MAKE | BUILD | HACK | CREATE

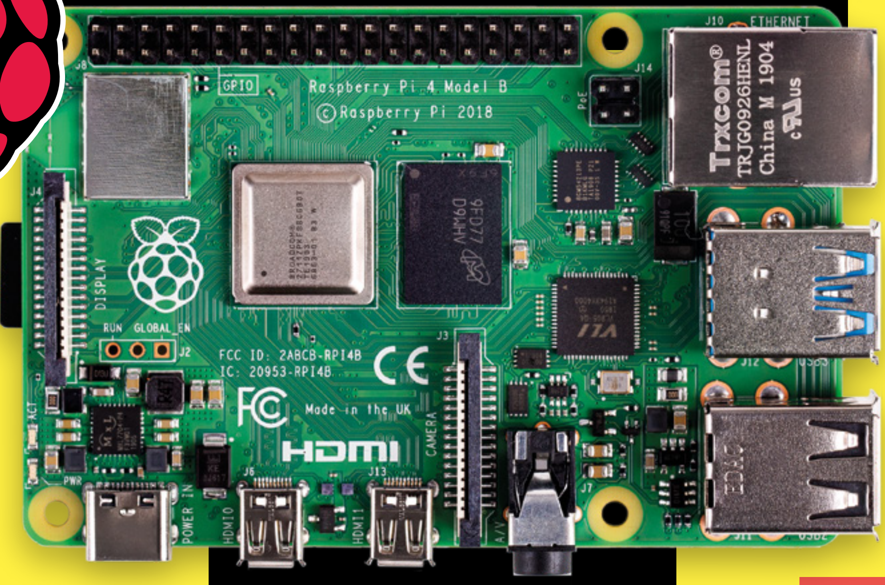
132 PAGES OF MAKING

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | February 2020 | Issue #27

BEST RASPBERRY PI PROJECTS



25 WAYS OF USING THIS TINY COMPUTER

+ ARC GENERATOR

+ NUTS AND BOLTS

+ ULTIMATE PONG

+ MITRE SAWS

+ ARDUINO POWER

DESIGNER JACKET

Personalise your clothes for a perfect fit

TALKING TOOLS

Adding speech output for callipers

MITCH ALTMAN

The Gandalf of makerspaces shares his wisdom

Feb.2020 Issue #27 £6

9 772515 514006 27

Raspberry Pi PRESS

PYTHON SPARKS POLYSTYRENE NUTS & BOLTS

Welcome to HackSpace magazine

Raspberry Pi released their first board only a couple of months after I started working on tech magazines. Back then, the board caused quite a stir in the tech world (and particularly the Linux-focused tech world that I was in). However, I doubt many people could have predicted quite how big it would become.

While the worlds of industry and education have embraced the board in many varied ways, it's in the makersphere that we've seen this little board take on the most exciting (we think) uses. This issue we're celebrating all the ways Raspberry Pi boards have been used to make the world a more interesting place. We love seeing all the different ways this hardware is used, and we'd love to find out how you use yours? Let us know on social media or by emailing **hackspace@raspberrypi.org**.

> **It's in the makersphere that we've seen** this little board take on the most exciting (we think) uses

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
f hackspacemag
🐦 hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

PAGE 48
SUBSCRIBE TODAY

EDITORIAL

Editor
Ben Everard
@ ben.everard@raspberrypi.org

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

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Harriet Knight, Ty Logan

Photography
Brian O'Halloran, Dominic Blewett

CONTRIBUTORS

Lucy Rogers, Drew Fustini, Jo Hinchliffe, Derek Woodroffe, Mayank Sharma, Andrew Lewis, Glenn Horan, PJ Evand, Simon Monk Will White Gareth Branwyn, Les Pounder, Marc de Vinck,

PUBLISHING

Publishing Director
Russell Barnes
@ russell@raspberrypi.org

Advertising
Charlie Milligan
@ charlotte.milligan@raspberrypi.org

DISTRIBUTION

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

SUBSCRIPTIONS

Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
☎ 01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

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

HackSpace magazine is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. ISSN: 2515-5148.

50

Contents

Tutorial

Make a made-to-measure jacket

82 Sew your own piece of sartorial excellence

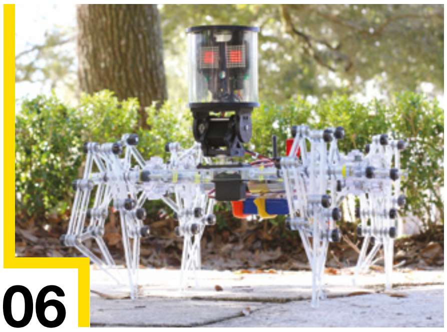
Cover Feature

34

124

Direct from Shenzhen

Arc igniter

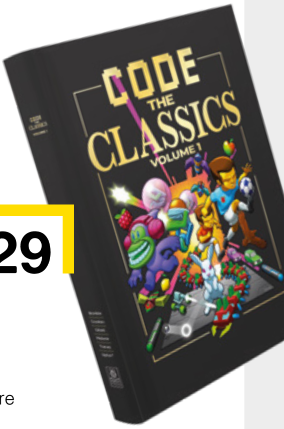


Interview

Mitch Altman

73 FORGE

113 FIELD TEST



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

BinDayCator

By Darren Tarbard

hsmag.cc/5LuK71

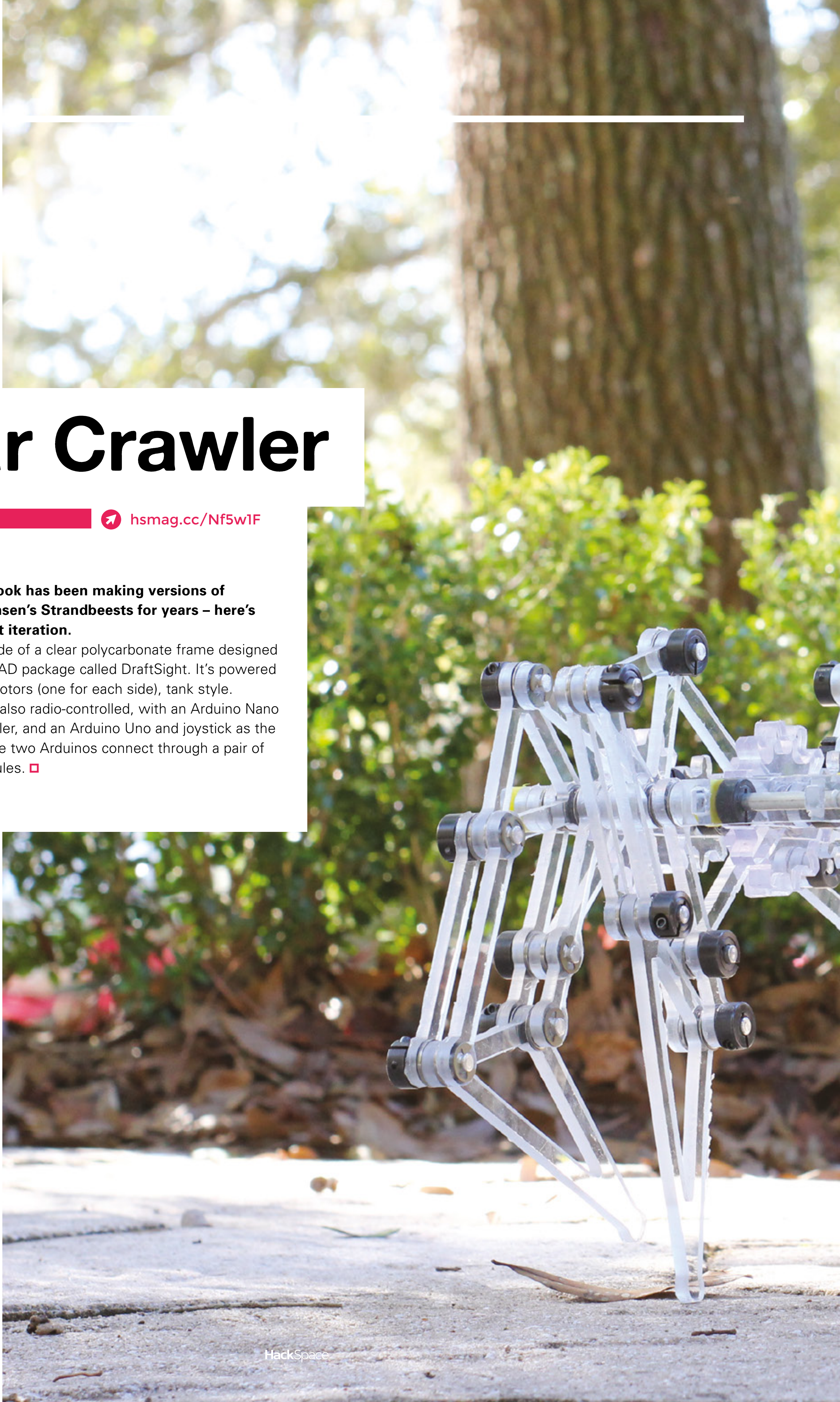
We have four different coloured recycling bins that are collected based upon a collection schedule created by the council. After a few too many forgotten bin collections (especially annoying around Christmas time, or in the heat of summer), I had an idea to display which bin to put out using an actual bin that literally changes to the colour of the required bin.

I designed, using Blender, and 3D-printed a bin that I could fit the electronics unobtrusively within for a clean design. The electronics consist of an ESP8266 programmed via Arduino and some NeoPixels. I made a YouTube tutorial to show people how to make their own here: **hsmag.cc/18w8TQ**.

The second part of the puzzle was creating a Node-RED flow which polls my local council website each day. The BinDayCator then uses this to automatically show the correct bin colour(s) and pulse when the bin needs to be put out. The tutorial for this part is here: **hsmag.cc/YIHom3**.

When I tweeted about my creation, I was amazed to find it struck a chord with lots of people, getting nearly six thousand retweets, two million views, and reshared on Reddit and Facebook where it got thousands of comments. □

Left Amid the post-Christmas detritus still covering Britain's streets, it's the BinDayCator's time to shine



Clear Crawler

By Jeremy Cook

hsmag.cc/Nf5w1F

Jeremy Cook has been making versions of Theo Jansen's Strandbeests for years – here's the latest iteration.

It's made of a clear polycarbonate frame designed in a 2D CAD package called DraftSight. It's powered by two motors (one for each side), tank style.

The Clear Crawler is also radio-controlled, with an Arduino Nano as the on-board controller, and an Arduino Uno and joystick as the remote control unit. The two Arduinos connect through a pair of nRF24L01+ radio modules. □

Right The ever-watching eyes of the Clear Crawler are provided by two 8×8 LED matrices

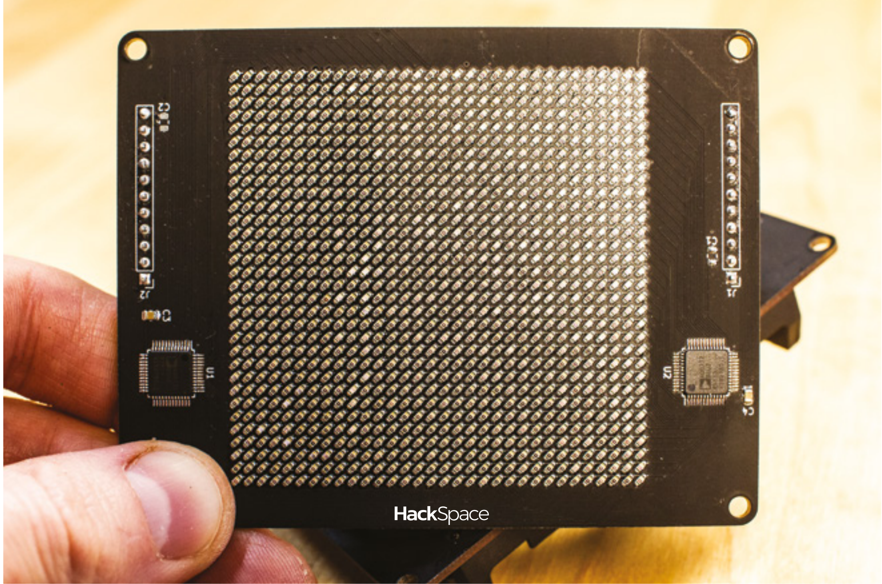
DIY Camera

By **Sean Hodgins** seanhodgins.com

With most of my projects, they're more of a challenge to myself than anything else. I've always been into photography and DIYing everything, so instead of making just a DIY film camera, I wanted to make the most basic form of a digital camera and make it in a way that is simple enough where almost anyone can understand how it works. I thought about getting an off-the-shelf image sensor, or hacking a computer mouse sensor, but I decided that wasn't simple enough and it also wasn't really building the digital camera 'from scratch', which at some point I thought was an important feature of the build.

I started taking some photos and scaling them down to ridiculously low resolutions to see what would be an appropriate size for an image sensor and still: 1) get a decent image, and 2) not completely break the bank in component costs (or my neck soldering it). I settled on the 32×32 array (60mm across) sensor.

What's great about this size (even if I didn't take advantage of it in this build), is that it's a standard in a lot of medium format cameras, so adapting a quality lens would be possible – simple, even. I originally wanted to use a pinhole lens (I'm still convinced this is possible without too much difficulty) but decided to settle on the hacked-apart Canon front glass for the time being. However, I am going to attempt to 3D-print a lens in a few weeks, whenever my SLA printer arrives. The images actually turned out way better than I was expecting. There will likely be a second version of the build; bigger image sensor, faster shutter times, and more complex circuitry, but it was great to start with the basics. □



Right
1024 phototransistors come together to form the full image

Left Sean tells us that 1024 is about the limit of components a human should populate a circuit board with in one sitting

Rain-powered generator

By Quint Crispin

hsmag.cc/eaUEh9

"**Ever since I can remember, I've been fascinated with machines.** That's what fuelled me to become an engineer. But fascinations don't stop just because the workday is over. That's why I built my own fab shop at home, so I can take my ideas from theory into practice. Plus I always learn a ton when I build something new. And thanks to the internet, I can share what I've learned to help others figure out what they can build."

Quint first built a standard, brushed motor to generate power from rain falling on his roof. This wasn't as efficient as he'd hoped, so he built a brushless version, shown here. The magnets spin, while the wire coils stay stationary, enabling Quint to capture the current generated without the friction that comes from a commutator.

If you've got a rainy afternoon to spend, we strongly recommend Quint's YouTube videos that explain the science of electricity generation. □

Right ↗ The generator (actually an alternator) puts out 0.6 watts from the rain collected on Quint's roof

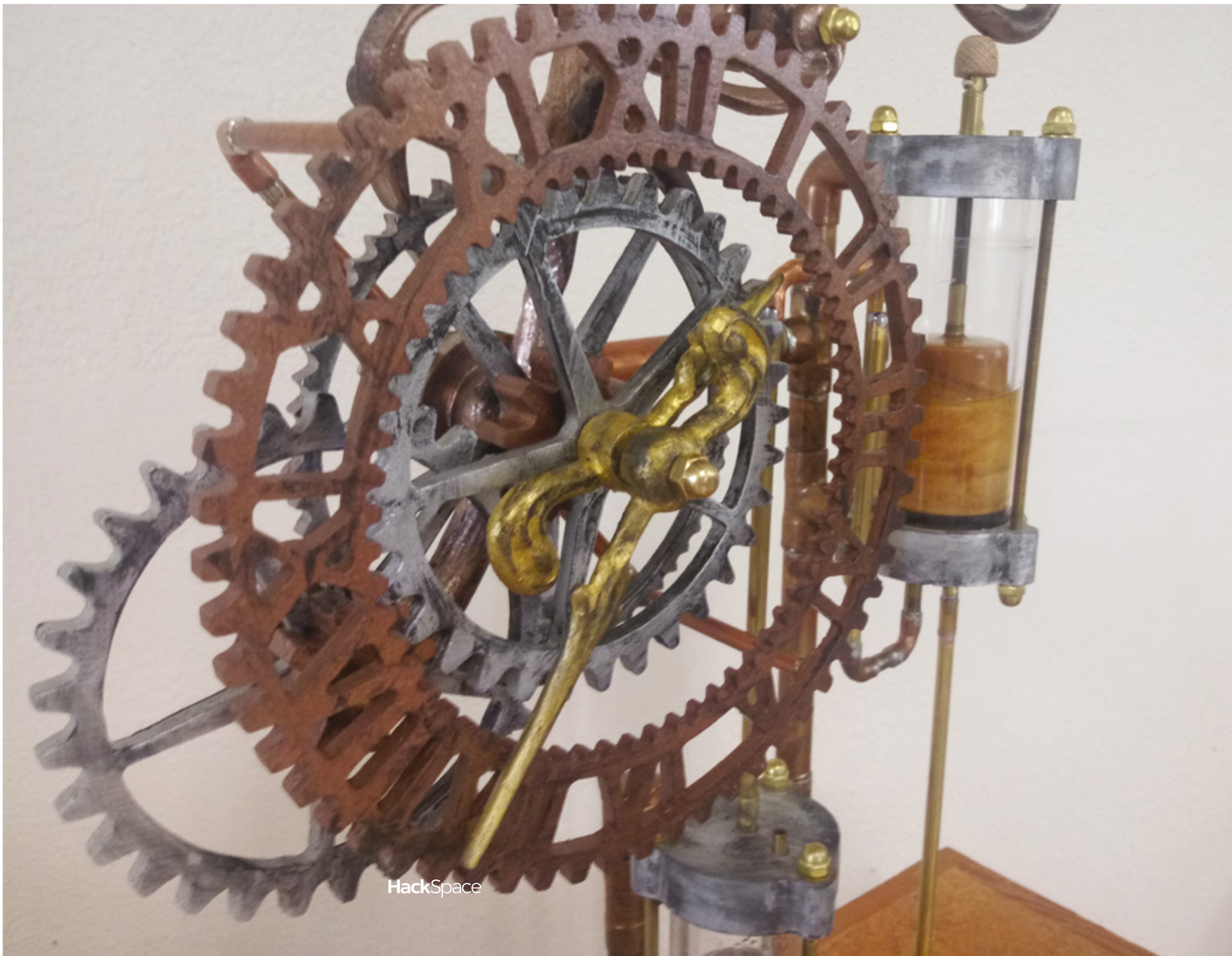
N
S
N
S
N
S
N
S

Steampunk clepsydra

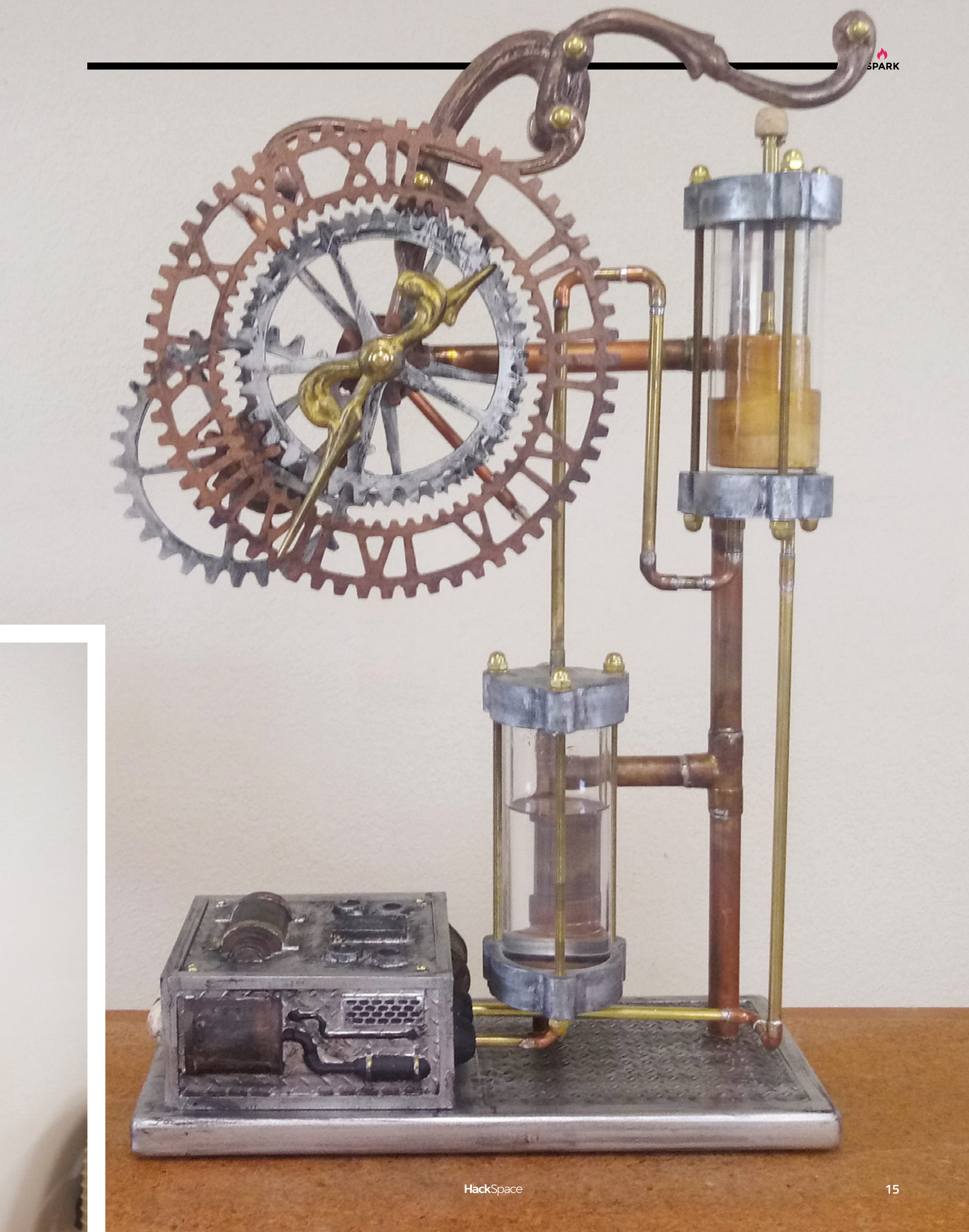
By Dick Bipes

hsmag.cc/H8zTri

I'm a retired electrical engineer with a long-time hobby of woodworking that started in my dad's shop when I was a boy. I enjoy taking on custom woodworking projects for myself, friends and relatives, and customers. Steampunk clepsydra is a water clock whose basic design dates back to the 3rd century B.C. A chamber is constantly filled with water and is periodically drained by a siphon. A float connected to a mechanism displays the time. I use an electric pump regulated with a microcontroller and software to ensure a precise one-minute cycle time and timekeeping accuracy. □

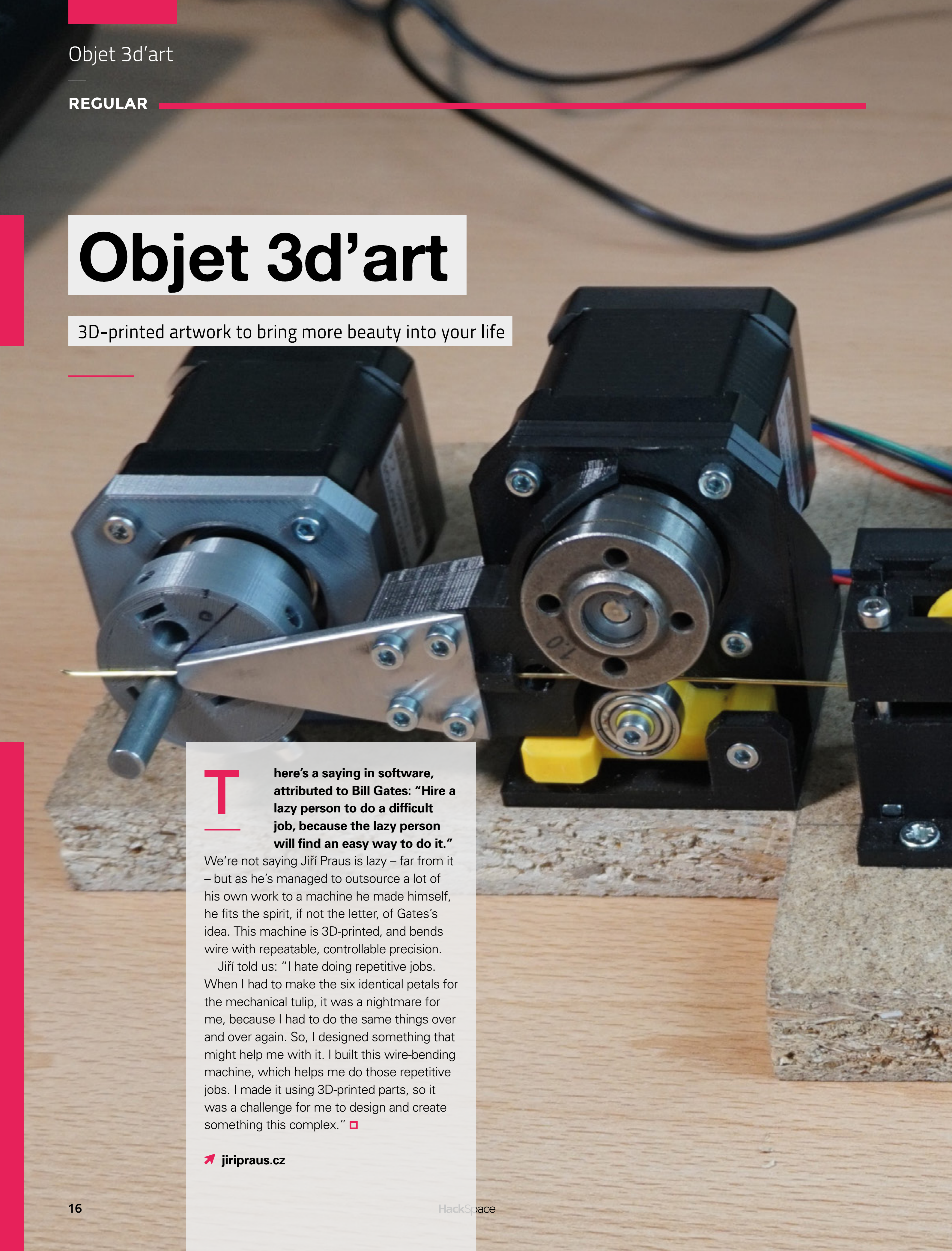


Right ↗
Water clocks have been found from several ancient cultures, from Korea to Persia to Egypt



Objet 3d’art

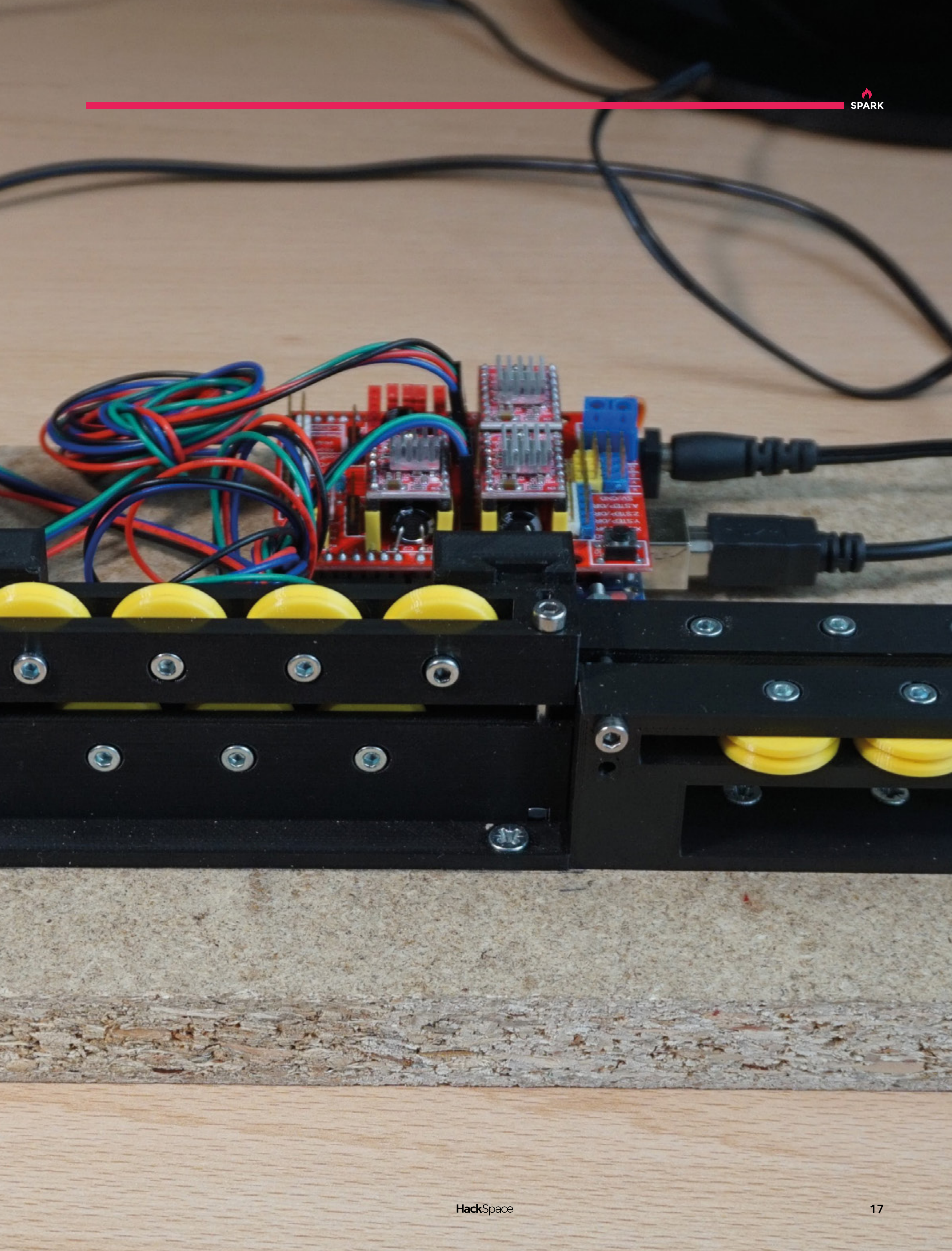
3D-printed artwork to bring more beauty into your life



There’s a saying in software, attributed to Bill Gates: “Hire a lazy person to do a difficult job, because the lazy person will find an easy way to do it.” We’re not saying Jiří Praus is lazy – far from it – but as he’s managed to outsource a lot of his own work to a machine he made himself, he fits the spirit, if not the letter, of Gates’s idea. This machine is 3D-printed, and bends wire with repeatable, controllable precision.

Jiří told us: “I hate doing repetitive jobs. When I had to make the six identical petals for the mechanical tulip, it was a nightmare for me, because I had to do the same things over and over again. So, I designed something that might help me with it. I built this wire-bending machine, which helps me do those repetitive jobs. I made it using 3D-printed parts, so it was a challenge for me to design and create something this complex.” □

jiripraus.cz



Meet The Maker: Jiří Praus

Programmer, artist, maker, and robot builder extraordinaire

Free-form electrical circuits tick all the right boxes for us. The creativity, the sculptural quality, the fact that you're passing an electrical current through a physical thing that you've made… all of this is great fun. Anyone we've spoken to who has had a go at this art form always mentions one name as an inspiration: Jiří Praus. We caught up with Jiří to find out what it is about free-form electronics that inspires him.

"By day I'm a full stack developer. I'm part of a small team creating a platform for creative teams who want to be more effective. Making things is my hobby.

"Three years ago, I needed a special regulator for my home heating system, and the only one I could find was ridiculously expensive. So I bought an Arduino kit and started hacking. It was a great challenge to do low-memory, low-fat programming for me, as a Java developer.

"I read a quote along the lines that, to be a great developer, you need to design your own hardware. That stuck with me as a great challenge. And that was my first project. It was fun, useful, and a great start.

"I started looking around on the internet for other things I could do with the Arduino. One day I saw this great guy, Mohit Bhoite, the father of free-form electronics for me.

"It was winter, two years ago, and I got the idea to create a snowflake with this free-form art. It combined all the passions I have: the programming, electronics, and 3D construction. I put all my heart into it and created this free-form snowflake.

> "My first project, the snowflake, combined all the passions I have: **the programming, electronics, and 3D construction**"

LEARNING NEW THINGS

"First, you need proper tools, and the right materials to make nice joins. And then, of course, you need a lot of imagination to make all the connections properly. If you miss something, it's really hard to repair – unlike a computer program where you can just fix the error. It's pure creativity, which is why I love it.

"The second one, which I think to this day is still the most famous of them on the internet, is the mechanical tulip I made for Valentine's day for my wife. That thing got a lot of attention.

"The Freeduino was a challenge of a friend of mine: 'Hey, I saw this Arduino somebody made from wires [rather than a PCB]. It's not that good-looking. What do you say?' I said, 'Challenge accepted, man!' It works. Everything I do has to serve some purpose. I don't like things that look nice but do nothing, so it actually works.

"For me, every project needs to give me something new. Some new concept or process, or something else new that I need to learn. For example, the tulip, there was a mechanical part. For the Freeduino, it was creating my first complicated circuit encased with epoxy.

"I was asked to do a workshop at Maker Faire Prague last year, so, to fit into one hour, I came up with these small badges, with a coin cell battery →

Above
"I love the freedom that comes with not doing it for money. I'm doing it for fun, so I only have to do the things that I really enjoy."

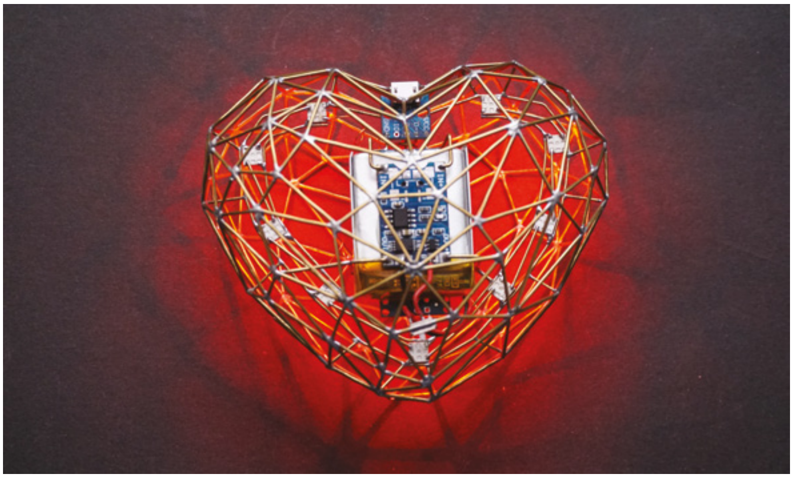
and a few LEDs in different shapes. You can make them in an hour, so they're ideal for anyone who wants to have a go at free-form soldering.

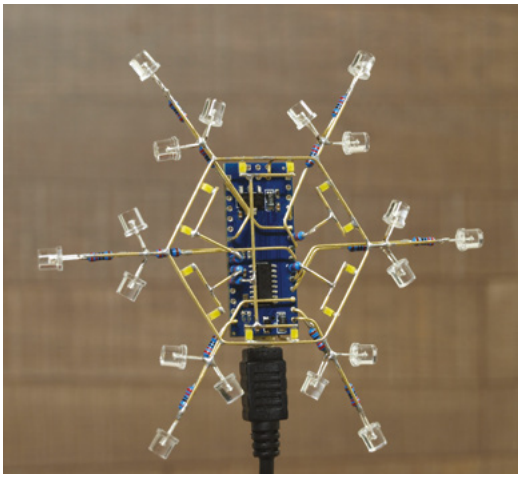
THE PLAN IS… THERE IS NO PLAN

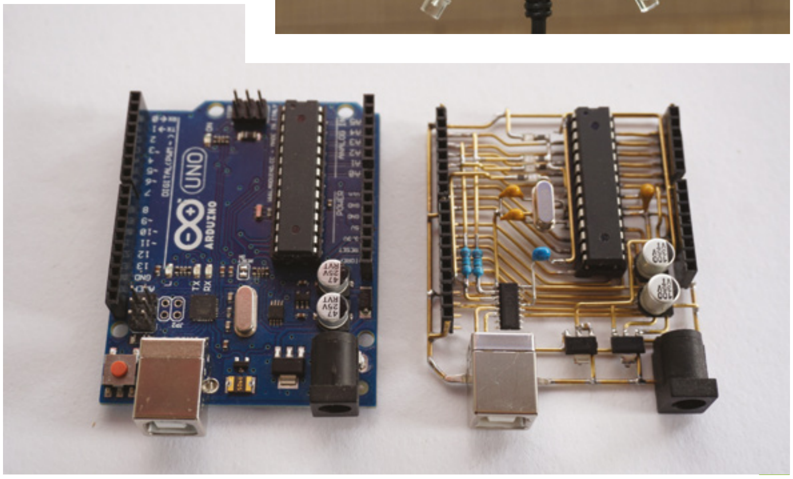
"What next? That's a tricky question. I don't make plans. When I plan things, it typically ends up a disaster. Or, I get bored and leave half-finished projects at the bottom of a drawer, because it's no longer fun for me. My ideas come to me when I'm not looking for them: in the shower, in bed, and as soon as the idea comes, I have to start working on it. That's what I do instead of planning for projects. I do plan to do more YouTube this year, though. I started two months ago, and I want to do more, and also to visit some more fairs and conferences this year.

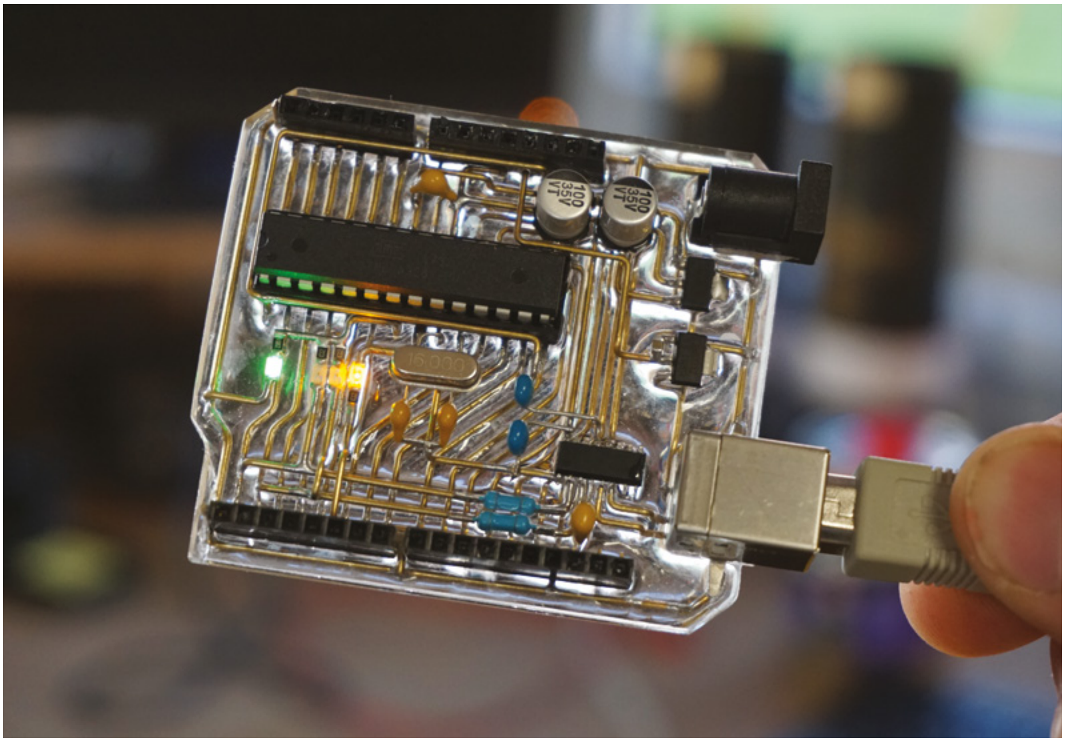
"Making totally clears my mind. It helps me focus on work, and work helps me focus on my hobby. Those two things are totally different. While one part of my brain is relaxing, the second one is working. I hate to sit at home in the evening and watch TV. I can do that for an hour, maximum, but then my hands start looking for something to do. I therefore spend four hours a day on my projects, usually in the evenings – not every day, but most of the week.

"My message would be not to be afraid of trying something new. Just do it! You will fail, I guarantee it. I've failed loads of times, but I only ever show off my successes, because that's what people share. That's another reason I don't like to share plans, because it normally ends up differently. But I fail big time. You'll fail, but eventually you'll make something great." □









Top
We still can't quite get over the fact that this immaculate snowflake sculpture was his first-ever free-form electronics project

Right
The Freeduino is a fully functional, much prettier, clone of an Arduino Uno

Left
His first Arduino make, his heating regulator, is still running to this day

When things don't work

Pushing the limits of what's possible

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**

It's all very well saying 'celebrate your failures' – when the failures don't matter too much – such as putting too much current through an LED and it going 'pfft', or a 3D print that ran out of filament. These things I happily share on Twitter and say, 'oops – silly me'. However, when I have to tell a client that I can't actually make the thing I said I could make, then pride, ego, and professionalism make things a whole lot harder.

I usually make one-offs and solve bespoke problems, and I don't usually start a project knowing quite how I am going to get to the result required. Usually, I search the internet in case something has already been developed that I can use or adapt – there is no point reinventing the wheel. Then ideas start to coalesce, and I will probably buy some bits to see if I can make a prototype. In most projects, I will be able to work my way around any difficulties, and even though it may take longer than I expected/budgeted for, I can make it.

I've had to go back to a client this month and say, 'I know I said I could do this, but actually, I've looked into it and I can't'. It took a while for me to actually accept that I couldn't make it – and then a deep breath before I told the client.

The project was to make 'coloured flame candles'. I remembered the flame tests at school, where different chemicals burn with a different coloured flame. A few years ago a friend bought some fir cones that gave off some colour when burnt in the fire – although we were disappointed with the results. So, I thought I could make some myself.

I bought the chemicals, and first set light to them in a tea light case with a small amount of methylated spirits – I made some lovely pinks, greens, and a bright orange.

It took a while for me to actually accept that I couldn't make it – and then a deep breath before I told the client

I then tried fir cones. I soaked different cones in solutions of the chemicals and dried them out. When I put them in a fire, the kitchen roll I had dried them on briefly burnt the bright colours. The fir cones, however, just had the normal yellow of burning wood – and, if you squinted, had a dark room, and a bit of imagination, the chemical colour could be seen. I tried making candles by soaking wicks in the solutions – but even those were pathetic.

Fortunately, I had not left the project to the last minute. And so when I did tell the client, they had time to find an alternative. They even thanked me for investigating the options and letting them know. My pride still hurts though! □

Open-source chips

Breaking free of chip design monopolies with RISC-V

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.

When we think about what open-source hardware means, we usually think about the board design being freely available. But what about the processor? Is there a way to make hardware that is truly open source? This month's column is dedicated to an exciting – and surprisingly political – development in chip design.

When you write a program in the Arduino IDE, it is compiled into instructions for the microcontroller to execute. How does the compiler know what instructions the chip understands? This is defined by the Instruction Set Architecture. The ISA is a standard, a set of rules that define the tasks the processor can perform.

Chances are that both your laptop and the data centre streaming your favourite movie are using an ISA owned by Intel or AMD. The processor in your smartphone is almost certainly using a proprietary ISA licensed from ARM. Proprietary standards can be overpriced, prevent innovation, or even disappear altogether when companies change strategy.

Enter RISC-V, a free and open ISA created by researchers at UC Berkeley, led by Krste Asanović and David Patterson. "We were always jealous that you could get industrial-strength software that was open," Patterson explained to VentureBeat at the RISC-V Summit back in December. "But when it came to hardware, it was proprietary. Now, with RISC-V, we get the same kind of benefit. It helps education, and it helps competition."

This open standard proved to be useful outside of academia. Nvidia and Western Digital are now shipping millions of devices with RISC-V processors. These companies have the freedom to leverage open-source implementations while avoiding ARM licensing fees – which can really add up when shipping large volumes.

The ISA is a standard, a set of rules that define the tasks the processor can perform

Nations such as India see the importance of being able to create processors that are not under the control of a foreign corporation, who may be forced to build in backdoors for their own government. There is also strong interest from chipmakers in China, especially now that US companies have been banned from doing business with Huawei.

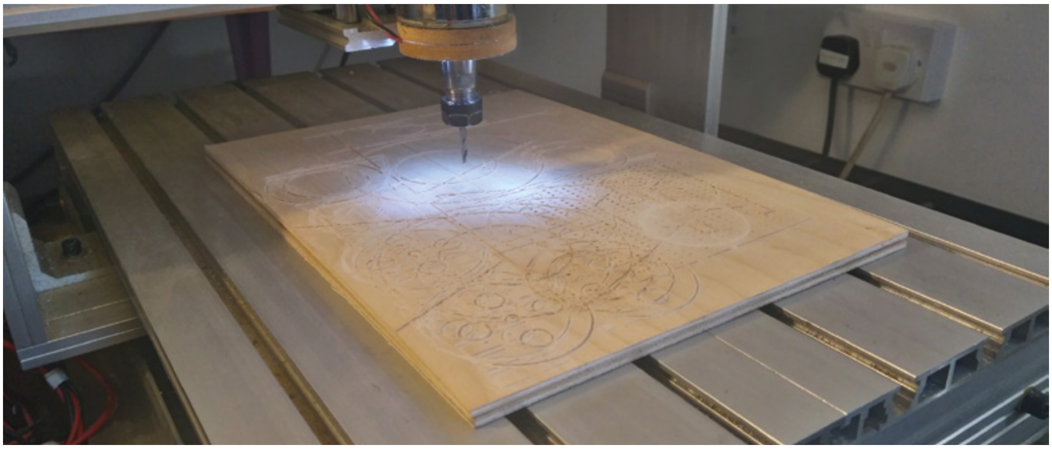
With these financial and political motivations, plus an increasingly mature software ecosystem, including Linux support, it won't be long before you have a device with a RISC-V processor in your home or pocket.

