pin going high or low lets us know if there's been any movement

BOOK OF MAKING
VOLUME 2

50 PROJECTS FOR MAKERS & HACKERS
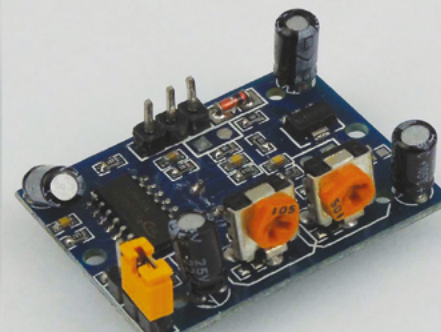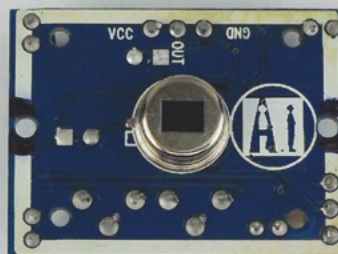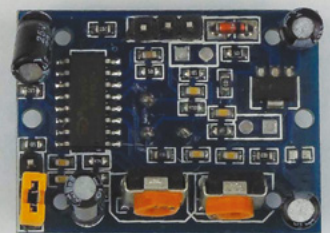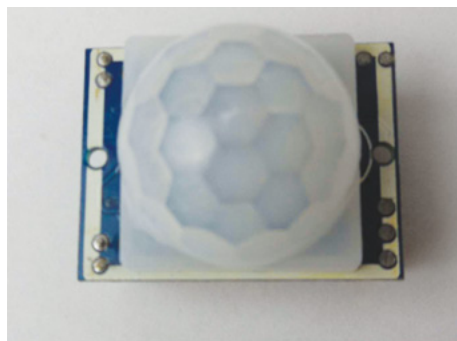
ONLY £10

WHSmith
BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

BOOK OF MAKING
VOLUME 2

MAKE ELECTRONICS CIRCUITS WITH PLAY-DOH

TAKE TO THE SKIES WITH YOUR OWN ROCKET

PLAY GAMES WITH A CUSTOMISED CONTROLLER

CREATE MUSIC ON A HOMEMADE SYNTHESIZER

FROM THE MAKERS OF HackSpace MAGAZINE

THE BEST PROJECTS FROM
HACKSPACE MAGAZINE
THE ULTIMATE SKILLS, TRICKS, AND MAKES

AVAILABLE NOW
hsmag.cc/store
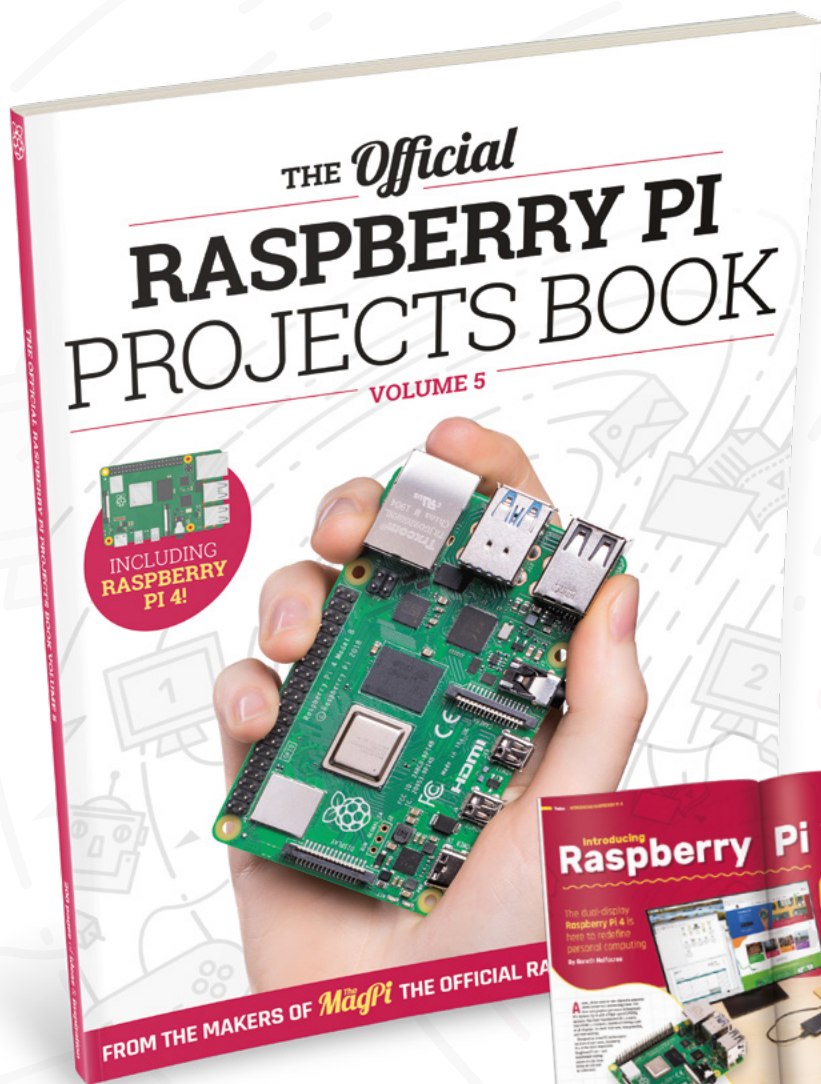
FROM THE MAKERS OF HackSpace MAGAZINE

OpenWeather desk display

An unassuming connected display

Marc de Vinck

🐦 @devinck

Marc de Vinck founder
of Panoptic Labs, llc –
panopticlabs.com

Above ↗
The weather's
looking good today!

Right ↘
You should find
a weather station
near you

Below ◆
You'll need
to create a
username to use
OpenWeather

Everything is
connected! And to
be honest, we find
a lot of connected
things useful, but
we're also bombarded
by LCDs and LEDs everywhere.
We like analogue, and we like the
Internet of Things, and that's where
this project comes into play. It's a pseudo-analogue
connected device that gathers the current weather
for any given location and shows you, in a very not-
so-digital way, the results. No LEDs, no LCDs, just a
simple, passive, wooden display.

The real point of this tutorial is to show you just how
easy it is to connect a microcontroller to the internet
and gather data. In this case, we'll be looking at the
OpenWeather API. They have a free tier, which is
great, and the structure of their data is really easy to
understand and implement. Think of this project as
a starting point for connecting the virtual world with
the physical. What you do next is up to you. Let's
get started!

STEP 1 GETTING ALL YOUR ACCOUNTS READY

Let's start with getting all our credentials ready for
the code. You will need three things to get your
microcontroller online. First, you need your WiFi
network name. Simple. Next, your WiFi password,
which hopefully you can remember. And lastly, you will
need access to the OpenWeather API by generating
what is called a 'key'. It's easy.

Start by going to **OpenWeatherMap.org** and create
a free account.

Once you are signed in, click on API Keys.
There you can generate a specific key for your
project. In our case, we created a new one called
'HackSpace Project'. You can copy the API key for
later – we will need it once we start programming our
Feather HUZZAH.

The free OpenWeather account
allows you to check a location for the current
weather up to 60 times per minute. We're only going
to check once every minute, so the free plan works
perfectly. If you decide to get a paid account, you can
easily modify the code to gather the future forecast
for a location and have your display tell you things like
'Bring an umbrella', or let you know snow is on the
way. But for now, let's stick with the free account and
just request the local conditions.

STEP 2 KNOWING WHERE YOU ARE IN THE WORLD

You will also need to know the location of where
you want to request the current weather conditions.
There are a few ways to accomplish this, but we
found looking up the closest weather station to be
the easiest.