You can learn more about the exciting possibilities that RISC-V unleashes from Dr. Megan Wachs by pointing your web browser to **hsmag.cc/qw4ed1**. □

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**



CNC

Thanks for the quick-start guide to CNC [written by Jo Hinchliffe, issue 26]. I've been using a CNC machine to cut parts for home projects, shelving, etc. I get it right after a few goes, but there's always an element of trial and error. Hopefully now I'll be making fewer errors, or at least understanding why I've gone wrong rather than wasting more plywood than I have to!

James FitzPatrick
Toledo, Ohio

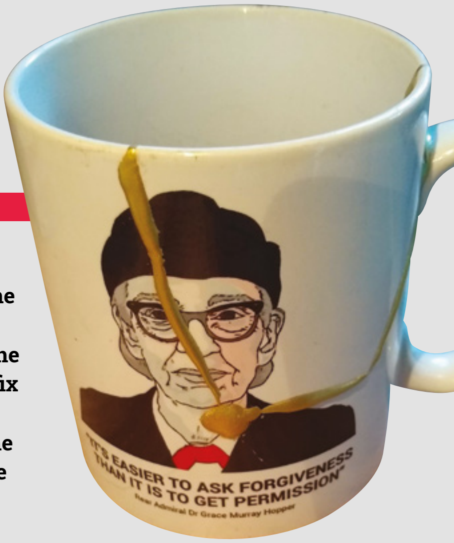
Ben says: In theory, every error is something to learn from. In practice, I find that in flitting from brewing/woodwork/electronics/programming, I sometimes forget what I did wrong the next time I come back to a technology. If I were using CNC every day, the information would go straight in; as it is, it takes a few goes before I remember what I'm doing wrong. Jo's guide has also saved me a few quid's worth of plywood!

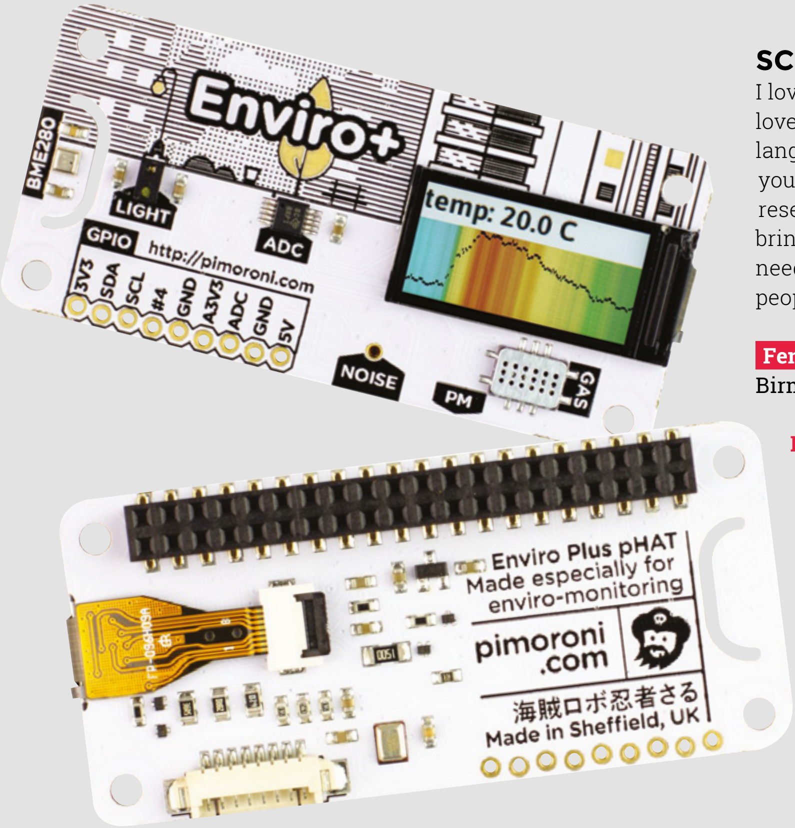
THE MUG THAT WAS BROKEN

Fixing things with gold glue is a great idea. I may accidentally on purpose smash my husband's disgusting old AC/DC mug so I can add some gold to it.

Sarah
Dover

Ben says: Steady on – Kintsugi, the Japanese art of decorative fixing, gains its power because we love the broken things enough to want to fix them. If you're smashing them on purpose just to fix them, what's the point? They may look cool, but the whole idea is that we cherish our things and look after them.


"IT'S EASIER TO ASK FORGIVENESS THAN IT IS TO GET PERMISSION"
Rear Admiral Dr Grace Murray Hopper



SCIENCE

I loved reading about the scientist you spoke to in issue 26! I loved science at school, but life gets in the way. It's not like languages, or electronics, or literature that you can pick up in your own time at home – to do science you need to be in a research company or a university. I take my hat off to those bringing science to anyone who is interested. As she says, we need to learn how to take care of the planet better, so the more people thinking about solutions, the better.

Femi Adebayo
Birmingham

Ben says: Yes! But also, no. Yes, because we do indeed need more scientists and Chan'nel is a shining example of someone making the world how they would like it to be. No, because there are a ton of citizen science projects that anyone can get stuck into! Check out the Luftdaten project – if you don't speak German, there's an excellent tutorial on the Pimoroni website: hsmag.cc/8UGNBq.

It would also be remiss of us not to mention the superb Pint of Science project, which every year (11–13 May this year) organises scientific talks in pubs. This year they're holding events in 40 towns and cities across the UK. There will be beer!

INFILL

I got a Prusa i3 not long ago. I've not bothered going in depth with any of the settings yet – it mostly just works, and if anything goes wrong, I Google the answers on a case by case basis. I'd never even thought about infills, just left it as the default. Well, thanks to you, I'm now thinking about what the right infill is, and looking forward to getting my teeth into the million other things I can change to get perfect, not just OK, prints.

Anthony Talbot
Kingston upon Hull

Ben says: The infill guide in issue 26 was a lot of fun to write. I had no idea of how deep you could go with different stresses, strength of filaments, speed of prints, etc. It's almost a shame I had to stop and write some of it down!

CROWDFUNDING NOW

ANAVI Miracle Controller

A programmable lighting board with options for expansion

From $25 | crowdsupply.com | **Delivery:** Mar 2020

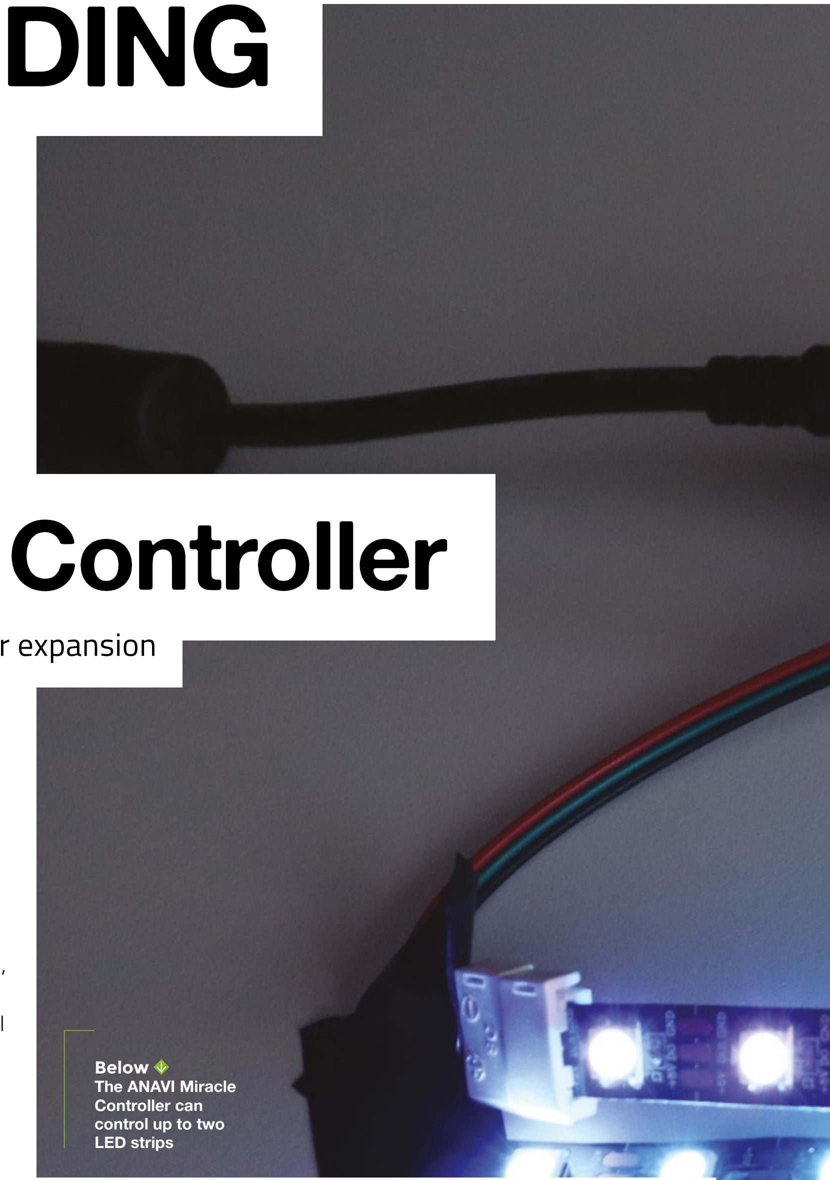
Lights, WiFi, and programmability make a great combination. You can make light shows, display information, change the colour automatically with time to help keep your body in sync with the natural cycle of daylight, and much more. The ANAVI Miracle Controller brings these three things into one board. It's based on the ESP8266, and takes power from either 5- or 12-volt jacks. Output comes via any supported three-wire addressable LED (for most people this will mean WS2812B, but some other models are also supported).

If you need a little more output, you can add an I^{2}C OLED display, and there's also space for up to three I^{2}C sensors.

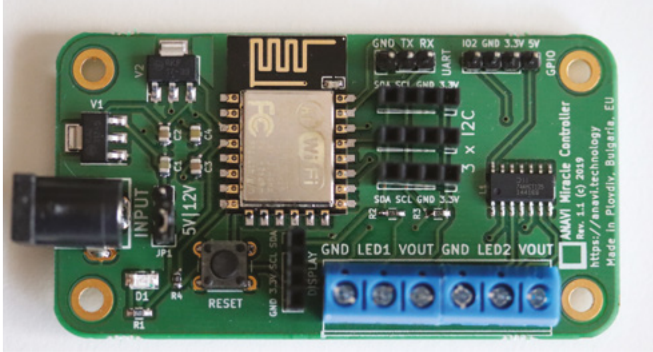
You can program this board via the Arduino IDE (you'll need a serial-to-USB cable), but out-of-the-box it comes set up to use Home Assistant, so if you have a hub running, you can control the lighting without having to touch the code.

The basic board is $25, or you can opt for a more fully featured package such as the Developer Kit which includes a serial cable, two LED strips, a 0.96-inch OLED display, and a selection of three sensors. All this comes in for just $59.

There's nothing you can do with the ANAVI Miracle Controller that you couldn't do with a bare microcontroller and a little wiring, but it does package everything up well and makes it a bit easier to plug everything together. Though we haven't got our hands on one to test, from the looks of things, it should be an attractive option for people who aren't confident working with microcontrollers, but want to design their own lighting system. □



Below
The ANAVI Miracle Controller can control up to two LED strips



BUYER BEWARE

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

> "You can control the lighting without having to touch the code"

Space of the month: 57North Hacklab

57North Hacklab

57north.org.uk

57NorthHacklab

57NorthHacklab

57North Hacklab is a small hackerspace in the north-east of Scotland, in the city of Aberdeen [writes Tom Jones]. It started up in 2013, when a group wanted to do something hackerspace-like, and we quickly found office space in the centre of the city. 57North is now on its third iteration, moving this autumn from the basement of a gaming cafe to some unused space in a commercial office building.

We offer a space for members and visitors to talk about technology, and access tools and components to make projects possible. The space has a large selection of tools, such as several 3D printers, and a laser cutter (though in our current location, not a lot of space for really messy activities).

Both tools and equipment are free for anyone to use, and you can just drop by. We have an open night every week on Tuesday, and this acts as the main meeting night for the space. We also have a large stock of components generated by members buying excess parts when working on their own projects. These components are made available for everyone, as we would much prefer you were able to finish your project today and if we have to look after some extra cool components to do it, all the better.

We are a space built and driven by our members, and a lot of our activity comes from their interests. MS0SCZ is a radio club growing within the Hacklab, with a wide selection of cool radio projects. The club does basic voice operations, and is expanding into performing data mode operations radio, on both HF and VHF. There is an effort now in the Hacklab club to help others get their licences in a modern, friendly environment.

The Hacklab has a lot of open-source activity, with developers for the FreeBSD and Debian operating systems amongst its ranks. Some open-source work tends to trigger more work – the members involved in open-source projects are eager to leverage their local network to get help with problems. A member of the Hacklab, who is a radio amateur, recently started the HamBSD (**hambsd.org**) project to build an open-source, well-designed and written APRS network stack.

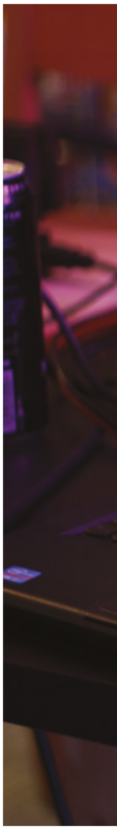
We like blinkenlights! Our previous space had an entirely home assistant-driven lighting system that integrated with commercial lights, as well as lights made by members from microcontrollers and LED strips. Our current space is getting an entirely new lighting system. →

Right Unicorns and soldering irons go together like a horse and carriage

57North
HACKLAB
JUAN
ZANDE

Out of the Hacklab, the Air Quality Aberdeen (**airaberdeen.org**) group started workshops to build air quality monitors for citizen science. Air Quality Aberdeen works with the global Luftdaten (**luftdaten.info**) network of air quality monitors, which gives a picture of air quality conditions around the world in the form of open data. The workshops introduce people to collecting their own air quality data through building sensors to deploy in their own homes/workplaces. The project uses off-the-shelf microcontrollers and sensors with open firmware.

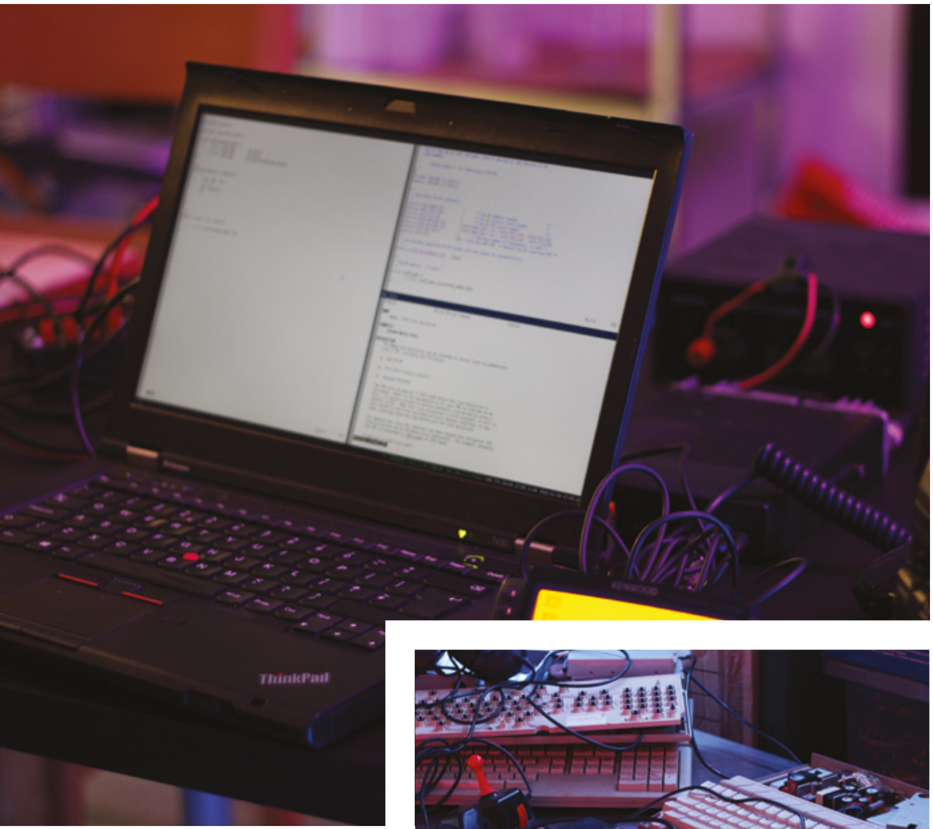
We are small, but have taken it on ourselves to represent Scotland at various hacker conferences, camps, and festivals around the world. If you are at an event, it might be worth searching to see if the Scottish Consulate has a presence. You should drop by if you are up in Aberdeen, or come and find us at Chaos Computer Club events in Germany, or at EMF Camp in 2020. □

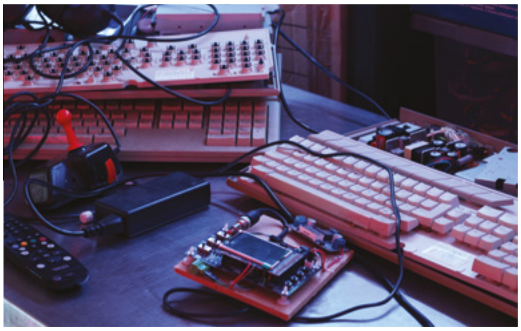


Right ⬗
For a small group, 57North sends attendees to a large number of events

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
The space has plenty of spare electronics components for members to use

meet
build
learn
play

HackSpace
TECHNOLOGY IN YOUR HANDS

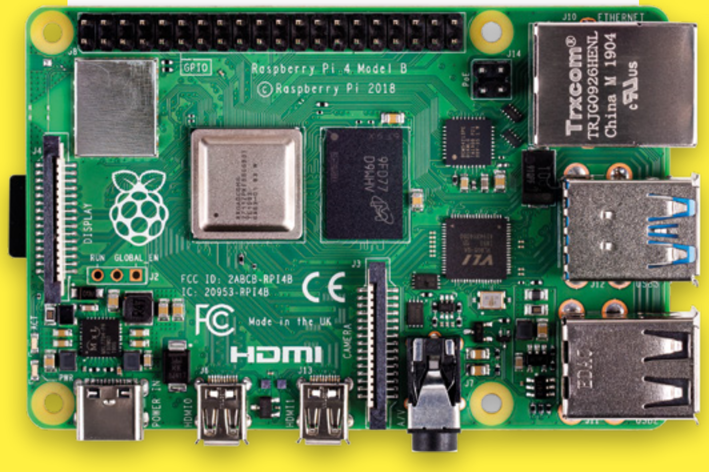
LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

BEST RASPBERRY PI PROJECTS

25 WAYS OF USING THIS TINY COMPUTER



Raspberry Pi is a small, Linux-powered computer that gives us a huge amount of computing power in a form factor that makes it easy to incorporate into our projects. The first version – released in 2012 – packed a 700MHz processor and 512MB of RAM, but by the latest, that's grown to a quad-core 1.5GHz processor with up to 4GB of memory.

As well as processing power, Raspberry Pi has a wide range of output options, from simple general-purpose input/output (GPIO) pins that can be turned on or off, to complex add-on boards (known as HATs) that give you new functionality. If physical connections aren't what your project needs, there's also WiFi and Bluetooth hardware which means you can control the board wirelessly or from the internet.

Perhaps the best feature of Raspberry Pi is the community that's built up around it. They range from expert engineers and scientists working at the cutting edge to beginners taking their first steps with computing. There's an almost endless range of projects, tutorials, tips, and tricks available to help you craft your own builds.

In this issue, we want to showcase the flexibility of the Raspberry Pi platform for makers. You can use this diminutive board to power an almost endless range of projects. Over the next few pages, we'll highlight some of our favourite projects. →

GAMING BUILDS

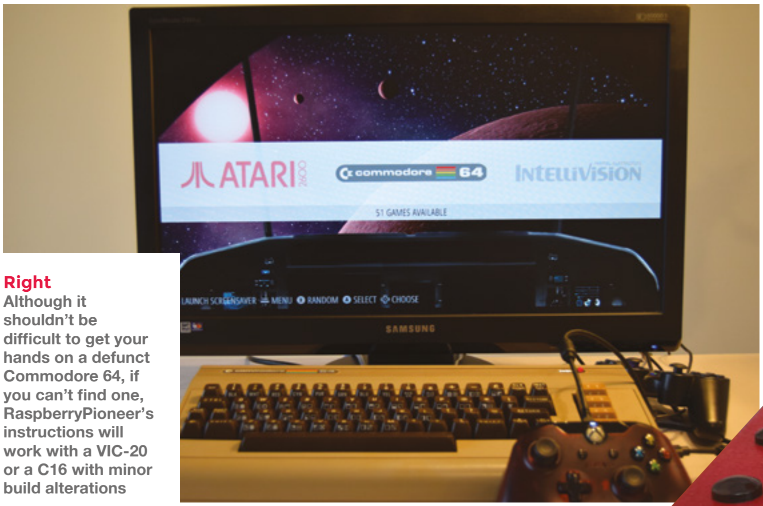
DIY RECREATION MACHINES

Outside of education, one of the most popular uses of Raspberry Pi has been as a retro-gaming platform. Thanks to projects like RetroPie, you can easily transform a modern, everyday computer like Raspberry Pi (RPi) into a retro-gaming machine.

For a richer experience, though, you can follow the lead of Stephen Williams, who decided to use a Raspberry Pi 3 to bring his old Commodore 64 back to life (**hsmag.cc/JN3GIS**). In addition to his Raspberry Pi that's repurposed as the motherboard of the Commodore, he's used an Arduino Micro to resurrect the Commodore's keyboard as a fully functional USB one. In true maker style, Stephen has used Lego pieces to pad the Commodore's internals. The build is straightforward but quite involved. Once you have all the components, Stephen says it shouldn't take more than a weekend to put it all together. Pay special attention when hooking up the Commodore, Arduino Micro, and Raspberry Pi with a breadboard and some jumper wires. A handful of keystone inserts give the old computer modern interfaces such as USB, HDMI, and network ports that are all wired to RPi. Once everything's connected, you can set up the Commodore 64 emulator called VICE in RetroPie, and you're good to go.

If you want to kick things up a notch, Cameron at School of Shop shares detailed instructions on building a bartop arcade machine powered by a Raspberry Pi 3 (**hsmag.cc/TBf81A**). A majority of the build involves woodwork, cutting through a medium-density fibreboard (MDF) to create your cabinet. As can be expected, the build requires an extensive list of parts, although Cameron has marked the ones that are optional, such as extension ports and cables. One of the things that Cameron constantly reminds us throughout the build is to plan ahead. It's a good idea to fetch and measure all the controls and buttons before you cut out spaces for them in the cabinet. Also remember to wear the proper protective gear when cutting and shaping wood. Besides a

Below
Cameron shares tons of useful tips for virtually every step of the build that'll not only be of use for makng the bartop arcade, but also for general woodworking



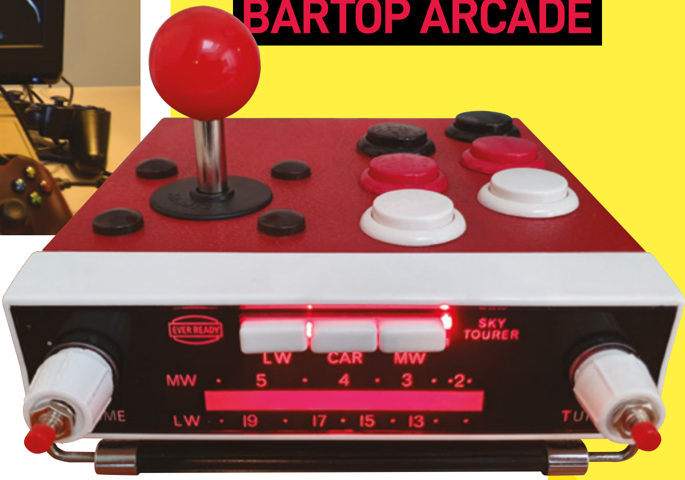
Right
Although it shouldn't be difficult to get your hands on a defunct Commodore 64, if you can't find one, RaspberryPioneer's instructions will work with a VIC-20 or a C16 with minor build alterations

Raspberry Pi 3, the other electronics in this build are a pair of Logitech speakers and a 19" monitor. Again, Cameron points out that the build is very flexible, and you can use different-sized components.

> THE BUILD DOESN'T INVOLVE AS MUCH WORK AS THE BARTOP ARCADE

TURN ON, TUNE IN

Does that sound like too much work? If you aren't comfortable working with wood, try Martin Mander's approach. The host of the 'Old Tech. New Spec.' YouTube channel, Martin loves to give a new lease of life to classic, broken, and discarded equipment using his favourite weapon of choice, Raspberry Pi. Martin wanted to build a bartop arcade console as well, but he enjoyed playing RetroPie games on his 28" TV and didn't want to build a separate cabinet. So instead, he repurposed a car radio from the 1960s into a portable RetroPie gaming console (**hsmag.cc/t2gPQi**) that can stay docked to the TV, but can also be unplugged for use elsewhere. His hack has got a microswitch joystick and six arcade buttons that are connected to an Raspberry Pi 3 via a Picade controller board. Martin has also replaced the volume and the tuning dials with buttons for Select and Start functions. The build doesn't involve as much work as the bartop arcade, and can be replicated without too much effort once you have all the components. Again, just like the previous build, you don't necessarily need the same car radio to create your portable gaming console. Instead, use Martin's procedure as a guide to squeeze in a small computer and the other arcade electronics inside any car radio or a similar piece of tech.



Above
Martin has built an extra USB port into the console to plug in an additional USB joystick, which he's created from a 1980s joystick retrofitted with a Pimoroni Player X controller board

When you're done building your consoles, and are in the heat of the battle, you can use Daniel Davis's hack to stream your games to Twitch with the push of a button. Daniel's hack, dubbed Twitch-O-Matic (**hsmag.cc/9S7dmm**), uses a Raspberry Pi Zero W, along with an HDMI to CSI-2 adapter to stream HD video to the video game streaming website, Twitch. The adapter costs $80 (about £69) and connects to the camera port.

To make it more intuitive and fun to use, Daniel then wires an Adafruit light up latching switch via a breadboard and some jumper wires to the Raspberry Pi's GPIO, along with two LED backlight panels. After testing his setup, he puts it inside a 3D-printed case that's connected to the TV and the gaming console via an HDMI splitter. With everything wired up, you can now press the button on the Twitch-O-Matic to begin sending video game streams to your Twitch account until you press the button again to stop the stream. →

Below
Once you have the CSI-2 adapter, you can easily customise this hack to stream to a different website like YouTube

AUTOMATIC FABRICATION

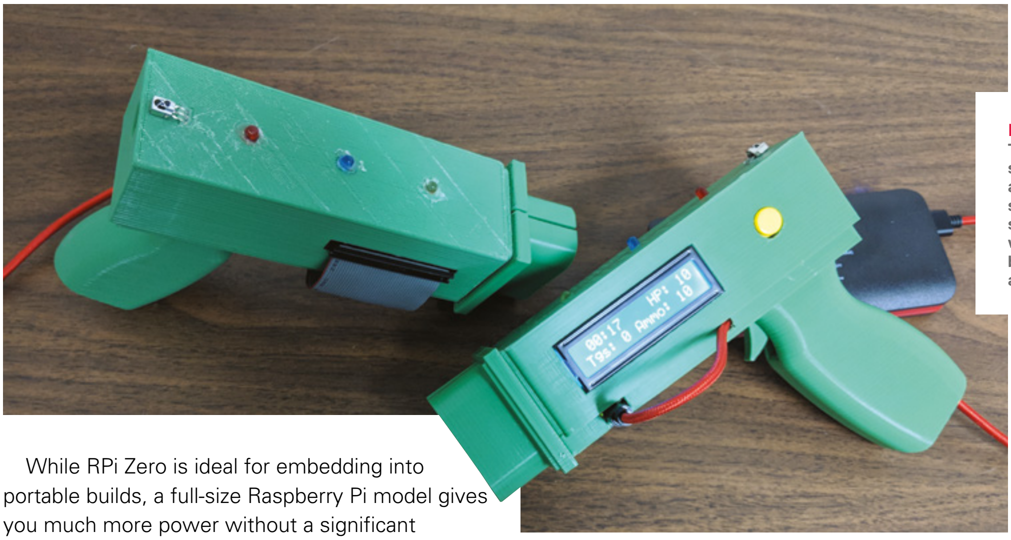
BUILDING THINGS AT THE PRESS OF A BUTTON

Makerspaces all around the world have reduced the barrier of access to high-end, specialised, and expensive fabrication tools and technologies like 3D printing and laser cutting. By democratising their use, the makerspaces have opened up a whole new dimension for trail-blazing makers to now fabricate unique and innovative hacks.

Rick Perotti made good use of a 40W laser cutter to fulfil everyone's childhood dream of owning a rocket ship control panel (**hsmag.cc/H1QBrS**). Rick has always been fascinated with switches and dials on control panels of all kinds, and his hack pays homage to several science-fiction movies we all love and admire. The control panel is driven by a Raspberry Pi 3 that works along with three Arduinos, a 9" TFT LCD, and dozens of LEDs, switches, and dials that he got from eBay, Amazon, and AliExpress. The hack took him almost a year to complete, partly because he was only working on it part-time, but also because this was the first time he was working with a laser cutter.

If you have a 3D printer handy, you can replicate Terran Gerratt's project and build yourself a few Raspberry Pi-powered laser tag guns (**hsmag.cc/ooRi9t**). Terran got the idea for the build while working on home automation projects that used infrared for communication. The guns have a Raspberry Pi Zero W inside them running Raspbian Lite, along with the MQTT broker service for communication, I^{2}C for the display libraries, and the Linux Infrared Remote Control daemon. Terran has used the IR LED transmitters from an old set of laser tag guns, but he suggests you can replicate the build using any IR transmitter. The 3D-printed guns also have an LCD panel that displays useful information during the gameplay.

Right
Although Kieran used a Bluetooth keyboard, he's attached it to Raspberry Pi via USB, since he had no intentions of removing it from the case



Left
Terran's laser tag setup requires a simple Linux server, and he suggests its work can easily be handled by a Raspberry Pi 3

While RPi Zero is ideal for embedding into portable builds, a full-size Raspberry Pi model gives you much more power without a significant increase in physical dimensions. As a Computer Science professor, Lewis Baumstark fully understands the capabilities of the small device. He took inspiration from the MSX computer of the 1980s to build a case for his Raspberry Pi 3+ (**hsmag.cc/u3nRGs**), although he reckons it'll work for a model 4 as well. He's combined it with a 7" TFT screen that comes with a small breakout board that has buttons for turning on the screen and for adjusting the display. In addition to fabricating a case to house the screen while exposing its buttons, Lewis has printed some of the intricate design bits of his 'FLIPT BIT' computer using a 3D printer and has also used some wood and PVC to space and fit some components like the keyboard and the trackpad.

Right
The light panels at the bottom of the rocket ship panel are actually just marbles with plastic tubing that are lit by conventional 3.3V Christmas lights

BOX OF DELIGHTS

Want something more classy? That's exactly what Kieran Donnelly thought when he picked up the HyperPixel 4" touchscreen from the Pimoroni stall at the Makers Central event in Birmingham, England. So he got to his workshop, picked up a chisel and a hammer and cut some wood into a tiny wooden laptop (**hsmag.cc/mDy2IP**). The build isn't particularly difficult to replicate, especially for someone comfortable working with wood, nor does it require too many tools. Follow Kieran's video closely as he crafts the pieces of wood into the box that will also require a custom 3D-printed hinge mechanism. Kieran has also included a UPS HAT with a 2200 mAh LiPo battery and charging circuit to make the device truly portable. The UPS HAT is attached to a power button and power socket with a micro USB lead to turn the computer on and off. After creating the case, Kieran glued the electronics to the case. →

THE BUILD ISN'T PARTICULARLY DIFFICULT TO REPLICATE

Below
Cartridge slots complete the retro look of FLIPT BIT, and can be removed to access RPi's GPIO pins

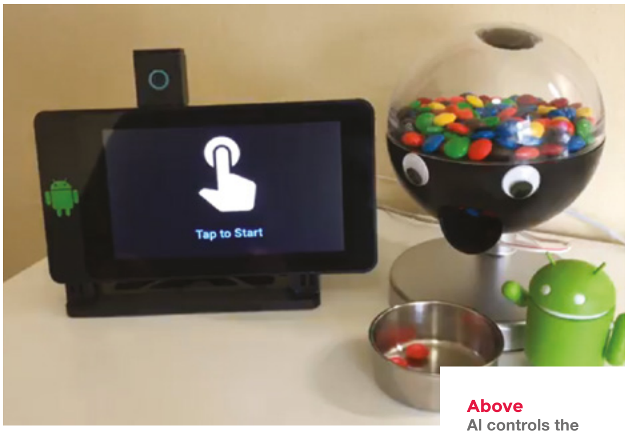
ROBOTS AND AI

ADDING A MIND OR MOVEMENT TO YOUR BUILDS

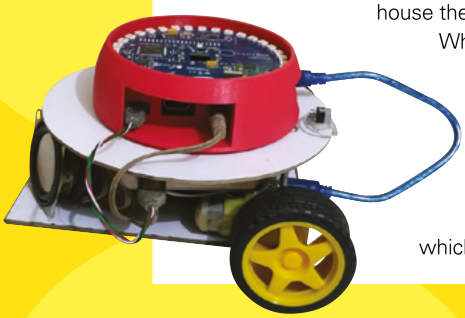
One of our favourite uses of a Raspberry Pi board is how it enables intrepid makers to create intelligent devices and robots. Md. Khairul Alam built one in his attempt to tear kids away from screens. His simple robot (**hsmag.cc/bf5qAc**) has two 3W speakers and a mini amplifier board. You can ask it to make different sounds – like those of ambulances, police cars, or various animals – and it can also tell stories, poems, jokes, and play music to keep the kids engaged. The robot can also align itself to any of the four cardinal directions. Khairul made the chassis himself using an ebonite board and a couple of DC gear motors that were driven by Arduino Uno. The main components of his robot are the Matrix Creator wired up to a Raspberry Pi 3. The Matrix Creator acts as a microphone and provides various sensor data. He's 3D-printed a case to house the Matrix Creator.

While Khairul's robot is good enough to keep the kids engaged, it lacks video. For that, you need Jorge Rancé's gatoBot, which is a live video streaming bot that you can control from the web (**hsmag.cc/DYaUcC**). He has combined an Raspberry Pi Zero W with a Camera Module atop a 2WD robot car chassis to keep an eye on his cats. The bot is driven by a couple of SparkFun L298 motors that are powered by four AA batteries. It's a straightforward assembly, and Jorge has shared the simple wiring schematics and all the code to create the simple web interface that has buttons to control the movement of the bot. While Jorge uses it for his cats, you could use it as a general-purpose security and monitoring bot.

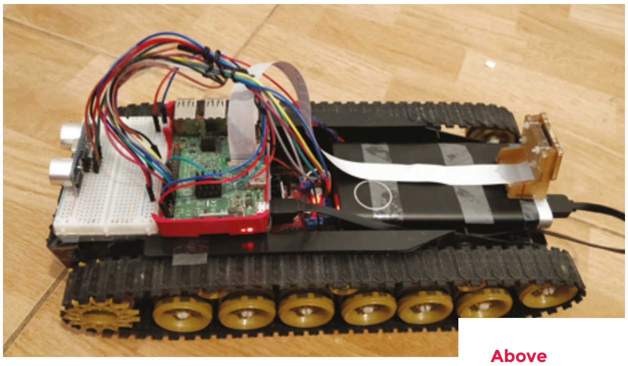
If you want something that looks a lot more rigid, look no further than Roi Segev's RaSpyTank (**hsmag.cc/LbHSK2**). He's used a damping balance



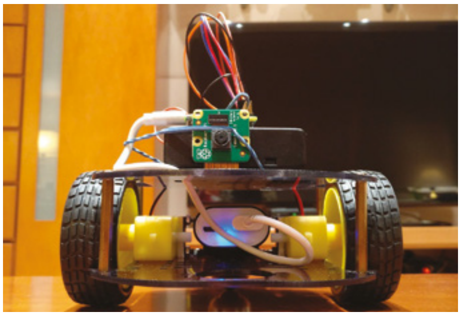
Above
AI controls the motor on the dispenser through an Explorer pHAT, but you can use the L9110 breakout board as well



Below
Khairul has developed an app with the free Snips AI voice platform to allow his robot to interact with the kids



Above
Roi has used the pistreaming Python interface for streaming video from his RaSpyTank to a web browser



Left
The gatoBot's RPi Camera Module live-streams the video and allows Jorge to control the robot without having to keep an eye on it

IT'LL SHOW THE DISTANCE TO AN OBSTACLE WHEN YOU REVERSE THE TANK

tank robot chassis platform that he bought from AliExpress together with a Raspberry Pi 3, NoIR Camera Module, and a few other electronics to create a robot tank that does everything Jorge's robot car does and can additionally measure the distance to an obstacle. Roi has used the HC-SR04 ultrasonic sensor pretty much like the reverse sensors on cars – it'll show the distance to an obstacle when you reverse the tank.

Between Jorge's and Roi's robots, you'll get a good idea of how to build yourself a video streaming robot that you can control remotely using a web interface. Roi's robot also shows you how to append information from additional sensors, which will help you tailor the robot for your specific application. Both the codebases are easy to tweak and can easily be customised as per your requirements.

SWEET BUILDS

Now it's time to reward yourself with some candy. Al Bencomo used a Raspberry Pi 3 running Android Things to build a smart sweetie dispenser that asks you to show it a particular image (**hsmag.cc/pXpMGa**). The dispenser then processes the image and dispenses candy once it's determined that you've shown it the correct image.

The dispenser has a touchscreen for displaying information, such as the image it needs to see to release the candy. A Raspberry Pi on the dispenser captures an image using a Camera Module, then processes the image through the TensorFlow Lite inference library for Android and matches it against the pre-trained ImageNet model. A successful match activates a motor to dispense the candy.

One of the most popular showcases for the OpenCV and TensorFlow libraries is the Donkey Car platform for building small-scale autonomous cars. Salma Mayorquin and Terry Rodriguez have extended the Donkey Car into a Wall·E-inspired robot that can identify and clean up small pieces of rubbish. Dubbed LitterBug (**hsmag.cc/vz2DWY**), the duo have designed their robot to work on the beach and have equipped it with a solar panel, a night vision camera, a motion detector, and other electronics to monitor and report littering in addition to clean-up.

As you can imagine, the build process is quite extensive and involves a fair bit of training as well. The two makers have explained the process in great detail, from their customisations to the stock Donkey Car, to the training of the vehicle to navigate and identify pieces of rubbish. Even if you don't intend to build LitterBug, there's quite a lot of information on the project's website for anyone who is interested in building autonomous vehicles. →

Below
The makers suggest using a PS3 controller connected to RPi over Bluetooth to train the LitterBug Donkey Car

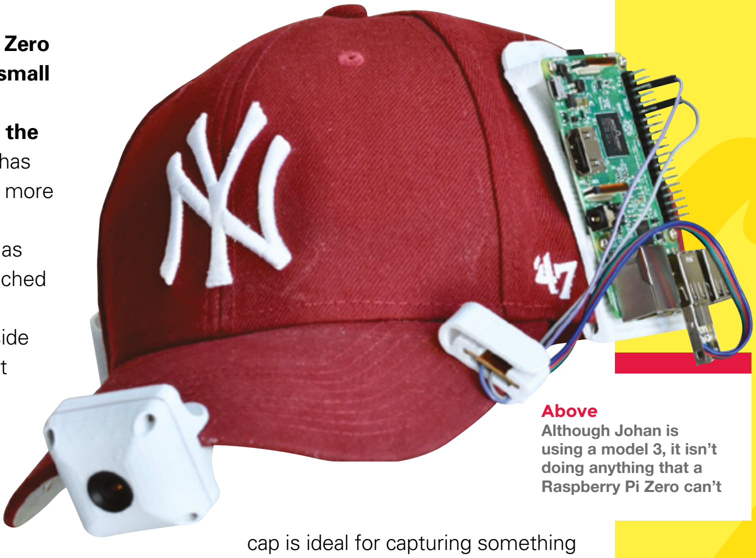
ELECTRONIC FASHION

BUILD YOUR OWN WEARABLE TECH

Even before the debut of the Zero model, a Raspberry Pi was small enough to fulfil all kinds of computing requirements at the edge. The Raspberry Pi Zero has just made the platform all the more embeddable.

So much so that Tyler Suard just uses one as his smartwatch (**hsmag.cc/CFLPog**). He attached a Waveshare 1.33-inch LCD screen to a Raspberry Pi Zero and put the contraption inside a 3D-printed case to create a smartwatch that has Bluetooth and WiFi. While the build isn't very difficult to replicate, it does involve some trickery. For starters, you'll have to first solder the headers to the Zero upside down to be able to connect it to the LCD screen: according to Tyler, the shorter ends of the headers fit the Waveshare screen perfectly. Secondly, besides the case, the build also requires 3D-printed spacers to separate the screen and the Raspberry Pi board.

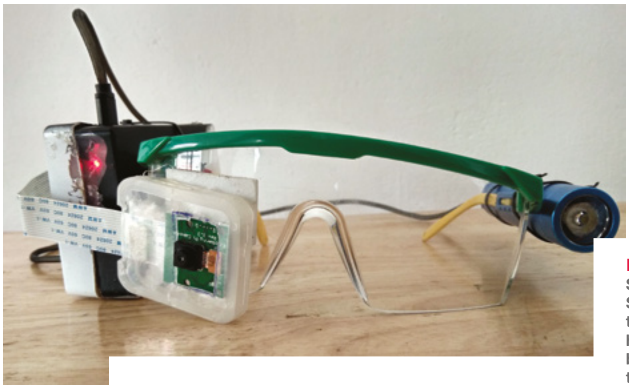
Johan Link has also used a Raspberry Pi 3 to make an interesting wearable (**hsmag.cc/XPRa8C**). His smart cap has aUSB camera that is constantly on. However, it doesn't fill your memory card with useless footage: the cap will discard all footage by default unless you press a button, in which case it will record the last seven seconds of video. This might sound strange, but it helps capture those events that don't give you enough time to hit the record button. For instance, Johan says his smart cap is ideal for capturing something like a passing meteorite. Besides the electronics, the build uses some 3D-printed parts to mount the Raspberry Pi board and clip the camera to the hat. It also uses a 5000 mAh battery, which should keep it going for several hours.



Above
Although Johan is using a model 3, it isn't doing anything that a Raspberry Pi Zero can't

You can also use Raspberry Pi boards to build intelligence into glasses. Salman Faris and Suhail have harnessed machine learning to create a pair of smartglasses for the visually impaired (**hsmag.cc/BWGXoV**). The design is pretty simple and straightforward to implement. They have mounted a Raspberry Pi 3, along with a Camera Module, on a pair of safety glasses. A SparkFun 12 mm push-button switch is wired to the

Right
Tyler spent weeks programming the Waveshare LCD, but when he couldn't, he asked the company for a disk image that he flashed to the microSD card



Left
Salman and Suhail want to replace the headphones with bone conduction transducers to make their smartglasses useful for hearing-impaired users as well

Left
Matthew also shows you how to replace the headphones on his PiGlass with Adafruit's Bone Conductor Transducer

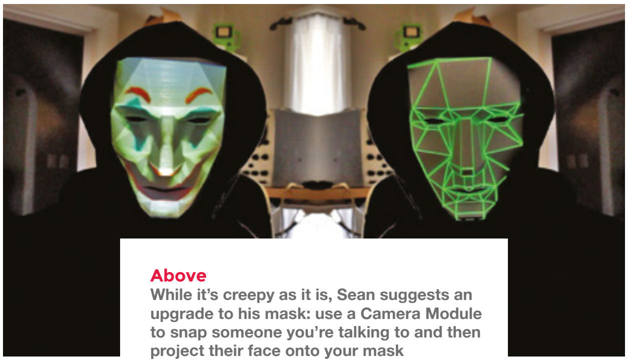
YOU CAN ALSO USE RPI TO BUILD INTELLIGENCE INTO GLASSES

Raspberry Pi's GPIO via a custom HAT created on a perfboard. When pushed, it'll take a picture and analyse the image using TensorFlow. Once the image has been classified, the information is relayed to the wearer via the connected headphones. While the first version of their smartglasses is powered by Google's Android Things, the makers are experimenting with a move to Raspbian.