Head over to **hsmag.cc/BHcs8a** and scroll to your
location. Make sure the 'CITIES' option is checked.
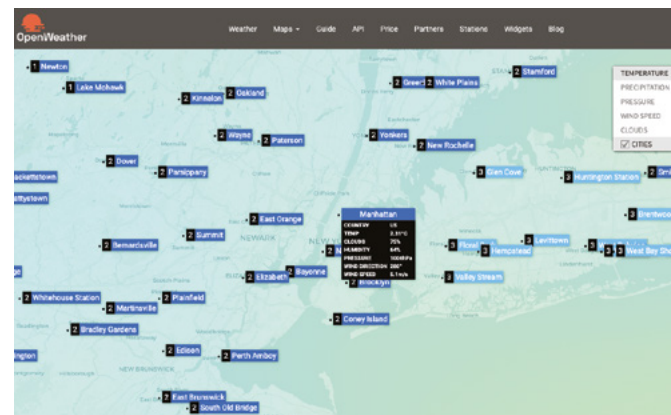
As you zoom in, you will find the local names you can use. In our case, we can use the name 'New York, US', or we can be a bit more specific and use 'Manhattan, US'. Click on the name, and you will see the country code listed. And don't worry if your specific location isn't listed: there is always a location that's close enough to get local weather.

STEP 3 BUILD THE CIRCUIT

Building the circuit couldn't be easier. We chose the Adafruit Feather HUZZAH because it has all the components we need to get connected to WiFi, and it can easily control a servo. Many other Arduino variants would work, as would many ESP8266 or ESP32 boards, but the Feather HUZZAH is our go-to board because it's so simple, and it even has options for battery power, allowing for portability.

The Feather has through-hole header pins already installed, so we simply put it into a breadboard to make it a bit more stable (we don't actually use the breadboard for anything other than some structure).

Hook up the servo to the Feather HUZZAH by attaching three male jumper wires to the servo wire harness. The brown wire gets connected to ground, the Red wire to 3.3 V, and the yellow wire to Pin 2. That's it!

STEP 4 BUILD THE PANEL AND INDICATOR

We used a laser cutter to cut out the board and the indicator. After they were cut, we used the laser cutter to engrave the weather symbols on the board. You could easily cut all this out by hand and paint or draw the symbol on the board. Use what you have – we have a laser.

STEP 5 ASSEMBLE THE WEATHER DISPLAY

We're not 100% sure what we'll add to this project in the future; we just know that we'll be adding features. Because of this, we didn't make a specific enclosure, and just used hot glue to assemble everything. This allows us to take it all apart really easily and modify it in the future.

We designed the display so the servo could be hot-glued to the breadboard and the display panel. A little dab of glue on the servo holds it in place. We then follow up with more glue to add extra support.

The last step is to add the indicator. We used hot glue, but eventually, we decided to drill a hole and screw the indicator onto the servo. It made it much more sturdy. That's it. Time for code!

STEP 6 CODE EXPLANATION

This is not an in-depth look at code, but rather a quick overview of a few specific blocks of code. We hope

Above
Laser cutters are quick and easy, but you can also use hand tools

that with this explanation, and comments in the code, you'll be able to modify it to make you own IoT display. We've based some of this code on Miles Nash's WeatherBox2.0 code – **hsmag.cc/551PE1**. Thanks Miles!

```
#include <Servo.h>
#include <ArduinoJson.h>
#include <WiFiClient.h>
#include <ESP8266WiFi.h>

Servo stateServo;
WiFiClient client;

const char* ssid     = "YOUR WIFI NETWORK NAME";
// Your WiFi SSID name
const char* password = "YOUR WIFI PASSWORD";
// Your WiFi password
const String APIKEY = "YOU API KEY FROM
OPENWEATHER";    // API Key from OpenWeather.org
int PingInterval = 60000;
// The amount of time in milliseconds between API
call requests (expressed in milliseconds)

//JSON Variables
String dt_txt0, dt_txt1, dt_txt2, dt_txt3, dt_txt4,
dt_txt5, dt_txt6, dt_txt7, dt_txt8, dt_txt9,
dt_txt10, dt_txt11, dt_txt12, dt_txt13, dt_txt14,
dt_txt15, dt_txt16, dt_txt17;
String currentWeather;

const String currentLocation = "Manhattan, US"; //
Enter Your location
```

The first part of the code loads up libraries to help connect the Feather HUZZAH to the network. It also gets the board ready for using a servo and declares →

YOU'LL NEED

- **Adafruit Feather HUZZAH** (or any ESP8266 Arduino-compatible board)
- **Servo** (any size)
- **3 × jumper wires**
- **Half-size breadboard**
- **Scrap plywood**
- **Micro USB cable and 5 V power supply**
- **Computer running Arduino IDE** (free download at arduino.cc)
- **Laser cutter** (or you can cut the wood by hand)

some JSON variables. We use JSON later to interpret the text.

```
void setup() {
// Let's get started!
  Serial.begin(115200);
  stateServo.attach(2);
// Servo attached to pin 2
  delay(50);
// Take a little break
  stateServo.write(10);     //Home position
  WiFi.begin(ssid, password);
// Start the WiFi connection
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected!");
  Serial.println(WiFi.localIP());
  Serial.println();
  delay(500);
// Take another little break
  stateServo.write(0);
// Initial position
}
```

Next is the setup. Here, we start our serial communications and connect the Feather HUZZAH to the local WiFi network. Not much else. It's fairly simple. To see what's happening, open up the serial monitor. It will display all the steps as they happen.

Below ◈
The large display keeps the electronics out of sight
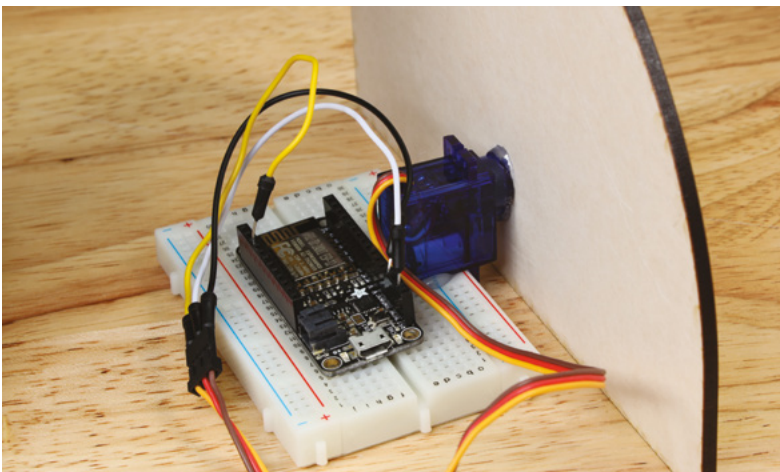

```
void loop() {
  String forecast = getWeather();
  parseWeather(forecast);
  Serial.println("Weather updated! Connection
closed.");

  Serial.println();
  delay(PingInterval);
// Maximum of 60 requests per minute on the free
plan
}
```