MORE WEARABLE TECH

If you want something more extravagant, there's Matthew Desmarais's PiGlass (**hsmag.cc/xloazX**). As its name suggests, it's a Google Glass-like device powered by an Raspberry Pi Zero W together with a pHAT DAC and a Vufine+ wearable display that's mounted onto a pair of safety goggles. The PiGlass can recreate many of the functions of Google's discontinued eyewear, in that they can take pictures, record videos, and play music. It captures video with the Zero Spy camera, which is then processed by OpenCV before being sent to a preconfigured Dropbox folder. You can also stream YouTube videos to the glasses with the Raspicast Android app, as well as live-stream videos from the glasses to YouTube.

While we love all the smart wearables, Sean Hodgins's projection mask (**hsmag.cc/zoKG6j**) has just blown us away. It's a brilliant concept that will give privacy advocates something to chew on. Sean interfaces a Raspberry Pi Zero W with the DLP LightCrafter Display 2000 projector to project faces of anyone on to a custom mask. The project uses a custom PCB in order to adapt the projector that's designed to interface with the BeagleBone, to connect with Raspberry Pi GPIO pinout. The mask itself is 3D-printed, along with four other parts that will house the electronics. Thanks to Sean's simple instructions, creating your own projection mask isn't the most cumbersome part of the project. However, you will have to spend some time creating and fine-tuning the images and videos you want to project on to the mask to make sure it stays in line while you are wearing it. →

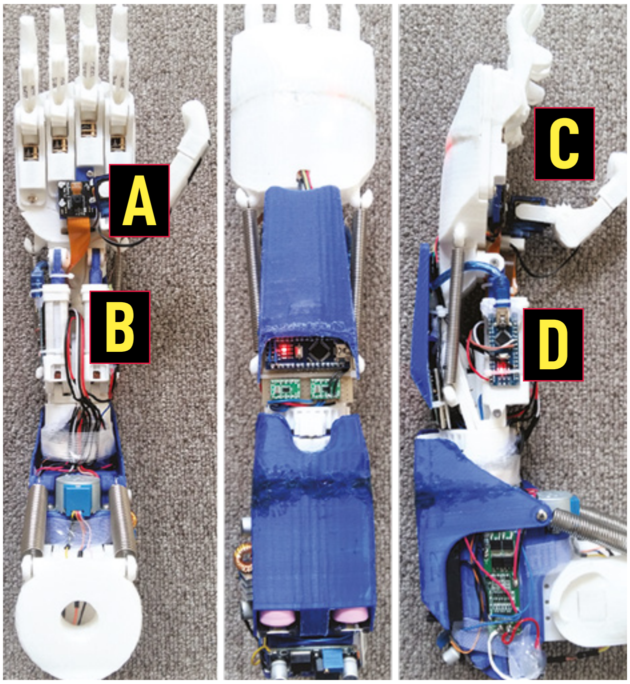


Above
While it's creepy as it is, Sean suggests an upgrade to his mask: use a Camera Module to snap someone you're talking to and then project their face onto your mask

PROJECT ARMED

AN OPEN-SOURCE SMART PROSTHESIS

When Vietnamese engineering student Duy Nguyen Phuong met Thuong, a victim of the dangerous herbicide and defoliant chemical known as Agent Orange, he decided to build her a low-cost prosthesis.



After two years of research, Duy has given Thuong a 3D-printed fully functional prototype (**hsmag.cc/nDogdc**) that's controlled by a Raspberry Pi board and several Arduinos. Interestingly, the project makes use of computer vision in order to detect objects and choose an appropriate grip pattern. Duy says the final design has ten degrees of freedom (DOF) and can perform complex movements. The best thing about Duy's project is that it's fully open-source.

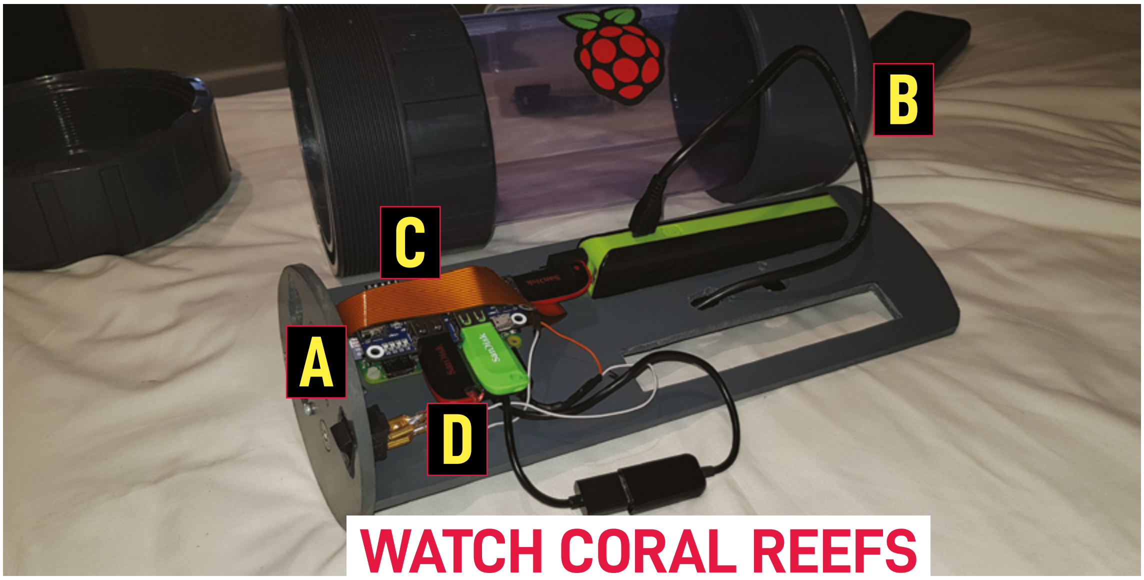
A The system implements a hybrid active vision system to recognise and track objects. The project uses TensorFlow to train and evaluate the object recognition model for the system.

B The wrist consists of two linear actuators and a third pivot point to bring about two DOF of tilting and waving. The rotator is added between the wrist and the forearm, providing the design with one more DOF.

C The hand module consists of four finger modules that are driven separately. Each of these has one DOF, and a thumb module has two DOF.

D The forearm holds all electronics and power supply for the entire system and is positioned beneath the spring-loaded elbow module.

PIPECAM



WATCH CORAL REEFS IN FAST-FORWARD

Fred Fouri is an electronics technician for an engineering firm in Cape Town, South Africa, that specialises in marine sciences. A long-term time-lapse video of corals fighting with each other for space piqued his interest in the study of biofouling – the accumulation of plants, algae, and microorganisms such as barnacles. Inspired by the OpenROV project, Fred decided to build an affordable underwater camera rig using Raspberry Pi Zero that he calls PipeCam (**hsmag.cc/CsbgjH**).

A Fred first used an Raspberry Pi 3 and then an Raspberry Pi 2 before switching to Zero to reduce power consumption. A cron job calls a Python script to check for available space on the attached USB drives, before it snaps a picture or records a video.

B The PVC pipe and the end caps protect the electronics from the elements. The leak-proof housing has a 10 mm Perspex lens that withstands four bars of pressure.

C PipeCam's Camera Module has been giving good results and surprised Fred with its underwater performance. However, the ribbon cable is a little delicate and he fried one during his tests.

D The device uses SanDisk Cruzer Blade USB drives plugged into a four-port USB pHAT. It uses `statvfs` to find the drive with the most free space whenever the cron job calls on the script. →

HOLOGRAPHIC AUDIO VISUALISER

THE ULTIMATE MUSIC ACCESSORY

Using some acrylic sheets mounted on a wooden frame, Aaron Tainter and Davis Lau, the hacker duo at Hacker House, have managed to transform a regular LCD monitor into a Raspberry Pi-powered 3D holographic music visualiser that's a sure head-turner at any party (hsmag.cc/AKD37I).

The device projects an image from a monitor down onto an acrylic pyramid, or frustum, which then creates the holographic parallax effect. To add to the futuristic aura of the device, Aaron and Davis have connected a Flick board from Pi Supply, which enables them to use Jedi-like hand gestures to control playback, volume, and visualisations. The Flick board is connected to a Raspberry Pi 3 that processes the gestures and sends them to the visualiser, which is running on a more powerful computer that can smoothly run WebGL graphics at a decent frame rate.

Q What was the inspiration behind the project?

A We acquired a gesture control board and we were brainstorming things to use it for. I had done some 3D audio visualiser graphics with WebGL in the past for a personal project. At the same time, we saw some videos online about how to make mini hologram projectors with a CD case and a phone. We thought it would be interesting to merge those things together to make a really futuristic visualiser.

Q What was the most challenging part of the project? Is there anything you would have done differently?

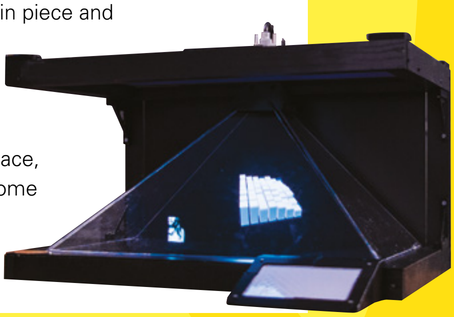
A The construction was pretty straightforward, but the code took quite a bit of time to write. I would have liked to add more animations or a simple interface for anyone at home to add their own visualiser.

Q Why did you use a computer to run the visualiser instead of the Raspberry Pi board?

A We initially tried running the WebGL graphics on the Raspberry Pi 3, but it wasn't able to run them at a smooth frame rate. Ideally, we'd power everything from one device, but the control board didn't have an SDK that could be run on a normal UNIX machine.

Q This project involves a lot of physical construction. What things should people keep in mind before replicating the project?

A The Plexiglas is very difficult to cut, so I would make sure to use a thin piece and create some paper templates to verify that everything fits properly. If they don't fit perfectly in place, you can always use some clear tape at the bottom to hold them together.

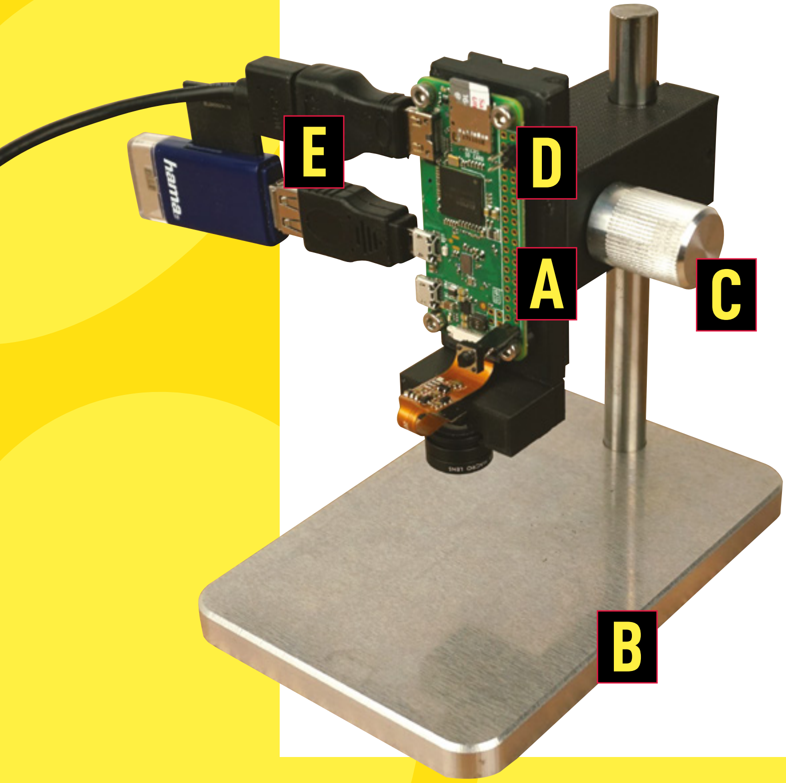


PI MICROSCOPE

MAKE VIDEOS OF TINY THINGS

Leon Brauns wanted a digital microscope that'd let him record videos for a reasonable price. But instead of hunting for bargains online, he decided to build one using a Raspberry Pi Zero W (**hsmag.cc/1IJR7O**). His microscope pairs an Camera Module V1.3, together with a cheap smartphone 20× macro lens. This is good enough to find defects in your 3D prints, inspect your electronics, or just get an interesting magnified perspective on everyday objects. While Leon is pleased with the traction from his setup, his design is adaptable, and you can also use it with other lenses.

THIS IS GOOD ENOUGH TO FIND DEFECTS IN YOUR 3D PRINTS



A Raspberry Pi and the camera are the main components in the microscope head. A script auto-starts the camera on boot.

B Leon has used some aluminium parts, particularly for the base, but these can also be 3D-printed like the other components.

C The contraption uses a rack-and-pinion design to be able to move the microscope head up and down along the steel rod.

D Leon has added a button on the last two GPIO pins. It's triggered by a custom shutdown script that powers off the Raspberry Pi Zero W.

E While he has connected a HDMI monitor to view the feed from the camera, you can stream the video instead using the UV4L streaming server. □

SUBSCRIBE TODAY

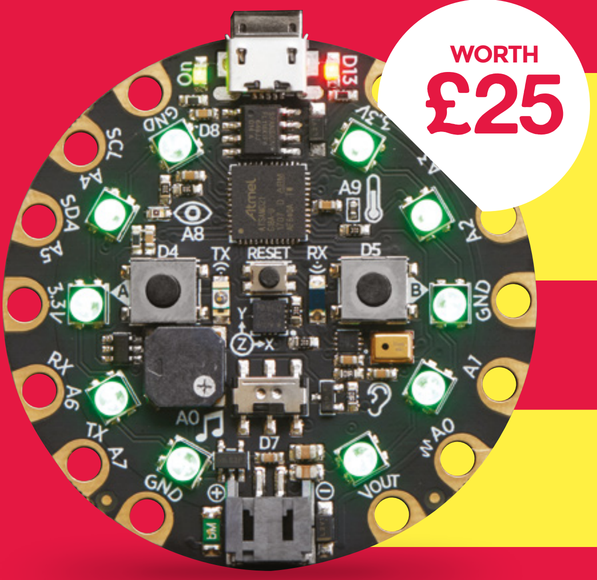
Subscribe for 12 months	Rolling monthly subscription

£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**



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

MAKE BUILD HACK CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
BEST RASPBERRY PI PROJECTS
PYTHON SPARKS POLYSTYRENE NUTS & BOLTS

How I Made
THE FIXIE CLOCK

A modern twist on a vintage display

By **Derek Woodroffe**

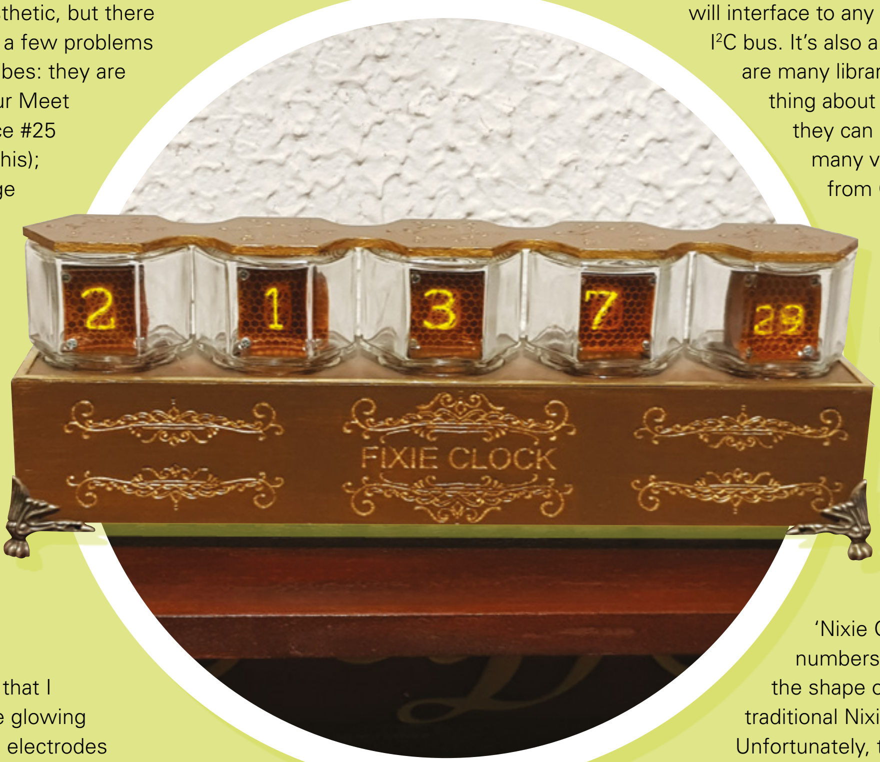
What is a Fixie? There are a lot of Nixie tube clocks available, and I really love the aesthetic, but there are a few problems with traditional Nixie tubes: they are relatively expensive (our Meet the Maker in HackSpace #25 charges $145 each for his); they need a high-voltage power supply; they only display numbers; and they need a large number of driver transistors per digit, making the circuit quite complex.

I also liked the challenge of making something Nixie tube-like, without using Nixies. So the Fixie (Fake Nixie) was born!

THE FIXIE

There are a few things that I needed to emulate: the glowing orange colour, the wire electrodes (that make the glowing numbers), the grid, and the glass envelope.

For the base of the display, I decided on a white monochrome 1.3" 128×64 OLED graphic display. This was partly because I'd used them before, but mostly because I had a couple of them available to experiment with. These small OLEDs are easy to drive as they have an I²C interface requiring only four wires: 0 V, +3.3 V, clock, and data, and will interface to any processor that has an I²C bus. It's also an advantage that there are many libraries available. The best thing about these displays is that they can be purchased from many vendors and very cheaply from Chinese suppliers.



My first test was to hook one of the displays up to a Raspberry Pi, and I quickly created a program using Python and the Adafruit SSD1306 library. The SSD1306 includes TrueType font support, and a quick search on Google found a TrueType font called 'Nixie One' that would output numbers which closely matched the shape of the glowing wires in a traditional Nixie tube.

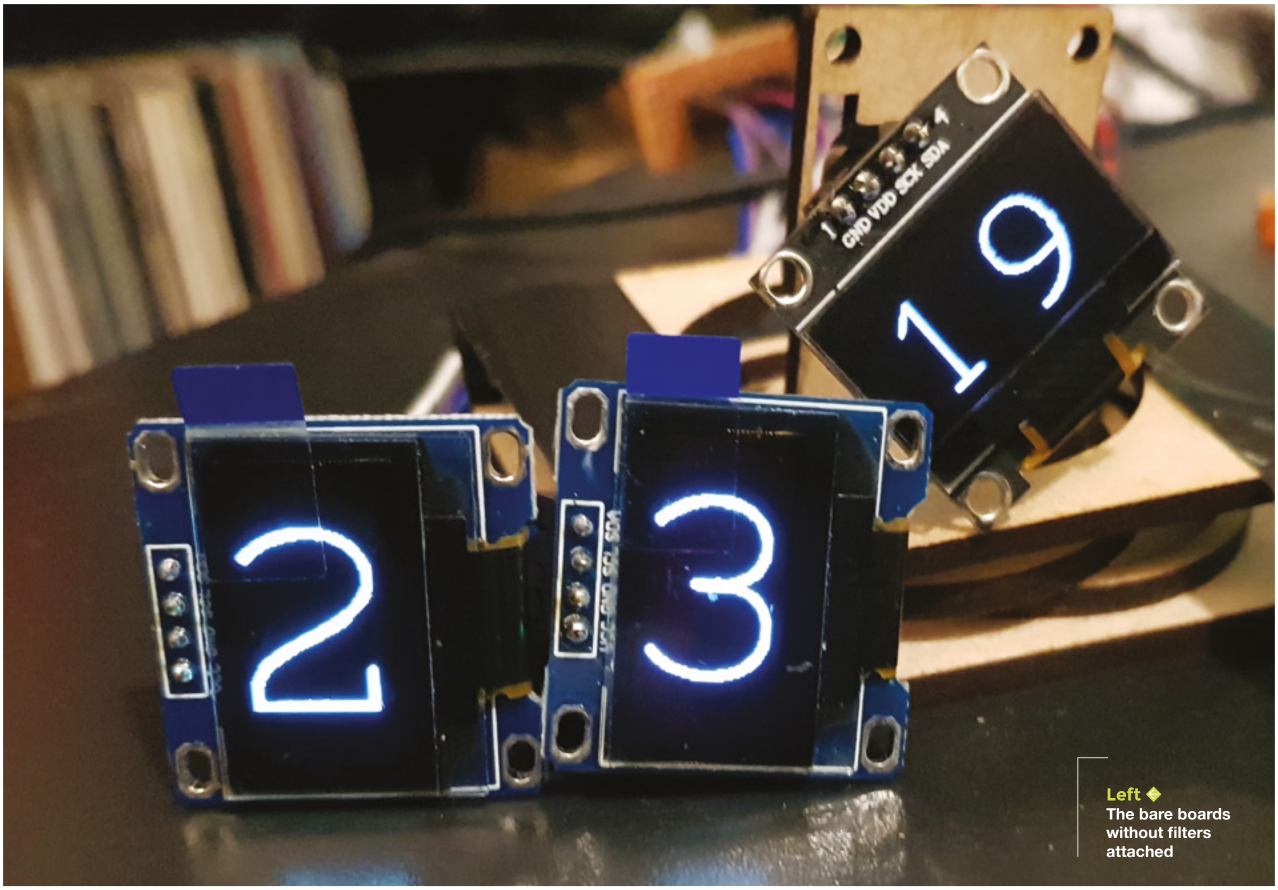
Unfortunately, they don't do these OLEDs in orange to match the warm glow of a real Nixie tube, but this is easily sorted by using an orange sheet of acrylic as a filter in front of the OLED. This has the added benefit of providing a surface that I can add the other details onto.

Adding the look of an electrode grid to the acrylic filter was easy: I lightly laser-engraved a honeycomb shape onto the acrylic. Etching the unlit wires seen in Nixies was more of a challenge. The problem was that the 'Nixie One' font was an outline of a number, not a single line, so when I tried to etch this font onto the acrylic I got two lines, an inner and outer line, not a single line. I could have traced the centre line of the digits to make a single line, but this would have been a lot of work. Twitter came to my aid, suggesting I use the single-stroke font 'Hershey Sans 1-stroke' from the EggBot Inkscape extension. With this font and Inkscape, I could etch the fake wire lines into the orange acrylic to give the effect of unlit Nixie wires. Unfortunately, the 'Nixie One' and 'Hershey Sans 1-stroke" fonts don't quite match, but no one has noticed yet and I'm not telling anyone, so I'm happy.

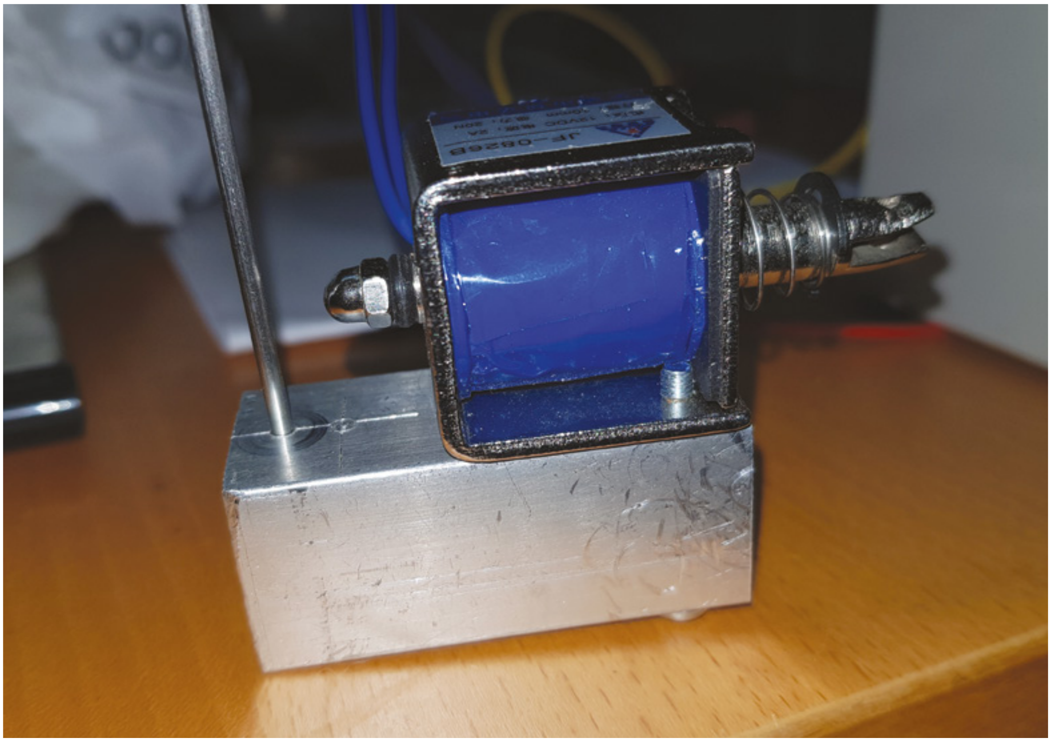
> I decided that I was going to use jam-jars as the glass envelope because I have zero glass-blowing skills

Early on, I decided that I was going to use jam-jars as the glass envelope because I have zero glass-blowing skills, and they are available in many shapes and sizes. The OLEDs dictated the size of the jam-jar as I needed to be able to get them through the mouth of the jar. Again I was led by what was easily available, and I managed to find some hexagonal jam-jars with a mouth of 47 mm and flat to flat 50 mm by 45 mm deep. This just allows one display in each jar. It would be well worth experimenting with other shapes and sizes of both jar and OLED, although I like the hexagonal look; it is possibly the most non-Nixie-looking bit of the design. I mounted the displays in a portrait orientation to get the digits as large as possible.

To mount the display, I laser-cut some strips of 3 mm MDF so they would go →



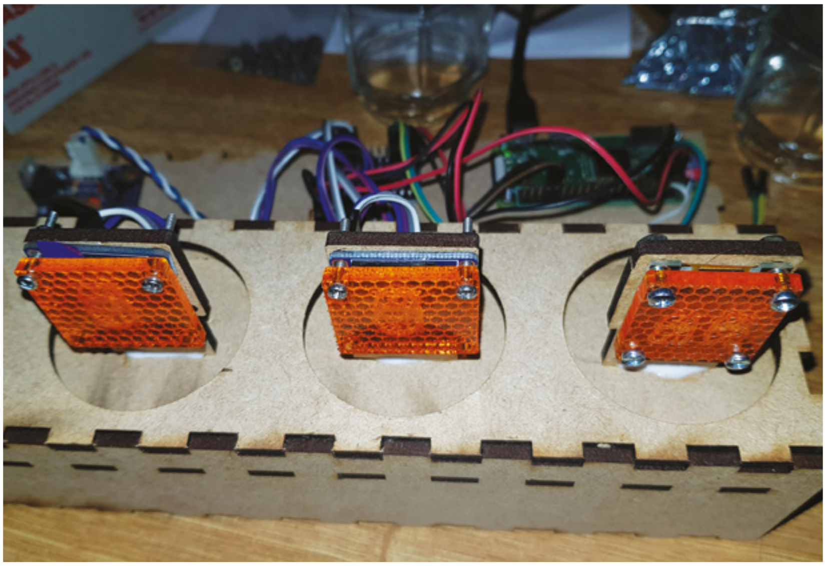
Left The bare boards without filters attached



Right A solenoid waiting to chime

Below Fixes mounted in place

> I had a total of five displays, so I decided to use two displays for the hours, two for the minutes, and one for the seconds

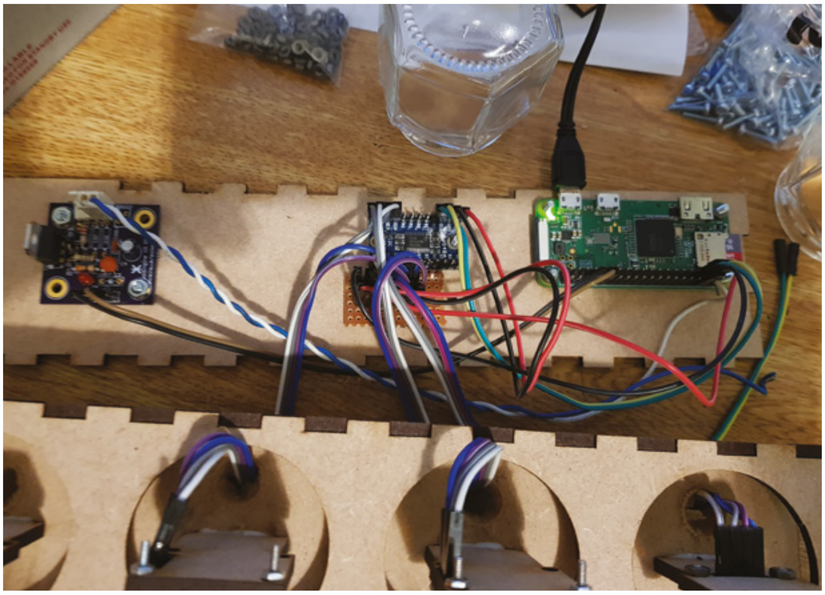


through the mouth of the jam-jar (with the jar upside down), the filter and display are screwed onto this with M2.5mm machine screws and nuts. The jam-jar is supported by another piece of 3mm MDF with a hole that allows me to tightly screw in the jam-jar mouth with its own thread.

THE CLOCK

So I have a working Fixie! There is, of course, only one killer app for Nixie tubes, and the same must be true for a Fixie: I needed to build a clock.

For a working clock, I needed to put four or more of the Fixies together. The problem with this is that the displays only have the options of responding to two I²C addresses, and the Raspberry Pi will only support a single I²C bus (in hardware). Luckily there is a simple solution: a TCA9548A I²C multiplexer. These are devices that allow the processor to switch the single I²C bus to multiple I²C devices and are available from a number of sources as a breakout to 2.5mm pins, making connection easy. Because you need to move a surface mount link to change the addresses of the displays and I had eight I²C channels available from the multiplexer, I decided to leave all of the



Left
The control electronics

displays on the same I²C channel and wire each of the displays to a separate bus from the multiplexer.

I could only buy the displays cheaply in multiples of four, and with my single white test display, I had a total of five displays, so I decided to use two displays for the hours, two for the minutes, and one for the seconds.

Because each display uses up to 40 mA, I needed a total of 200 mA for five OLEDs. This is more than the current available at 3.3 V from the Raspberry Pi's regulator, so I added a separate 3.3 V regulator to supply all of the displays.

I wired up all of the display's power to the external regulator and each of the I²C buses to the multiplexer, plus, of course, the master I²C from the Raspberry Pi to the multiplexer. I used a Raspberry Pi Zero W for its size, and the WiFi allows the clock to connect to the internet to get its time as the Raspberry Pi natively doesn't have an internal real-time clock, although this is easy to add. One advantage of using internet time (NTP) is that the clock is self-setting.

TOO MANY PINS ON A NIXIE

As I only had five OLEDs, the second display shows seconds in double digits from 00 to 60. This is done by having a smaller filter etched with twin digits, and the software sends two digits to the OLED. I've also spun the OLED around into a landscape orientation to allow the two digits to be larger. The unintended result of this is that, when Nixie tube owners look at the clock, they will often question how many pins the 'seconds' display Nixie tube has (real Nixie tubes have eleven pins per digit). The answer of four causes some interesting conversations.

BONG!

So, I thought this had gone too easily so far, but what the clock needed was a chime.

After a number of experiments, I found that a single 2 mm piece of brass rod tightly fixed into a largish piece of aluminium made quite a nice sound when struck with a screwdriver. So I fixed a push-type solenoid to the aluminium, then supplied a short pulse to the solenoid causing it to bang against the rod. It bonged, but badly.

The problem was that the strike must pull back from the rod quickly after the strike. I have mostly solved this in software, but a proper hammer mechanism rather than a heavy solenoid armature and spring would improve this. →

Software

The software is completely written in Python and should work with both version 2 and 3. It runs on a Raspberry PI Zero W with Raspbian Buster Lite.

Internet connectivity needs to be set up either by a direct connection or WiFi, and the Raspberry PI needs to be able to set its own time via NTP (the default configuration for Raspbian).

The software is available at **hsmag.cc/8BXu9E**. The files below need to be saved in the same directory.

Config File – oledclockconfig.py

Routines – oledclockroutines.py

Main – oledclock.py

The Nixie One TTF can be downloaded at **hsmag.cc/4ga57C** and must be saved to the same directory as the other Python files.

The libraries that need to be installed for the OLEDs:

With Python 2

```
sudo python -m pip install --upgrade pip
setuptools wheel
sudo pip install Adafruit-SSD1306
```

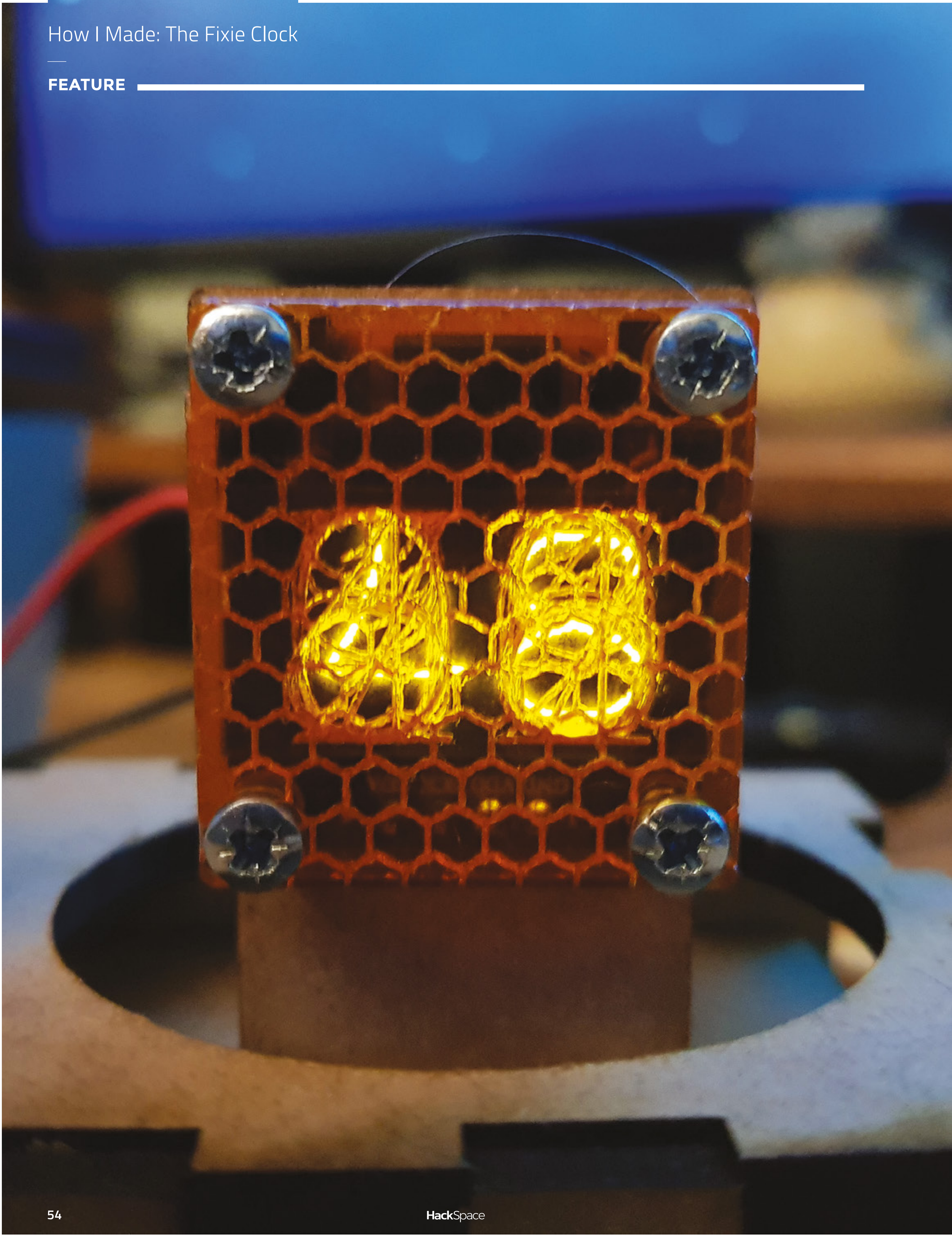
With Python 3

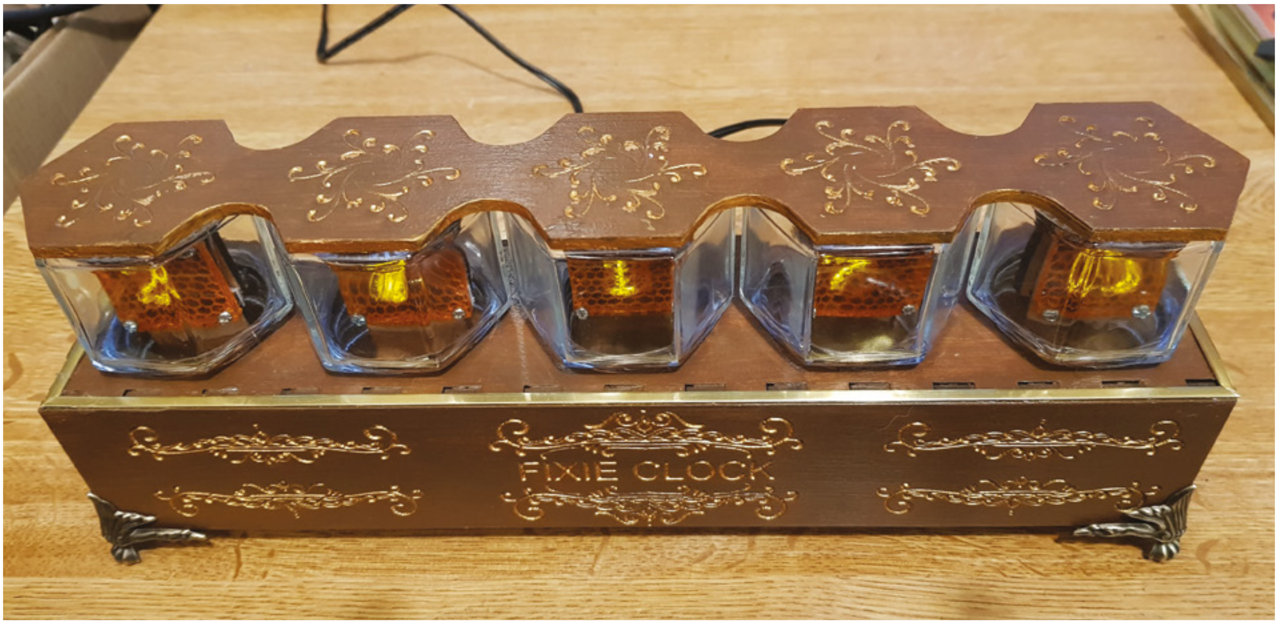
```
sudo python -m pip3 install --upgrade pip3
setuptools wheel
sudo pip3 install Adafruit-SSD1306
```

The software will need to be run as root.

To run the software automatically on power up, edit the **etc/rc.d/rc/local** file and add in the lines below immediately before the exit 0

```
cd /directory_with_files/
/usr/bin/python3 oledclocknag.py
```





Above The final clock

The solenoid I had to hand was 12 V, but all of the rest of the clock ran from either 5 V or 3.3 V. I didn't want more than a single power supply/cable to the clock, so I needed to get 12 V from the existing rails. To provide this, I bought a cheap 5 V to 12 V DC-DC converter. To drive the solenoid, I fitted a TIP121 NPN transistor driven from the Raspberry Pi's GPIO and a diode/resistor snubber to remove any spikes.

So now every time the solenoid engaged, the Raspberry Pi crashed. I suspected a current overload on the 5 V supply. My first attempt at resolving this was adding a reservoir cap to the 12 V supply; this made things worse. I figured this was due to the DC-DC converter doing exactly what it was designed to do. When the 12 V supply dipped, because the solenoid took current, it would take 2 A or more from the 5 V 2 A power supply to charge it back to 12 V again.

I eventually solved this by inserting a 5 ohm resistor in the 5 V feed to the DC-DC converter to limit its maximum draw on the 5 V supply. It now takes longer to charge back up, but much less than the gap between chimes.

THE CASE

Making a case was relatively easy as the box needed was rectangular, with five holes in the top for the jam-jars and a support for the displays. I laser-cut this from 3 mm MDF using tabs to align the sides. The Raspberry Pi, multiplexer, display PSU, chime PSU, and the chime and solenoid are all screwed directly to the bottom of the case. The chime rod sticks out of the back of the case, and there is also a hole in the back of the case to allow the USB power connector to the Raspberry Pi.

The case is stained with a dark oak wood stain. The detailing on the case front and on top of the Fixies is achieved by deeply laser engraving the design onto MDF, then staining, and filling the engraved areas with gold acrylic paint. The case was then finished off with a brass strip, glued around the top to cover up the visible tabs and four moulded brass feet screwed to the bottom. The feet allow the bottom of the clock (with all the screws heads holding the electronics) to stand away from the surface, as scratching the shelf it sits on was not worth my life.

I hope this has given an overview of my Fixies and the Fixie Clock. It would be great to see more designs for Fake Nixies, and maybe fake VFD displays using modern, cheap displays. □

> The case was then finished off with a brass strip, glued around the top to cover up the visible tabs

Downloads/information

Nixie One font from Google Fonts: **hsmag.cc/4ga57C**

Adafruit driver details SSD1306: **hsmag.cc/36K3Sk**

EggBot Inkscape download: **hsmag.cc/gzVeo2**

IN THE WORKSHOP: 3D printers

By Ben Everard

The by-products of 3D printer reviews

Over the last couple of months, we've been testing out a lot of bits with 3D printers. Partly, this was for the feature in issue 26, partly for printer reviews (see the CR-10 V2 later in this issue), and partly this has been preparing things for future issues. This means that we've had to run a lot of test prints. Some of these have been the classic Benchies, temperature towers, and other such tests, but we've also just run a lot of models through the printers to test settings and see how they run, day in day out.

It's been a great chance to run through some models from model sharing websites, and see what works and what doesn't.

Plantygon

There's something about the juxtaposition of industrial shapes of hard lines with the natural leaves of a plant that is just appealing. This planter is designed to allow many similar ones to slot together to build up an arrangement of plants as large as you like. There is quite a small amount of space for soil, so they're best suited to plants such as succulents that cope well with such conditions. The standard model includes a drainage hole, but we used the adapted model without one, as we'll be using it inside.

This model also serves as a stringing test, with the added advantage of making something useful at the end of it. The example shown here was printed on a Creality CR-10 V2 which – with its Bowden tube – can be susceptible to stringing. However, as this example shows, with the right settings in the Creality Slicer, it doesn't have to suffer from this.

thingiverse.com/thing:2448911

Vase

Search for vases on a 3D printing model site, and you'll quickly become inundated with them. There are good reasons for this: they're easy to print, use little material, and can show off your design skills around the home. This is just one example, from the plethora available.

The important thing with printing vases is to make sure you slice them in the correct way. Many (but not all) are designed to be printed in 'vase mode' where a single bead of filament is fed around in a spiral, making a vase with one perimeter. This thin wall will let a small amount of light through, giving a shimmer effect. Printed in PLA, they're strong enough to hold light objects upright, but aren't usually fully waterproof. You can add a waterproofing sealant if you want to use it for live flowers.

This particular design comes included with a Prusa MK3S kit, but many other designs are available on the various model sharing websites.