Next is the loop. This repeats over and over. Basically, it starts by running the `getWeather` code which connects to the OpenWeather API. It then gathers the current weather based on the location we defined, then it figures out what was returned from OpenWeather and waits for a set period of time, in our case 60 seconds, before it does it all again.

```
String getWeather() {
// Grab the weather data from OpenWeather

  Serial.println("Connecting to OpenWeather
API...");

  if (client.connect("api.openweathermap.org", 80))
{
    Serial.println("Requesting data...");
    client.print("GET /data/2.5/forecast?");
    client.print("q=" + currentLocation);
    client.print("&appid=" + APIKEY);
    client.print("&cnt=20");
    client.println("&units=imperial");
    Serial.println("Receiving data...");
    client.println("Host: api.openweathermap.org");
    client.println("Connection: close");
    client.println();
  }
  else {
    Serial.println("Unable to connect");
  }
  String line = "";

  //read, store, and return the forecast JSON
string

  while (client.connected()) {
    line = client.readStringUntil('\n');
  }
  return line;
}
```

After the loop, you will find the `getWeather` string. This is called in the first line of the loop and returns the

complete current weather conditions. It's actually more information than we need.

```
void parseWeather(String json) {
  //parses forecast weather JSON string and updates
global variables

  const size_t bufferSize = 20 * JSON_ARRAY_SIZE(1)
+ JSON_ARRAY_SIZE(20) + 16 * JSON_OBJECT_SIZE(0) +
55 * JSON_OBJECT_SIZE(1) + 21 * JSON_OBJECT_SIZE(2)
+ 20 * JSON_OBJECT_SIZE(4) + 2 * JSON_OBJECT_
SIZE(5) + 2 * JSON_OBJECT_SIZE(7) + 25 * JSON_
OBJECT_SIZE(8) + 13 * JSON_OBJECT_SIZE(9) + 6460;

  DynamicJsonBuffer jsonBuffer(bufferSize);
  JsonObject& root = jsonBuffer.parseObject(json);

  if (!root.success()) {
    Serial.println("parseObject() failed");
    return;
  }

  const char* d = root["list"][1]["weather"][0]
["main"]; // this can be a "description" too see --
https://openweathermap.org/weather-conditions

  currentWeather = d;
  Serial.println("The current weather in " +
(currentLocation) + " = " + currentWeather);
```

After the **getWeather** string, we have the **parseWeather** code. This code looks at the returned data, or list, and finds the 'weather' description called 'main'. There are several different 'main' weather categories. You can also change the code to look for 'description', which would give you even more specific details, but in my case the 'main' weather info is fine.

```
  if (currentWeather == "Clouds") {  // case
sensitive!
    Serial.println ("It's cloudy out now!");
    stateServo.write(123);
    delay (1000);
  }
  else if (currentWeather == "Rain") {
    Serial.println ("It's Raining out!");
    stateServo.write(100);
  }
  else if (currentWeather == "Drizzle") {   //
Drizzle is considered rain too
    Serial.println ("It's Raining out!");
    stateServo.write(100);
  }
  else if (currentWeather == "Clear") {
```

```
    Serial.println ("It's Clear out!");
    stateServo.write(78);
  }
  else if (currentWeather == "Snow") {
    Serial.println ("Yea! It's Snowing out!");
    stateServo.write(53);
  }
  else if (currentWeather == "Thunderstorm") {
    Serial.println ("It's Thundering out!");
    stateServo.write(25);
  }
  else {
    Serial.println ("I have no idea what's
happening outside!");
    stateServo.write(0);
  }
}
```

The last bit of code is certainly not as efficient as it could be, but we wrote it to be easily understood by even a beginner. Basically, it is looking for a keyword and if it is found, it rotates the servo to a specific value. If the value isn't found, it keeps looking. Eventually it finds the matching keyword and rotates the servo to a specific location that indicates the 'main' status. It's a very simple block of if/else code.

And that's it! So simple, and fun. No LEDs, no LCDs, but very useful. We've been using it for a few days now, and it already alerted me to some rain starting which prompted this author to quickly grab the dogs and get them out for a quick walk before it started raining harder. It worked! □

QUICK TIP

Head over to **openweathermap. org/weather-conditions** to see how the data is structured and all the possible results.

Pick the perfect infill

Give some body to your prints

Ben Everard

🐦 @ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

When 3D printing, your slicer will put one or more shells around the object, then fill it to a particular percentage, known as the infill percentage. However, there are many different ways you can set the infill pattern. Let's take a look at some different patterns, and find out what they're good for. Before we get too deep into this subject, it's worth remembering that this is an optimisation – the default infill pattern should work well for most cases, and there's no need to obsess over it, unless you want to.

We've used PrusaSlicer for this, and most other slicers have comparable infill options (though sometimes under different names). The times given are the times to print a Benchy on a Prusa i3 MK3S, with 25% infill and 0.2 mm (quality) layer height settings. ▢

Below ◈
Infills (from left to right): Archimedean chords, Hilbert curve, octagram spiral, and concentric

FLEXIBLE PATTERNS

When printing with flexible filament, you usually have a different set of requirements because you want the infill to flex. However, depending on the particular object you're printing, you may want it to flex in different ways. These can all be used to produce squishy objects with varying degrees of squish.

Concentric time	1 h 39
Hilbert curve	1 h 49
Archimedean chords	1 h 41
Octagram spiral	1 h 46

FAST INFILL

Above ↗
from left to right:
lines and rectilinear

Long straight lines are quicker to draw than corners, and these infill designs maximise this. The result may not be as strong as some other patterns, but for prototypes, that may not be a problem. While it's only slightly faster than some others, this is quite a small model, and the time will scale up as the print gets bigger.

The difference between rectilinear (here) and grid (2D pattern) is that rectilinear goes in one direction for each layer, so it's a mesh, while grid goes in both directions on each layer, so it's a solid pattern.

Lines time	1 h 37
Rectilinear time	1 h 36

2D PATTERNS

These patterns are quick to slice, print quite well, and are strong, but with one caveat – they're stronger in some directions than others. Typically, they'll be able to take more force vertically than in any other direction. This can be a good point if you know which direction the force will be applied to your object, but they can be quite weak against perpendicular forces.

Honeycomb time 1h 54   **Grid time** 1h 38   **Triangles** 1h 40   **Stars time** 1h 40

Above
from left to right:
triangles, stars,
grid, honeycomb

Right
from left to right:
gyroid, cubic

3D PATTERNS

This isn't just a repeated pattern, but one that varies each layer to create a 3D structure. The resulting infill typically won't be as strong as a 2D pattern with vertical forces, but will be much stronger with perpendicular forces.

Perhaps the most unusual pattern here is the gyroid. It's a bit hard to describe, but the curves bend in 's' shapes in all three dimensions. It's not a pattern invented by 3D printer designers, but by nature, and it's this shape that gives butterfly wings their structure. It also has very even strength across all directions, so it's a great choice for general use if you don't know the direction of force.

Cubic time   1h 40
Gyroid time   1h 40

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

REGULAR

USB power supply

Get adjustable power output from an old phone charger

By **Ben Everard**   @ben_everard

Supplying power to your projects can be deceptively tricky.** If you just need 5V or 12V, you can probably find an old wall wart in your bag of old power supplies for gadgets you no longer use (or is it just us that keeps these?) But, if you need something outside of these two voltages, it can be a little difficult. Bench power supplies are great, but they take up bench space, and can be expensive.

We tested a USB power supply from diymore Alice1101983 Store on AliExpress for £6.93, including UK delivery. This works much like most other bench power supplies, in that you can adjust the voltage output or maximum current, and it will send out power according to these requirements. Standard USB supplies 5V, but this device can also use some 'fast charger' power

supplies to get up to 9V – this isn't the same as power distribution (PD) in USB-C, and is more common on older, micro USB phone chargers.

The power supply is able to boost the input voltage up to 30V, and can supply up to 15W of power, and a maximum of 2A.