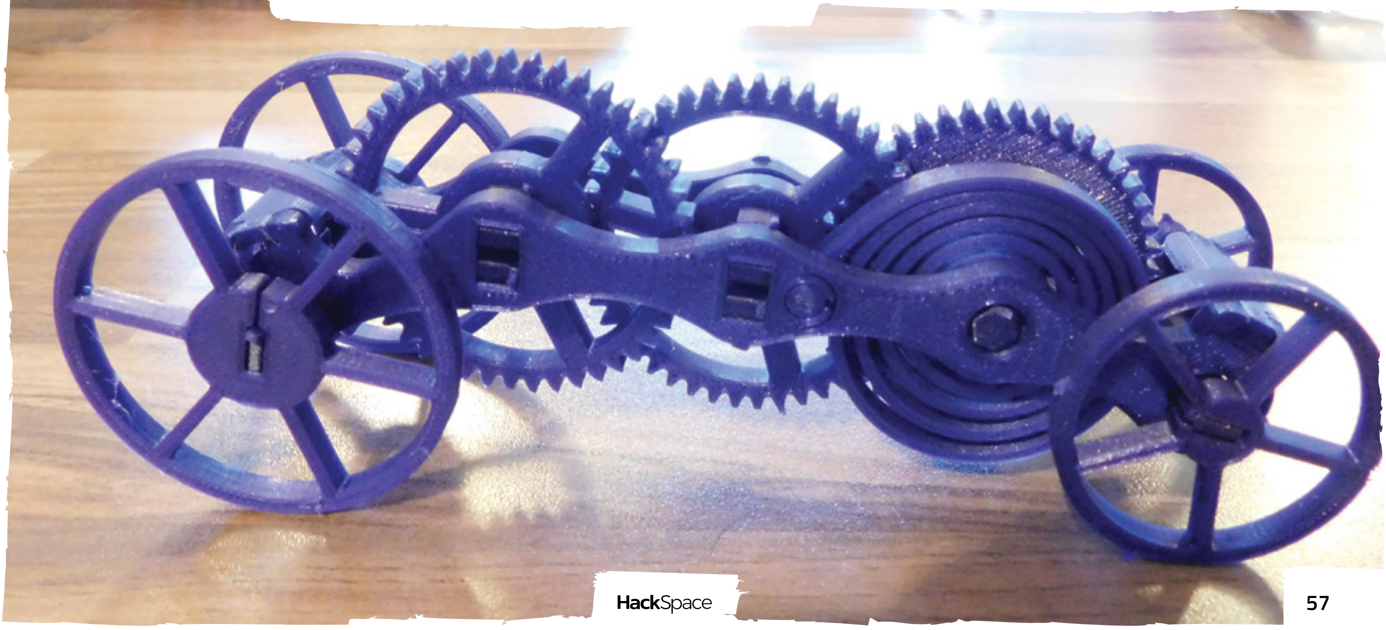
prusaprinters.org

Wind-up card car

This car prints as two panels, sized to fit in a typical greetings card. The parts can then be snapped out and assembled. The instructions leave a little to be desired – the pictures make it a little hard to see the difference between some similar pieces, and the spacers are omitted entirely (but do make it run much better when in use). That said, it's not too complex to work out how to put it all together. Our advice is to start by looking at how the gear train lines up, and then see what bits you need to hold it in the right place, and work from there.

Once assembled, the car works as expected. Turn the key (you can get about one revolution before the spring has fully wound), put it down, and watch it go. The only problem we found was with the wheels slipping on smooth surfaces. On a hard, smooth surface, it tends to wheel spin quite a lot, but as long as the wheels grip, it shoots forward.

thingiverse.com/thing:3308710



You will always need to store lots of little things

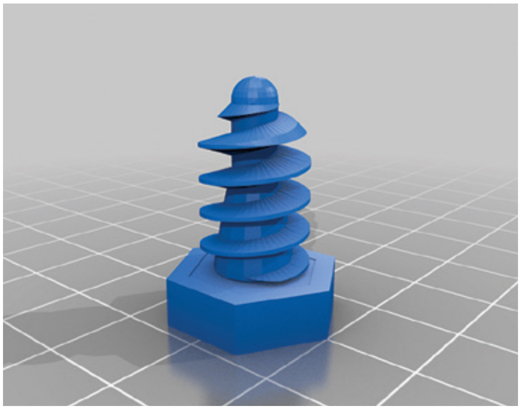
Cardboard screws

Online shopping has become an indispensable part of life for many people, and that often means piles of cardboard. While cardboard can be recycled, why not give it an extra bit of life before putting it out for your waste collection? These screws are designed to be compatible with both the Makedo system (in fact, the hinge shown here is a model shared by Makedo), and the Lego Technic system. This author's living room has been taken over by a large cardboard house that he made with his daughter – he's particularly proud of the braced roof, and is wondering if it's capable of supporting a second story. We suspect that he had more fun building it than she does playing in it.

Poke a small hole in a sheet of cardboard, and you can screw these in to attach sheets together to build almost anything. There's a range of other bits and pieces, such as right-angled brackets that you can add in to your creations.

The screws are also a good test of your printer. Although there's a steep overhang, we were able to print without supports on the Prusa MK3S (and when we tried with supports, we couldn't remove them).

thingiverse.com/thing:3867969

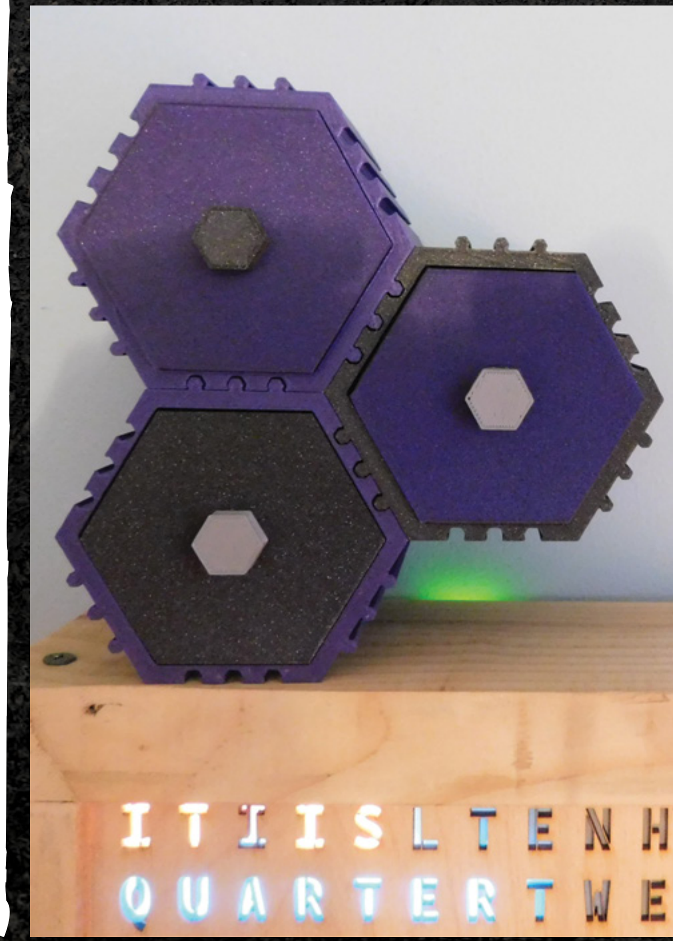


HIVE

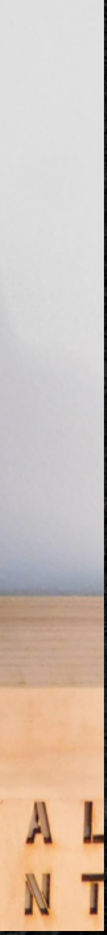
It's more or less an inevitable part of being a maker that you will always need to store lots of little things: wood workers have screws, sewers have cotton reels, electronics enthusiasts have components. The HIVE system is made up of hexagonal drawers that you can slot together to build a storage system of any size you need. The ridge-and-groove edging doesn't quite hold everything together, but with a drop of glue, you can firmly attach everything together.

There's a couple of different options for the drawers – we recommend the deep version, as that gives you more space per module. There's also some brackets for wall mounting, but it can also be free-standing. There is a danger with a system like this that you'll end up with a storage system that continuously grows, which does raise the question as to whether you need a storage system for the storage system.

thingiverse.com/thing:1743145



Let's be honest, who doesn't need a few more clips?

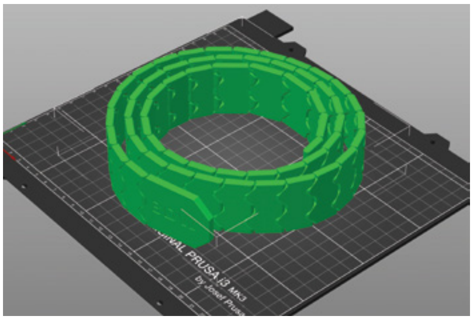


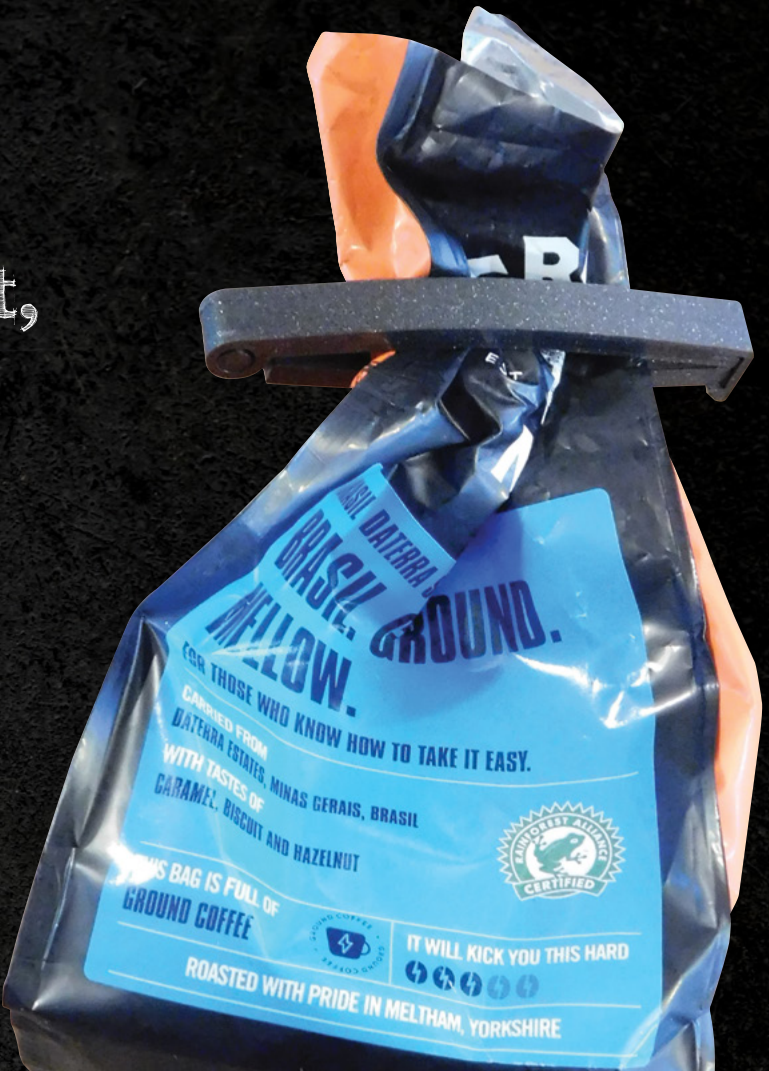
Belt

There's not really a good reason to think that 3D-printed PLA is a good material for a belt. It's stiff and brittle. However, that hasn't stopped the enterprising 3D printer community from designing one. We printed this out and wore it for about a week before it broke. Unfortunately, it broke as this author was attempting to corral a toddler through the caves at Wookey Hole. Stooping amongst the stalactites isn't the easiest thing to do when trying to hold up your trousers, and help a two-year-old around.

While it was an interesting experiment, we can't really recommend this, unless you're looking for an embarrassing way of wasting filament.

thingiverse.com/thing:3270948




BRASIL DATERRA
BRASIL. GROUND.
MELLOW.
FOR THOSE WHO KNOW HOW TO TAKE IT EASY.
CARRIED FROM
DATERRA ESTATES, MINAS GERAIS, BRASIL
WITH TASTES OF
CARAMEL, BISCUIT AND HAZELNUT
RAINFOREST ALLIANCE CERTIFIED
IS BAG IS FULL OF
GROUND COFFEE
IT WILL KICK YOU THIS HARD
ROASTED WITH PRIDE IN MELTHAM, YORKSHIRE

Clips

OK, they're not the most exciting item in the kitchen, but let's be honest, who doesn't need a few more clips? Keep your coffee fresh, stop your biscuits going stale, and much more. There are loads of different designs available, but these basic ones printed well for us. They're quite small, so the perfect way to finish off a bit of filament that you've got left on a roll.

thingiverse.com/thing:330151

HackSpace magazine meets...

Mitch Altman

Nomadic open hardware champion and inventor...

Open hardware refers to hardware whose specifications are shared and licensed in such a way that anyone with the wherewithal can copy, learn from, and reproduce the original as they see fit. It is a different form of intellectual property to the traditional model of patents and closed innovation, and it has given rise to countless spin-offs tailored to the individual maker's preferences. By taking innovation out of the hands of large R&D departments and giving it back to home tinkerers, open hardware is slowly but surely changing the world.

One person who has a huge impact on open hardware is Mitch Altman. His most famous creation, TV-B-Gone, enables the user to turn off TVs in public places. It wouldn't have had anything like the success it did, had it not been open hardware, and after its release in the mid-2000s, it showed makers that opening up your hardware for the world to learn from made sense.

We spoke to Mitch a while back to ask him about the inspiration behind TV-B-Gone, how the makerspace movement got started in the USA, and what he's working on next. →

Above

Mitch started Noisebridge with help from people at Metalab in Vienna and Chaos Computer Clubs of Köln and Berlin

HS You were instrumental in bringing hackerspaces to the USA, where the idea has really taken root. How have things changed since you started Noisebridge in 2007?

MA There already were places calling themselves hackerspaces in the US before Noisebridge; they were the L0pht, in Boston and the Walnut Factory, which was in Philadelphia. And there's also Halfway Hacker House in New York. But that was in the 1990s, a time when a whole bunch of mostly adolescent boys were playing with their computers and "Oh, this can hook up through your phone to other computers, that's fascinating".

Some of it came out of the phone phreaking days, where people were helping each other figure out how the phone system works. A lot of people came in because they got a computer and a modem, and some of them were finding they could get into computers because there was no security then, or not enough to keep them secure.

Some of these kids were just playing around – they didn't know what they were doing. They found the word hacker from MIT, which meant something totally different then.

The negative aspect of it came from mainstream US media, because that's what mainstream media does: make people afraid of 'them'. People were getting arrested, the laws were brand-new and coming in, and it didn't really create an atmosphere of trust. So, these early hackerspaces were very elitist, mistrusting, adolescent. Brilliant people, but if a new person came along, you had to prove how cool you were, and it was all about software.

The scene went around the world, and people in Germany picked up on it and changed it. Little groups were forming around Germany. Once a year they would have a congress, where people would come along and compare notes. The first Chaos Communication Congress was 1984. Germans did a lot of this exploration, and through a bunch of different sets of circumstances, they actually became very respected for the most part in German society.

HS How did the Germans change it from the American version?

MA The thing is, these people were the freaks of Germany. Germany was forced after the war to stick its nose in its own [excrement], and a lot of people learned from that. In the UK, the Yanks come in and save the day, so the US never had to stick its nose in its own [excrement]. It would do us some good.

In Germany, people who grew up with parents who lived through the war in any form did not want to have that happen again. So, learning from the past, they wanted to forge new directions forward – and here's this cool technology, so they loved playing with the technology and making it accessible to everybody. And in actuality, not just for marketing purposes.

> "As we talked more and more about structure, we started to narrow down what our vision was"

So they made it welcoming, and it was mostly still just computer software people, but hardware started creeping in. They were much more welcoming than the Americans were.

In 2006 I was invited to my first hacker conference in NYC. It was HOPE – Hackers on Planet Earth, held once every two years in New York, in the summer. Also attending were people from CCC [Chaos Computing Club] in Germany who were helping. They told me, "If you like this, you'll love the ones we do in Berlin". They were right. I got invited to talk about TV-B-Gone at the 23rd Chaos Computing Congress. Seeing all these people doing stuff that they really loved exploring and playing with. In Germany though it's kind of weird, because people have this dividing line of the stuff that's wonderful and cool on the one hand, and work on the other hand.

This was one of the first times I ever felt really super-wonderful just being in my tribe. I was an introverted geek, and here were thousands of introverted geeks. You didn't need to be socially awkward, but it was OK to be that, because we were all socially awkward. It was totally cool to talk geek and learn from everyone.

HS What made you take that ethos to San Francisco?

MA One of the talks at the summer Chaos Camp was about how to set up your own hackerspace. And that gave me super inspiration. All my life I've done things with community. That's the kind of energy that I really needed and wanted in my life, and it was lacking. These hacker conferences gave it in abundance, but only for a short period. But the idea of a hackerspace that I could just walk to whenever I felt like it, with loads of super-cool, introverted, awkward people who are my tribe… what would it be like if one of those existed in San Francisco? I wanted to find out.

There were other people there with the same idea – we hung out and talked about it. When we got home, we started putting the word out. In New York, that led to NYC Resistor; in DC, that led to HacDC.

Anyways, back in 2007, I put out the word and soon we were having weekly meetings of 50 people. As we talked more and more about structure, we started to narrow down what our vision was, our mission, how to structure ourselves. And as we were a bunch of hippie, weirdo, anarchist punks, whatever, we created a hippy, weirdo, anarchist punk kind of a →

Above

"There are no rules about how to start a hackerspace… the whole idea is to organise yourself in the way that works for you."

Above

"Even when we disagree it's important that we respect one another and figure things out in a way that works enough for everyone."

hackerspace. No leaders, one rule – be excellent to each other – and run by consensus decision-making. It's a do-ocracy, which was a concept borrowed from the Germans.

HS That's a great concept. If something needs doing, do it.

MA Yep. And be excellent to one another while you do!

At Chaos Computer Club, they embrace chaos as one of the forces that are really important for a creative environment, not to banish chaos. But Germans being what they are, they're very good at organising as well. Not the overbearing type-A organisers who squash the chaos and squash any kind of creative atmosphere. They create a place that's organised enough for anarchy to flourish.

You're not alone because you have the help of all the other people, seeing what works and sharing it, because we're not just working on individual projects, we're hacking on ourselves and our communities along the way.

HS Do you consider yourself a Godfather-type figure?

MA Other people have called me that. Maybe not Godfather; maybe like a Yoda figure.

I'm 62 now, so I'm older than most people in the scene. But age doesn't mean anything unless you learn from your experience along the way. I've learned a bunch of things. A lot about community in particular; what works, and what doesn't.

I'm totally into supporting people. Even if they're starting a hackerspace that I don't want to be a part of, because it's too hierarchical or whatever, because I don't flourish in those kinds of places too well. But I like going to these places, visiting; as long as it's cool for the community there, I'm happy. If people there take what I share and do something cool with it, that makes me happy. I know anywhere in the world that I go, pretty much anywhere in the world that anyone can go, there are communities that anyone can pop into and say hello.

HS It's curious that you mention community – it feels to me like a lot of people making their own hardware do it not because it's cheaper, or better, but because they can share it with other like-minded people.

MA It's more fun, and you learn a lot. And you gain from that. The main thing is, you're doing something that you enjoy right now. The only way the world's going to get better is if people find ways to do things that are meaningful, spend time in ways that are meaningful, being supported in a community.

> "It's totally up to us what we choose to do with our time. There are consequences to the choices we make"

HS Do you think that means there's more growth to be had for hackerspaces?

MA Sure. There's more growth to be had until the Earth gets sucked up by the sun. We've got a good 15 billion years left – we may as well make use of it. However long we have in our own personal lives, however long or short, when we're alive is when we have the opportunity to do cool stuff. We also can choose to just watch a screen. It's totally up to us what we choose to do with our time. There are consequences to the choices we make, and we have to learn from those choices, and make better choices as time goes along. If we support each other in that, great: even better!

HS Speaking of watching screens, can I ask you how TV-B-Gone came about?

MA I grew up horribly depressed, beaten up every day by bullies, often as the gym teacher watched, and as a little kid, what do I know? I figured I deserved this kind of treatment. I hated myself, my parents couldn't handle it. And TV became my first addiction.

I tried to escape all of my pain through the screen. The nature of addiction is that it seems to work while you're using. But then once you stop, everything comes screaming back and you want to do even more. With TV, I'd be comparing myself to the kids on shows and getting ever more depressed because there was no way I could match up to them. They had wonderful lives; their problems would resolve easily in a half-hour chunk, and in my life, they could not. I was like, what's wrong with me? I just blamed myself more.

Years went by of being unhealthy, depressed, and getting beaten up by bullies, of staying away from people, being introverted, being fat, being a geek, being gay, and being everything bullies like to target little queers for. It was horrible. Years went away that way.

At age 23, I went cold turkey. When I was 23 there were no TVs in public places; I could just get rid of the one in my apartment and I was free of it. But then they started appearing in public places, and this horrible force that I got rid of in my life was suddenly invading me again. And I hated it. I hated it, and I still hate it.

I was in a Chinese restaurant in 1993 with friends I hadn't seen in two or three years. There were a few of us who met in university, and in 1993 were all in Palo Alto. Back then there weren't TVs in public places, except in horrible Chinese restaurants. We were in a horrible Chinese restaurant that had one thing on the menu we all liked, but we couldn't pay attention to each other because the TV was on. We all kept gravitating to look up in the corner where the TV was. We weren't there to watch TV. Even though I'm telling myself, 'Stop this, pay attention to your friends', →

Right ↘
On the subject of TVs taking over our minds, Mitch recommends a documentary called *Stare into the Light my Pretties*

I kept doing it. So, anyways, we were talking about that, and then I said, wouldn't it be cool if you could just get rid of these things? I instantly came up with an idea of how to do it, and one of my friends came up with the name right then and there. But then it took ten years to stop working long enough that I had time to play with electronics, rather than just work with it all the time.

In 2003 I quit consulting – I didn't do it to work on TV-B-Gone, I did it because I was tired of doing all this stuff for other people. Everywhere I've ever worked, without exception, the military would come in and want to buy it. And in all cases but one, the bosses said 'Yeah! It'll make us money, great!' And then I had to quit… I'd put all my time and energy and life into this project that's now being used by the military to make war on people, destroy people, destroy parts of the planet. And that's not why I do what I do.

The only one where the bosses didn't say yes was a virtual reality product. The military ended up getting it anyways of course, and used it for a World War III training simulator. On a system that I spent three months of my life putting together with my own hands.

HS Isn't World War III going to be very, very short?

MA Who the hell knows? If it's up to the US military, it's going to be horrible for everybody. So anyway, I just had to quit to explore what it might be like to make a living doing things that I found truthful and wonderful, things I loved doing. And in my year of exploration, I was lucky enough to be able to do a lot of cool stuff. One of the things was TV-B-Gone, which then ended up being exciting enough for other people to want to do it. But, I really only made it for me. Friends who were excited about it were encouraging and helping me in many ways. I made one for everybody – friends, friends of friends, friends of friends of friends…

One of the things I learned while being a consultant for large companies was how to manufacture. So I thought I could manufacture there. And it took off, partly due to an article on **Wired.com**. Evidently, enough people paid attention to it.

HS I guess when you threaten TV companies' revenue streams…

MA It wasn't a threat, though! There are, what, 13 billion TV sets in the world, and I sold almost 700,000 TV-B-Gones. That's just noise. It doesn't stop people watching. It makes some people question it, which I'm really glad for; that's why I put it out into the world, to let people have a fun way of realising that you actually have a choice whether the TV is on or off. It doesn't have to be on all the time.

TV's only gotten worse since then, and since I came up with the idea of TV-B-Gone, the internet has become the worst piece of mind control ever devised, and spying and bullying and life-destroying. It's also a powerful tool that can be used for all sorts of cool things. But like anything that's addictive, there are pluses and minuses, and the minuses win.

> "That's why I put it out into the world, to let people have a fun way of realising that you actually have a choice whether the TV is on or off"

HS What you said about watching TV and looking at people's perfect lives that you compare yourself to. You've sold 700,000, which obviously means people like it. What are you working on now?

MA Now it's kind of a transition time for me. I don't know what project I'll be doing next. I'm still all over the world giving talks and workshops, and I love doing that.

The project I have been working on for the last three years is pretty mature now – that's my synthesizer. I've done a lot of stuff with brain waves which kept me occupied for a long time. But there are no ideas that have been percolating in the back of my mind for a long time that are dying to get out. So I don't know what it'll be next.

It's kind of weird because Noisebridge has been a really important part of my life since 2007. And for the last year, I haven't been part of Noisebridge. And that made me realise that there's not a whole lot keeping me in San Francisco. The owners of my building in San Francisco offered me some money to move out, because they're selling the building. I accepted, so the city I've been in for 33 years and the apartment I've been in for 23 years is not going to be home any more. A bunch of stuff is up in the air. One of my favourite states is having finished all my current plans and deciding what to do next. And that means I can choose absolutely anything I want! □

Above
When we spoke to Mitch he was mid-way through a world tour of maker spaces

POLYSTYRENE FOAM

IMPROVISER'S TOOLBOX

It's time to forge from foam

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.

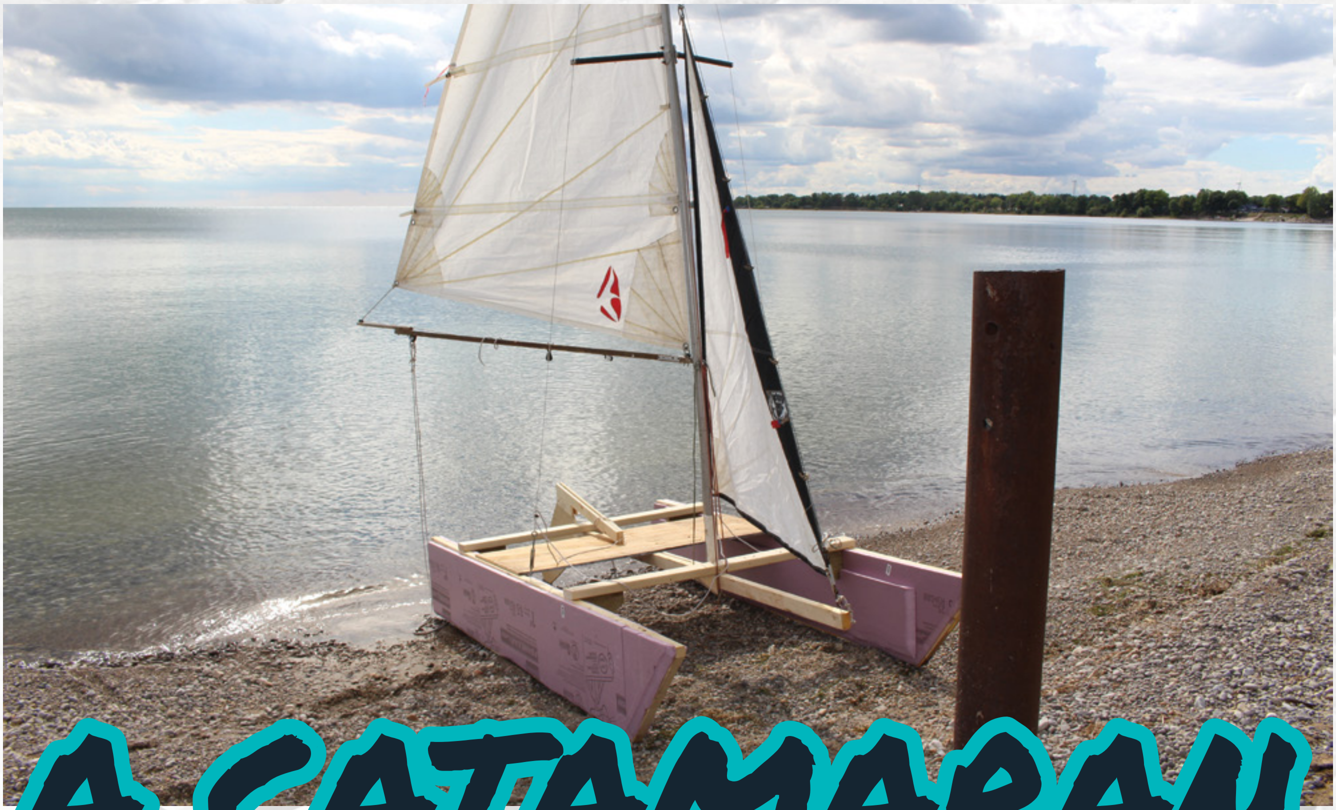
Although it's all around us, there aren't many who refer to the commonly used packaging material by its chemically correct name. Polystyrene is produced by the polymerisation of styrene, and is available in the form of a solid plastic and as a rigid foam material. The foam form of polystyrene is widely employed in the food-service industry in the shape of rigid trays and containers, disposable eating utensils, and foamed cups, plates, and bowls.

The polymerisation of styrene dates back to 1839 and German pharmacist Eduard Simon. However, the material couldn't find commercial acceptance for almost a century, mostly because of its brittleness. That changed when American chemist Robert Dreisbach modified the production process to create a more durable version in 1937. A few years later, in 1941, Dow Chemical Company invented a proprietary process to make their trademarked and well-known polystyrene foam product they called Styrofoam.

There are two types of polystyrene foam, and the one that we most commonly encounter is expanded polystyrene (or EPS foam). It is manufactured by expanding spherical beads in a mould, using heat and pressure to fuse the beads together. EPS foam was developed by the Koppers Chemical Company in Pittsburgh, USA. Anywhere between 96% to 98% of the foam is air, and the remaining is polystyrene. EPS granules are heated with steam in such a way that they expand rapidly and form a low-density block. EPS foam was earlier made with the aid of chlorofluorocarbon blowing agents, before they were banned for their detrimental effect on the environment. These days EPS is foamed by other gases, such as carbon dioxide.

The main reason for the popularity of EPS foam is that it's inexpensive and light, yet rigid enough not to lose its fundamental properties with time. While the material can be completely recycled and reused, it isn't looked at very favourably by environmental groups. Not only is EPS foam extremely slow to biodegrade, but it's also one of the major contributors to outdoor litter. According to the US Environmental Protection Agency (EPA), several billion EPS utensils are tossed into US landfills every year, and lots of it ends up as oceanic garbage. No surprise then that several countries are drawing up legislations to curb its use. While its use as a packaging material is on the decline, it'll be sometime before you stop receiving goods inside protective EPS housing. Until that time, here are some interesting ways you can upcycle the cushioning material.

> "ANYWHERE BETWEEN 96% TO 98% OF THE FOAM IS AIR, AND THE REMAINING IS POLYSTYRENE"



A CATAMARAN

Project Maker
Joshua Donald

Project Link
hsmag.cc/oHph5r

Instead of spending $2000 on a catamaran, Joshua, who runs the JoshBuilds YouTube channel, decided to build one himself using some pieces of wood and lots of Styrofoam.
Joshua's boat can sail with two people on board, and costs about a tenth of the price of a similar store-bought catamaran. As you can imagine, it's an involved process and requires lots of sawing and drilling. He starts by using some structural lumber to build a frame for two pontoons. Both the pontoons have struts at both ends and are filled with insulating foam. He then glues sheets of Styrofoam on top of the frames. Finally, he adds a wooden triangle at the top that will be used for mounting the frame before adding some more foam for buoyancy. Joshua then builds the frame by first connecting the two pontoons using cut pieces of wood. The keel and rudder are cut from a piece of plywood. Follow his Instructables for illustrated details of the build, as well as a detailed list of components and the complete schematics for the design. □

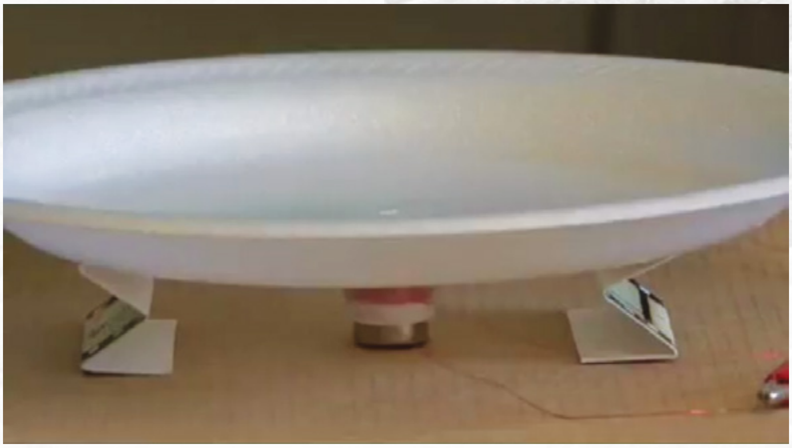
> "As you can imagine, it's an involved process and requires lots of sawing and drilling"

Above
The boat took two full days to put together, and Joshua spent one full day to test his catamaran

PLATE SPEAKERS

Project Maker
JOSE PINO

Project Link
hsmag.cc/l5dcds



Left ⬗
Make sure the coils of the speaker have lots of turns and are only connected to low-power amplifiers, or else they can become a fire hazard

Do you know how you hear sounds? Inside your ear is a thin piece of skin (called an eardrum) that vibrates, and our brain interprets the vibrations as sound. What speakers essentially do is take an electrical signal and translate it into physical vibrations to create sound-waves for our brains to interpret. Speakers do this conversion using a suspended diaphragm, a voice coil, and a magnet. Jose Pino replicates this design with a foam plate, some paper, a cardboard sheet, a couple of business cards, some enamelled copper wire, and a small neodymium cylindrical magnet. His instructions are fairly simple, and each step is nicely illustrated. It involves using two strips of paper to create a gap between the magnet and the coil. The business cards are used to help suspend the speaker plate over the cardboard base. You'll need to use an ohmmeter to make sure the coil has at least 7 ohms. You'll also have to remove the enamelled ends of the coil with either a wire stripper or a lighter. Jose also shows you how to connect the wire to either a mono or a stereo plug. □

"THE BUSINESS CARDS ARE USED TO HELP SUSPEND THE SPEAKER PLATE OVER THE CARDBOARD BASE"

PORTABLE AIR CONDITIONER

Project Maker
Prajjwal Nag

Project Link
hsmag.cc/0HJe9s

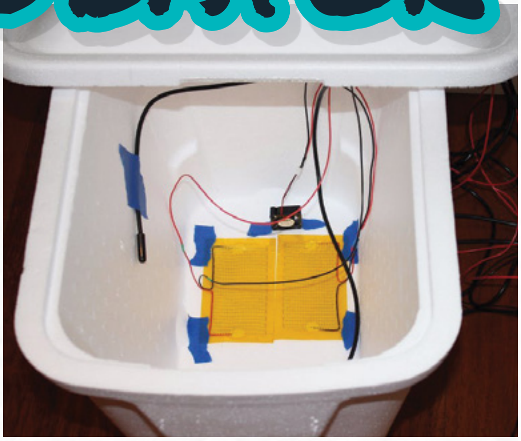
Looking to cool yourself on a hot day? While it isn't a replacement for regular air conditioners, Prajjwal's hack works on the same principles. Above all, it doesn't take much doing, and it's pretty effective. He uses a foam box and a couple of salvaged computer cooler fans to put the principle of convection heat exchange into practice. The fans are soldered together and attached to the top of the box. He then uses two lengths of flexible PVC pipes, and connects them to the side of the box towards the bottom with the help of some acrylic putty. That's about it. Now the box is filled with ice before the fans are fired up. The idea is that warm air will be sucked in through the fans, cools inside the foam box, and comes out via the outlets at the bottom. In case you want to replicate Prajjwal's build, he advises working in an environment with good ventilation, since foam can release harmful gases. □

Right Prajjwal cuts the foam box using his hot wire foam cutter (hsmag.cc/ifZdYI), which is pretty useful if you need to cut through foam regularly

BACTERIA INCUBATOR

Project Maker
Prasantha Jayakody

Project Link
hsmag.cc/A1bST3



Right The electronics are all mounted on a breadboard. Prasantha suggests using any breadboard that's large enough to have power rails down the side

Several decades ago, Lyon Creek, which flows into Lake Washington, was full of salmon. In his bid to reintroduce the fish to the creek, Prasantha started monitoring the stream. After carrying out basic temperature monitoring, he conducted several water chemistry tests to test the creek's alkalinity, pH, hardness, and such. Next up was bacteriological monitoring, to look for E. coli and other coliform bacteria. This required an incubator, and though he could have grabbed one off-the-shelf, like any good maker, he decided to forge one instead. And of course, Prasantha's version is smart and will message if a monitored parameter goes out of range. His incubator is a foam cooler that has a couple of heating pads, a fan, and temperature sensors that are all controlled by a SparkFun ESP8266 Thing dev board. You can easily adapt Prasantha's design to create a different kind of incubator or monitoring chamber. □

HackSpace
TECHNOLOGY IN YOUR HANDS

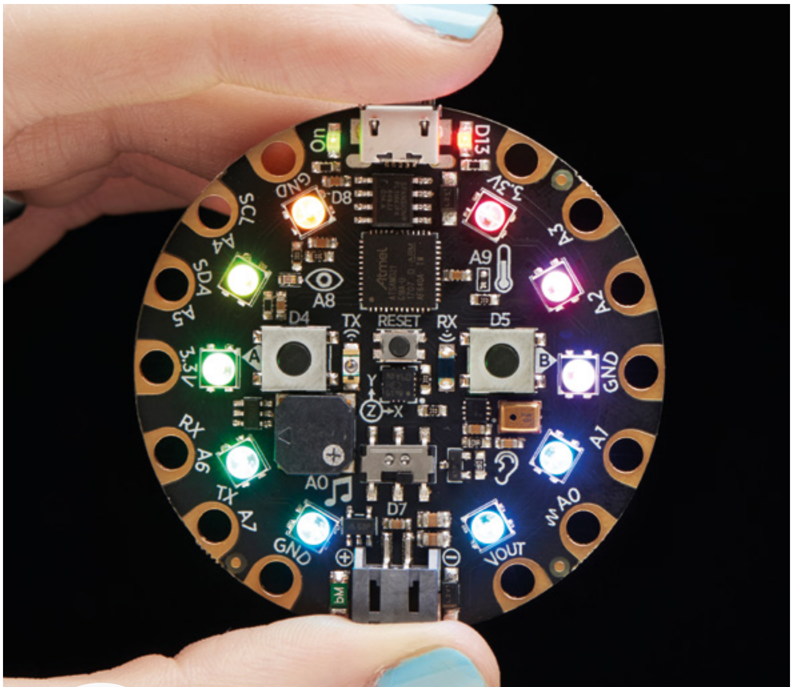
FORGE

HACK | MAKE | BUILD | CREATE

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

Three ways to blink an LED

Use CircuitPython to handle intermittent tasks



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.

Above Using these techniques, you can add complex LED animations to your microcontroller code without affecting how other parts of your code will run

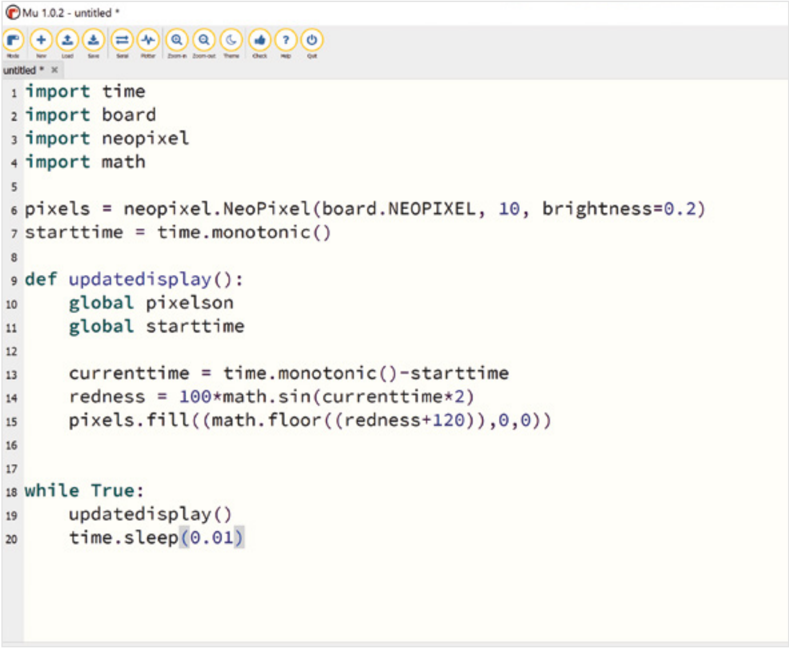
It's quite common with embedded programming to have a task that you need to perform every second, or every ten seconds, or once a minute. There are a few ways of doing this, and they each have pros and cons. Let's take a look at them.

By far the easiest way of running tasks periodically is using the **sleep** function to pause the program for a set amount of time. For example, we can blink the NeoPixels on a Circuit Playground Express with the following code:

```
import time
import board
import neopixel

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10,
brightness=0.2)

RED = (255, 0, 0)
OFF = (0, 0, 0)

while True:
    pixels.fill(RED)
    time.sleep(1)
    pixels.fill(OFF)
    time.sleep(1)
```

As you can see, a **while** loop runs indefinitely, and it uses the **time.sleep()** function to pause for one second between each action. This process works, and it's really easy to do, but it has one fatal flaw – you can't do anything else. You could fit a bit of extra code around the **sleep** function, but when it's sleeping, it's not running any other code. If you've got more complex code, it can be difficult to juggle the other processing you need with this type of loop. Let's take a look at an alternative way of coding this that gives you more freedom to run other code.

```
import time
import board
import neopixel

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10,
brightness=0.2)

RED = (255, 0, 0)
OFF = (0, 0, 0)
pixelson = False
lastflip = time.monotonic()

def flippixels():
    global pixelson
    global lastflip
    if time.monotonic() - lastflip > 1:
        if pixelson:
            pixels.fill(OFF)
        else:
            pixels.fill(RED)
        pixelson = not pixelson
        lastflip = time.monotonic()

while True:
    flippixels()
    time.sleep(0.1)
```



```
import time
import board
import neopixel
import math

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=0.2)
starttime = time.monotonic()

def updatedisplay():
    global pixelson
    global starttime

    currenttime = time.monotonic()-starttime
    redness = 100*math.sin(currenttime*2)
    pixels.fill((math.floor((redness+120)),0,0))


while True:
    updatedisplay()
    time.sleep(0.01)
```

Above This code is slightly more complicated than a naive sleep approach, but it's far more flexible

In this example, we call the function **flippixels()**, which checks how long it has been since the last time the pixels turned on or off. If it's been more than a second, it flips them; if not, it carries on. We can call this function as often as we like and it will continue to work normally. In our main loop, we've just got a **sleep** function, but that's just because this is an example. This could be any code you like. The only thing you need to remember is that the flashing of the lights won't be exactly every second, it'll be somewhere between one second and one second plus the time it takes the main loop to run. In this case, that's between 1 and 1.1 seconds.

The **time.monotonic()** function is a bit unusual in that the only thing we know about it is that it counts up in seconds. We don't know (and don't care) what time it counts up from. For uses like this where we only need to know the amount of time that's passed, it works well, but if you need to know the actual time, then you'll need some way of finding out the actual time (either via the network or a real-time clock).

The above method works well for things like blinking that aren't super-dependent on time, but some things, such as animations, require a slightly different approach. In this case, rather than look at whether enough time has passed and take an action based on that, we'll look at how much time has passed and adjust the action we take based on this.

```
import time
import board
import neopixel
import math

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10,
brightness=0.2)
starttime = time.monotonic()

def updatedisplay():
    global pixelson
    global starttime

    currenttime = time.monotonic()-starttime
    redness = 100*math.sin(currenttime*2)
    pixels.fill((math.floor((redness+120)),0,0))

while True:
    updatedisplay()
    time.sleep(0.01)
```

This doesn't blink the LEDs on and off, but 'breathes' them in and out by adjusting the brightness based on a sine wave of the time. There's a bit of adjustment to speed up the breathing and make the sine wave appear at the appropriate levels for them to be visible. This same technique can be used whenever you're adjusting something on a continuous scale, such as movement based on velocity.

The crucial difference between this and the previous example is that this one will – when averaged over a longer time period – always be accurate. In 100 seconds, this will blink 100 times, while the previous one will blink between 90 and 100 times. Again, we're using a **sleep** call in the main loop, but this can be any code, and the only thing it might do is stutter slightly if the loop takes too long to run. □

TIME-CRITICAL APPLICATIONS

We've looked at things that need to take place within a scope of a hundredth of a second, but you might be wondering about things that need to take place far more frequently than this. If you need to send data to another device at kilobits per second, you need to accurately switch a digital pin incredibly quickly. The slightly evasive answer to this is that it's not really possible in CircuitPython – at least not within Python itself. You can write modules in C that take advantage of a whole bunch of things with more accurate timing, and then import them into CircuitPython, but that's beyond the scope of this article. When you want to go even faster – megabits per second – you generally need dedicated hardware on the chip that responds automatically without the need for code, and many microcontrollers have this for protocols such as SPI and I^{2}C. This does all the timing-dependent stuff without relying on code running on the main processor.

TUTORIAL

Bolts and other threaded fasteners

Let's explore how bolts and other fasteners are made, chosen, and identified

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!