Obviously you need a USB source that can handle the input power which, at 15W, is 3A at 5V, or 1.7A at 9V. This is more than many USB power supplies can handle, so you might find out that the limiting factor is the power source, not this power supply. We'd very strongly recommend using this with 9V 'fast charge' chargers, and you'll know that this is working properly

Above
An easy-to-use power supply in a tiny package

because the blue 'FC' LED will light. Although power can come in via USB-C, it doesn't support the USB-C power distribution modes, so even if your power source is capable of supplying more voltage, it may limit itself to 5 V.

Power comes in via USB, micro USB, USB-C, or solderable pads, and goes out via press-to-fit connectors. There are knobs to adjust the output voltage and current, and a rather confusing set of buttons to control the supply (for the most part, you won't need these, but 'Trig' can be used to manually select 'fast charge' mode).

> **Given the cheap price of this, you could use this to convert almost anything into a USB-powered device**

The voltage and current controls require a small amount of turning before they start to register a change. This is to prevent small knocks changing the output, and potentially damaging any connected electronics.

Constant current mode is where you set a maximum current for the output, and if the connected circuit attempts to draw more than this, the power supply will reduce the voltage to keep the output below the maximum current. This change in voltage might cause some odd behaviour on your circuit, but it will hopefully prevent any damage to sensitive components.

SMALL CURRENTS

On this power supply, the constant current selector going down to around 10 mA, and we found that it could protect fairly fragile components such as LEDs (we wouldn't trust this for protecting valuable equipment though). It's not particularly easy to control the current at low amperages as the step size seems to vary arbitrarily but, with a little care, it does seem to work.

There are no leads supplied, so you'll need to add your own. For many projects, this just means using ordinary wires or male jumper connectors. Perfect for a breadboard project, but if you need a particular connection type, you'll need to create a lead for it. The push-to-fit connectors are easy to use and solid, so it's easy to attach or detach almost anything.

Given the cheap price of this, you could use this to convert almost anything into a USB-powered device (in case you've lost the original cable or don't want to use batteries). No soldering required – just cut and strip the power wires, adjust this to the appropriate voltage, and connect. You'll need to mount it in such a way that the voltage selector can't be accidentally knocked.

This is a great little device for powering small circuits. It's small enough to not require permanent shelf-space, and can supply a wide enough range of voltages for many small projects. It doesn't have the features of a good bench power supply, but it's a fraction of the cost, and a fraction of the size. We'll certainly be keeping it in our toolbox, and expect to get plenty of use out of it. The big caveat is its ability to gobble up more current than the USB standard requires, so don't take it as a given that you'll be able to get the full 15 watts out of it in all cases. ◻

DIRECT FROM SHENZHEN

ONLY THE BEST

Soldering. It's easy and fun!

An inspiring collection of kits for small to large groups of future tinkerers

By **Marc de Vinck**   @devinck

For some people, jumping into soldering is as simple as picking up some parts and figuring it out. For others, a more guided approach is needed. This author has taught thousands of people how to solder over the years, and he's learned a lot about what works and more importantly, what doesn't work.

Around 2010 he designed his first learn-to-solder pin, and it was a huge success. Thousands of them were used at the weekend event. Since then, he's created at least another half-dozen pins for various events all over the world. It's been a lot of work, and a lot of fun. He's also led smaller learn-to-solder workshops with slightly larger budgets where we could spend more time and ultimately learn a bit more and have more fun. It turns out that there's not a perfect learn-to-solder kit, and different groups (and different teachers) prefer different kits.

In this Best of Breed, we're going to look at a few of our favourite kits for learning how to solder, and great kits for teaching soldering. Ultimately, which kit you choose depends on three factors. How many people? How much time? And how much of a budget? The good news is, there is a great solution for any situation. We've been in every situation, and it's always somehow worked out.

Below ◆
Try to avoid breathing in solder fumes

Day of the Geek Soldering Badge **vs** The Learn to Solder Robot Badge

Quick kits for getting started

DAY OF THE GEEK ◆ $8.95 | adafruit.com   **LEARN TO SOLDER** ◆ $3.95 | spikenzielabs.com

Although this isn't the least expensive pin on our list, and there are plenty of kits out there that aren't mentioned here, this kit is still our go-to kit when it comes to learning to solder in a large group. The Day of the Geek Soldering Badge comes with a beautiful skull PCB, pin back, and CR2032 battery holder. You'll need to pick up one CR2032 battery per pin as it's not included. Don't forget! Fortunately, they are very inexpensive when purchasing in bulk. You can usually pick them up for £0.25 each, or less.

What we really like about this kit is that it also includes a variety of LEDs allowing the user to add a little bit of customisation. You can pick red, blue, white, or any combo. Yes, this kit is a little more expensive, but yes, it's worth it. The quality of the PCB is stunning, and it never disappoints. Remember to look into bulk pricing too! □

Above ◨
Our go-to learn to solder pin

Above ◨
Show off your new skill with this geeky fashion accessory

SpikenzieLabs has a ton of great kits for learning to solder. And if you are teaching a really large group, this little robot head might be the perfect choice. It's fairly affordable, fun, and easy to solder. Unlike many other learn-to-solder kits, this one doesn't require the pin back, which is always the hardest part to solder.

Another interesting part about this pin, is the fact that some of the components are already soldered onto the PCB. OK, so soldering those components could be fun too, but they are surface mount and really small. It's a great opportunity to talk about other types of soldering process, and to showcase just how small components can get when space is tight. Be sure to check out the how-to guide with this kit. It's a good idea to print out a few, and have them handy at your learn-to-solder workshop. □

VERDICT

Day of the Geek

The most stylish way of learning to solder. Make the badge, then show it off!

10/10

Learn to Solder

Wear your bot with pride because you made it. Well, most of it!

9/10

Soldering. It's easy and fun!

—

SparkFun WeevilEye – Beginner Soldering Kit

Get started in style

SPARKFUN ◈ $9.95 | sparkfun.com

Right ◈
The WeevilEye is
a through-hole
soldering kit

VERDICT

Unusual and fun

9 /10

T he WeevilEye from SparkFun is a really well-thought-out, through-hole soldering kit designed for beginner soldering enthusiasts. And once you successfully put it together, it has a bit of interactivity. The eyes of the Weevil will glow more or less, depending on how much light is present. The instructions for the kit even include a brief how-to solder guide in the beginning, along with some fun facts about electronics as you go through the steps.

This kit comes with all of the parts you need to build your own WeevilEye, including the battery, which isn't always the case with other kits. You'll also receive an instructional kit card that will guide you through the assembly process. Just grab your iron and get soldering! ▫

SparkFun Basic Flashlight Soldering Kit

Build yourself a useful tool

SPARKFUN ◈ $4.95 | sparkfun.com

Right ◈
Nicely detailed PCB

VERDICT

A lot of thought
went into this
little PCB.