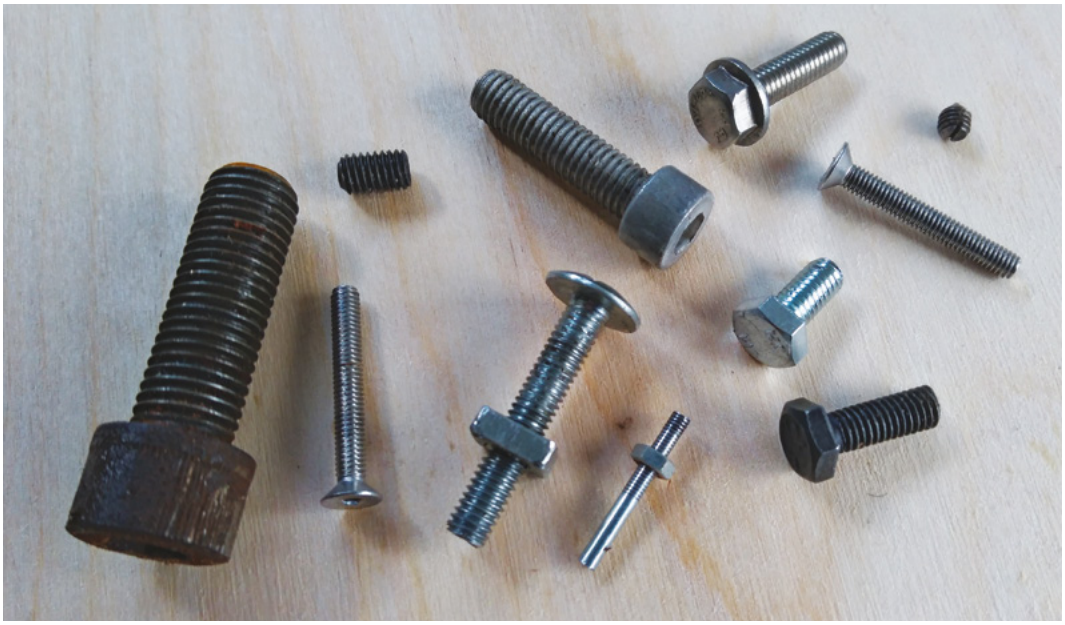


Figure 1 A collection of different types of threaded fasteners in a range of sizes and a range of head types

YOU'LL NEED

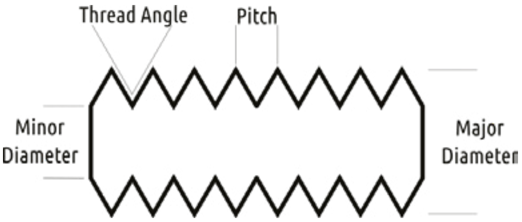
- Digital callipers
- Drill-press
- Drill bits
- Die and die set
- Counterbore tool
- Countersink tool

Bolts, machine screws, and other threaded fasteners are everywhere and are an important part of many mechanisms and structures. At first glance, there may seem to be little to know or explore about these items, but some knowledge of how they are formed and used may help us to select the right item for a task, or help identify an unknown bolt for replacement.

SO WHY 'FASTENERS'?

There are many types of bolts, but first, we need to make a distinction between the term 'bolt' and 'machine screw'. Machine screws are threaded items that are made to go into a threaded hole that is formed in one half of the workpiece or item. Bolts are used to go through holes without thread, commonly called a 'clearance hole', and are secured using a nut. In practice, a machine screw can often be used as a bolt and, as such, any threaded item often gets called a bolt. In this article we will often use the term 'fastener', which could mean a bolt, machine screw, or threaded stud, and we will use 'mating component' to either mean a nut or an internally threaded workpiece.

Lots of differing standards of fasteners exist, both past and present. Commonly, metric bolts are named using the letter M and then a number which represents the threaded section's outer or 'major' diameter (**Figure 2**). So, for example, 'M6' would refer to a 6 mm diameter bolt. However, there are two families within the M series, designated as 'coarse' or 'fine'. Coarse and fine refer to the thread pitch which is the spacing between the peaks of the thread turns (**Figure 2**). So, for example, an M6 fine bolt would have a thread pitch of 0.7 mm, whereas an M6 coarse pitch would have a thread pitch of 1 mm. Neither of these could replace the other, and the threads may be damaged if you tried. It's common practice, if you see terms such as M6 or M3, that they refer to the coarse



pitch versions which are more often used; however, it's a useful thing to keep in the back of your mind when something doesn't fit! Fasteners are made in a wide variety of designs, all with their own intended use cases. The familiar hexagonal bolt made from steel is perhaps the most common, but Allen bolts, slotted head screws, carriage bolts, grub screws that have no head, and many more exist for a variety of fastening tasks (**Figure 1**).

A LARGER BOLT ISN'T ALWAYS STRONGER!

The strength achieved by threaded fasteners is linked to the amount of thread engagement within the structure of the joint. Thread engagement is a complex subject, but basically is the measure of how much contact area there is between the fastener and the mating component. This is affected by a couple of parameters: the length of the bolt that is engaged with the mating component, and the pitch of the thread. When selecting a fastener for a design, often we might presume that fitting a larger diameter fastener is stronger, but this is not always the case. If we fitted an M10 bolt with the regular coarse thread pitch of 1.5 mm into a tapped hole in a 2 mm sheet of material, it would result in less than two turns of the thread engaging with the mating component. This would potentially create a flimsy joint. If we made the same design with an M2 bolt with a 0.4 mm pitch, we would create a smaller, lighter, and stronger joint as we have five turns of thread engagement into the mating component. Whilst we encourage you to explore the maths/geometry of thread forming and the stresses and strains in joint creation further, a good rule of thumb for choosing a fastener for a task is to choose one with a ratio of around 1 to 1.5 →

> "Fasteners are made in a wide variety of designs, **all with their own intended use cases**"

COUNTER**BORING**

Counterboring is used to create a hole for a bolt where the hole is widened to receive the bolt cap and so is used with round-headed bolts, such as Allen bolts. After drilling the clearance hole for the bolt, the counterbore tool (on the right-hand side of the image) is used to cut a wider hole concentric to the main hole and with a flat base. Counterboring the hole to the depth of the bolt cap, or lower if the design requires, creates a pleasing finish, with the bolt fully recessed into the workpiece.

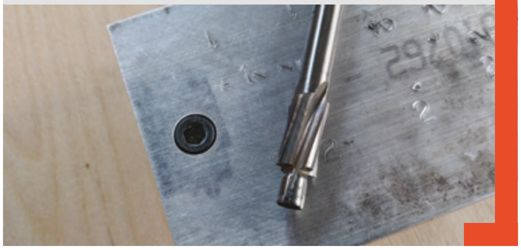


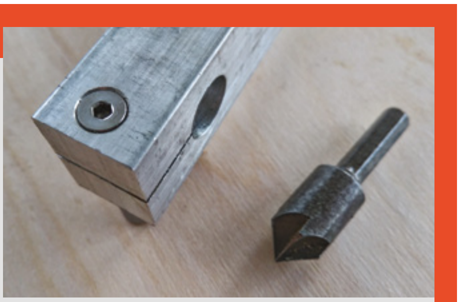
Figure 2 ↖
Features of a thread we will refer to in this tutorial

Figure 3 ↙
Using an M3 die in die holders to cut a thread into a short stud; similar to tapping an internal thread, the die must be kept horizontal in relation to the workpiece

STANDARDS

There are many families of thread sizes, and many use imperial units. British Standard Whitworth (BSW), Unified National Coarse (UNC), Unified National Fine (UNF), British Association screw threads (BA), British Standard Pipe (BSP) threads, and many more exist, and are all different. One major difference is that metric threads define the thread pitch as a distance between the peak of each thread, whereas imperial threads are identified as a number of turns of thread per inch (TPI). It's also worth noting that occasionally you may come across a custom fastener that doesn't adhere to any known standard. It's common, perhaps for someone working with a lathe on a single item, to make a bespoke thread size, as they are machine-turning the internal and external threads, so they don't always need to stick to a standard.

COUNTERSINKING



Countersinking is perhaps best known when using wood screws, but is commonly used with ranges of fasteners designed for countersink use in metalwork. The chamfer angle of countersunk bolts varies, depending on the family of bolt. ISO metric countersunk bolts tend to have a chamfer angle of 90 degrees.

When preparing a hole to receive a countersunk bolt, a countersinking tool (right-hand side of the image) is used that will produce the correct angle. Creating the countersink hole is slightly complex in that it can be hard to define a depth to cut to that will give the desired diameter of the countersink hole. If precision is required, then some relatively simple trigonometry can be done to work out the depth of cut, but a common approach is to set a calliper to the desired diameter of the countersink required, and then take small cuts and use the calliper as a comparator tool to eventually bring the hole to the desired diameter. Countersinking creates an attractive finish, whilst also increasing the surface contact area of the fastener, creating a strong joint with distributed load.

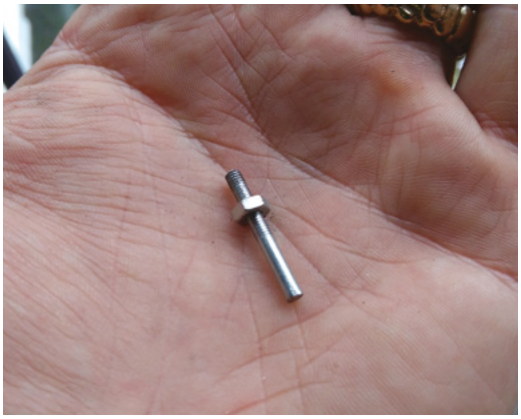
> "The commonest way to create a thread by hand **is to use a die and a die holder**"

QUICK TIP

Whilst there are many engineering books with tables of thread dimensions, often an internet search of the pitch and major diameter will identify the fastener.

major diameter when compared to the thickness of the mating component. So, for example, an M3 or M2 bolt into a 3 mm workpiece.

Often when using a nut and bolt, it's simply a case of drilling a clearance hole through the two surfaces to be joined and tightening the bolt into a nut. Whilst in the home workshop, we might just grab a drill bit the same size as the outer diameter of the bolt, there are official clearances stated for each size of bolt, and you can look these up online. For example, for an M3 fastener, a clearance hole diameter of 3.4 mm is suggested. There are a few other considerations or options when we are drilling holes to receive bolts, however, such as counterboring, countersinking, and spot facing.



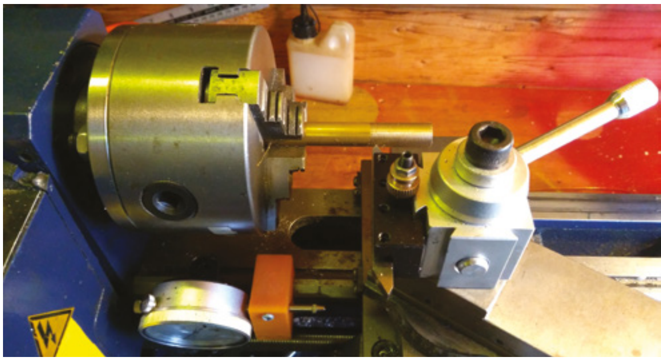


Figure 4 Our mini lathe is set up to create a thread onto this piece of brass with a 1 mm pitch thread

Right The finished thread section cut onto the stud is checked for fit with an M3 nut

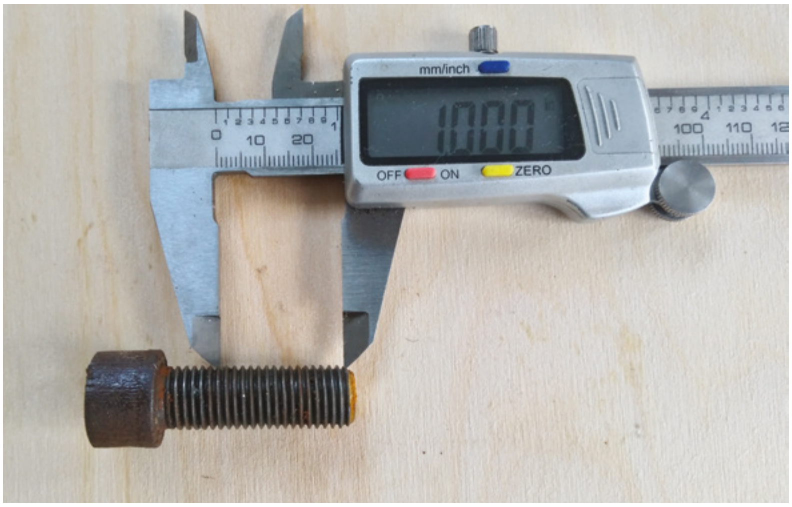
We looked at how threads are cut internally into objects to either make a nut or a threaded hole in the School of Making article on 'Drilling and Tapping' in issue 20. So, in this article, we will just explore how external threaded fasteners are made. It can be extremely useful to be able to fabricate a bolt or machine screw for a project. The commonest way to create a thread by hand is to use a die and a die holder (**Figure 3**). Similar to tapping a hole, the die must be kept horizontal to the workpiece so that the thread forms correctly. It's a good idea to set the rod in a vice using a set square to make sure it is vertical. Fit the die to the die holders and apply some cutting fluid. Introduce the die to the rod and begin to turn the die whilst applying downward pressure. Once you feel the die begin to cut the thread, ensure that for every half turn you rotate the die backwards a quarter turn: this clears the debris/chips out of the die cutting surfaces and ensures it won't clog/jam and so will cut correctly.

Creating a thread on a lathe is a common way to create a fastener – we covered lathes in issue 15, if you need a refresher. Thread cutting on a lathe is worthy of a full standalone tutorial, but an overview is useful. Not every lathe is capable of cutting a thread, but those equipped with a thread-cutting gearbox can do so with some simple tooling and a bit of practice. The gearbox can be reconfigured with different gear wheel combinations that can result in a known amount of travel of the cutting tool per revolution of the chuck. Lathe manufacturers will supply instructions as to how to achieve different ratios for creating different thread pitches or TPI. Often, a high-speed steel cutting bit, ground to the correct pitch angle (60 degrees for metric threads, but different families use different pitch angles), is inserted in the tool-post and the compound slide is set at 29 degrees to the right (**Figure 4**). As the passes are cut into the material, the compound slide feeding in at this angle allows the left-hand side of the tool to do the majority of the cutting, with the right-hand side of the tool clearing the cut thread.

SOME DETECTIVE WORK!

Learning a little about how threads are designed and made is useful when identifying an unknown bolt. The first port of call is to see if there are any details about the item online that mention the fasteners it uses. Next, consider where the object was made and its age. Metric screw threads originated in 1947, so anything before that will be imperial. However, after 1947 many fasteners in the UK were still imperial.

Next, measure the outer (major) diameter of the fastener's threaded section. If you have a pair of digital callipers that can switch between metric and imperial, start by measuring in mm. If it is very close to a metric bolt size, there is a fair chance you have identified your bolt. If the diameter on the calliper doesn't correspond with a metric size, swap the callipers over to imperial and have a look if the fraction of an inch corresponds to a known imperial size – an obvious one is 0.250" is likely to be a ¼ inch. Once you have identified a likely diameter, try to work out either the pitch of the thread or count the number of threads turns per inch (TPI) (**Figure 5**). Using a set of callipers, set at an inch (or a known fraction of an inch) and count from the peak of one thread how many there are. An internet search on the major diameter and the pitch or the TPI is often enough for you to find the family of bolt/fastener you have. You can, of course, identify a missing fastener from the internal threads of its mating component. It's harder to find the major diameter, but you can usually use callipers to identify its minor diameter, which may return some search results. A good way to ascertain the thread pitch is to insert a thin object (e.g. a matchstick) into the hole with some Blu Tack on the side of it, pressing it into the threads. Removing this can then give you an impression of the threads to measure and count thread turns. If identifying threaded fasteners is something you do frequently, it's a great idea to get a set of pitch gauges, or even a few sets of pitch gauges, that cover the different families of fastener. Pitch gauges (**Figure 6**) are small, flat, metal objects with the profiles of threads cut into the side of them. To identify a fastener, you simply present each gauge up to the thread and hold it up in front of a light. When you find a gauge that fits the thread exactly, with no gaps and which doesn't clash with the fastener's threads, you can read the pitch or TPI off the fastener label.



SO MANY FASTENERS TO CHOOSE FROM!

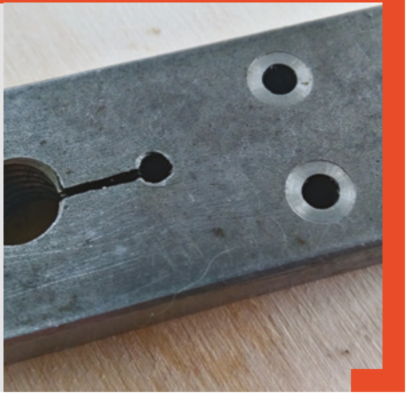
Finally, whilst the steel nut and bolt are a first 'go-to' item, it's worth considering the range of materials that fasteners are available in. Aluminium bolts may be lighter but with a lower shear strength than steel, but also may be available in a range of colours. Titanium bolts are strong and light and are becoming more reasonably priced. Many fasteners are available in plastics, with nylon bolts being the commonest. However, there are many retailers selling fasteners made from more modern plastics, like Ultem or PEEK, which can be used at a very high temperature. □

Figure 5 Using callipers to try to count the turns per inch of an unknown imperial bolt

Figure 6 Pitch gauges are a useful tool to help you accurately identify an unknown fastener

SPOT FACING

Spot facing is used to make the area around a bolt hole flat, meaning that the head of the bolt lies flush against the surface of the workpiece. Usually used in surfaces where the surface finish of the rest of the item is non-critical, spot facing can be achieved using either a counterbore tool or an end mill. A small cut is made until the cut surface forms a full circle concentric to the through-hole. It's usual to try to remove a minimum of material to achieve the flat finish required.



Mitre saws

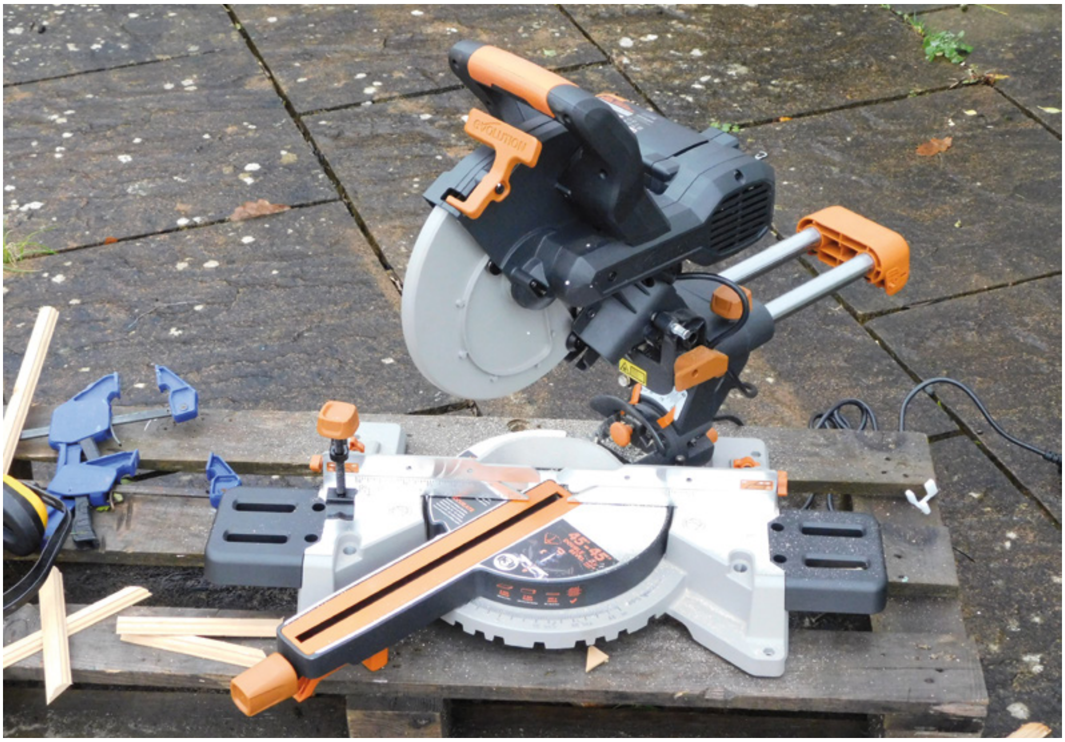
Cutting at this angle and that angle

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.



Above This is a particularly large mitre saw – a more compact option may fit your workshop and budget better

If you consider the hierarchy of complexity with regard to spinning saw blades, circular saws are the most basic. You have a motor, a blade, and a handle – it's up to you to make sure you come into contact with the thing you want to cut. These are great for portability and cutting long sections, but they're not always very precise or controlled.

If you mount this spinning blade on a lever so it can cut down on a base, you have a chop saw. These can cut accurately and, typically, are powerful saws using grinding wheels rather than toothed blades. If you want a 90-degree angle cut quickly and accurately in a hard material, they're great. However, in woodworking, we often need to cut other angles, and for this, we can mount the blade on a lever that can be rotated to cut at different angles – and this is called a mitre saw. If the lever can slide in and out, you can cut larger pieces of wood, and this is known as a sliding mitre saw. The final layer of complexity in spinning wheel saws is bringing the saw down at an angle, and this is known as a compound sliding mitre saw. If you can rotate the angle in both directions, it's a double-bevel compound sliding mitre saw.

Let's unwind all this and look at why you might want a mitre saw, and which one you might want. A mitre saw is generally used for cross-cutting bits of wood. It's a quick and fairly safe way of doing this. Since the wood should be clamped in place before starting and the blade follows a very fixed track, we can keep our hands well out of the path of any spinning metal. You should wear all the usual protective equipment (particularly goggles, and possibly ear defenders). The only slight trap is that you should wait for the blade to stop spinning after cutting before lifting it up, as this prevents the risk of off-cuts flying up at you.

Many people use their mitre saws almost exclusively as chop saws, just cutting planks of wood at 90 degrees. However, if you're making something like a picture frame, you can use the ability to cut at an angle (known as a mitre cut) to make accurate angled cuts. For a picture frame, this would typically be 45 degrees.

A bevelled cut is where you also cut at an angle, but in the vertical plane. In essence, the only difference between the two cuts is rotating the wood 90 degrees, so you can do both types of cuts with a plain mitre saw, though it's less convenient, and the size of wood you can cut may be different. Speaking of size, sliding mitre saws let you cut considerably larger pieces of wood than non-sliding version. However, a significant amount of the use most mitre saws get is on wood small enough to be cut without sliding, so whether or not this is useful to you will depend on your particular use case.

With a compound mitre saw, you can make a cut that is both mitre and bevelled (a compound mitre). These are mostly used for crown mouldings, and are less common in woodworking.

A few more useful things to look out for:

- **Blade size** – In mitre saw terms, this equates to the depth of wood you can cut. Typically, saws come in 216 mm, 255 mm, and 305 mm (or 7, 10, and 12 inches). Remember that you can't quite cut the radius as there's also the drive mechanism in the middle.

- **Positive stops for easily hitting common angles** – This is where the saw clicks into place, making it easy to precisely hit angles such as 45 degrees.

- **Laser guide** – Many have lights to show where the cut will be, but to be honest, we prefer lining the blade up with the saw not spinning. This way, we're more confident that everything is lined up properly.

Left
A bevel cut rotates the blade around the horizontal axis

Above
A mitre cut rotates the blade around the vertical axis

ON TEST

To look at all this in practice, we got an Evolution R255SMS-DB 255 mm Double-Bevel Sliding Electric Mitre Saw. At £199, it isn't the cheapest option (the basic mitre saw from the same manufacturer is £59), but it is a versatile tool. The included saw blade can cut through wood, plastic, aluminium, and steel, so can find a place on many projects. The maximum cross-cut size is 300 × 80 mm (but bear in mind that the maximum mitre cut size will be less than this, depending on the angle cut).

Our only slight niggle with this saw is the clamping ability. The one supplied clamp holds material down, but a more flexible clamping system would make it easier, especially when cutting wood that's not rectangular. You can use additional clamps to hold wood in place though. The sloping fence back makes this a little more tricky than it should be, but it works.

> "For cross-cutting and mitre cutting, **there really is nothing as quick and easy as a mitre saw**"

For such a large and powerful machine, it's surprisingly light to use, and with everything set up, it doesn't require much force or precision to cut through even stubborn materials. For cross-cutting and mitre cutting, there really is nothing as quick and easy as a mitre saw, and with a sliding cutting head, you can use it on surprisingly large pieces of wood as well. We tested the machine with both wood and mild steel. The supplied blade cut through everything without major problems, but you may benefit from a more specialised blade, depending on what you want to use it for.

If you find yourself cutting a lot of timber with widths less than 30 cm, then a mitre saw is quick, accurate, and (if you're sensible) safe. □

Sew your own jacket for the perfect fit

Use FreeSewing's open-source pattern for ultimate customisability

Below It's great to be able to make clothes that fit, without needing to know how to alter patterns

Digital technology is making sewing more accessible for beginners and easier for experts. Software can automate the difficult and tedious task of pattern alterations, so you can input your measurements and create patterns for clothes that will fit perfectly. For this tutorial, we've chosen a jacket design from FreeSewing because the platform is open source and easy to use.

FreeSewing has over 20 designs, from swimwear to a Sherlock Holmes-style coat. When patterns are added, they don't always have sewing instructions, so this guide covers the whole process of making their hooded jacket. This author has also made a video of the jacket at **hsmag.cc/GXBEUU**.

DRAFTING THE PATTERN

First, sign up for a free account at **FreeSewing.org**, and create a new model. Choose 'Filter by Pattern' to avoid giving unnecessary measurements, and select 'Huey', which is the pattern we'll be using for the tutorial.

WHAT IS FREESEWING?

FreeSewing is an open-source platform for made-to-measure sewing patterns. Joost De Cock started the site because, at 6' 6", he struggled to find patterns that fitted him. It's since grown into a community with over 12,000 users.

You can ask for advice or contribute to the community at: **hsmag.cc/6hbqkP**. If you'd like to add a new pattern, and you can code in JavaScript, there's a step-by-step guide to getting started. They're also looking for people to help translate the site into other languages.

Joost says the main challenge for him is burnout, so donations are great for morale. All the money received goes to Médecins Sans Frontières, and last year the charity received over €4000.

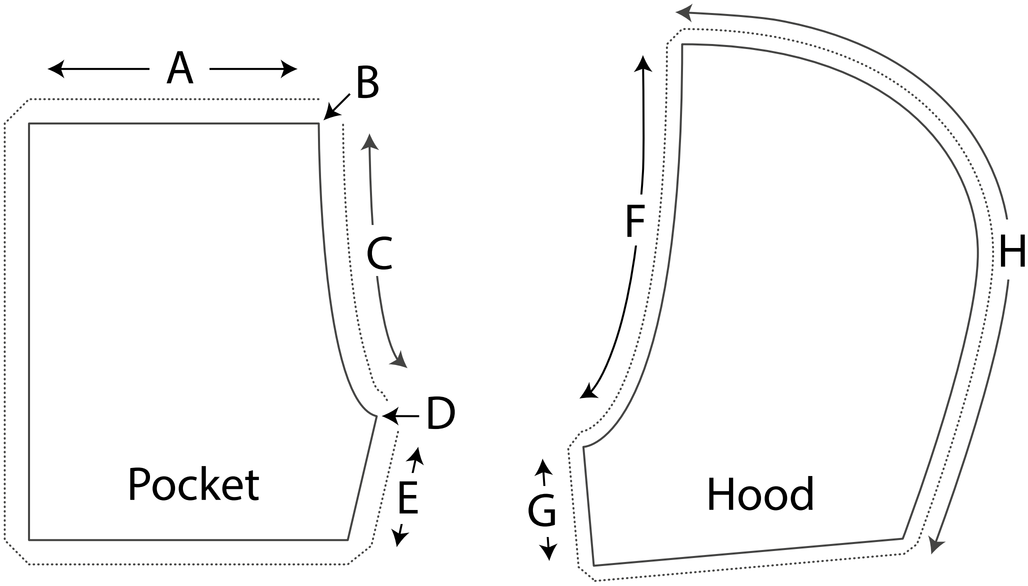


Figure 1
Edges and corners diagram

After adding your measurements, go to New Pattern, choose Huey again, and select your model. The pattern will be drafted automatically, but you can customise several features using the Pattern Options in the sidebar. You'll probably want to adjust the ease settings to give room for movement because the default creates closely fitted garments.

Click the Compare button to make sure the pattern looks right. If it seems surprisingly different to common sizes, ask someone on the FreeSewing chat room to check it. As the founder, Joost, says, "People who are familiar with the system will spot issues, like faulty measurements, with one glance."

Keep the recommended seam allowance of 1 cm, and when you are happy with the design, click Export Pattern. We exported it as an A4 PDF, but choose the maximum size for your printer. Make sure the scale is set to 100% before you print, and measure the test rectangle on one of the pattern pieces when it's printed.

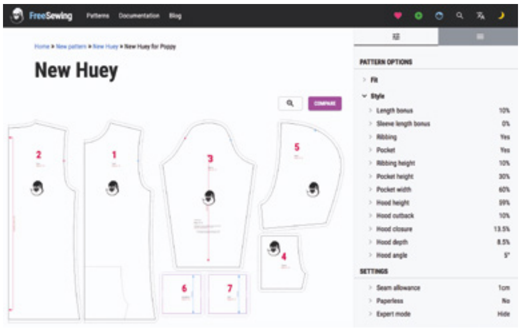
FOLLOW THE DOTTED LINES

Cut the paper pattern pieces out along the dotted seam allowance lines. For the front section, cut along the solid vertical line. If you printed each shape over several sheets of paper, tape the connecting sections together.

Fold your jersey fabric in half so you can cut through two layers at once. It's important to fold it widthways and lay your pattern pieces correctly, to make use of the natural stretchiness of jersey. The garment needs to stretch across the shoulders and sleeves, not along its length.

Position the back pattern piece on the fold of the fabric, with the arrows pointing to the fold. Place a cutting mat underneath and small weights on top to hold the paper in place. Using a rotary cutter, cut around the pattern, except along the fold. Mark the notch by making a tiny cut.

Then cut out the fronts, sleeves, hood sections, and pockets through both layers of fabric to get mirror images of each piece. Where there are grain lines marked on pattern pieces, they must be parallel to the fold. Make small snips to mark the notches, making sure you can tell the difference between →



New Huey

PATTERN OPTIONS

Fit

Style

Length bonus	10%
Sleeve length bonus	0%
Ribbing	Yes
Pocket	Yes
Ribbing height	10%
Pocket height	30%
Pocket width	60%
Hood height	99%
Hood cutback	10%
Hood closure	13.5%
Hood depth	8.5%
Hood angle	5°

SETTINGS

Seam allowance	1cm
Paperless	No
Expert mode	Hide

Above
The preview image will update when you edit the pattern options, so you can see the effect of any changes

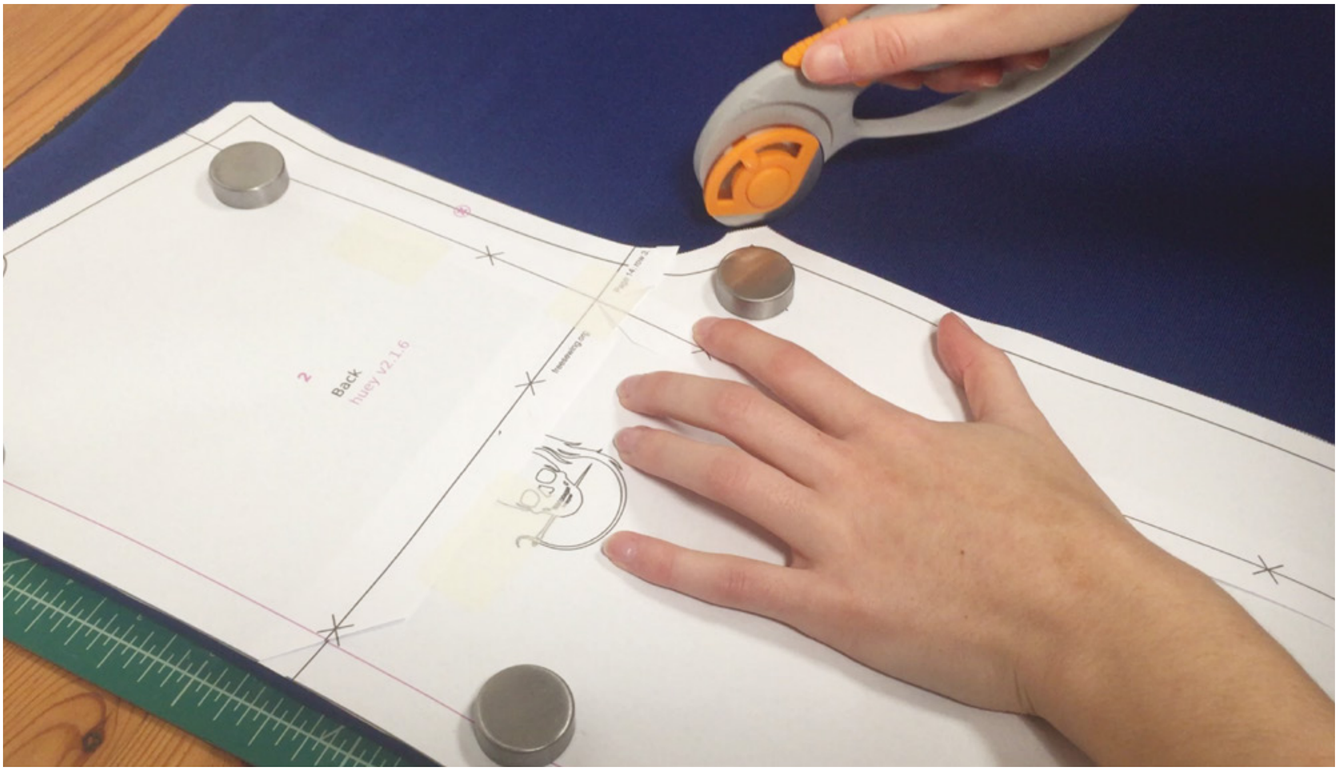
Poppy Mosbacher

@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.

YOU'LL NEED

- Sewing machine with a stretch stitch option
- Zipper foot attachment for sewing machine
- Thick jersey fabric
- Lining fabric
- Ribbing fabric
- Open-ended zipper
- Reel of thread
- Flexible tape measure
- Printer and paper
- Masking tape
- Rotary cutter
- Large cutting mat
- Small weights
- Clips and pins
- Needle
- Scissors
- Iron



Above By using a rotary cutter, you don't have to pin the pattern to the fabric. But take care – the blade is sharp

the front and back notches on the sleeves. This will make it easier to position the sleeves later. The cuffs and waistband are simply rectangles that can be measured directly on the ribbing fabric. Add a 1 cm seam allowance on each end. Cut two cuffs and one waistband from the ribbing fabric, paying attention to the stretch direction. Cut two hood sections from lining fabric.

START TO SEW

Set up your sewing machine, and select a 3 mm straight stitch. If you don't know how to do this, check out last month's sewing tutorial.

Using clips, fold the 1 cm seam allowance on edges A, C, and E of the pockets (**Figure 1**). Then sew along the fold as one continuous seam, removing the clips as you sew. Pin the pockets to the front sections, as indicated by the faint dotted line on the front pattern piece.

Sew along edges A and E between the previous line of stitches and the edge of the pocket. Don't sew the pocket opening (edge C).

Bar-tack corners B and D by sewing forwards and backwards a few times at a diagonal. This adds strength around the opening.

> "Sew along the seam lines using stretch stitch. Unless specified, **use this stitch from now on**"

Putting right sides together, line up the top edges of the front pieces with the shoulder edges of the back piece. Sew along the seam lines using stretch stitch. Unless specified, use this stitch from now on. As the name suggests, the thread is less likely to snap when the garment is stretched.

Put the right sides of the hood lining together and sew along edge H. Repeat this process for the outer parts of the hood.

Open the hood and hood lining, and place right sides together. Line up the seams you just sewed, and sew the hood and lining together along edges F and G.

Turn the hood the right way round and iron flat. Then, using a straight stitch, create a line of top-stitching by sewing along edges F and G 1.5 cm away from the edge of the fabric.

Use clips to hold the centre seam of the hood to the centre of the back piece with right sides together. Add more clips to position the corners of

QUICK TIP

If your material looks the same on both sides, use a piece of masking tape to identify the 'wrong' side of each piece.

HOW TO CHOOSE THE ZIP LENGTH

Measure the length of the front pattern piece and half the width of the waistband, minus the seam allowances. If your measurement isn't the size of a standard zipper, adjust the width of the waistband, or cut a longer zipper to size and remove a few teeth from the top with pliers.

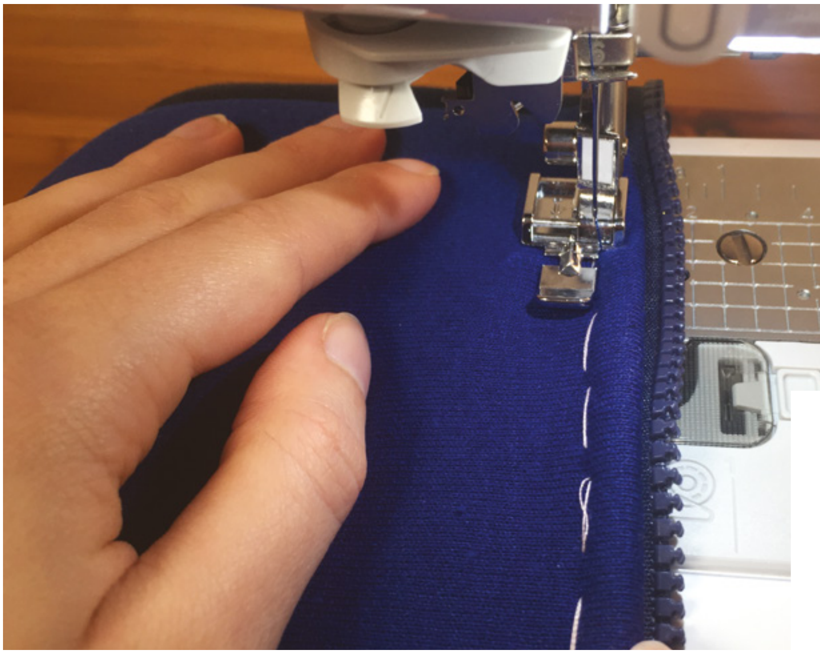
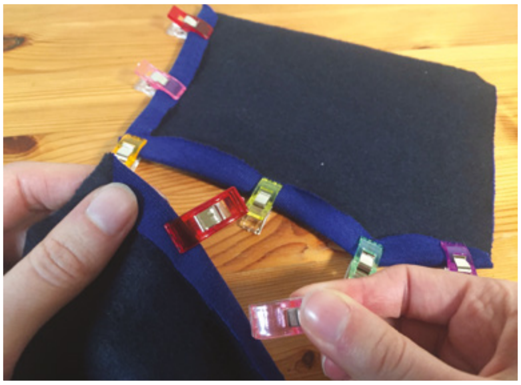


Figure 2
Top-stitch in a straight line between the row of tacking and the zipper teeth



Below
Fabric clips are often quicker and easier to use than pins

the hood 1 cm from the edge of the corresponding front pieces, and add a few more clips to secure the neckline. Then sew a seam all the way along. Change to a wide zigzag stitch, and add another row of stitches over the raw edges of the neckline you've just made. This makes a neater edge.

Temporarily, fold a sleeve in half to mark the centre point of the curved edge with a pin. Using the notches that indicate front and back, work out which side of the jacket each sleeve needs to be attached to. With right sides together, line up the centre point on the first sleeve with the shoulder seam of the jacket and use clips to hold in place. Match the notches and clip the top of the sleeve to the jacket. The arm hole is intentionally smaller than the sleeve head, so you will need to stretch it slightly to fit. Then sew the sleeve in place and remove the clips. Repeat with the other sleeve.

TAKING SHAPE

Turn the jacket inside out. With right sides together, sew a seam that attaches the edges of the front and back sections together and along the length of the sleeves. Repeat on the other side of the jacket.

Fold the waistband in half lengthwise, wrong sides together, and mark the centre point with a pin. Also mark the centre point along the base of the jacket. Lay the folded waistband on the base of the jacket, right sides facing, with the raw edges lined up. Match the centre points you just marked and use clips to secure them. Stretch the waistband evenly along the bottom edge of the jacket, using clips to hold it in place. Then sew. Make the cuffs the same way, but sew the ends together first and don't mark the centre points.

Separate the zip into two sections. With the jacket facing up and the zip facing down, use clips to secure both parts in place. Where the zip goes over the waistband, spread the layers of fabric out so they're not too bulky to sew.

ZIP IT UP

Do the zip up to check that both sides are aligned at the top, pockets, and waistband and adjust if necessary. Tack the zipper in place and unzip the jacket. Then sew, using the zipper foot and a straight stitch. Remove the tacking.

Fold the fabric edge of one side of the zipper flat against the wrong side of the front and tack in place. Then use a straight stitch to sew a line of top stitches (see **Figure 2**). Repeat with the other side and remove the tacking.

FINISHING TOUCHES

Fold down the fabric at the top of each half of the zipper and secure with a few hand stitches. Cut any loose threads and try it on! □

QUICK TIP

Ironing the seams at each stage gives a neater finish.

The Maker's Formulary

Create a recipe book for DIY paints, finishes, glues, solvents, cleaners, and more

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:.

When we need a special chemical or treatment for a project, most of us go shopping online to find the commercial product designed to fill our need. But in many instances, a home-brewed solution can serve just as well (and can often be cheaper). Here is a little shop cookbook with some of our favourite DIY concoctions.

Make sure you're familiar with the necessary safety precautions when working with any chemicals, particularly concentrated ones. Some require ventilation, some require protection against contact with the skin, and don't forget your eye protection!

NON-SLIP TEXTURED PAINT

- **Gloss paint**
- **Sand**

Pour sand over wet gloss paint. When dry, paint over the top to make a non-slip textured paint.

FABRIC PAINT

You can turn any acrylic paint into a fabric paint simply by adding textile medium to it. You can find this at most art and crafts stores.

PAINT WASHES

- **Acrylic paint** (colour of choice)
- **Matte medium**
- **Distilled** (deionised) **water**
- **Flow Aid** (optional)

Washes are used in a lot of hobby painting, such as tabletop miniatures, scale modelling, and cosplay costuming. Washes add realistic weathering, ageing, and deep, contrast shading. To make your own, all you need is a few drops of your desired paint colour, some matte medium, and distilled water. There are many wash recipes online. Some add Flow Aid (used in airbrush painting) to the mix to help the wash resist clinging to high surfaces, and flow better into nooks and crannies where you want it.

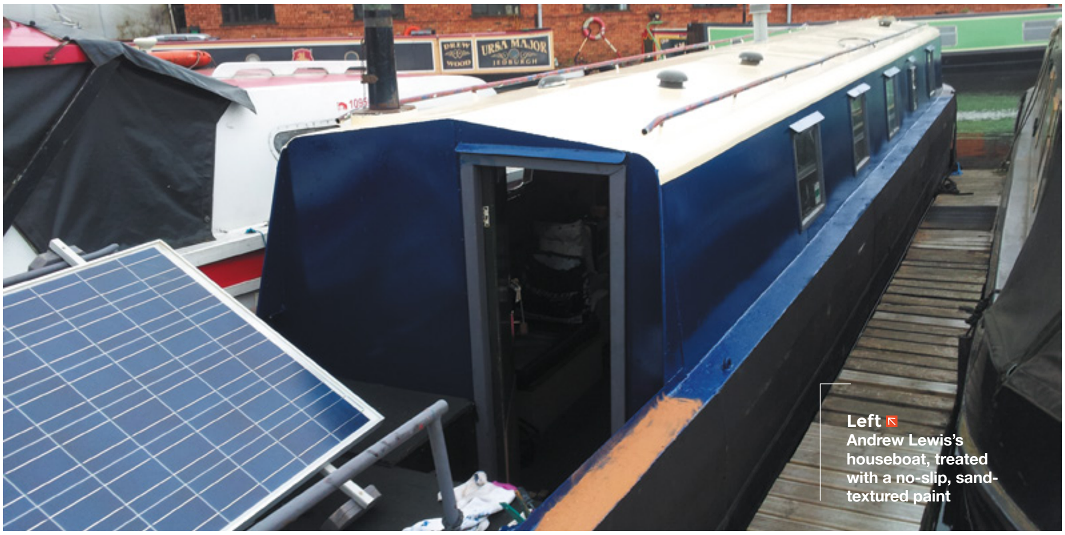
MATTE GLAZE MEDIUM

- **Distilled water** (say 180-240 ml)
- **Matte medium** (4–5 drops)
- **Flow Aid** (a few drops – optional)

Thinning paints in water tends to separate the water and pigment, making it difficult to get a thin coat with adequate coverage. Enter matte glaze medium. It is designed to maintain the integrity of thinned paint for smoother coverage. There is no set recipe here.

QUICK TIP

You can find a home-brewed version of nearly any formulation you can think of simply by doing a web search on 'DIY (and the product you're looking for)'. Make sure to read up on what you find, and be mindful of any cautions and what users of the formulation say about its effectiveness.



Left Andrew Lewis's houseboat, treated with a no-slip, sand-textured paint

Experiment with the amount of matte medium (and Flow Aid) until you get a nice, thin but pigment-rich deposit of paint.

ACRYLIC PAINT THINNER

- **Distilled Water** (175 ml)
- **91% isopropyl alcohol** (75 ml)
- **Glycerin** (10–15 drops)

COFFEE STAIN FINISH

- **Freeze-dried coffee**
- **Water**
- **Wood glue**

Hans Gerhard Meier writes: “Coffee stain is my favourite colour when finishing up my box builds. It gives an aged and weathered look that I really love. There’s no set recipe. Experiment with a mixture of freeze-dried coffee, water, and wood glue (for binding). I usually finish the boxes with clear varnish when I’m done. Plywood becomes aged wood." You can also use black tea as a stain/finish.