10 /10

S There are a lot of things to like about the SparkFun Basic Flashlight Soldering Kit. First, just like their WeevilEye kit, this one comes with the CR2032 battery, making a truly complete kit. Next, their PCB is labelled clearly as to what all the parts do, and where they need to be placed. And lastly, the back of the PCB has the schematic, showing how you would design this circuit.
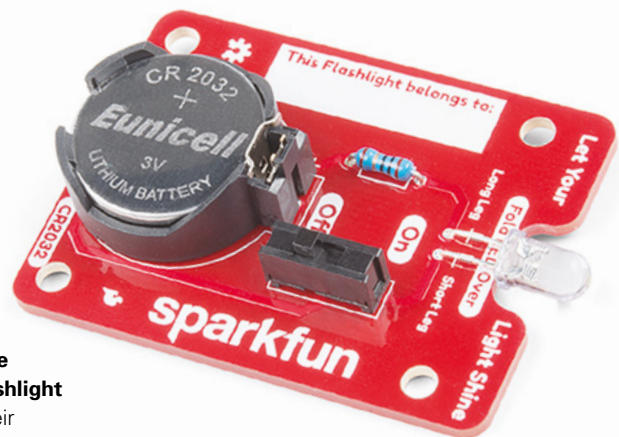
It's hard to go wrong with this affordable kit. It's also useful once completed, which makes it fun to use for a longer period of time. Be sure to check out the product page for a video how-to and really nice step-by-step instructions, with tips and tricks for beginners. ▫

SpikenzieLabs Dice Kit

Plenty of soldering to hone your skills

SPIKENZIE ◈ $19.95 | spikenzielabs.com

Not everyone is teaching a large group of people how to solder, and if it's just a few people, or even one, a little more challenging project is appropriate. We really enjoy the Dice Kit from SpikenzieLabs and have used it on a few occasions – it's always a huge hit.

Although the kit has more components, all the LEDs are the same, and it gives the people learning some extra practice. The one thing to note is the IC isn't socketed, so be sure your students have the confidence to install it properly and solder it in a timely manner. You might think this is a bit tough for a beginner, but they will have already had many resistors to practice on before they need to tackle the IC. Be sure to check out the product page for other colours available, and the complete instructions on how to build the kit. ◻
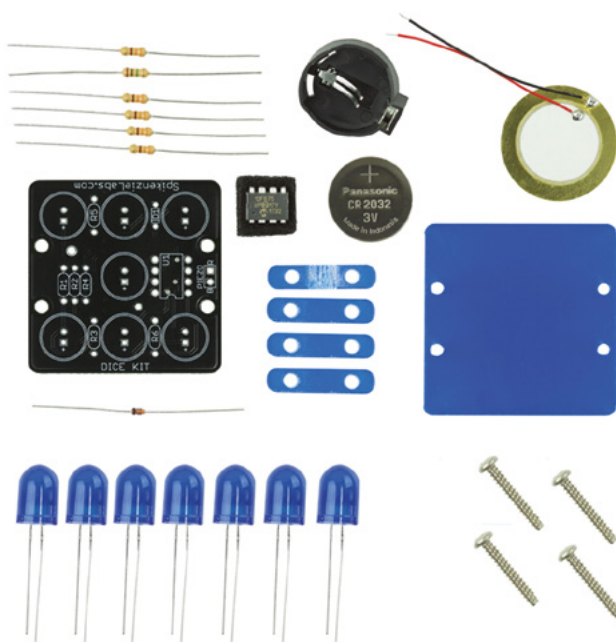
> **"** Although the kit has more components, **all the LEDs are the same** **"**

Above ◈
The completed kit ready for a roll

Left ◈
That's quite a lot of parts to practice your soldering with

VERDICT

When you have more time and budget, this is a great option.

9/10

Drawdio kit

The most unusual kit on test

DRAWDIO KIT ◆ **$17.50** | adafruit.com

Another, more complicated, kit that makes the list is Drawdio. It's not going to be the kit we would use for teaching hundreds of people in a workshop, but again, with just a few students, or even one, this makes a great choice. It's fun, easy to put together, and the end result is entertaining. We have even hacked a few kits to make other interesting musical instruments. You can see a video of '*Drawdio meets Unruly*' on the Adafruit Drawdio kit website.

But what is Drawdio? Drawdio is an electronic pencil that lets you create music as you draw! It's a basic music synthesizer that leverages the conductive properties of the pencil lead and can generate musical tones based on its resistance. It's a lot of fun to draw a piano and be able to play the notes, once you complete the circuit. It runs off a single AA battery, so no need to source a strange cell size. And if you already know how to solder, go ahead and grab this kit anyway. It's that fun! □

Above ◥
Drawing and music in one

VERDICT

Fun to build,
fun to draw!

9/10
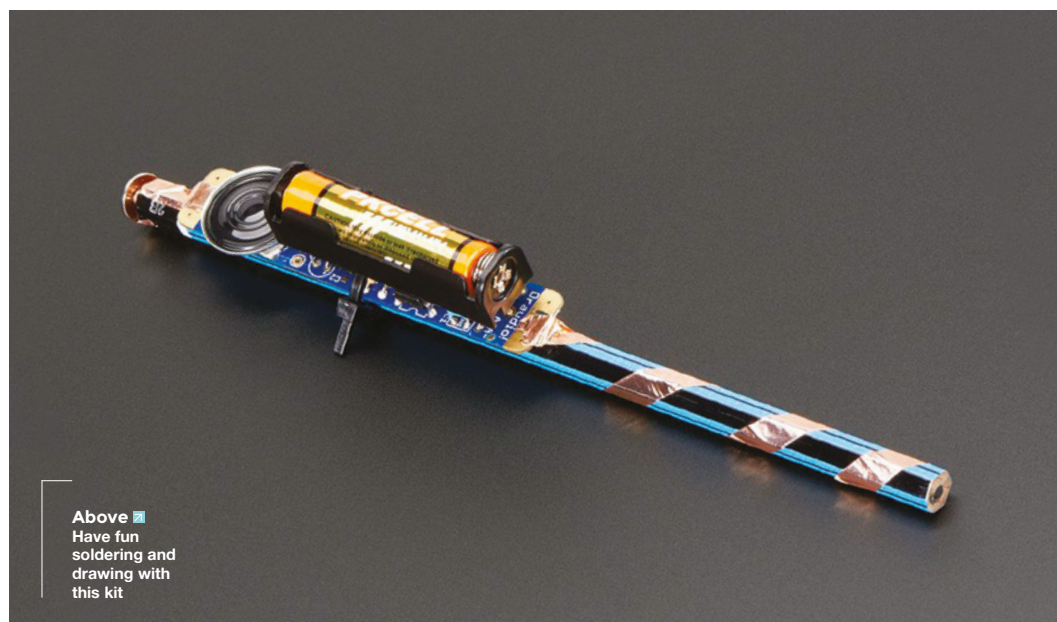
Above ◹
Have fun soldering and drawing with this kit

Can I Hack It?

An electronics lab kit?

Can we hack a child's electronics lab kit?

Les Pounder

🐦 @biglesp

Les Pounder loves taking things to pieces and seeing how they work. He teaches others how to be makers and tinkerers at events across the UK. He blogs at **bigl.es**

Chances are that if you are reading HackSpace magazine, then you may have some electronics knowledge, and from that group of people, a large number will have learnt the basics from a kit like this typically bought from Tandy/Radio Shack. This author did – back in the 1980s he had an electronics kit from Argos, which included a crystal radio and made lots of noise! These kits gave us the components to build something fun while secretly learning new skills. So when we happened upon a modern version of these kits, we decided it was time to take one apart and see just how it is built.

Made from a formed thin plastic, the electronics lab can be hacked with hand tools. If using a power drill or rotary tool to cut the plastic, take care as it will crack if too much stress is placed upon it. The top and the bottom of the lab see sections of card used to hide the components from view. The card sections are held in place with small cross-head screws.

On the top left of the lab is a spinning light which is built from a kit in the box. It contains a DC motor with a worm screw attachment and a bright white LED. Coloured reflectors shine the light from the LED as they spin around. At the top of the board are two sections, one for batteries and another for a speaker. Both of these sections can slide out for removal or replacement. Just be careful with the speaker as the wires are attached!

This electronics lab kit is designed for children to create simple projects, and as

such, we see a plethora of components such as resistors, reed switch, seven-segment display, LEDs, transistors, speaker, etc. The selection of components is plenty for those starting with electronics. They are held in place on the card, and this reduces the chance of them being lost or ingested.

So how can we make something using this electronics lab kit? There are a series of springs around the board, and these springs are connected to the components under the card layer using a mixture of friction fit, solder, and hot glue. By leaning the spring to one side, we can insert a single-core jumper wire into the spring and use that to connect the components, as per the instruction manual which comes with 50 experiments.

Power is supplied by 2 × AA batteries, providing a nice, safe 3V for experiments including the motor and LEDs. If using rechargeable batteries, note that the rotating LEDs will not work if used with the motor!

MEMORIES OF YOUTH

The Tronex Circuit Lab is a modern version of the electronics lab which many readers growing up in the 1970s–1990s will have used to learn their skills. But do we need kits like this now? What with access to boards such as Raspberry Pi, micro:bit, Arduino. etc. can we just teach kids via those boards? The answer is yes we can; but then we are mixing coding and electronics into one lesson. By separating the two into their own lessons, we can learn how electronics work – the theory behind how a circuit works is much simpler with an LED, a resistor, some wires, and a battery. Once we grasp the basics, and we do mean just a few key concepts and not a full curriculum, we can move on to create exciting and expansive projects where code controls the GPIO and enables us to take our ideas further, faster!

In conclusion, learning to light an LED is truly a 'light bulb' moment!

HACKABILITY

This is so hackable! OK, this might not be something to add to your cosplay, or provide massive 'bang for your bucks' project enhancement. But what we can do with this kit is add an Arduino, micro:bit, or Raspberry Pi and use a little coding to enhance the learning offered by the board.

For example, we connected up a micro:bit using 4tronix's Bit:170 breakout board. This gave us access to all the GPIO pins of the micro:bit using a breadboard format. The jumper wires which came with the electronics kit fit perfectly into Bit:170, and with some MicroPython code written using the excellent Mu editor, we managed to flash the large, yellow LED. With that test successful, we moved on to control the motor via an NPN transistor whose collector and emitter was connected to the VCC side of the battery and the + terminal of the motor. By turning the base
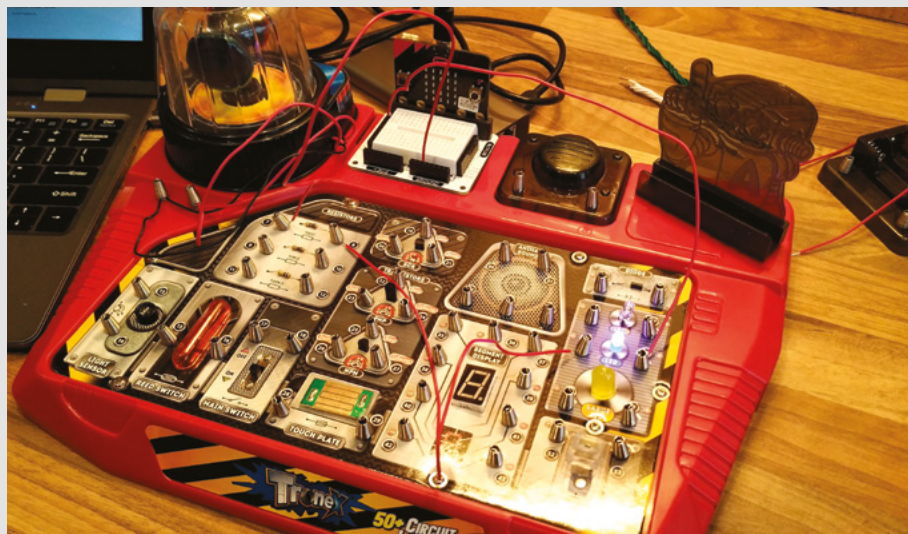
pin on via a resistor connected to a pin on the micro:bit, we could start and stop the motor with ease.

Using the electronics lab kit with micro:bit was very straightforward thanks to Bit:170; in fact, the board fitted neatly into the space where the batteries were housed.

Can we hack this any more? Yes! We can add our own components, requiring only that we carefully make space for the parts, and add springs which can be found cheaply online. So, we could add a NeoPixel LED to replace the RGB flashing LED, an OLED screen, or simple components such as capacitors. We can add anything we desire!

If we wish to take the hacks even further, we could integrate a Raspberry Pi Zero in USB gadget mode into the kit, and then code the GPIO and electronics via another computer.

The purpose of this kit is to inspire the next generation of makers, hackers, and tinkerers to learn a new skill and make cool projects. On its own, it can do this, but by adding a microcontroller/microcomputer we can take this to the next level. Learning to code and learning electronics in one simple package is now possible for less than £30! □

Above
We did a quick hack with the kit and connected a micro:bit via a Bit:170 breakout board. We then controlled an LED, then a motor
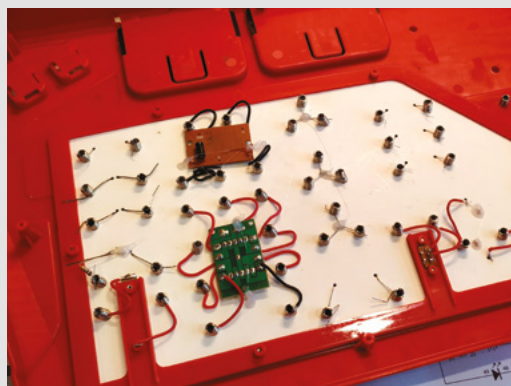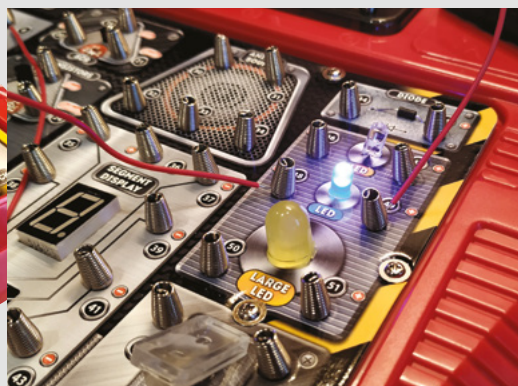
Far Left
The printed card cover explains how to connect to each component, and is numbered for reference in the manual

Left
The bare minimum is spent to secure the components, but rather than a negative, this is a big help to our hacks

Gas-powered mini soldering torch kit

Solder your projects without electricity

FERREX ◈ **£12.99** | **aldi.com**

By **Ben Everard**   🐦 **@ben_everard**

We're no strangers to cheap electric soldering irons – we've tried out many of them both on these pages and in our test lab. However, this is the first time we've put a gas soldering iron through its paces.

The unit consists of a body which holds a small amount of butane (or butane-propane mixture) that can be refilled using common cigarette lighter refills (though in the 15 years or so since we last bought one of those, they seem to have become much less common in the UK, and the only shop we found selling them was a kitchen supply store). There's a safety button, and with this depressed, you can slide back the trigger which both releases gas and fires the igniter. You can hold this back while you solder, or there's a 'continuous' slider that locks it down.

It's bulkier than most electric irons, but it's still comfortable in the hand, and it's surprisingly light, so we didn't have any issues holding it or using it as a tool, even over an extended time.

There are three different setups: blowtorch, which sends out a small flame; hot air, which acts as a mini hot-air gun; and soldering iron with three different tip shapes (chisel, cone, and needle).