RUST

- **Steel wool**
- **White vinegar**

Place the wool in a jar and cover with vinegar. Shake from time to time. In a few weeks, you will have a jar of liquid rust that you can use in modelling, dioramas, or other situations where you want to add rust/weathering.

RUST REMOVER

- **Water** (50%)
- **Vinegar** (30%)
- **Hydrogen peroxide** (20%)
- **Enough salt to dissolve**

Replenish the hydrogen peroxide periodically. Not cost-effective for large de-rusting/restoration work, but if you just need to soak a few small parts…

CYANOACRYLATE (CA) ACCELERANT

- **Baking soda** (½ teaspoon)
- **Purified water** (¼ cup)

Brush/spray this mixture on the dry side of your join. This recipe will allow a several-second working time. If you want an immediate bond, increase the baking soda (to 1 tsp.).

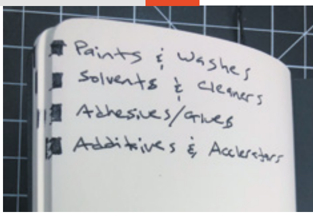
HOT GLUE ACCELERATOR AND DE-BONDER

You can use compressed canned air as both a hot glue accelerator and de-bonder. To accelerate hot glue drying time, hit it immediately with a blast of air from the can. To break a hot glue bond, freeze it by turning the can upside down and spraying the join. It should then be able to break.

MAKE YOUR OWN **RECIPE BOOK**

If you want to keep your recipes and application notes handy and organised, make yourself a Maker’s Formulary notebook. This author uses the blank-page Moleskine pocket notebooks. This way, you can keep all of your recipes handy. I attached a looped string to mine so that I can hang it from my pegboard.

You can also create an index on the back page so that you can more quickly find the formulary section you’re looking for. All you have to do is list the sections you want to create on the back page, and then mark the pages of that section along the edge so that the marks correspond to the index listing in the back.



ADHESIVE PUTTY

Sugru is a well-known 'mouldable glue' or adhesive putty made from a durable silicone rubber. There are a number of recipes for making your own. Oogoo is a silicone clay that’s similar to Sugru. You can find instructions on making it at **instructables.com**, ('How To Make Your Own Sugru Substitute'). Also search for other 'DIY Sugru' recipes online.

MOULD RELEASE AGENT

- **White petroleum jelly**
- **Naphtha**

Dissolve white petroleum jelly in naphtha until saturation is reached. Apply to finished silicone moulds, wait for the naphtha to evaporate, leaving a very even, usually satin, finish that will easily release your cast silicone parts. Suitable for releasing Addition Cure silicone from identical Addition Cure silicone.

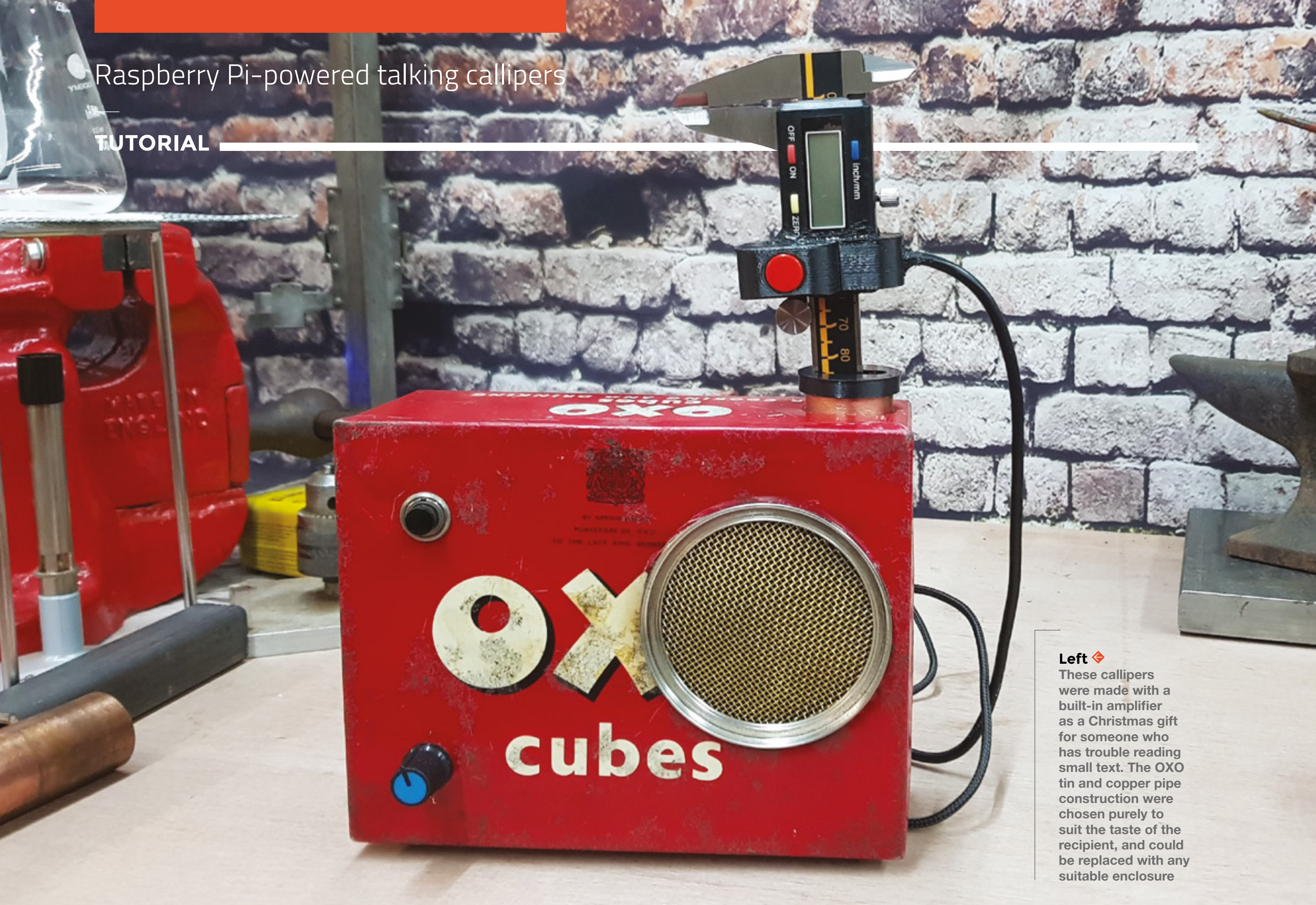
SOLVENT CEMENT

Unless your application is critical, cheap, liquid paint stripper (not the gel, paste, or colour-changing kind) is a fine substitute for commercial acrylic solvent cement. Commercial products might have a few chemical improvements for forming a tighter bond, but for many low-stress applications, this is a much cheaper alternative. □

Left A bit of coffee can liven up your woodwork

THANKS

Matt Schaid, Andrew Lewis, Hans Gerhard Meier, Dave Dalton, Ross Hershberger.



Left These callipers were made with a built-in amplifier as a Christmas gift for someone who has trouble reading small text. The OXO tin and copper pipe construction were chosen purely to suit the taste of the recipient, and could be replaced with any suitable enclosure

Raspberry Pi-powered talking callipers

Interface a pair of digital callipers with Raspberry Pi to make them read measurements out loud

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.

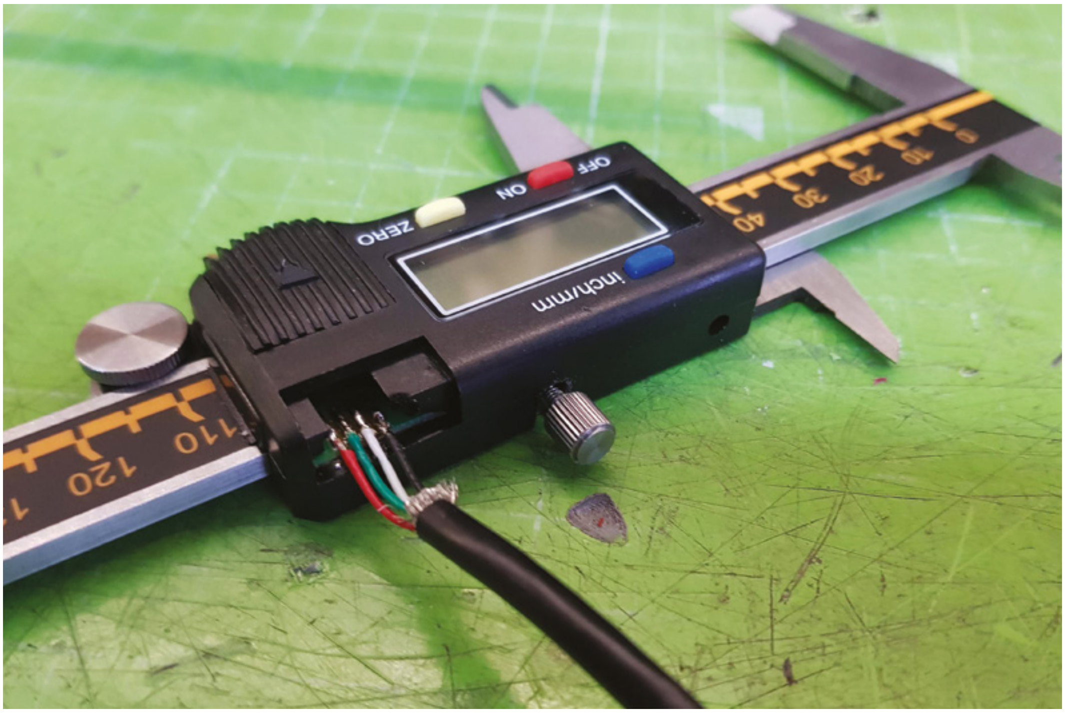
The ability to measure things accurately is something that we all take for granted in the modern world. However, it's not always easy to read a digital scale. The goal of this project is to interface a Raspberry Pi to a set of digital callipers and use it to read measurements out loud.

It turns out that making a pair of cheap digital callipers talk like this isn't as difficult as you might imagine, although it isn't a beginner's project. With a little bit of modification, you could adapt the project to log measurements, or even use the callipers as a control method for servos or other devices. The callipers in this project have a button to make them speak, a button to select between metric and imperial measurements, and use an amplified speaker to verbalise the measurements from the callipers.

Completing this project means overcoming a couple of challenges. Firstly, you need to be able to capture data from a pair of digital callipers and turn it into meaningful information. Secondly, you need to take that meaningful information and use Raspberry Pi to turn it into spoken words.

DIGITAL IS JUST ANALOGUE WITH LUMPS IN IT

The outward design of cheap digital callipers can vary, but most of them have the same circuit board inside. On some callipers, you can find a clip-off



Left
This digital calliper has a panel on the casing that can be removed to reveal the connection points on the circuit board. As you can see, the ground (black) and 1.5 V (red) wires are soldered onto the outer pads, while the inner clock (green) and data (white) are on the inside

> The clock and data lines send two 20-bit chunks of data **with a short gap between them**

section of the case that allows access to four mysterious copper pads, while on other models you'll need to disassemble the callipers completely to find the same pads. It's worth understanding what is happening with these pads, even though the nitty-gritty of the interfacing has been dealt with in the Python code that you can download for this project (**hsmag.cc/issue27**). Reading across these pads with an oscilloscope will reveal that the outer pads are connected to the 1.5 V and GND of the battery, and the other two pads are sending out some sort of signal. Closer inspection shows that one of the signals is a repeating square wave pattern that doesn't change. This is a digital clock signal. The remaining pad is a data signal, and probing it with an oscilloscope reveals that the signal changes when the callipers are moved. It doesn't take much of a leap to realise that this is a digital signal, and with a little bit of mental exercise, you can determine that the signal contains the measurement from the callipers in microns, with a little bit of padding at the end of the data packet to deal with negative numbers. The clock and data lines send two 20-bit chunks of data with a short gap between them. Then there's a longer gap, and then the signal repeats.

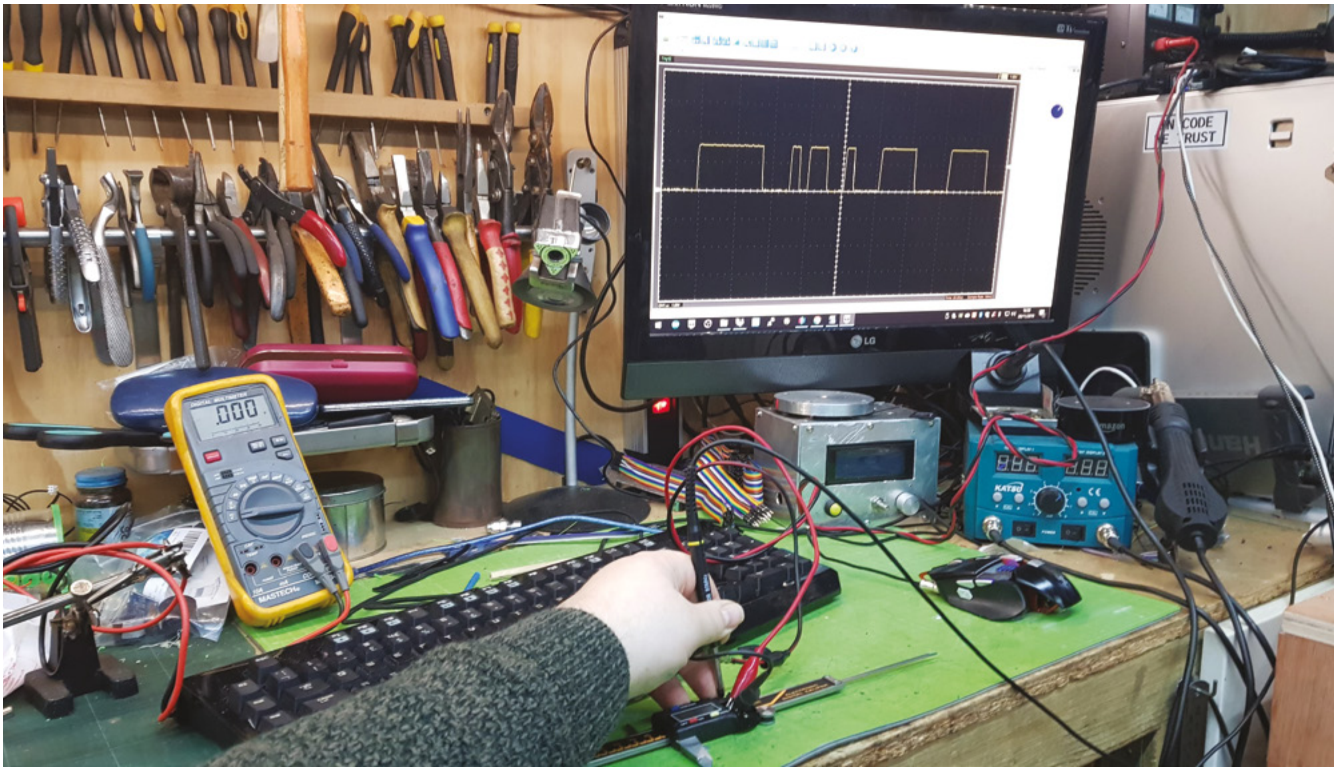
Capturing the digital signal from the callipers and getting them into your Raspberry Pi presents a hardware and a software challenge. Raspberry Pi uses 3.3 V on the GPIO, and the digital callipers use 1.5 V. You're only interested in unidirectional communication (you never want to send data to the digital callipers – you only want to read data from them), so a simple transistor circuit is all you need to shift the logic level from 1.5 V to 3 V. The circuit used in this project has the side effect of inverting →

INTERRUPTIONS

If you're not familiar with interrupt-driven applications, the Python code might look a little bit strange to you. The main loop of the Python application doesn't have much code inside it other than a sleep command. This is because the Python code is relying on callback functions that get executed when a certain condition is met, such as a rising edge being detected on a particular pin. This means that the main loop is essentially redundant, and all of the important code is happening in the callbacks.

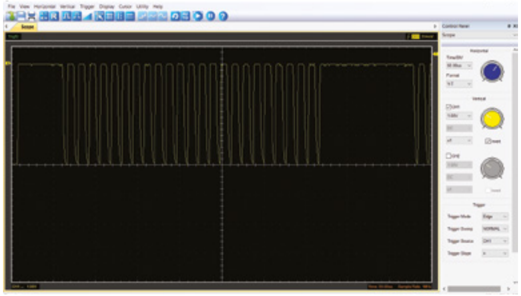
YOU'LL NEED

- Raspberry Pi 3
- Pair of digital vernier callipers
- 2 × BC548 NPN
- 4 × 10 kΩ resistors
- DC-DC converter (5 V to 1.5 V)
- 2 × push-button switches
- 3 W speaker (optional)
- 3 W audio amplifier (optional)



Above Here, you can see the more irregular (and changing) pattern from the calliper's data pin

Right And this shows the regular wave produced by the calliper's clock



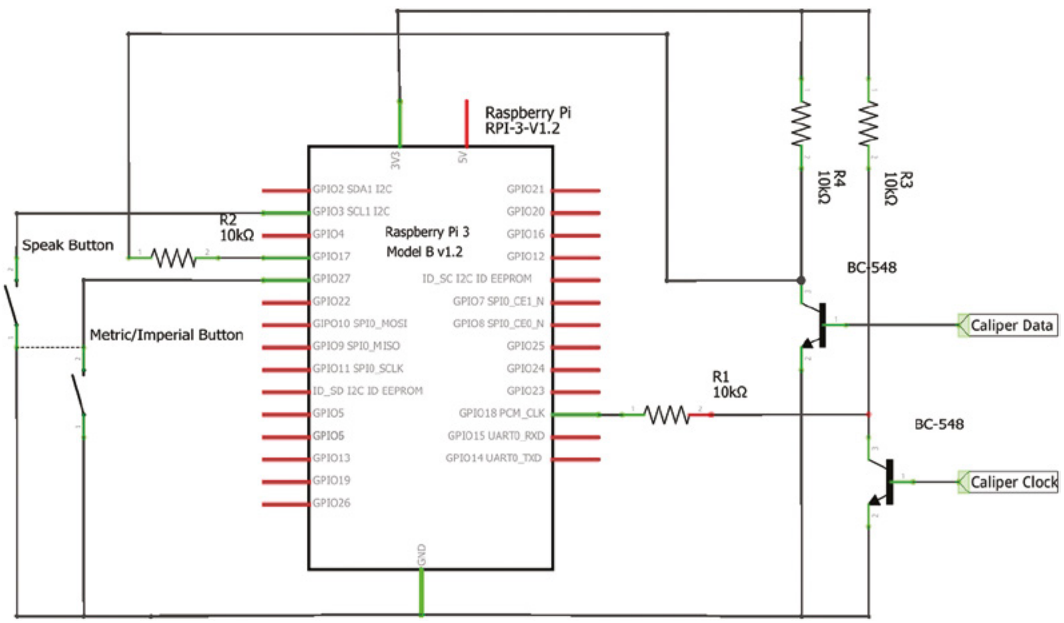
the signal, but this is easy to account for in software and doesn't need to be dealt with in hardware. As you're soldering connections to the callipers, you can also take the opportunity to remove the battery from them and use a DC-DC converter to provide 1.5 V power from Raspberry Pi's 5 V source. We used a variable buck converter, but a simple potential divider would probably be OK. If you're attaching a button to the callipers to trigger the speech process, it's a good time to add this, too. Follow the circuit diagram for the project to connect the callipers and buttons to Raspberry Pi.

RUNNING AUTOMATICALLY

To run the digital calliper application automatically when Raspberry Pi starts, you can add it to your **rc.local** file after you run pigpio. You need to make sure that the application doesn't try to access pigpio until the daemon has started. We added a two-second delay at the beginning of the code to accomplish this. Don't forget to append the & symbol to the end of the line when you execute the Python script from **rc.local**, or your app won't run in the background, and Raspberry Pi won't boot correctly.

With the hardware connected, it's time to look at how to read the data from the callipers using Raspberry Pi. Raspbian isn't a real-time operating system, and that poses a problem when you're trying to process a streaming signal reliably. Operating system functions like updates, garbage collection, and any number of system processes can cause your program to halt or slow down, which means that accurate timing is almost impossible. Thankfully, you can use pigpio (**hsmag.cc/b0hDTn**) to overcome this problem. Pigpio is written in C with a Python interface, and uses a daemon to access the GPIO with a high degree of accuracy. To use pigpio in this project, you need to download and install it using the instructions at **hsmag.cc/MRU8V9**, and then set the pigpio daemon to autostart when Raspberry Pi boots. The easiest way to do this is to add the command `sudo pigpiod -s 1 -b 1000 &` to your **rc.local** file just before the end. If you don't run the daemon, Python won't be able to access the GPIO pins with pigpio.

Using pigpio, the Python app can use interrupts to monitor the clock and data signals from the callipers, and run code every time the values change. The Python application also uses timing information to detect the gaps between the different packets of information, so you can detect the start and end of each packet.



Left The transistors shift the voltage level to let the 1.5V callipers trigger the 3V GPIO pins

MEASURE TWICE, SPEAK ONCE

Making your Raspberry Pi speak is relatively simple in comparison to the process of getting it to read the digital callipers. The Festival speech engine can be installed from the standard repository on Raspberry Pi using the command **`sudo apt-get install festival`**. There are other more natural-sounding text-to-speech implementations available on Raspberry Pi, but they typically require a connection to the internet to process speech. One of the big advantages of Festival is that it is a standalone application that doesn't need to connect to a remote server. There are Python wrappers for Festival, but it's easy enough to use the os module in Python to execute an echo command, and pipe that into the festival application like this: **`os.system("echo 'ALL OF THESE WORLDS ARE YOURS'|festival -tts")`**.

> **Using pigpio, the Python app can use interrupts to monitor** the clock and data signals from the callipers

MAKING YOURSELF HEARD

You'll need to add some sort of audio amplifier and speaker to the project so that you can hear Raspberry Pi speaking. We used a simple 3W audio amplifier with a speaker plugged into one of the channels, and the amplifier connected to the audio output of Raspberry Pi. Depending on how you intend to use the callipers, you could also connect to headphones or a Bluetooth speaker.

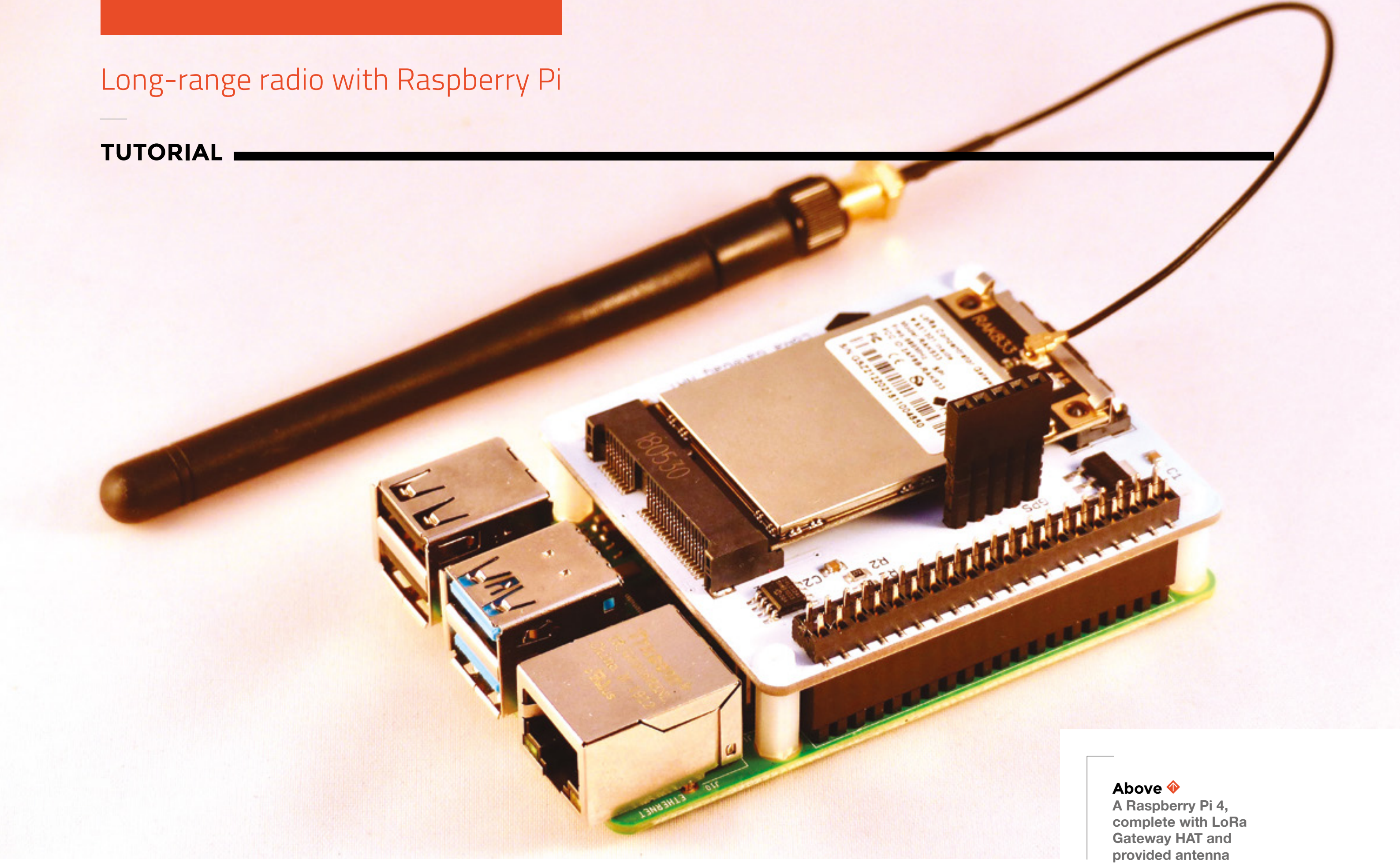
Taking the data from the calliper and sending it out to the system using appropriate language is easy when you are working in millimetres. Imperial measurements are a little bit more complicated, and need some extra processing. There's a function in the Python application called **`sayFractions`** that takes care of this for you and makes spoken fractions sound a little bit more natural. For example, saying "One point two five inches" doesn't sound correct, because it's more usual to say "One and a quarter inches".

MEASURE THE POD BAY DOORS, HAL

So, you've seen how to capture data from a digital calliper and use a transistor to level-shift it into a format that Raspberry Pi can work with. You've also seen how you can use the timing of the calliper's clock signal to synchronise the packets of data and decode measurement information using pigpio and Python. Finally, you've seen how to install and use Festival to get your Raspberry Pi to speak from Python. This project could be adapted to suit many purposes, including an audio-visual DRO (digital readout) for a lathe, milling machine, drill-press, or other tools. It could also be used to provide feedback from a linear actuator or be modified into a water level indicator. □

QUICK TIP

The exact purpose of every bit transmitted from the callipers isn't important, as long as you can recover the measurement data.



Above A Raspberry Pi 4, complete with LoRa Gateway HAT and provided antenna

Long-range radio with Raspberry Pi

Use Raspberry Pi to connect sensors to the internet, even if they're miles away!

PJ Evans

@mrpjevans

PJ Evans is a developer and wrangler of the Milton Keynes Raspberry Jam. He runs a LoRa gateway, which is probably the nearest he'll get to his own radio breakfast show.

Many of the projects likely to be found in these pages will involve hooking up some kind of sensor to the internet and recording the results. This normally implies some kind of WiFi capability, which reduces range and increases power requirements. What if you wanted to receive data from a mile away, or from a point where you can't easily supply power or networking? In this tutorial, we're going to solve this using long-range radio (LoRa) with Raspberry Pi.

We're entering an age where the Internet of Things (IoT) is commonplace. You no longer have to be a computer to be on the internet – you can be a washing machine, toaster, or a car. A common problem, especially in the agricultural world, is the infrastructure required to connect large numbers of sensors wirelessly. After all, your fanciest WiFi router won't cover several acres of farmland, and covering a large city takes a multitude of hotspots and is fraught with logistical challenges. A popular solution is long-range radio (LoRa), a networking solution designed for low-power remote devices. Commercial devices using LoRa can achieve wireless connectivity over several miles, with none of the costs or power requirements of using a mobile data network.

SMART FARMS AND CLEVER CITIES

LoRa has found applications in farming for monitoring soil quality and in urban planning as an inexpensive way of monitoring traffic, population, and air quality. In recent years, the Milton Keynes Smart City project used LoRa extensively to network smart water meters, and implement thousands of

Above
Mounting your gateway outside can increase the range by miles. This example from Sixfab (hsmag.cc/3orDvd) is powered by Raspberry Pi

A LITTLE GOES A LONG WAY

LoRa is designed for small amounts of data over long distances. It's not fast, and there's no practical application even for images, never mind audio or video. However, with a bit of planning, you can achieve long distances: 1–2 km is considered 'easy' and with the right antenna in the right place you can get 25+ km. The really determined in perfect conditions could challenge the record of over 700 km. All of this without any data costs, and very low power requirements. Using deep-sleep modes, some battery-powered LoRa sensors are rated to last over ten years.

YOU'LL NEED

- **Raspberry Pi 4**
- **Pi Supply LoRa Gateway HAT**
- **micro:bit** (optional)
- **micro:bit LoRa Node** (optional)

parking sensors across the city. Now, prices are coming down, and LoRa technology is within the grasp of the maker, we can add LoRa sensors ('nodes') to our own projects.

ALL THE THINGS

LoRa is foundation radio technology; on its own it's not very useful. LoRaWAN (Long-Range Wide Area Network) is a software layer that sits on top of LoRa and gives us a common protocol to work with. The final step is to get the data somewhere useful, like the internet. To do that we need a LoRa gateway, a device that can receive data from nodes and then forward them to a server so we can do something with the data. A popular community-led project is 'The Things Network' (TTN), and we'll use this service to get our data where we need it.

> "LoRa is foundation radio technology; **on its own it's not very useful**"

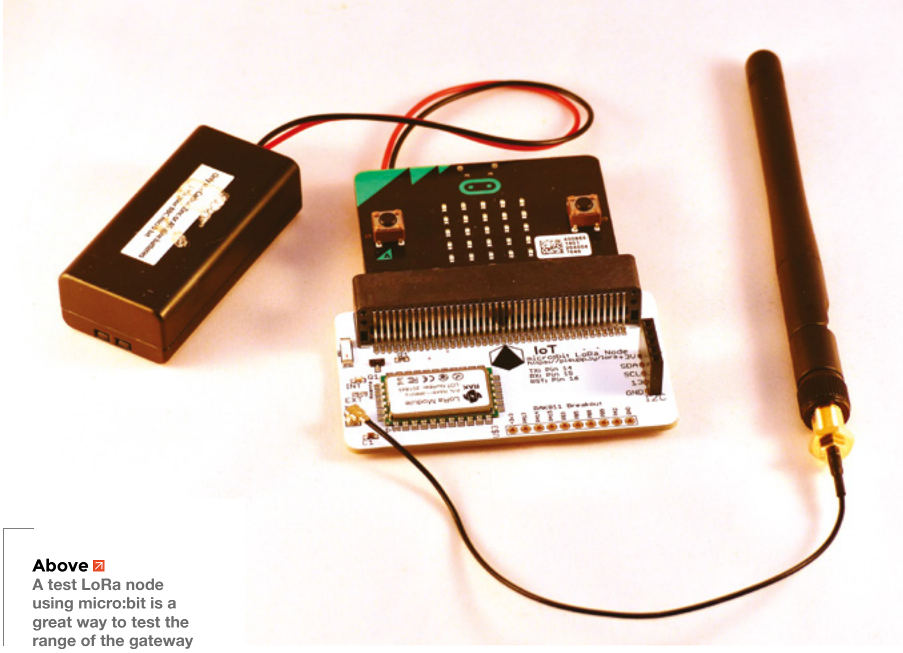
LoRa has been brought to Raspberry Pi via a successful Kickstarter campaign by Pi Supply. Rather than buy expensive commercial LoRa gateways, we can now lay our hands on a LoRa gateway HAT for getting data to the internet, and LoRa Node pHATs (for Raspberry Pi Zero) for building low-cost nodes. We also get the benefit and flexibility of the gateway running on Raspbian. We'll start with the gateway, then use a micro:bit with a Pi Supply add-on to create a simple temperature sensor.

Start by attaching the LoRa Gateway HAT to your Raspberry Pi 4. Pi Supply thoughtfully provides standoff posts to prevent wobble. You should also connect the antenna at this time. The supplied antenna is good enough for close proximity projects, but you can investigate adding a high-gain external antenna should that be required. To ease installation, Pi Supply provides a custom image of Raspbian Buster with everything you need, ready to go. Burn this to a microSD card using your favourite method (ours is balenaEtcher). You should now be able to boot up.

We recommend wired Ethernet for long-term installations, but WiFi is fine for most projects. Either way, you'll need to get your Raspberry Pi 4 →

QUICK TIP

Take the high ground! LoRa networks benefit from line-of-sight, so your gateway antenna needs to be as high as possible.



Above
A test LoRa node using micro:bit is a great way to test the range of the gateway

TAKE ON THE WORLD

Where LoRa becomes exciting is when you think of the applications. With a good location and a decent antenna, you can place nodes in previously inaccessible places. Simple 'black box' LoRa temperature sensors can be found online for as little as £15, and will last for years on a single charge. The Flood Network (**flood.network**) is using LoRa devices to measure river levels by bouncing ultrasonic pulses off the water surface and measuring time-of-flight. These cost a fraction of traditional systems and require no maintenance. At home, how about outdoor environmental sensors, or adding GPS to create a bicycle tracker? As the network grows, this is the opportunity to be a part of a great citizen-led project.

QUICK TIP

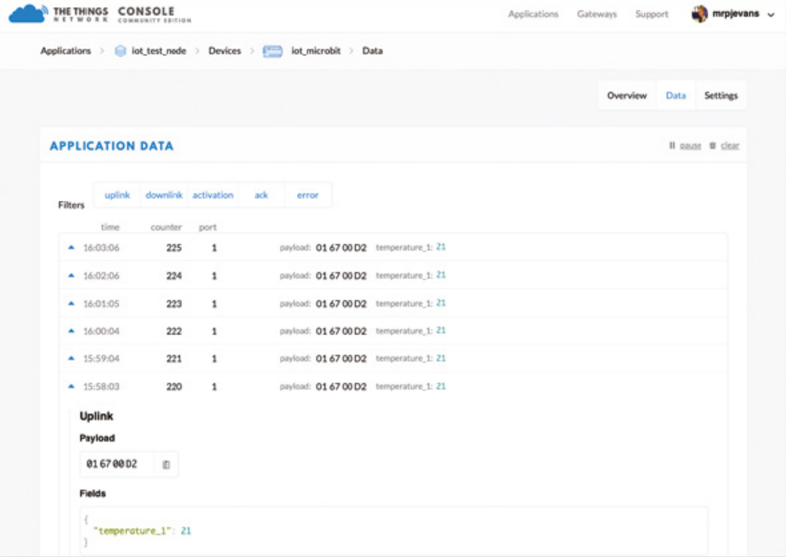
Less is more. You can choose what data you send, but the shorter the message, the quicker it will arrive. The Cayenne standard is a good place to start.

on the internet before going any further, and we recommend doing a `sudo apt update && sudo apt upgrade` to get your system up to date with the latest software. Placement of your Raspberry Pi 4 is critical to the success of the LoRa network. LoRa frequencies can be attenuated by buildings, so line-of-sight plays an important part. As a rule-of-thumb, the higher the better.

IT'S GOOD TO TALK

Now register for a free account on The Things Network (**thethingsnetwork.org**). An account allows you to join a global network of LoRaWAN gateways. Your nodes are now permitted to use other gateways to relay information to you, critical in applications such as vehicle tracking. Also, you can allow others to use your gateway if their nodes come into range. The aim is to cover the world in LoRa 'cells'. Once your account is set up, you can add a new gateway in a matter of clicks. The Things Network has a tutorial on how to configure everything at **hsmag.cc/njgdOb**.

Below The Things Network bridges the gap between nodes, gateways, and the internet



```
time       counter  port
16:03:06   225      1      payload: 01 67 00 D2   temperature_1: 21
16:02:06   224      1      payload: 01 67 00 D2   temperature_1: 21
16:01:05   223      1      payload: 01 67 00 D2   temperature_1: 21
16:00:04   222      1      payload: 01 67 00 D2   temperature_1: 21
15:59:04   221      1      payload: 01 67 00 D2   temperature_1: 21
15:58:03   220      1      payload: 01 67 00 D2   temperature_1: 21
```

Uplink

Payload

```
01 67 00 D2
```

Fields

```
{
  "temperature_1": 21
}
```

Above As part of Milton Keynes' smart meter project, LoRa gateways were mounted to lamp-posts across the city

Once your gateway has been created on TTN, you'll need to configure your Raspberry Pi 4 so it can connect securely. Again, refer to the tutorial on how to do this as there are a few things that need to be set up for successful communication. By the end of the process, TTN should be reporting that your new LoRa gateway is checking in (look for the 'last seen' parameter). Now we can set something up to send data to the gateway.

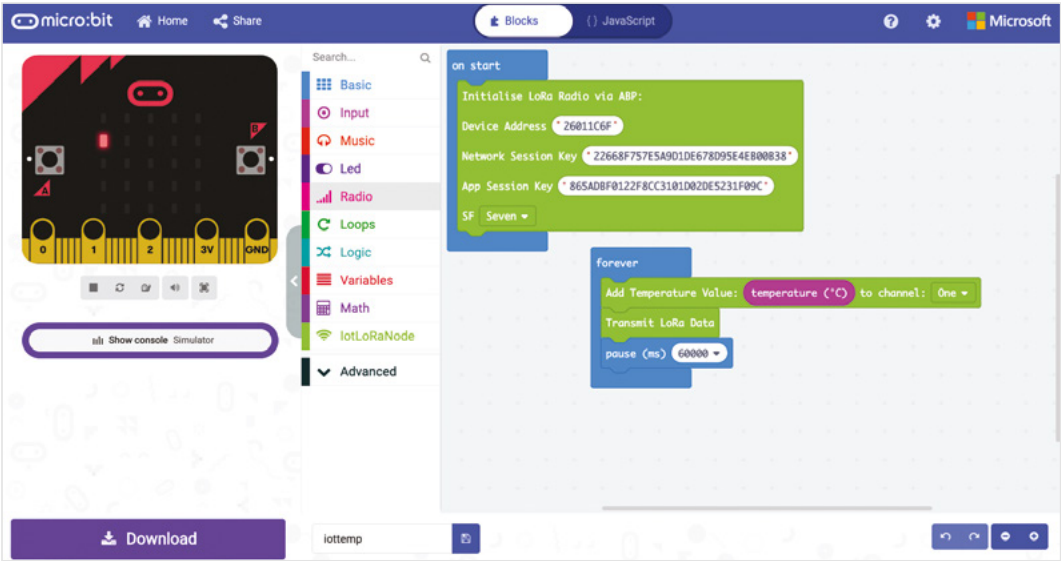
A KEY STAGE

Start by creating a new 'application' on The Things Network. You'll be assigned an access key. Then you can add any number of devices to the applications. These are the LoRa nodes that will relay information to your gateway. Each one you add will be assigned an additional access key. You'll also need to find out the LoRa device's unique address. This process differs from device to device, so check the documentation. You'll need both access keys and the device address for the next step.

There are many different ways to build a LoRa node. Microcontrollers are an obvious choice due to their low power requirements. Pi Supply offers a 'pHAT' form factor LoRa node add-on, suitable for Raspberry Pi Zero, but with that comes the need for a

CHEAP SENSORS

If you want to build a Raspberry Pi LoRa node, Pi Supply has a pHAT that is perfectly suited to the Raspberry Pi Zero. As you may not need WiFi, you can opt for the standard computer which can be purchased for as little as £4.50. Sending data is just a few lines of Python, and you can have the flexibility of the GPIO to connect any manner of sensors. It's an ideal choice if supplying power isn't an issue.



```
on start
  Initialise LoRa Radio via ABP:
  Device Address "26011C6F"
  Network Session Key "22668F757E5A9D1DE678D95E4EB00838"
  App Session Key "865ADBF0122F8CC3101D02DE5231F09C"
  SF Seven

forever
  Add Temperature Value: temperature (°C) to channel: One
  Transmit LoRa Data
  pause (ms) 60000
```

Left The 'Blocks' version of our temperature sensor code for micro:bit, created with MakeCode

wired power supply. Instead, we're using Pi Supply's board designed for the micro:bit. Setup is a matter of moments. Slide the board into the connector, attach the antenna, and you're ready to program. We were able to construct a simple temperature sensor in minutes using Microsoft's MakeCode, which offers full support for the board. For full details on setup, look at this tutorial: **hsmag.cc/zyAg1C**.

SIGNAL ACQUIRED

Once powered up, your micro:bit should now be transmitting data. Once a minute, a temperature reading is sent out. You may notice you didn't need to tell the node about the gateway. The gateway detects the signal and automatically forwards it to The Things Network. If you watch the 'data' page on the site, you'll be able to see the data packets coming in. In order to save as much bandwidth as possible, these are encoded and are not easily readable. If everything is working, you'll see something like this:

> "We were able to construct a simple temperature sensor **in minutes using Microsoft's MakeCode**"

```
01 67 00 DC
```

This particular setup uses a standard called Cayenne to format data, which can help us understand what type of data is coming in. This example is four bytes long and shown in hexadecimal format. The first byte means 'channel one'. LoRaWAN permits multiple channels per device so you can have multiple sensors. The next byte, 67, indicates that the following two bytes are a temperature reading in Celsius to one decimal point. We combine the final two bytes which, when converted to decimal, become 220. So, our temperature reading is 22.0°C.

Congratulations! You now have a data flowing over LoRa to The Things Network. So now what?

TTN is a broker, a way of getting data from LoRaWAN to you. To make use of your received data, the next and final step is to add 'integrations'. These are various services to which TTN will send your data for further processing. The most simple is to send to a webhook of your choosing. If you don't fancy getting into a bit of web programming, TTN can also transmit your data to services like IFTTT and OpenSensors. Alternatively, TTN offers free data storage for seven days that can be retrieved using its API. □

QUICK TIP

For the best possible range, you're going to want to mount your gateway outside. Make sure you use an IP67-rated case to keep everything dry!

Left An inexpensive temperature sensor node can live in traditionally difficult-to-access places, like a fridge

Getting the most out of Cura

How to tweak the settings to improve print quality

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.

The sheer number of parameters available in Cura can make it an intimidating bit of software. However, you don't have to tweak every little bit. Here, we'll give you a detailed tour of the latest Cura UI (Cura 4.4) and showcase a selection of our favourite settings, starting with the deceptively important basics, followed by some more specialised 'advanced' settings, finishing with some of the parameters from the ominously titled experimental section of Cura.

By the end of this article, we hope to have improved the quality of your prints, added to your knowledge of why different print parameters are important, and armed you with more tools to best navigate different scenarios that you might encounter while 3D-printing.

When we click the Custom button from the Print Settings tab, we are faced with an intimidating number of parameters. Thankfully, many of these are fine when set to their default, and others are only for very specific situations and will rarely (if ever) be used by most hobbyists. By clicking on the Menu button beside the search bar, we can select how many of these settings we want to be visible – our personal favourite is 'Show all settings', which does exactly what it says on the tin and gives us the full suite of options guaranteed to satisfy even the most fastidious of tinkerers.

THE BASICS

Printing temperature should be every printer's first parameter to play with. You can save temperature settings for each material by clicking Manage Materials from the drop-down menu in the Material tab. The settings may vary wildly, depending on what material you're using. If you wanted to be more meticulous, you could have unique settings for each brand, each roll of filament, and even between different colours, as the amount of pigment can affect the thermosensitive qualities of your filament. These differences can affect the amount of stringing, layer adhesion, bed adhesion and, of course, overall quality of your print. Just five degrees can make a big difference, so we recommend printing a temperature tower for each new filament you get.

Speed is perhaps the next most important property. A printer with a more rigid frame will allow you to print more quickly with minimal loss of quality. It's worth noting that if you increase the print speed, you may need to increase the temperature slightly to make sure enough filament is extruded. Experiment with different speeds while printing calibration cubes/Benchies to see which speed works best for your printer. If you are less confident about the strength of your frame, you can also reduce jerk acceleration (how fast the print head accelerates between movements) to improve print quality at the cost of slightly slower prints.

> **"The settings may vary wildly,** depending on what material you're using"

Adhesion to the build plate can be a problem on some printers. If you are struggling with your models coming loose from the build plate, we highly recommend a brim. This setting increases the surface area of your print by printing a layer of plastic around the base of the model. If your models are dislodging from the build plate due to warping (the corners begin to lift off the build plate before it comes off completely), then a raft may do the job. This setting →

YOU'LL NEED

Min specs for Cura:

- **OpenGL 2 compatible graphics card**
- **Intel Core 2 or AMD Athlon 64** or newer
- **At least 4GB of RAM** (8GB or more is recommended)
- **A 64-bit processor**

FINDING **YOUR WAY AROUND**

1 Load your STL files from here.

2 **Printer** – Clicking here will allow you to select and manage your printer profiles. Printer profiles are how Cura knows your printer's build volume is, as well as where you can specify custom start and finish G-code.