Right ◈
The soldering iron comes with a carry case and a selection of tips, but no gas

We tested out the functions in that order. The blowtorch is too small for anything we generally blowtorch (you'll struggle to solder copper pipes, and it's not hot enough to braze). The packaging states that his mode is for jewellery making. While this isn't an area that this reviewer is particularly experienced in, he did try it out. The small flame is easy to direct, and heats small areas of metal well. However, we don't feel qualified to give a verdict on this tool for jewellery.

A LOAD OF HOT AIR

The hot air comes out at up to 600 degrees C (or so the packaging claims; we weren't able to verify this). There are two attachments – a nozzle that fires a narrow jet of hot air, and a scraper for removing paint. We were interested to see if the nozzle attachment could be used for surface mount rework, and it turns out that it can. At least, after a fashion. It's on the hot side for soldering work, and you risk cooking your components if you're not careful, but we were able to both solder on and remove components using it. For larger items, such as ICs, you're probably better only using this to remove them.

The soldering iron heated up quickly and performed well. Our biggest issue is that the tip was very prone to oxidisation, and we had to work hard to keep the tip clean. A simple tin-and-wipe approach that works on our other irons only managed to keep a small area of this tip clean. Fortunately, it was the small area at the very tip, so we were able to use it, but it required closer attention to cleaning than most irons we've worked with. This might be because it was getting too hot (it was definitely on the hot side), or it might be the metallic makeup of the tip, but either way, it was a bit of a pain to use.

The tips screw into the iron, and look very similar to those on the HS-1115K iron (however, the screw threads are different, so you can't use replacement tips for HS-1115K on this iron).

BURNT FINGERS

An important safety note when using this iron – hot air is ejected near the tip – this is the exhaust from the burning gas that's heating the tip. The shape of the iron means that our fingers were never close to it, but we did manage to burn a few things by setting

> **It's on the hot side for soldering work,** and you risk cooking your components if you're not careful

the iron down next to them. The glass-half-full take on this is that there's a heat-shrink-shrinker built into your soldering iron for keeping your joints tidy.

Overall, this soldering iron isn't great. The lack of accurate heat control in any action means it's a bit of a blunt instrument, and the difficulty of keeping the tip clean is annoying. However, it is very versatile. The ability for a portable iron to do surface mount rework (even at a basic level) could be extemely useful, in the right circumstances.

We wouldn't recommend this as a first soldering iron, or an only soldering iron, but we can certainly see its place in some toolkits. We'd have an easier time recommending this iron for soldering if the tips weren't so prone to oxidising. ◻

VERDICT

A highly versatile tool, but lacks finesse.

7 /10

Let's Start Coding

Microcontroller kits for young makers

LET'S START CODING ◈ $19.99 | letsstartcoding.com

By **Ben Everard**   🐦 **@ben_everard**

Let's Start Coding produces a series of programmable boards aimed at young makers aged 8–15. Their kits fit into two broad categories – the 'code' kits that don't have any expandable outputs, and the 'base' kits that do. The former includes things like the code speaker, code piano, and code car. As the name suggests, the 'code' kits help children learn to program (as there are no electronics to fiddle with).

Programming the kits is done through the Let's Code app, which uses the Arduino C++ programming language, but wraps it into a series of exercises and example code, so users aren't confronted with an empty screen, but bits of code that they can tweak and adjust. Once they're done with that, scroll down the page and you uncover the next lesson, and the next bit of code is introduced. There are also videos to show exactly what they need to do. Arduino C++ wouldn't be our first choice for a beginner programming language, but this setup shows that it can work well. The hardware feels incredibly robust. These are PCBs you could use to hammer in nails if you wished, and they're probably the thickest we've come across. Everything feels solid, and they feel like they should stand up to the rigours of use, whether that's in a classroom or at home.

In some ways, the microcontroller on these is a bit of a gimmick – it would be perfectly possible to create exactly the same environment without the physical hardware and have, for example, the piano work with software buttons rather than physical ones. Since there's no ability to connect up additional hardware, there's not much need to have the code run on a separate processor. However, by putting the code on a physical thing, it helps the

> **"** Let's Start Coding claims that these kits have **between 10 and 20 hours of learning in them "**

learners take ownership of it. Tangible things with real buttons and real lights can be more engaging than other programming environments.

Let's Start Coding claims that these kits have between 10 and 20 hours of learning in them, and that seems about right, based on our experience. They're not endlessly expansive like many programmable boards are – they have a very fixed set of uses, and once you've gone through them, you'll probably want to pass them on to another learner and try something else.

EXPANDABILITY
The Base kits do have headers that you can plug extra hardware into, and they come with a few
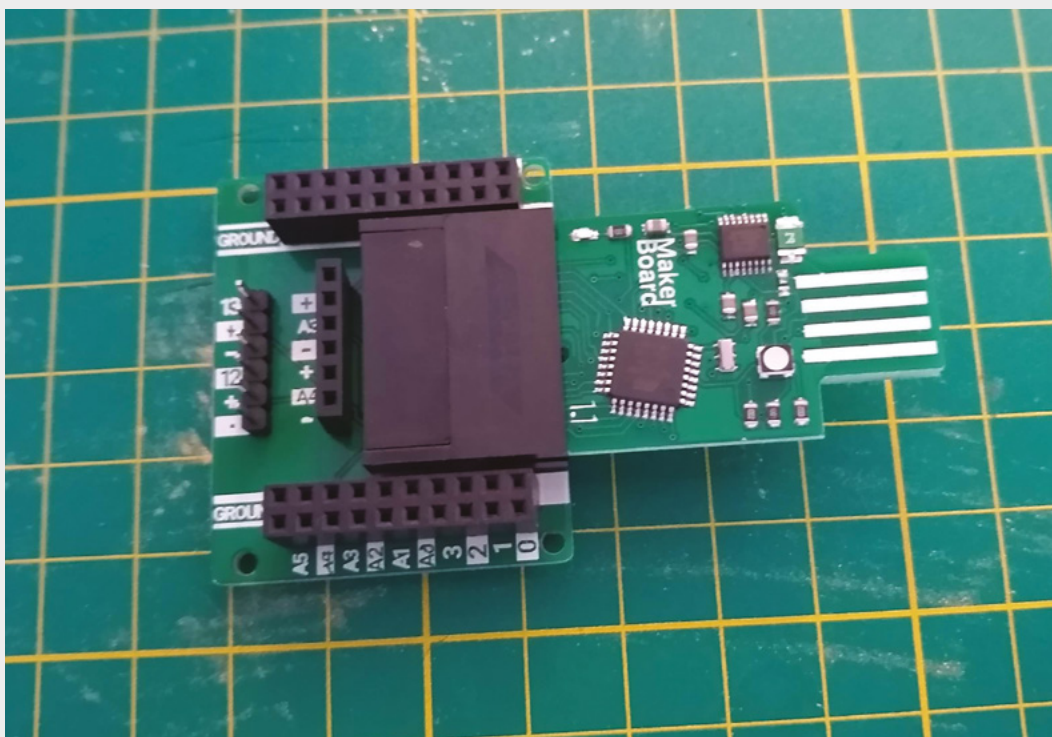
Below ◈
Each kit comes in a plastic pack with a set of cards to help you remember how it works

bits to get you started. These aren't huge kits with vast amounts of bits and bobs to play with. Even the 'Ultimate Kit 2' comes with just a handful of components, including four sensors, a few LEDs, and an LCD. As with the coding kits, the aim here isn't to create a vast number of things that you can build anything with, but create a small, curated kit with a good guidance to help you get started. In principle, you can use these as you would almost any microcontroller board. Once you've worked through the examples, you can start to design your own projects based on them, though it may be a bit of a leap for some learners to go from the cosy-hand-held world of lessons to free-form projects. How much use they get once the lessons are exhausted will depend on the learner, and some may prefer to move on to platforms with more support for independent projects.