3 **Material** – You can select and manage your printer material from here. Cura has an impressive list of well-known filament brands and materials (PLA, ABS), along with predefined printing temperatures for each.

4 **Preview tab** – Here you can preview your model after you've sliced it. You will be able to see where the supports have been generated (if there are any), see all of the travels your printer will be doing, visualise the feed rate, layer thickness, and more.

5 **Monitor tab** – If you followed our OctoPrint guide from issue 26, this will be very familiar to you. The monitor tab allows manual control and live monitoring of any printer connected to your computer via USB.

6 **Print settings** – This is the good stuff. Clicking on this section gives us a small selection of print parameters, but the real power of Cura is unlocked when you click the Custom button and unveil the full suite of print settings.

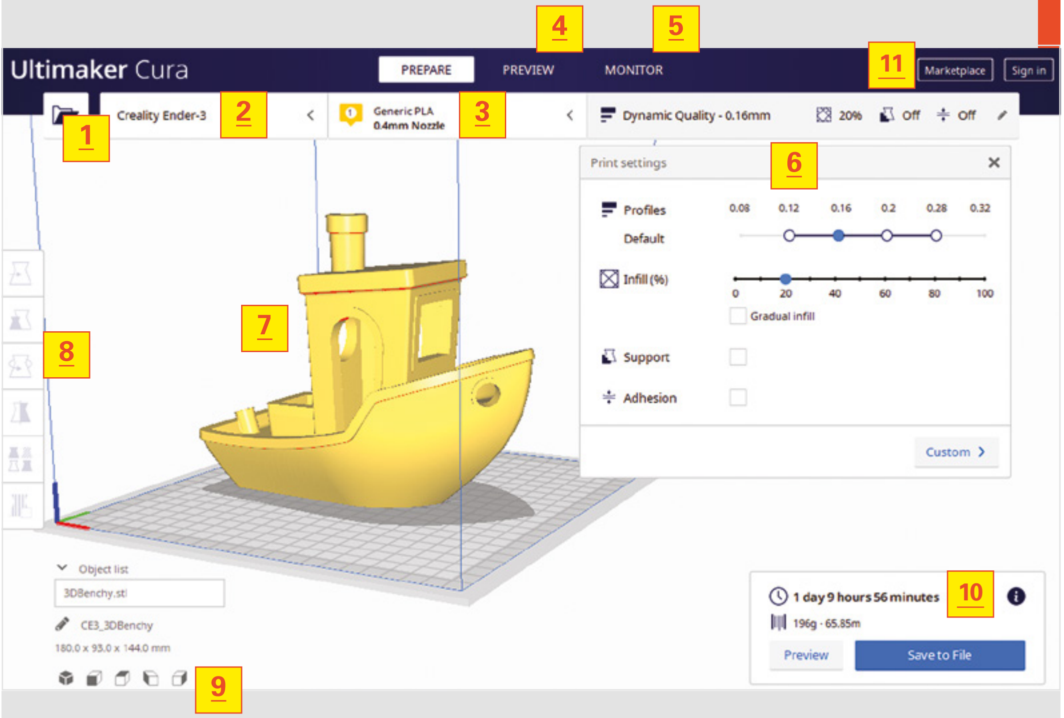
7 **Preview of the model** – This allows us to easily see how big the model is on the build plate, as well as get an idea of all its overhangs (the areas highlighted in red).

8 **Control panel** – Here you can easily rotate, scale, move, etc. the model to get it to the correct size/orientation on the build plate before slicing it.

9 **Camera controls and info** – We can use the quick view buttons along the bottom to orientate the camera to specific points. We can also change the model name and view a list of the models on the build plate.

10 **Print information** – After your model has been sliced, this is an estimate of how long it will take to print and how much material it will take to print.

11 **Marketplace** – Here you can find a plethora of useful plug-ins for Cura, as well as a number of material profiles from different manufacturers.



Creality Ender-3

Generic PLA 0.4mm Nozzle

Dynamic Quality - 0.16mm 20% Off Off

3DBenchy.stl

CE3_3DBenchy

180.0 x 93.0 x 144.0 mm

1 day 9 hours 56 minutes

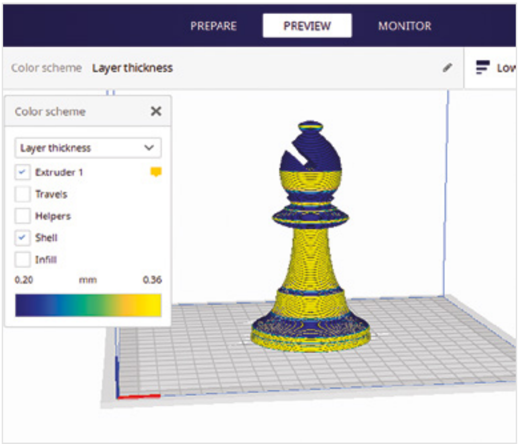
196g · 65.85m

QUICK TIP

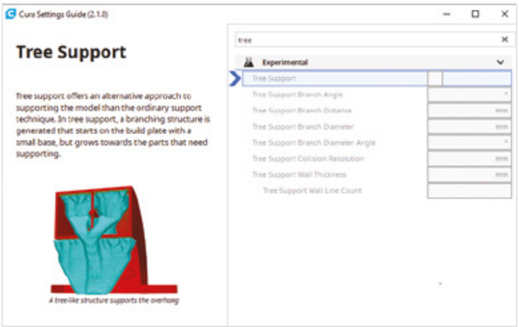
Can't find a setting? Use the search bar in the custom print settings box.

WHAT IS CURA?

Cura is a piece of open-source software from Ultimaker that proudly boasts the accolade of the world's most popular slicer. Its popularity is likely due to its easy-to-use UI, frequent updates driven by a passionate community of contributors, and its price (free!). An ever-growing list of useful plug-ins has meant that it remains popular today, even in the face of fierce competition.



Right We can use the Adaptive Layers Maximum Variation setting to set how big of a difference there is in layer size between areas of low and high detail


Cura Settings Guide (2.1.0)
tree
Experimental
Tree Support
Tree Support Branch Angle
Tree Support Branch Distance
Tree Support Branch Diameter
Tree Support Branch Diameter Angle
Tree Support Collision Resolution
Tree Support Wall Thickness
Tree Support Wall Line Count
°
mm

Tree Support

Tree support offers an alternative approach to supporting the model than the ordinary support technique. In tree support, a branching structure is generated that starts on the build plate with a small base, but grows towards the parts that need supporting.

A tree-like structure supports the overhang

Above You can find the Settings Guide plug-in in the Marketplace on Cura. Once installed, you can access it by clicking on Extensions in the menu bar

QUICK TIP

Stringing can be greatly improved by reducing temperature or enabling retraction.

prints a honeycombed base for your model to be printed on top of. The fact that it's not solid allows for a little bit of give if your material begins to warp, preventing it from peeling off the build plate.

That's the basics covered. Now let's look at some slightly more advanced stuff.

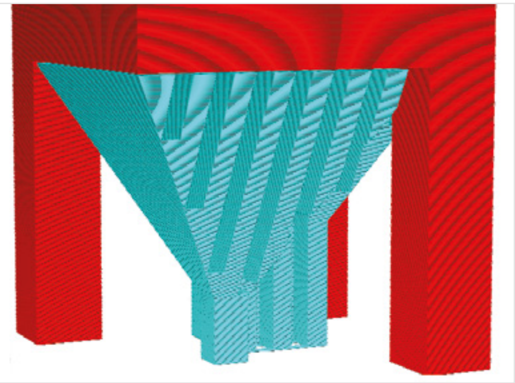
Nobody likes supports; they use up material and they make ugly interfaces where they are removed, but they're a necessary evil if we want any complexity in our prints. While supports are pretty basic, getting them right is anything but. We recommend first using the control panel on the left of the UI to orientate your print in such a way that support structure is minimised; or, if possible, such that the support surfaces are in a part of the print that won't be seen (or are in an area of low detail).

> While supports are pretty basic, **getting them right is anything but**

There are dozens of support settings, but we like reducing the support overhang angle – by default it's set to a conservative 45 degrees, but if you're using a cooling fan, you could probably push this up to at least 60 degrees (this will depend on the model). Support Z distance will affect how easy supports are to remove from a print, although if you make it too big, then the support won't do its job. Adding a support interface can result in much better-looking surfaces where you remove the support. It works by printing a roof onto the support which should break off as one clean piece from the print (in theory).

TEMPERATURE TOWERS

Temperature towers are structures that are split vertically into sections, each with their own print temperature. The idea is that with one print you can find out exactly what temperature is optimal for that filament. Typically when you slice a model, it will be printed in just one temperature. Fortunately, there are some unsung heroes on Thingiverse who have uploaded customised G-code for pre-sliced temperature towers that have the temperature settings for each layer already keyed in.



Above The Conical Support Minimum Width setting refers to the minimum size of base for the support structure (image from Cura Settings Guide)

If you want a smoother top surface on your prints, then combing is the solution. It's a setting that recently graduated from experimental into the Travel section, and it makes the print head avoid moving across the model when it's travelling from point A to point B. This prevents any 'scars' being formed from the hot end moving across the printed surface.

Have you ever had an otherwise great print display a 'scar' from the top to the bottom of your model? This is likely the area where the print head both stops and starts the outside wall of a given layer. If this area is in the same place each layer, it results in a scar known as the Z-seam. Cura version 4.2 introduced a feature known as Z-seam hiding (find it under Shell, and select Smart Hiding under the Seam Corner Preference setting) which, where possible, hides the seam by having it right on the inside/outside corners of your print.

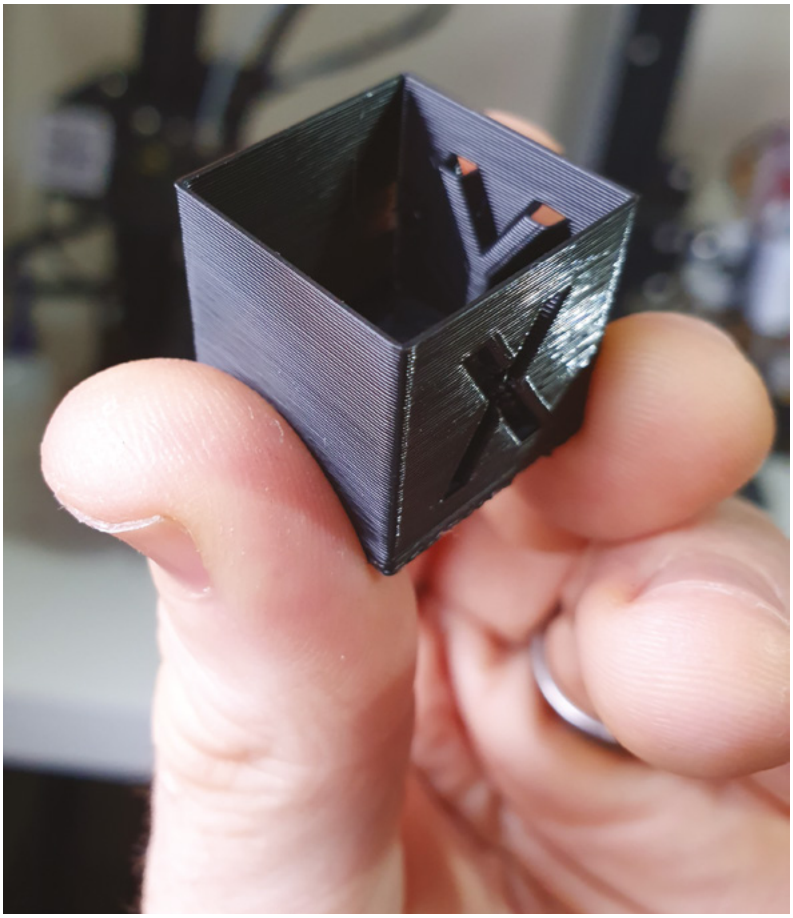
Spiralize Outer Contour results in a model with a floor and walls just one layer thick. Unsurprisingly, this results in very fast prints relative to the size of the model and is particularly useful for small component drawers and, of course, vases. It can be found under Special Modes in Cura. Why is it called Spiralize Outer Contour? Instead of printing layer by layer, this setting prints in one continuous line, slowly moving up the z axis as it prints.

GETTING EXPERIMENTAL

The Adaptive Layers feature analyses the level of detail in different parts of your model and uses a bigger layer size (lower resolution) for areas of low detail and a smaller layer size (high resolution) for detailed areas. The idea behind it is that it may reduce print times by printing most of the model in a low resolution.

Many people swear by tree supports. As the name suggests, tree supports start off on the bed as a thick trunk that splits into smaller branches to support overhangs in the model. The trunk and branches tend to hug the shape of the model rather than filling all the space between the build plate and the overhang, reducing the amount of material needed. They also tend to have fewer points of contact with the print, making them easier to remove and keeping the surface quality intact on the interface between support material and the print.

Conical supports are another relatively new feature to Cura. If you are not a fan of tree supports, conical supports are a great way to save on support material. Similar to the tree structures, conical supports don’t fill the entire space between the overhang and the build plate. The supports start off with a small amount of contact with the build plate, then expand as they reach the overhang to give a large amount of contact with the surface to be supported. This not only cuts down on print time, but also significantly reduces the amount of material needed. □



CREAWESOME

Creawesome started its life as a plug-in for Cura that primarily added print profiles for Creality printers that promised excellent-quality prints. It was fully integrated into Cura in version 4.2 and, if you are one of the many people who use the incredibly popular Creality Ender 3 (or any Creality printer for that matter), you now have all these fine-tuned settings as default. Check out the details here: **hsmag.cc/YB8BNu**.

QUICK TIP

The Cura Settings Guide plug-in will give you detailed descriptions of virtually all of the parameters.

Above You can use Vase Mode on virtually any model, with wildly variable results. Here it is with a calibration cube

Left The Z seam is harder to hide in circular models. One technique is to randomise the placement of the Z scar from layer to layer to prevent one continuous line on the print

The Joule Thief

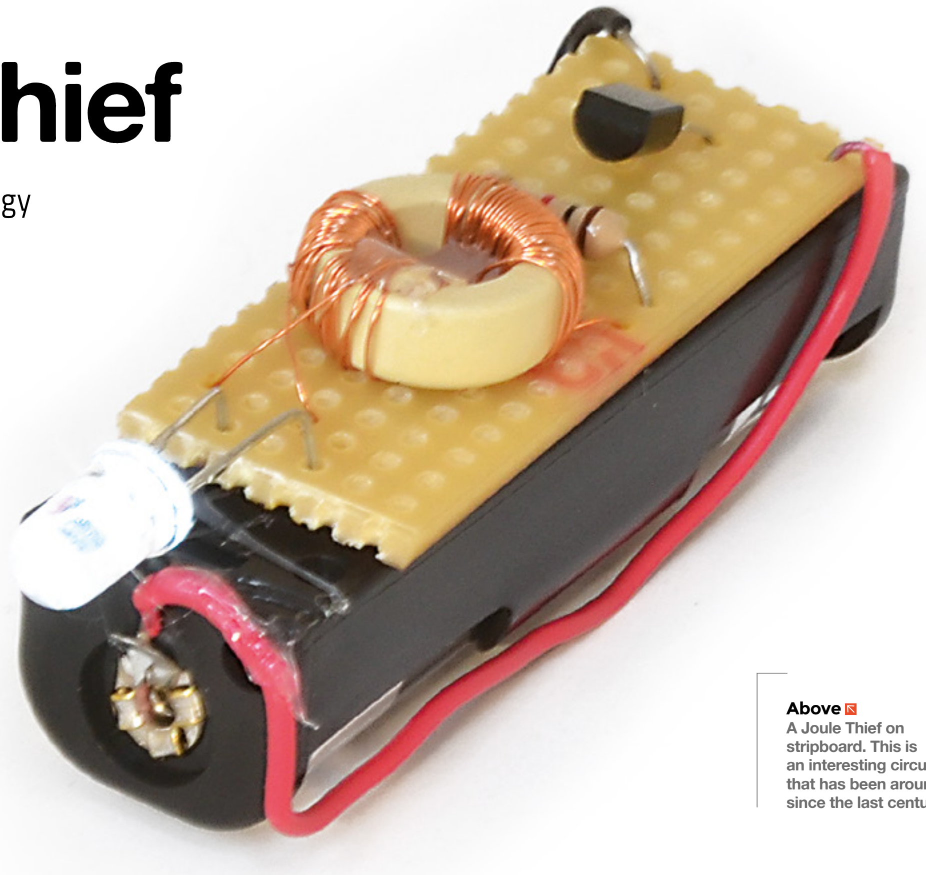
Squeeze every joule of energy from a single AA battery

Simon Monk

@simonmonk2

Simon divides his time between writing books and designing products for Monk Makes. Some of his better known books include *Hacking Electronics* (TAB) and *Electronics Cookbook* (O'Reilly).

simonmonk.org



Above A Joule Thief on stripboard. This is an interesting circuit that has been around since the last century

This well-known electronic circuit is called a Joule Thief because it is said to steal every last joule of energy from a battery. A fresh alkaline AA battery will register a voltage of about 1.6V. This will quite quickly fall to about 1.2V when the battery is in use. It will still contain useful energy right down to about 0.6V, but most things that you might want to power from an AA battery will usually give up once the battery is much below 1.2V – wasting energy. This circuit serves two functions:

- It allows you to use almost all of the energy stored in an AA battery
- It boosts the meagre 1.5V battery so that it can light white and blue LEDs that require 3V or more to light

This project uses a tiny, low-power transformer that you are going to make by winding two coils of wire around a toroid (doughnut shape) made of a magnetic material called ferrite. **Figure 1** shows the finished transformer.

Start by cutting two lengths of wire, each about 40cm (16 inches). The number of turns and the thickness of the wire are not critical.

Notice how we have marked where the wires leave the toroid as Under or Over. Pay particular attention to this so that your finished transformer ends up looking like **Figure 1**.

Leave a centimetre or two (half an inch) of wire extending from the transformer for each winding. The transformer is a bit delicate unless it's fixed to something rigid. So, this project uses a small piece of stripboard (also known as Veroboard) to provide a bit of protection to the transformer and also to connect together the other components in the project.

QUICK TIP

Stripboard is a type of general-purpose circuit board designed for making prototypes and home projects like this.

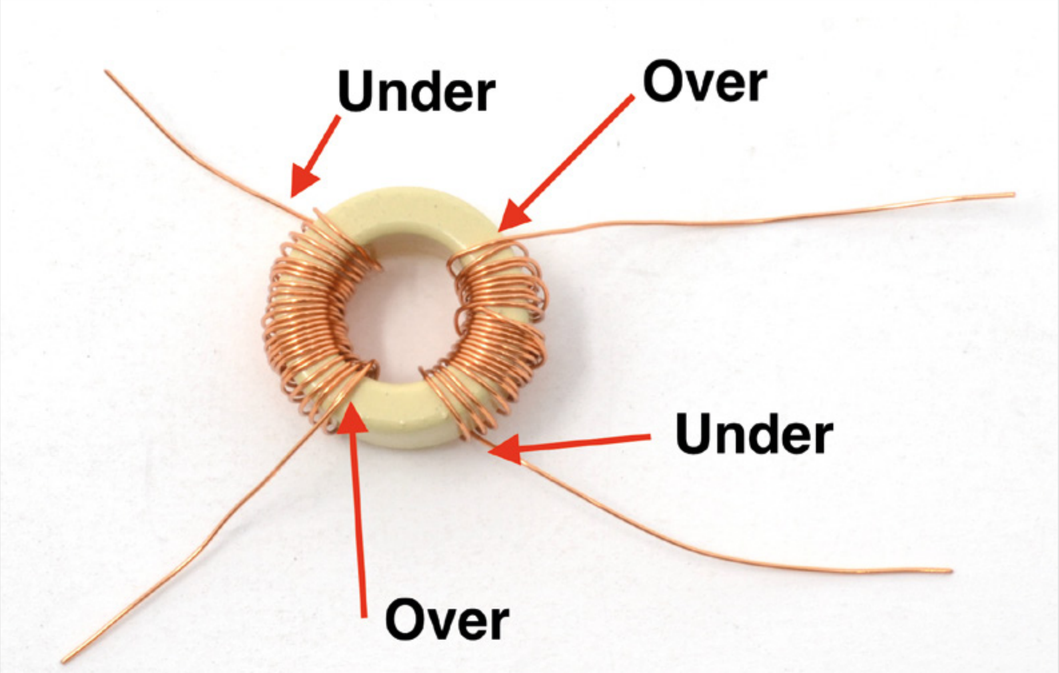


Figure 1 The finished homemade transformer. It doesn't matter if it's a bit untidy

LINING UP

Stripboard is made up of a perforated board that has strips of copper running along the board. The idea is that you can push the legs of your components through from the non-strip side of the board and then solder the leads to the strips on the underside.

You can break stripboard down to the size you want by scoring a line with a sharp knife and then breaking it over the edge of your table.

Use **Figure 2** as a guide to where you should be inserting the component legs and transformer wires through the top of the stripboard. Starting with the transformer, push the wire ends through the holes on the top side of the stripboard. Notice how the transformer wires on the right of the stripboard cross over.

The insulation on the transformer wire is quite thin and delicate, so rather than trying to strip it away from the copper wire underneath, if you keep the soldering iron on the wire a bit longer than normal, then the insulating enamel layer will quickly burn away. Do not try this if you are using wires with plastic insulation. In this case, strip the insulation off the wires using pliers or wire strippers.

Remember to keep the soldering iron in place for a few seconds extra to allow the molten solder to burn through the transformer wire's insulation; otherwise, the solder may grip the wire by its insulation and not make a good electrical contact. Once the joint is made, you can cut off any excess transformer wire.

You can now solder the other components in a similar way by pushing their legs through from the top of the stripboard. Start with the resistor, which can be either way around. →

> "**The transformer is a bit delicate** unless it's fixed to something rigid"

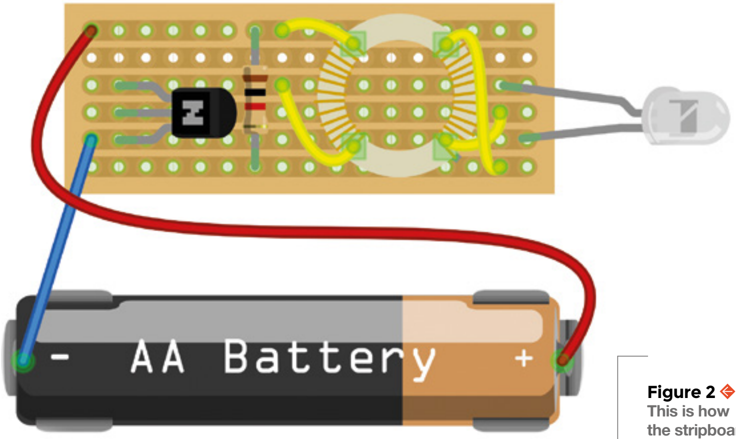
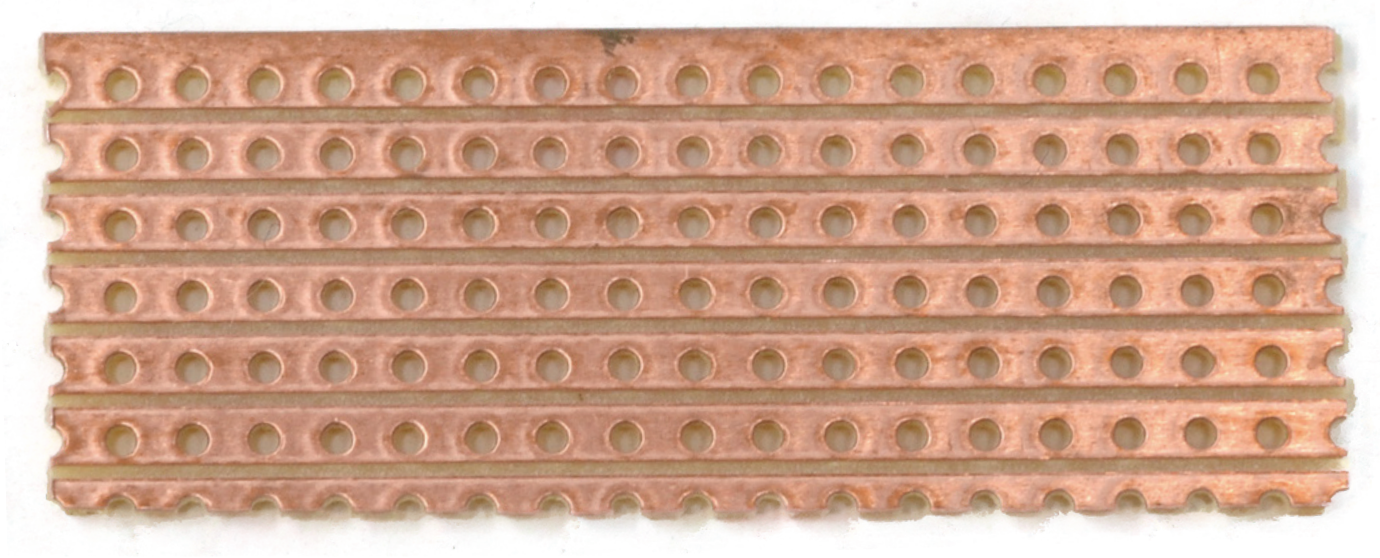


Figure 2 This is how the stripboard is arranged

YOU'LL NEED

- **Soldering equipment**
- **Glue**
- **2N3904 NPN bipolar transistor**
- **LED** (any colour)
- **1 kΩ resistor**
- **Ferrite toroid** We used this 13 mm one from CPC (FT01589), very similar to this one from Mouser (205-MP1303P4AS), but almost any toroid will work. The bigger the toroid, the easier it is to wind the wire onto it
- **Roughly 1 m (3 feet) of enamelled transformer wire** We used 34 SWG (32 SWG, 0.234 mm diameter) but any insulated copper wire that can fit two windings of 25 turns each will work fine
- **Red and black hook-up wire**
- **Small piece of stripboard** (5 strips by around 17 columns) – Mouser 854-ST1, CPC PC01601
- **Single AA battery holder with solder tag terminals** Like this one from CPC (PP002050) or this from Mouser (12BH311-GR)



Above Stripboard is great for making prototypes

Right With the transformer wires poking through from the top side of the board, flip the board over and solder the transformers to the stripboard by pressing the soldering iron and solder to the point where the transformer wire emerges from the stripboard hole

When it comes to the transistor, make sure that it's the right way around, with the curved edge pointing to the right of the board (see **Figure 2**). The LED also needs to be fitted the right way around. Its positive lead is slightly longer than the other lead. The positive LED lead should be to the top of the board, as shown in **Figure 2**. Leave quite long leads on the LED so that when you've soldered it, it can be bent through 90 degrees to point outwards like a flash-light.

When all the components are soldered into place, the top and undersides of the stripboard should look like **Figure 3** and **Figure 4**.

Solder lengths of hook-up wire to the stripboard, as shown in **Figure 5**. These need to be long enough to reach the battery box, so make the red lead a bit longer as this is going to have to reach to the positive terminal of the battery.

Before you finish the soldering and gluing the stripboard onto the battery box, test the project by holding an AA battery to the wires. The positive end of the battery should be connected to the red positive wire.

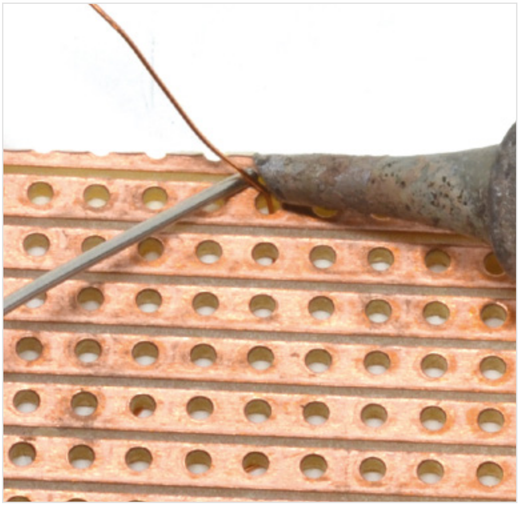
If all is well, the LED will light and you can solder the ends of the hook-up wires to the battery box. Make sure that you get the black (negative) wire to the negative end of the battery box (this end will have a spring).

Finally, put a blob of glue into the toroid to fix it onto the stripboard and then glue the stripboard to the flat side of the battery box. □

HOW DOES IT WORK?

Figure 6 shows the schematic diagram for the project. Technically what we have here is a self-oscillating boost converter. The transistor acts as a switch that turns on and off very quickly. A small current flowing into the base of the transistor turns the transistor on, allowing a current to flow the collector to the emitter. This causes pulses of power through the left-hand coil of the transformer that boost the voltage so that it is sufficient for the LED to light.

The reason that the circuit oscillates is that the base of the transistor is connected to the right-hand winding of the transformer. The resistor R1 provides a positive voltage supply to the right-hand winding and also limits the current flowing into the base, as too much would damage the transistor. Looking at the oscilloscope plot of **Figure 7**, the top (orange) plot shows how the voltage across the LED varies over time, and the bottom (blue) plot shows the voltage at the base of the transistor. You can see how the base voltage mirrors the collector voltage, effectively turning the transistor on as soon as it's off and vice versa. The plot also shows that the LED is turning on and off at a frequency of 134.2kHz. That's 134.2 thousand times a second!



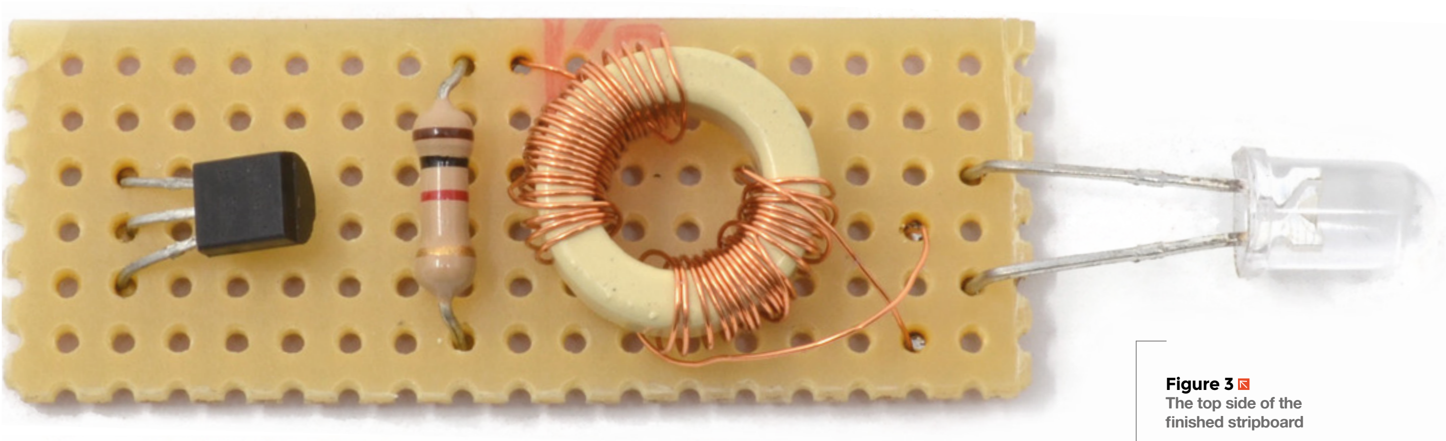


Figure 3
The top side of the finished stripboard

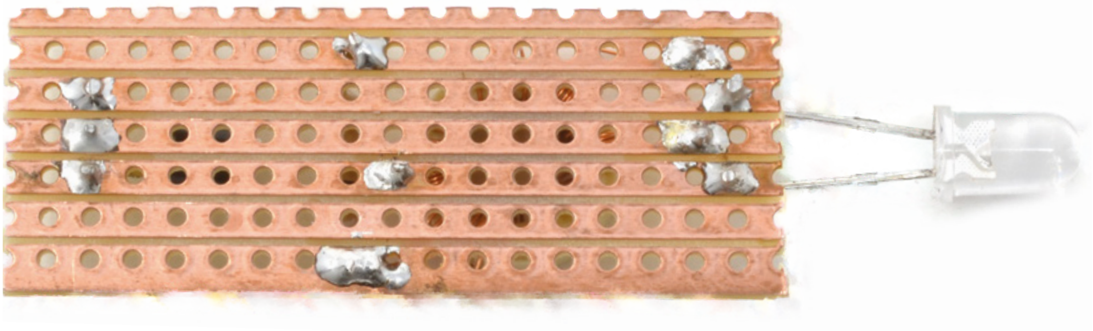
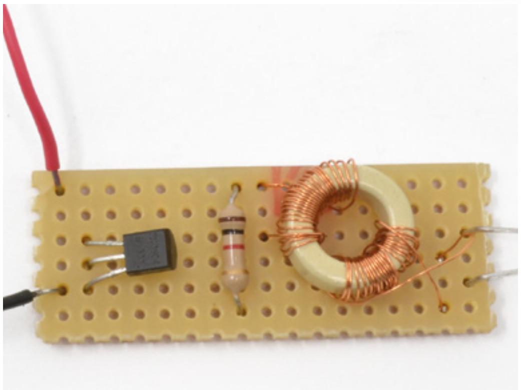


Figure 4
The bottom side of the finished stripboard (flipped top to bottom). Make sure that you don't get blobs of solder between the strips



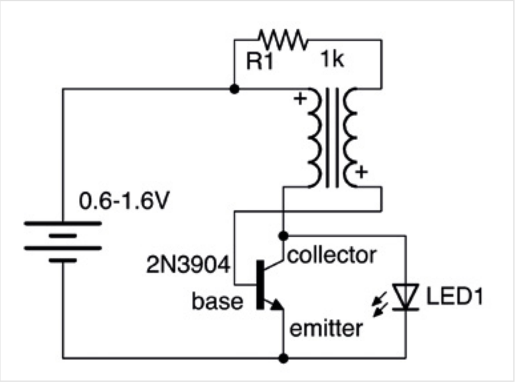
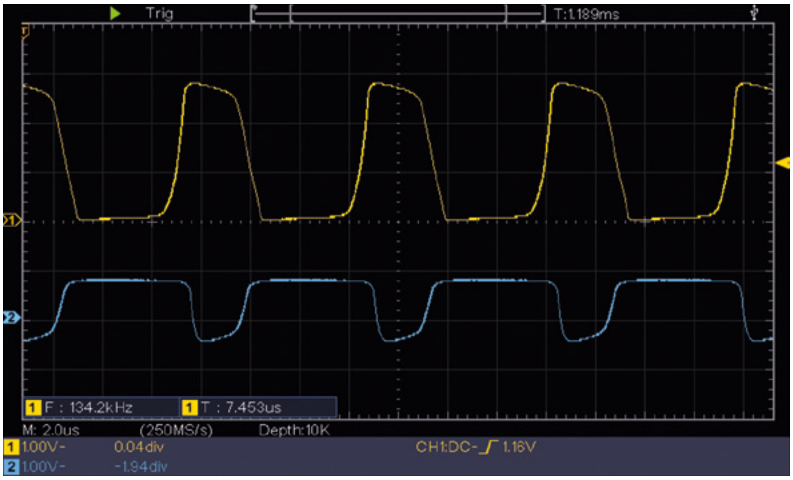


Figure 5
Hook-up wires soldered to the stripboard

Figure 6
The schematic diagram for the project makes it easier to see how the circuit works

Figure 7
The oscilloscope plot showing the oscillation

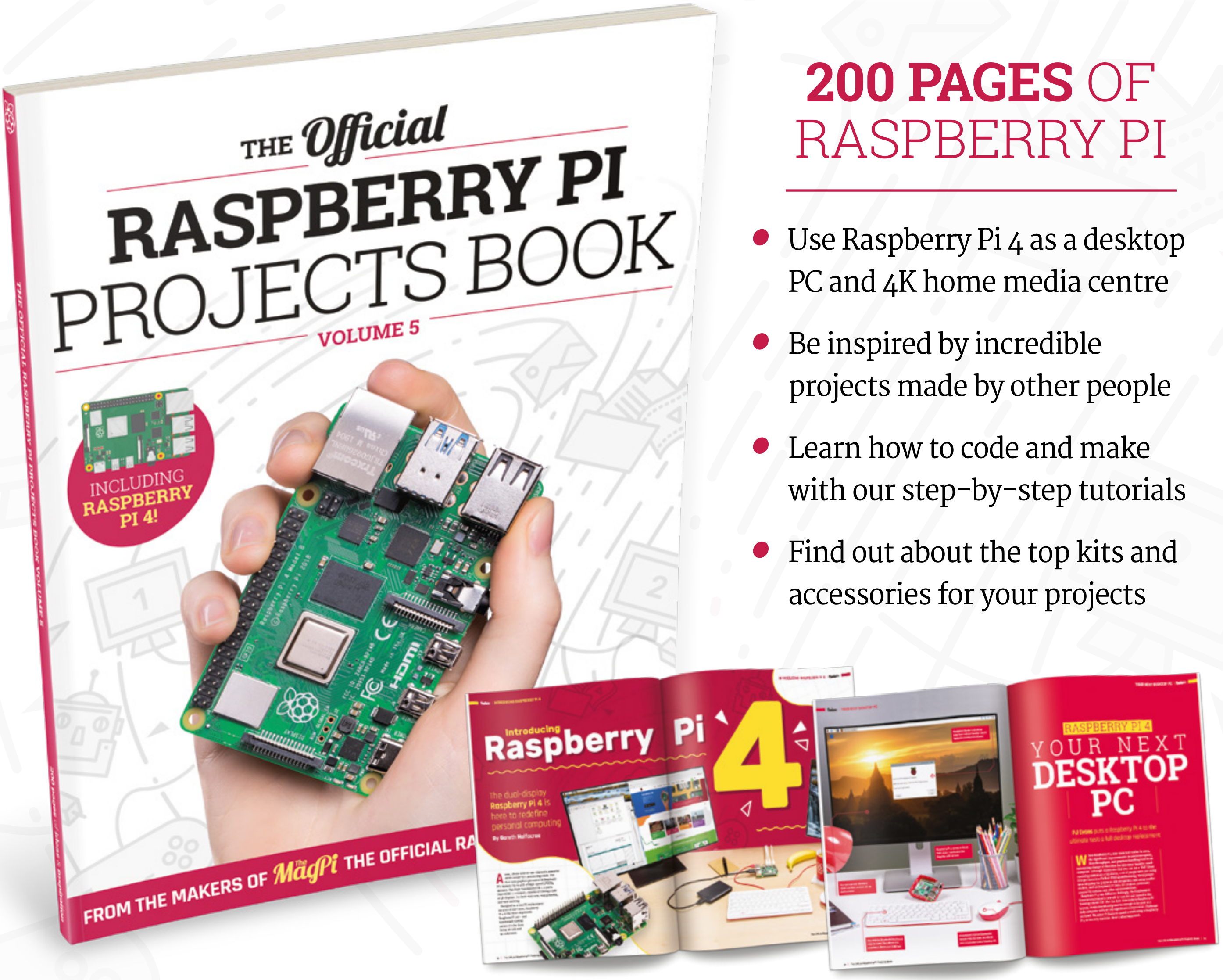


QUICK TIP

Aim for 25 turns for each winding, but if you are using thicker wire and can't physically fit 25 turns onto the toroid, then you will probably find that 20 or even 15 turns will work. The best wire for this job is transformer wire. This wire has a very thin, transparent insulating layer covering copper wire. Generally, you have to buy this in whole reels of wire, much more than you will need for this project.

THE *Official* RASPBERRY PI PROJECTS BOOK

VOLUME 5


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 5
INCLUDING RASPBERRY PI 4!
FROM THE MAKERS OF The MagPi THE OFFICIAL RA
Introducing Raspberry Pi 4
YOUR NEXT DESKTOP PC

200 PAGES OF RASPBERRY PI

- Use Raspberry Pi 4 as a desktop PC and 4K home media centre
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store

Programmable power supply

Use an Arduino to set your voltages

Will White

Will is a maker who's past projects include converting a 3D printer into a CNC mill

Sometimes you need to step down the voltage within your project. This is when voltage regulators come in handy. The three main types of regulators you will probably come across are: linear regulators, buck converters, and boost converters (we looked at these in more detail in issue 21).

You can buy these voltage regulators as modules with all the components soldered onto a board, ready to go. Many modules come with either a large potentiometer or a smaller (often blue) trimpot for manually setting the output voltage.

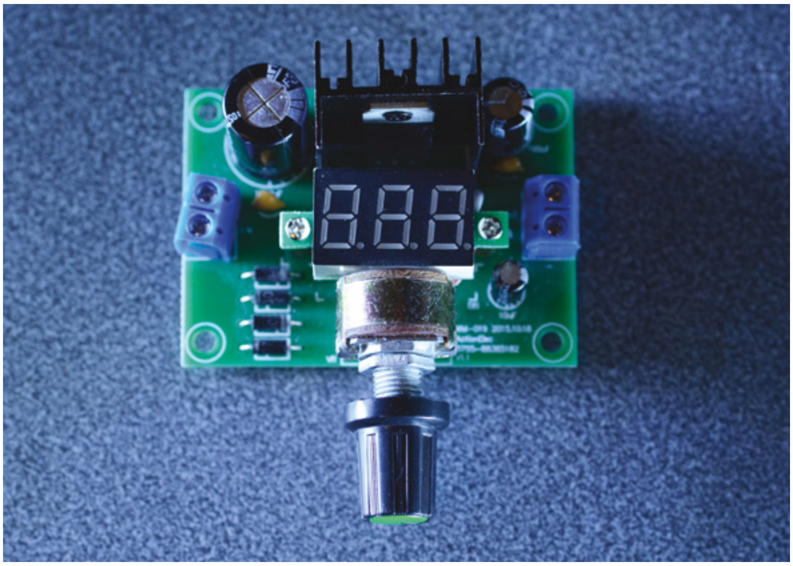
But what if you want your microcontroller to set the voltage for you?

What if you wanted an Arduino-controlled bench power supply, or a remote project that ramps up power as and when it's needed?

There are many ways you can do this using digital potentiometers, or the PWM signal from your microcontroller. We're going to use a digital-to-analogue converter to set the voltage and let the voltage regulator deal with the load regulation, leaving the microcontroller free to run other stuff.

For this tutorial, we're going to use an LM317 voltage regulator module, a 12 V 1.5 A power supply, a TLC2272 op-amp, some resistors, some 10 kΩ multi-turn trimpots, an MCP4725 digital-to-analogue converter (DAC), and an Arduino Uno. You will also need to be happy de-soldering some components – the data sheet for your voltage regulator will be useful, and you'll need a multimeter.

The LM317 linear voltage regulator has three pins: input, output, and feedback. By applying a voltage to the feedback pin, we can set the voltage of the output. The LM317 has a dropout voltage of about 3 V. So, if we supply it with 12 V, the maximum we'll get out is about 9 V, and the minimum should be the reference voltage of 1.25 V. However, in reality, we got much closer to 2 V due to limitations of our DAC.



Below The power supply module we started with had a manual adjuster

PICKING PARTS

When buying a voltage regulator module, look for one with a through-hole potentiometer – it will be easier to modify compared to one with surface-mount components.

First, we're going to remove the large 10 kΩ potentiometer from the module. It will later get replaced with a smaller multi-turn pot. This stage is optional, but it will make fine control easier.

Looking at the underside of the module, we can see that the central pin and one side of the pot are connected together. On our module, this was ground, and the remaining side of the pot connected to the feedback pin of the LM317.

Solder wires to both these positions, one to GND and one to the Feedback.

On a breadboard, add the DAC and connect Vin to the Arduino's 5 V pin, GND to the Arduino's GND, SDA to A4, and SCL to A5.

Then connect Vout from the DAC to the non-inverting input of the op-amp (pin 3).

On the op-amp, connect Vin (pin 8) to the 12 V supply to the voltage regulator; connect the output (pin 1) to both the central pin of a 10 kΩ multi-turn trimpot and via a 1 kΩ resistor to the feedback pin of the regulator. One of the remaining pins on that trimpot will connect to the inverting input of the op-amp (pin 2); add a 10 kΩ resistor between the inverting input and ground (pin 4). The 10 kΩ resistor and the trimpot set the gain of the op-amp; if you only want the output voltage the same as the output of your DAC, omit these two resistors and connect the output from the op-amp to its inverting input.

GET CONNECTED

If you removed the potentiometer earlier, add a 10 kΩ multi-turn trimpot between the feedback and ground of the regulator. Finally, check that all ground pins are tied together.

Now the regulator is set up, we need to add some feedback to the Arduino. Create a voltage divider between the output of the regulator and ground: connect two 10 kΩ resistors in series between the regulator's output and ground, then the A0 pin from the Arduino to the junction between the two resistors. This drops the voltage from 10 V to 5 V for the **`analogRead`** function at a ratio of 2 (Vin/Vout). If your output is higher than 10 V or your microcontroller uses 3.3 V for its logic, you'll need to change these resistors.