The hardware in these kits is, from a features perspective, fairly unimpressive. You can get far more powerful microcontroller boards for less money. However, that's missing the point of these kits – they're not about the hardware, but the software and lessons that come with it.

We've used some electronics with good documentation before, and there are certainly some other excellent tools available to help children learn to code, but we can't recall any other kits that provide this level of direction. You start at the beginning, and it takes you through a series of steps to learn different concepts. While more self-directed learners may struggle with this and therefore prefer kits with more off-piste options, we suspect that many learners – particularly those who are being exposed to programming for the first time – will respond well to this hand-holding approach.

These kits, combined with the app, are really impressive and will help children with both self-directed learning and learning with a parent who isn't a confident programmer. ◻
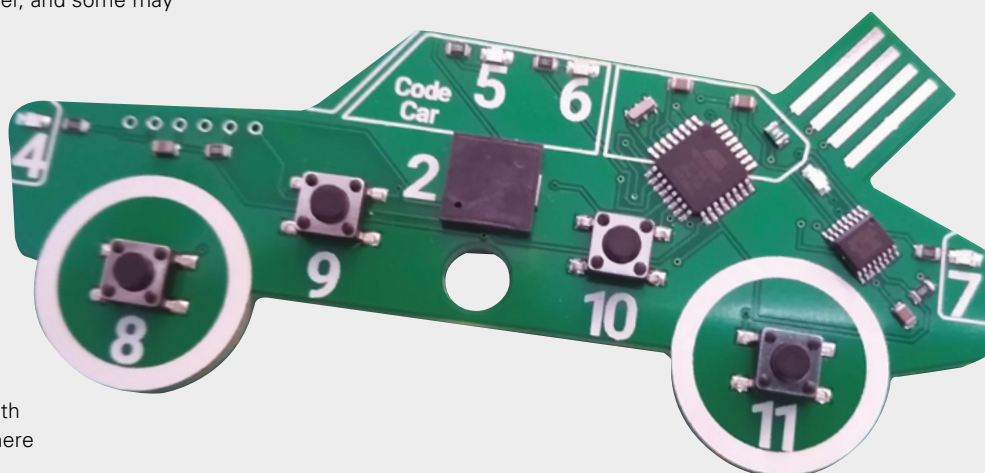
All About Circuits: Tools

Doing your maths automatically

ALL ABOUT CIRCUITS ◆ free | **allaboutcircuits.com/tools**

By **Ben Everard**   🐦 **@ben_everard**

Electronic circuits are bound by physical laws. You're probably familiar with Ohm's law for calculating any one of voltage, current, and resistance, providing values for the other two, but there are many more, and some are more esoteric than others. Of course, it's useful to understand the underlying formulas, but be honest, can you really remember exactly how to, say, calculate the resistor values for an inverting op-amp? And even if you can, it's not exactly an easy formula to put into a traditional calculator. Fortunately, the good people at All About Circuits have gone through many of the most common formulas required in electrical engineering, and converted them into handy online tools.

> ❝ Can you really remember exactly how to calculate **the resistor values for an inverting op-amp?** ❞

In each case, you can just enter the details, press Calculate, and out pops an answer. However, if you need a little more info, there are also details of the formula used and a little background information on the subject.

There are lots of similar tools on the web, but we like All About Circuits' offering because it's easy to use, yet still has lots of detail that you can dig down into if needed, and there's a great breadth of different calculators for almost any situation you find yourself in. The calculators even link through to the relevant pages in an online textbook, if you want to fully understand what's going on.

Whether you need to get the numbers quickly, or are just looking for a refresher on how to calculate them, the All About Circuits: Tools page is a great place to get what you need. We use these all the time because they're just far less prone to accidental mistakes than calculating the results by hand. ◻

Twisted-Pair Impedance Calculator

A tool designed to calculate the characteristic impedance of a twisted-pair cable

Inputs

Diameter of Wire

Seperation Between Wires

Substrate Dielectric

Above ◈
Just enter the data and press Calculate

Left ◪
There are diagrams, formulas, and more information if you scroll down

Equations

$$Zo_{twistedpair} = \frac{120}{\sqrt{e_r}} \cdot \ln\left[\frac{2s}{d}\right]$$

$$delay = 84.72 \cdot 10^{-3} \cdot \sqrt{e_r}$$

$$L_{twistedpair} = 10.16 \cdot 10^{-9} \cdot \ln\left[\frac{2s}{d}\right]$$

$$C_{twistedpair} = \left(\frac{.7065}{\ln\left(\frac{2s}{d}\right)}\right) \cdot e_r$$

VERDICT

A set of easy-to-use tools, with plenty of information to drill down to.
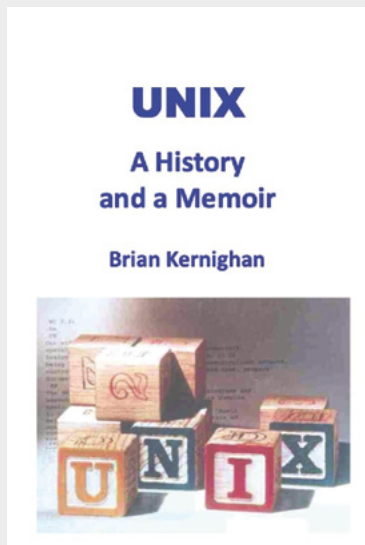
9/10

Unix: A History and a Memoir

Brian Kernighan ◆ **£15** ❙ **Independent**

By **Ben Everard**   🐦 **@ben_everard**

Unix is one of the most influential pieces of software ever written. It – or something very similar to it – runs on almost every supercomputer, the majority of smartphones and, of course, Raspberry Pis. However, it has slightly unusual beginnings. It came out of Bell Labs, the research arm of a telephone company.

While this book does have a compelling first-hand account of the creation of the software, it is let down by its cloying sentimentality. Kernighan comes across as a dyed-in-the-wool company man. If we are to believe him, Bell Labs was a utopia of perfect engineers and perfect management. While we don't doubt it was a good place to work, a more detached point of view would hopefully be able to provide a better context for the work.

Without this critical eye, the characters in the development of the system come across as wooden. We are all human and flawed, and it is these flaws that give us character. Perhaps it's merely a reflection on the fact that this is part-memoir, but the effect of hiding parts of the personalities of those involved makes it hard to relate to them, and the whole thing comes across as one-dimensional. ▫

> **If we are to believe him,** Bell Labs was a utopia of perfect engineers and perfect management

VERDICT

A contemplative look inside the creation of Unix, but ultimately unsatisfactory.

6/10

next month

RASPBERRY PI PROJECTS

ALSO

→ **ELECTRONICS**

→ **MAKING CLOTHES**

→ **SAWING**

→ **3D PRINTING**

→ **AND MUCH MORE**

DON'T MISS OUT

hsmag.cc/subscribe

Benchy

Benchy, the jolly 3D printer torture test, is an almost ubiquitous 3D-printed item. It makes a good test object because it has features that will only print well if your printer's set up properly, including overhangs and bridges. Almost as importantly, it's now been printed on almost every printer ever created, so you can compare results.

If you're having printer problems, there's even a Benchy Facebook group, dedicated to helping people get perfect little boats every time. Head to **facebook.com/groups/3DBenchy** and upload an image of your Benchy attempts, and hopefully someone can point you in the right direction.