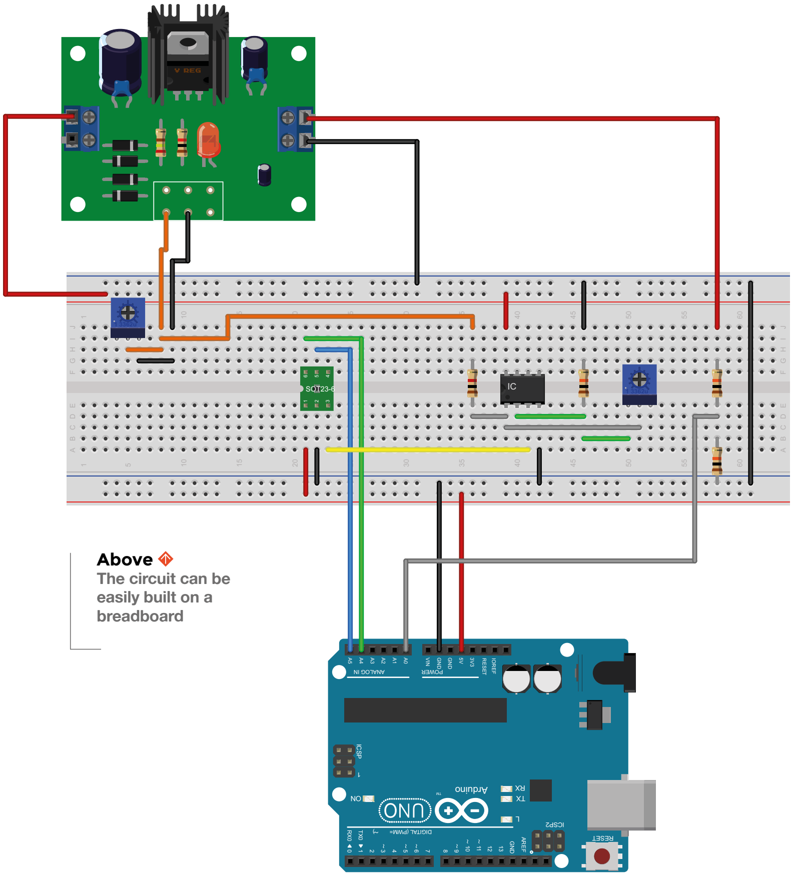
> " Once you have done both potentiometers, go back and **check them again as one will affect the other** "

In the Arduino IDE, check you have up-to-date versions of the Wire and Adafruit MCP4725 libraries.

First, run the **i2c_scanner** sketch (File > Examples > Wire > i2c_scanner) and make a note of the address for your DAC (ours was 0x60).

Temporarily pull the wire from the DAC's Vout and connect it to the Arduino's 5 V pin, or set the DAC's output to 4095, then connect a multimeter to the output of the regulator.

Now adjust the potentiometers, starting with the one for the regulator, until the voltage is only just at its maximum or your desired maximum voltage. Once you have done both potentiometers, go back and check them again as one will affect the other.



Above The circuit can be easily built on a breadboard

If you removed the DAC's output wire, reconnect it. You can download the Arduino sketch from **hsmag.cc/issue27**. Don't forget to change the DAC's address in the setup loop: **`dac.begin(0x60)`** – and the **`Voltage_Divider_Ratio`** if you are not using two identical resistors.

Open the serial monitor on the Arduino IDE and set the Baud rate to 9600, line ending to 'no line ending'. Type in a voltage between the minimum and maximum voltages and watch what happens.

Place your multimeter probes on the voltage regulator's outputs, and check to see how close the Arduino's readings are to the correct voltages. If the readings are out, try measuring the Arduino's 5 V pin and put that reading into the sketch: **`float Vref = [multimeter reading];`**. The sketch works by taking an approximation of the set voltage, then adjusts the output until it's within tolerance.

Now we have something that will set the voltage; we could combine it with a rotary encoder to adjust the voltage, or use the values in an array in order to make the response much quicker and cut down the hunting for the desired voltage.

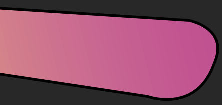
When using this trial-and-error style to building circuits, be careful not to accidentally overload something. □

CODE
THE
CLASSICS
VOLUME 1
CODE THE CLASSICS VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²
Raspberry Pi PRESS

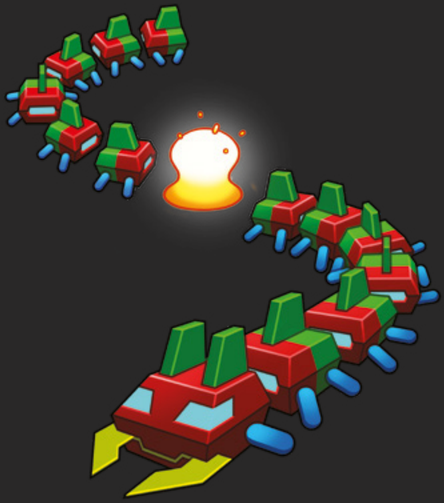
CODE THE CLASSICS

VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.



- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

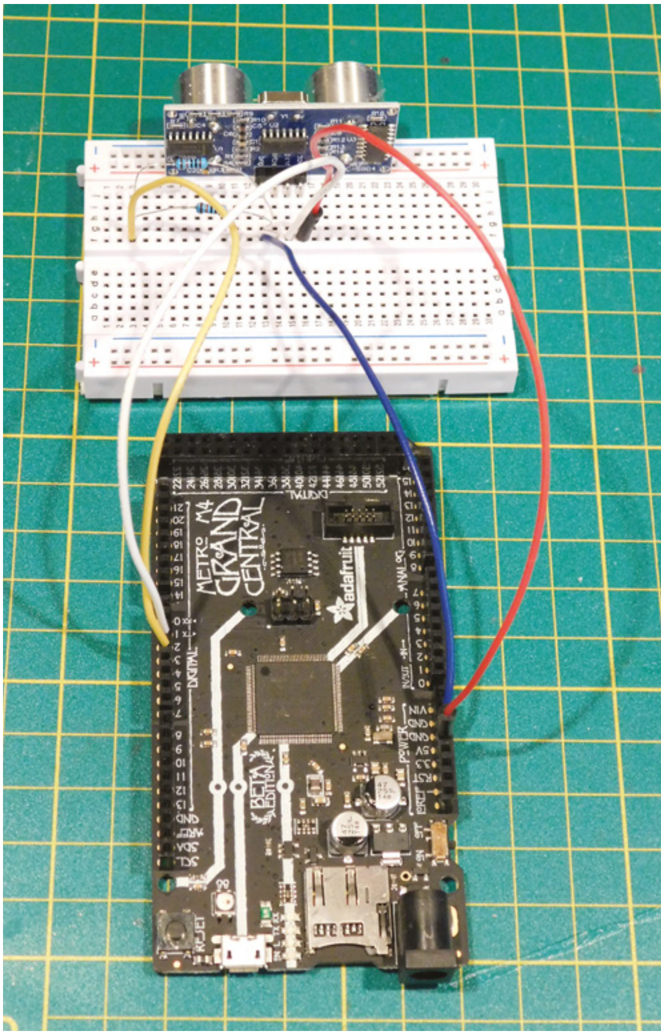
Sensing your surroundings: distance

Use ultrasound to see how close objects are

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.



Above: You can connect everything up with two resistors and four wires

Ultrasonic distance sensors work in the same way that sonar works. A high-pitched pulse of sound is sent out, it echoes off nearby objects, and the sensor listens for its return. The longer the time between pulse and echo, the further away the object is. The common HC-SR04 sensor is widely available, and often costs just a pound or two.

There are four pins: VCC, Trig, Echo, and GND. Trig triggers the sound pulse, and Echo is triggered when the sensor receives an echo. So far, so simple. The one slight catch is that this is a 5 V module, yet most microcontrollers run at 3.3 V. That means you either need to use a 5 V microcontroller (such as the Arduino Uno), or have a way of converting the signals between these logic levels.

Fortunately, this is quite straightforward. Each wire only sends data in one direction – the Trig pin takes a signal from the microcontroller to the ultrasonic sensor, and the Echo pin goes in the other direction.

> "If you go above the voltage **level of a microcontroller, you risk frying it**"

The 5 V input to the Trig pin doesn't have to be exactly 5 V. We've been unable to find any documentation on exactly what the limit should be, but experimental evidence suggests that a 3.3 V output from a microcontroller is enough to trigger the HS-SR04 module. It's quite common for 5 V devices to be able to take input from 3.3 V devices. For example, the Arduino Uno needs at least 60% of the microcontroller voltage to switch an input – that's 3 V – so as long as you're only getting input to the Uno, you can connect it to a 3.3 V device.

The same, however, isn't true the other way. There is generally much less tolerance of overvoltages than undervoltages. While you can safely experiment with providing lower voltages (the worst that will happen is that you might not trigger the sensor), if you go above the voltage level of a microcontroller, you risk frying it.

However, since we only need to get data from the sensor to the microcontroller, we can use a simple voltage divider circuit. This uses two resistors to split the voltage output from the sensor by a particular ratio. The easiest way of doing this is with two resistors (as shown in **Figure 1**). This will output half the input voltage – 2.5 V – which is safe for our microcontroller. The black wire goes to GND, and the yellow wire goes to the microcontroller input pin. The exact values for the resistors aren’t critical, except that they have to be the same. However, if they’re too low, you’ll waste power; if they’re too high, you might not get enough power to trigger the input. For this usage, 10 kΩ works well, but don’t worry about being too precise with this.

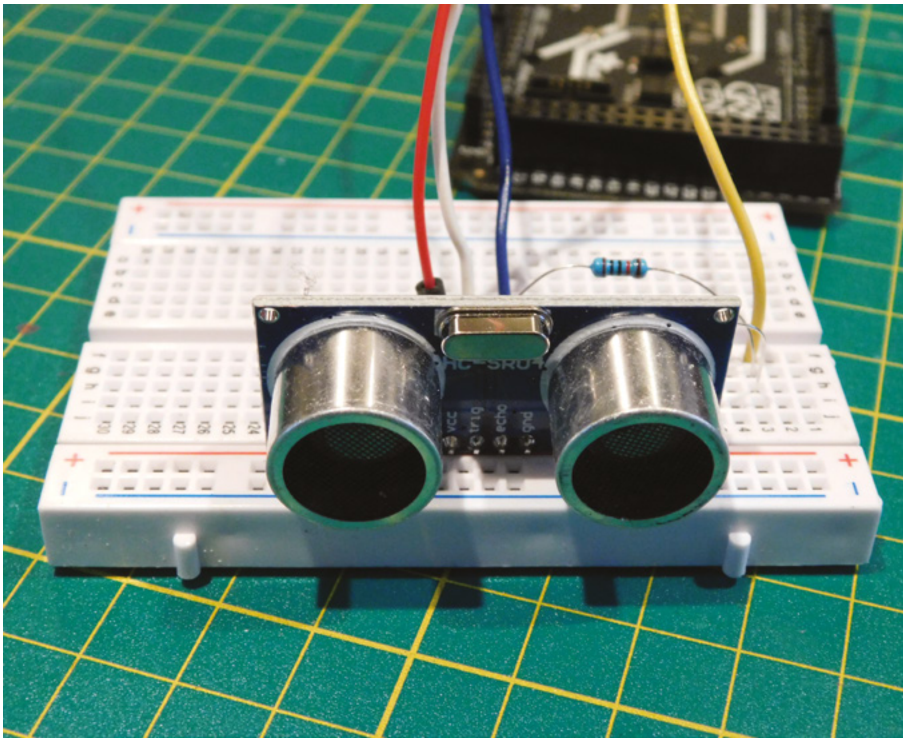
You can use this technique to connect many 5 V modules to 3.3 V microcontrollers, provided that each wire only sends data in one direction.

PUTTING THEORY INTO PRACTICE

Let’s now take a look at the code. It’s all fairly straightforward, but there are a couple of timing functions used that, while part of the standard Arduino language, are a bit unusual.

- **delayMicroseconds** The standard Arduino delay function takes an input in milliseconds, which is fine for many purposes, but when we’re timing the speed of sound, we need to be a bit more accurate.
- **pulseIn** This measures the time until an input pin changes value.

```
int trigPin 2
int echoPin 3

long pingTime;
int distance;

void setup() {
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
// Don’t think these are needed - yes they are,
for the first run?
 // digitalWrite(trigPin, LOW);
 // delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  pingTime = pulseIn(echoPin, HIGH);
  distance= (pingTime/2)*0.034;


  Serial.print(“The distance is: “);
  Serial.print(distance);
  Serial.println(“ cm”);

  delay(500);
}
```



That’s it all working in Arduino. We can do the same thing in CircuitPython using the following:

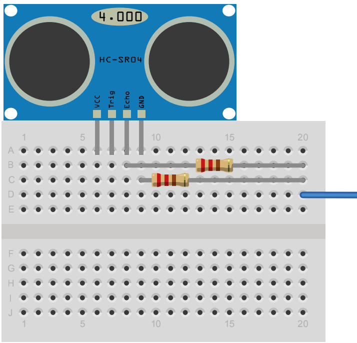
```
import board
import time
from adafruit_hcsr04 import HCSR04
sonar = HCSR04(board.D2, board.D3)
time.sleep(2)
try:
    while True:
        print(sonar.distance)
        time.sleep(0.5)
except KeyboardInterrupt:
    pass
sonar.deinit()
```

This code uses a module, as Python’s timing isn’t precise enough to measure the echo in code.

That’s all you need to get one of these cheap sensors up and running. You can use them to add a little intelligence to your robots, add a DIY reversing sensor to your car, detect if something passes close to your project, and much more. □

Above You can use these ultrasonic sensors as ‘eyes’ to anthropomorphise your robots

Figure 1 The resistor divider circuit cuts the voltage in half (or to some other proportion of the voltage depending on the values of the resistors)



FREE! RASPBERRY PI 4 STAND
The MagPi
The official Raspberry Pi magazine
Issue 90 | February 2020 | magpi.cc
FREE GIFT!
COOLING STAND
What's the weather like today?
Thursday, January 30, 2020
09:28 54
8.4°
Feels like 6°
BUILD THE ULTIMATE MAGIC MIRROR
CONSTRUCT AND CUSTOMISE
What's on my calendar?
OVERCLOCK RASPBERRY PI 4
Run at 2.1GHz with a 750MHz GPU
COOLING CASES TESTED!
Discover the best cooling solution
NEW YEAR, NEW PROJECTS!
Build something amazing in 2020
Issue 90 February 2020

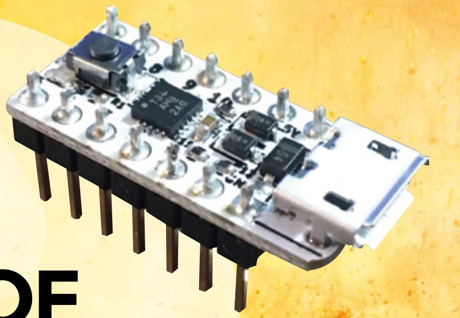
WORTH £20

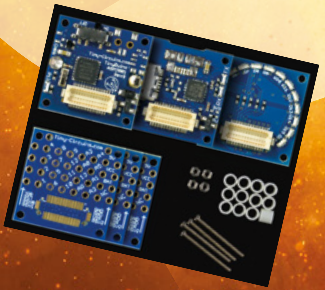
HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated





REVIEWS

Arc Igniter Kit

Let sparks fly with this soldering set

By Jo Hinchliffe @concreted0g

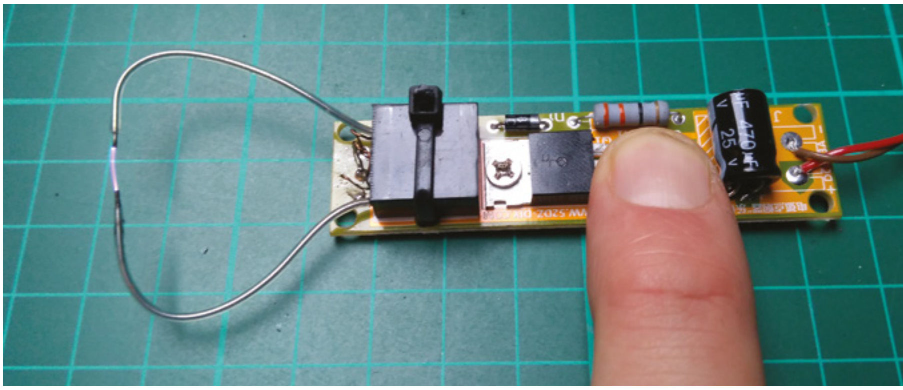
Above
A small bag of components with no instructions! Helpfully, the PCB is reasonably well labelled

There are so many small, cheap kits for all manner of random circuits available on direct-from-China websites, and this one caught our eye. Available for less than £3 from many online retailers, a kit that promises to make high voltage arcs appear between two wires definitely pushed our maker buttons.

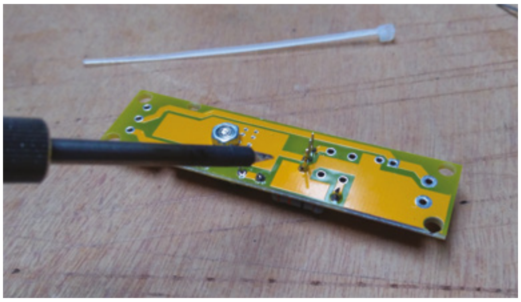
Simply packaged in a plastic bag, the PCB is well-made and, thankfully, reasonably labelled on the silkscreen, as there are absolutely no instructions supplied. All the components are through-hole, and things like the polarity of the capacitor and the orientation of the diode are clearly marked on the silkscreen. Looking around different vendors, this kit appears to be available in a couple of different form factors and different colours of PCB. Most pictures of the kit online do show similar components, and all show the kit assembled with a socket supplied for the power connections. However, ours arrived without the socket, so we soldered power supply wires directly to the board.

Building this little kit didn't take too long and, as it is all through-hole, it is pretty simple for a beginner to solder together, and it's pretty cheap if it goes a bit wrong! We soldered it together using some common 0.8 mm solder and a cheap USB soldering iron with no problems, apart from when we came to fit the small transformer. Inspecting the transformer, there are two stiffer wires with insulation coming out which will act as the electrodes between which the arc will be generated. There is also, frankly, a bit of a mess of copper wires hanging out of the transformer, with some single strand wires and some bundles of wires

Above The small, but extremely hot, arc!



Above After working out where everything goes and how the wires from the transformer work, we tentatively pressed the button, and success!



Right We found this PCB very easy to solder too, and the assembly for most parts was quite straightforward

twisted together with the tips soldered. We untangled these wires to reveal, on our version of this kit, a central pair of wires twisted together and soldered together, a single thicker wire, and a thinner single wire. If we viewed the transformer with the side up that allows us to see the coils inside, and with the wires pointing away from us, the thicker single wire is on the right, the combined wires are in the middle, and the thinner wire is on the left. This orientation is correct for when wiring these wires to the PCB. Another issue with these wires is that they are around 6 cm long, which is too long. After positioning the transformer in place, we cut the wires down to a more sensible length to solder them to the PCB. It's important at this point to realise that these copper wires are the ends of the coil wires, and that coil wire is coated with a thin shield of epoxy resin so that the coils don't short out against themselves. To be able to solder them, we needed to clean off a short length (3 to 4 mm) of the epoxy. It's a little fiddly, but we managed to sand the epoxy off the ends of the wires gently using an emery board.

> " If you continuously drove the arc for around ten seconds, it **would get quite hot** "

AC/DC

Commonly these kits are listed as requiring 3–5 V input. On our silkscreen, however it stated 5 V and, out of curiosity, we checked to see if it would run off 3 V, which it did not. Once hooked up to our bench power supply, supplying 5 V, a press of the button produced a fine arc between the electrodes. We found that the arc was pretty consistent and would jump between the electrodes, even if the electrodes were spaced up to about 1 cm. Much beyond that, and the arc wouldn't strike. The heat-sink performed well. However, if you continuously drove the arc for around ten seconds, it would get quite hot. Indeed, some vendors state in the description that they shouldn't be used for more than ten seconds continuously.

We aren't sure what the intended use of these kits originally was, but they are fun to assemble, and you could definitely use the arc to ignite something. They are certainly capable of burning wood and paper and lighting a match, and we found this a fun kit to build, even without any instructions.

The PCB was of good quality and easy to solder and, apart from working out the wiring of the transformer, it was a straightforward project, capable of creating small, bright sparks, which is always fun! Do stay safe though. The arcs this kit produces involve high voltages, which can be dangerous, so make sure you understand the risks before building a kit like this. □

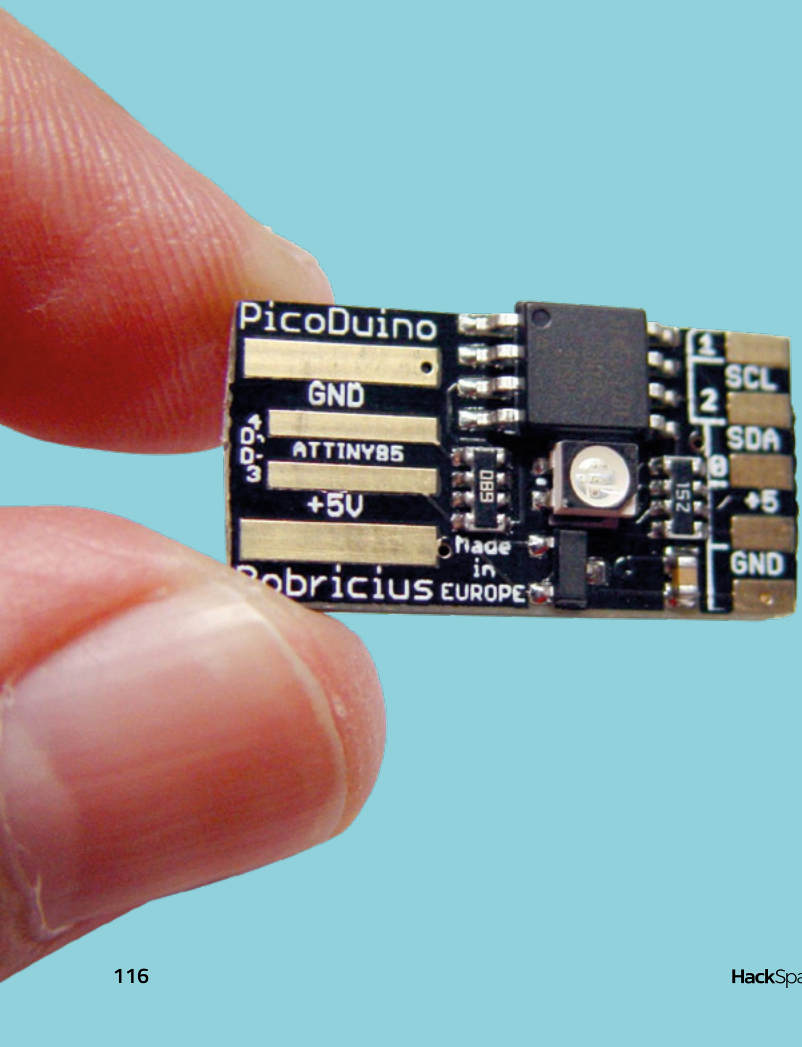
DIRECT FROM **SHENZHEN**

ONLY THE BEST

Teeny-tiny, itsy-bitsy, little microcontrollers

A collection of unimaginably small microcontroller development boards

By Marc de Vinck @devinck



Microcontrollers. We love them! Those little PCBs that can control our world. From something as simple as blinking an LED, to more complicated tasks like connecting to the internet or cellular network, microcontrollers can do it all. That's not to say there isn't a lot of love for single-board computers, like the Raspberry Pi boards. But in many situations, a microcontroller is all you need.

In this Best of Breed, we'll be looking at some microcontroller development boards – not simple breakout boards, but fully-fledged dev boards that truly earned their name. They are micro, tiny, and just amazingly small. These boards will make an Arduino Uno look huge. Even the incredibly popular Feather format, created by Adafruit Industries, is XXL compared to some of these boards.

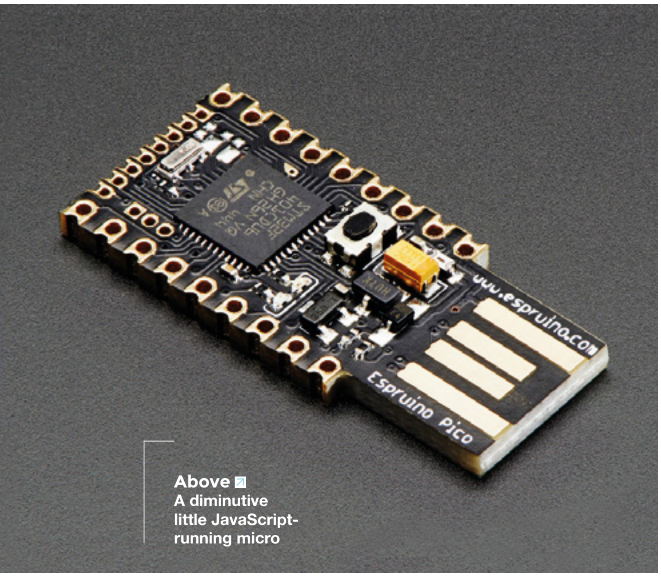
Which one is the smallest? It seems to change every day. Although several claim to be the smallest microcontroller development board, there is always a new one released, or about to be released, that's even smaller. There are dozens of micro-microcontroller boards out there, and we can't cover all of them, but we can pick a few that have been on our radar.

Espruino Pico vs Serpente

ESPRUINO PICO ◆ $24.95 | adafruit.com

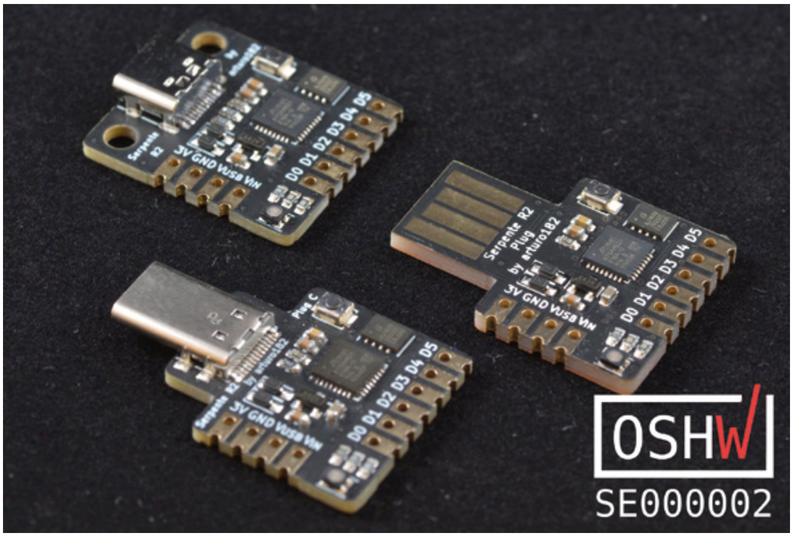
Web coding your thing? With the Espruino Pico, you can control electronics via JavaScript by plugging the PCB into a USB slot on your computer. The board features an STM32 microcontroller that has been preprogrammed with Espruino, so you'll be up and running in no time. You'll have access to 22 GPIO pins, two on-board LEDs, and one button. All controlled via an STM32F401CDU6 CPU.

There is also an on-board 3.3V 250mA voltage regulator and on-board FET which can be used to run high-current outputs. You can control the Espruino from almost any device, including a PC running Windows, a Mac, a Linux computer, your favourite Raspberry Pi, or Android phone. If your device can talk to a USB device, you can most likely use the Espruino Pico. If you love JavaScript, you'll love this board. □



Above A diminutive little JavaScript-running micro

SERPENTE ◆ $15 | tindie.com



Above Available in a variety of USB connector types

The Serpente development boards are low-cost microcontrollers that feature an ATSAMD21E18A 32-bit Cortex-M0+ running at 48MHz, and are designed to be used with Adafruit's CircuitPython. The boards come in a variety of USB form factors, depending on your project's needs. All three boards are virtually identical, except for the USB connector type. The boards are available with a female USB Type-C connector, a male USB Type-C connector, or a version where the board itself is used as a Type-A USB plug.

All three varieties feature the Cortex-M0, 256kB flash, 32kB RAM, 4MB flash for storing files and CircuitPython code. They also have six GPIO pins, a user-controlled RGB LED, and castellated edges so the boards can be used as modules and easily integrate into your project. Not many boards feature this amount of flexibility when it comes to the type of connector. If you need a specific type of USB support, this board may be for you. □

VERDICT

Espruino Pico

A good choice for Java developers.

9/10

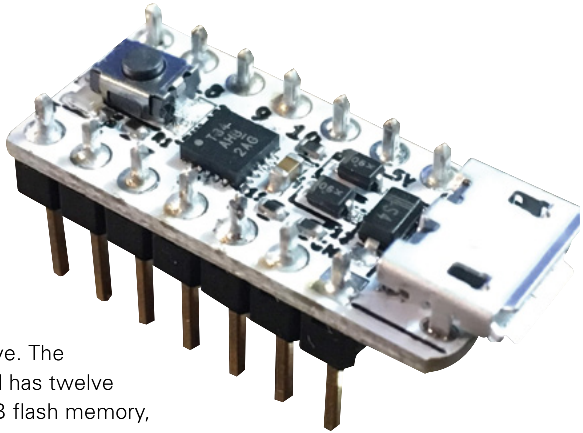
Serpente

So small, and it runs Python.

10/10

Tiniest USB development board with Micronucleus

TINIEST USB ◈ $ 7 | tindie.com



Left ◈ This board features an ATtiny84A

Once you get over the unusual name and claim, the Tiniest USB development board with Micronucleus is a really handy development board. The size is certainly appealing when working in tight spaces, but the specifications are also impressive. The board features an ATtiny84A, which is low-power and has twelve GPIO pins, if you disable the reset pin. It also has 8kB flash memory, 512kB RAM, and 512kB EEPROM. In addition, the board has a built-in button used as an interrupt for waking up the board when in low-power mode, and on-board status LED, and it comes preprogrammed with the Micronucleus bootloader, so you'll be up and running in no time. □

VERDICT

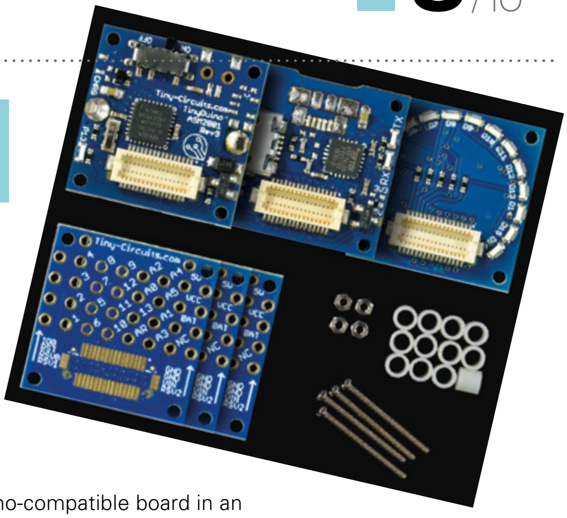
Tiniest USB

An interestingly named board, with an equally interesting micro.

8/10

TinyDuino Starter Kit

TINYDUINO ◈ $54.95 | tinycircuits.com



Right ◪ The basis for a variety of tiny projects

The TinyDuino Starter Kit is a bit more than just a single development board, which may make it an unfair contender in this Best of Breed. However, the success of the TinyDuino and available accessories and kits makes it worth mentioning. This author originally ordered the kit many years ago thinking it was a neat novelty. But little did he know that just a few short years later there would be a massive ecosystem around this form factor. At the last count, there were 138 products listed on the Tinycircuits website. From kits that feature a tiny TV with remote control, a tiny arcade machine, and a lot of breakout boards with all the sensors that you need for your next project.

TinyDuino is an Arduino-compatible board in an ultra-compact form factor. It's similar in power to an Arduino Uno board running an ATmega328P microcontroller, allowing most Arduino sketches to work as-is. This TinyDuino Starter Kit comes with the TinyDuino Processor Board, a TinyShield USB Board, the TinyShield 16 Edge Green LED board, three TinyShield Proto boards, and a hardware mounting kit. The boards all stack together, making for a really small form factor. This author's first project with the TinyDuino was building a watch. It was a fun, no soldering required experience. □

VERDICT

TinyDuino

A great ecosystem for teeny-tiny projects.

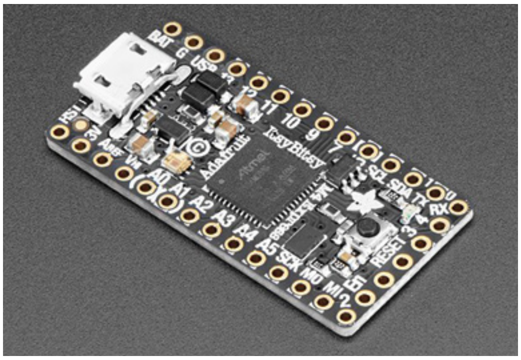
10/10

Adafruit ItsyBitsy M4

ITSYBITSY M4 ◈ $14.95 | adafruit.com

The Adafruit ItsyBitsy M4 Express, featuring the Microchip ATSAMD51, is one of the larger boards in this Best of Breed. It's also powerful, running at 120MHz. Compare that to several of the other boards that run between 8MHz and 16MHz. The ItsyBitsy also has floating point support and 512kB flash and 192kB RAM, again crushing a lot of similar dev boards.

You'll have access to six power pins, and 23 digital GPIO pins. Yes, 23! And even more impressive is that 18 are PWM. You can use the Arduino IDE to program the board, but you can also use the increasingly popular CircuitPython. This allows you to start programming as soon as you plug the board in, as it will show up as a USB drive. Just edit the **code.py** text file, and get going! Another great feature is when you unplug the board, the code is still there. Open it up on another computer and keep editing. It makes it so simple and intuitive. Check out the product page for a lot more information, and tutorials on how to use the ItsyBitsy M4 Express. □



Left ◈
A small-sized board that packs a lot of power

VERDICT

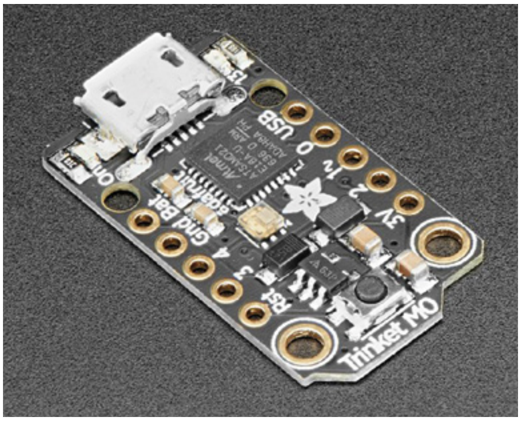
ItsyBitsy M4

Compact and full-featured.

9/10

Adafruit Trinket M0

TRINKET M0 ◈ $8.95 | adafruit.com



The Adafruit Trinket M0 is a small, but powerful, development board. It features an Atmel ATSAMD21 which has 256kB flash, 32kB RAM, all running at 48MHz on a 32-bit processor. That is exponentially faster than an ATtiny85, featured on many other small form factor boards. There are also several built-in LEDs, including an RGB DotStar for feedback and control. It has native USB support, five GPIO pins, three of which can be used as capacitive touch, and one has a true analogue output, allowing you to play 10-bit audio files.

You can use it with Arduino IDE and CircuitPython, making it really versatile. You connect to it with a micro USB, and it shows up as a drive when using CircuitPython, making programming on the go really easy. No installs, no IDE or compiler is needed. You can use it with almost any computer, even a Chromebook. Just open the text file and get programming. □

Left ◈
Run Python in a small form factor

VERDICT

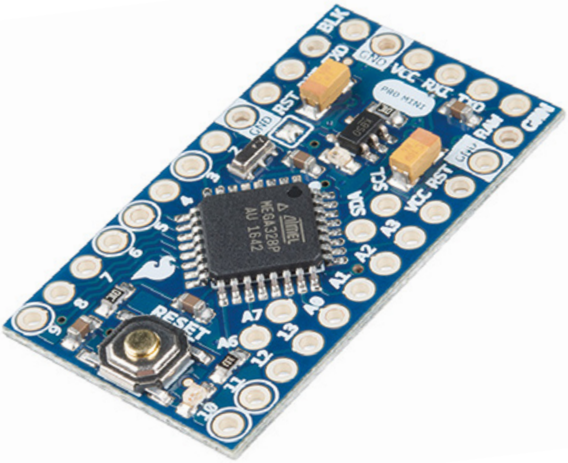
Trinket M0

About as versatile as a tiny dev board can get.

10/10

Arduino Pro Mini 328

ARDUINO PRO ◆ $ 9.95 | sparkfun.com



Left ◆ The only board in this round-up that bears the official Arduino moniker

It's tiny and thin – it's the Arduino Pro Mini! The only development board in this article that can sport the official Arduino name and logo. This 5V version of the ever-popular Arduino ecosystem is running at 16MHz on an ATmega328 IC. The board features twelve digital I/O pins and a few additional features. It's one of the few boards that still need an FTDI breakout board, or cable, to program it as it does not have native USB available. If you know the limitations, and have a programmer, you can save on space and on price. □

VERDICT

Arduino Pro

A very small Arduino board.

8/10

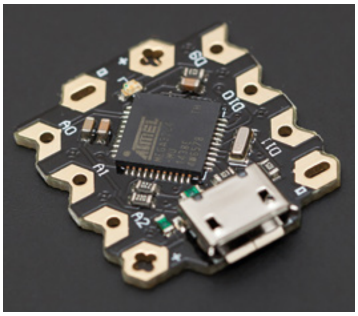
Beetle

BEETLE ◆ $7.90 | dfrobot.com

Below ◆ Great for students just starting out

Right ◆ Can a dev board get any smaller? No wait, we're afraid to ask!

The Beetle is one of the smallest and most inexpensive Arduino-compatible development boards. The board features an ATmega32U4 running at 16MHz. You'll have access to ten GPIO pins, three analogue and seven digital, making it a great alternative to other boards that may have fewer pins and features. And programming the Beetle is simple thanks to the on-board micro USB interface. Just plug it in and get connected.



The layout of the board makes it really versatile, from using it as is, adding header pins, or even as a wearable. Check out the product page for a lot of projects and tutorials on how you can integrate this little board into your next project. □

VERDICT

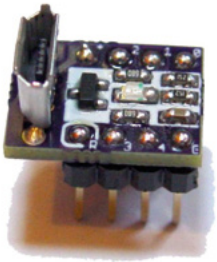
Beetle

Limited available pins, but very affordable.

9/10

BeanDuino

BEANDUINO ◆ $ 19.99 | tindie.com



You might have to give up on a few features since the BeanDuino runs on an ATtiny85, but you won't have to give up much space! Yeah, that's a micro USB port stacked up to save space. This board is small, at only 11 mm × 20 mm, yet it still has access to five to six GPIO pins, depending on how you configure it. Just keep in mind that two are used for USB communications when programming.

There is even an on-board LED. When space is a consideration, or you just want to have fun and show off the smallest of small microcontrollers to your friends, then the BeanDuino is a perfect choice. □

VERDICT

BeanDuino

Ridiculously small, yet it runs.

8/10

GET STARTED WITH ARDUINO
Robots, musical instruments, smart displays and more
Create AMAZING projects with this programmable controller
FROM THE MAKERS OF HackSpace MAGAZINE

Can I Hack It?

An RGB light kit?

How hackable is an RGB light 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**

The holiday season is but a memory, yet the nights are still long and dark. So adding a little light to our home is important, and using custom RGB lights via a Bluetooth app is the best way to do it. So how hackable is a set of RGB LEDs embedded into a ten-metre string of copper wires? Well, for that we need to take it apart.

GENERAL CONSTRUCTION

The largest part of the lights is the ten-metre lengths of copper wire that are used to connect each surface-mount LED in a large chain which can be cut to length. The copper wires go to a simple plastic enclosure via a sheath which is used to mechanically hold the wires, and to provide an external power connection. The plastic used for the enclosure is thin and easily worked with hand tools. The mechanical sheath for the wires is also very strong – much stronger than the wires, so take care. The enclosure also has three plastic buttons which are made in a way that they will press a momentary switch and then spring back into place.

ELECTRONICS

The brains of this product is a TLSR8267 Bluetooth (BLE) system on a chip (SoC) from Telink Semiconductor. The SoC comprises a 32-bit CPU running at up to 48MHz, 512kB of flash memory, and 16kB of RAM. There are also up to 37 GPIO connections, but these are not broken out to the PCB. The SoC also comes with a Bluetooth Low Energy (BLE) transceiver, which is also Bluetooth 4.2 and enables connection from a smartphone via a dedicated app. Power is supplied via a USB connection offering 5V 500mA, but the unit rarely uses even half of the available power. The board has multiple solder points, some with pre-soldered connections for power input (6V5 which provides 4.8V) and three connections for red, green, and blue outputs. The voltage provided to these pins varies as the 'mix' of colours is chosen.

YOU'LL NEED

Govee 10 m RGB Copper Wire Lights With App Model: H7312

COST

£8.99

WHERE

hsmag.cc/jodc8d

Right
Cost-effective and simple lights for your project, all wrapped up in a nice package and controlled via an app

The unsoldered pins are GND, TX, RX, and 3V3, which offer a serial connection for a dedicated development kit. Soldering pins to these connections is trivial for experienced users, but it won't do much good as no matter what we used to monitor the serial connection, no data was transferred to or from the board. Another set of unsoldered pins are for a single-wire communication system developed by Telink. 'SWS' refers to a single-wire slave device which requires a connection to a single-wire master device in order to see any data.

The colour and patterns created by the lights can be controlled directly from the plastic control unit. This offers limited control, but enough for basic use. And we can easily solder our own breakout buttons to the board, useful for cosplay use or a more robust interface. For more advanced users, the Govee Home app – available for iOS and Android – is required and provides a much more advanced way to control every aspect of the lights, from simple colours to elaborate animations. These LEDs may be 'smart', but they are not individually addressable, so the colour or animation you choose for the lights will be seen in every single LED, unlike NeoPixels and other addressable LEDs which enable individual control. What's great about this kit is that no matter how we use the lights, the control board will remember our preference ready for the next time they are used. Handy for decorating your home.

HACKABILITY

For quick and simple LED lighting, this unit is perfect, more for the price than function. It is hackable and can be easily integrated into a project with very little effort. Relying on USB power means we can power the lights from any USB power bank, and even with a 1200 mAh dollar/pound/euro store battery, we can get a few hours of light for our cosplay. The free app is a nice bonus and works reasonably well. In the latest update, there is a sound-to-light option for sound-reactive LEDs. But this relies on your phone's microphone to detect sound and control the lights accordingly. So if you are concerned about your privacy, best not use this option.

When it comes to using the lights with an Arduino or Raspberry Pi etc., this is possible as the LEDs are simple RGB LEDs. Just ensure that you measure your voltages and take the appropriate steps to safeguard your boards.

CONCLUSION

Cost-effective and easy-to-use illumination for projects great and small. This would be particularly effective in cosplay, as the control board can be hidden from sight and accessed using custom buttons. Using the lights in the home for decoration is a great idea as they can be wrapped around objects and retain their shape and strength thanks to the copper wire. Great fun, and ideal for the kids.

Happy hacking! □

ONE BY ONE

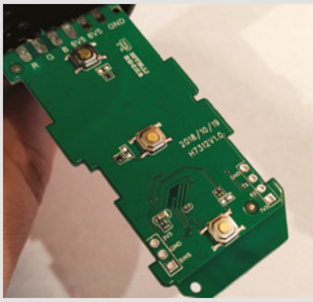
Addressable LEDs are a fascinating part of electronics. They come in single units, rings, squares, matrices, and self-adhesive tapes of varying lengths. In our collection, we have 'fairy light' versions which are used to decorate our Christmas tree and provide gentle illumination in our home. The most popular addressable LEDs are NeoPixels, a brand name from Adafruit for WS2812B LEDs. These LEDs feature three LEDs for red, green, and blue, and these colours are mixed to produce any desired colour. NeoPixels can be used with many different boards, and they are often used with Arduino for embedded installations. They are simple to learn and offer a great way to add interest to a project. The biggest drawbacks to WS2812B are that it requires exact timing in order to show colours, and that it can flicker, which is not ideal if using them in props for television or movies.

There are alternatives to WS2812B, such as APA102 and the SK9822 which both use the SPI interface (again, most boards have this) to achieve a similar effect; using SPI also means they do not rely on timing and so there is no flicker.

Above
The surface-mount LEDs are connected using four insulated copper wires, and each LED is covered in a plastic shell to keep water out

Left
The chain of RGB LEDs has decent colour reproduction, and each LED in the chain shows the same colour

Below
The control unit is very hackable; we can solder new buttons to control the unit, and we can solder header pins to flash the board, if we can source the correct software

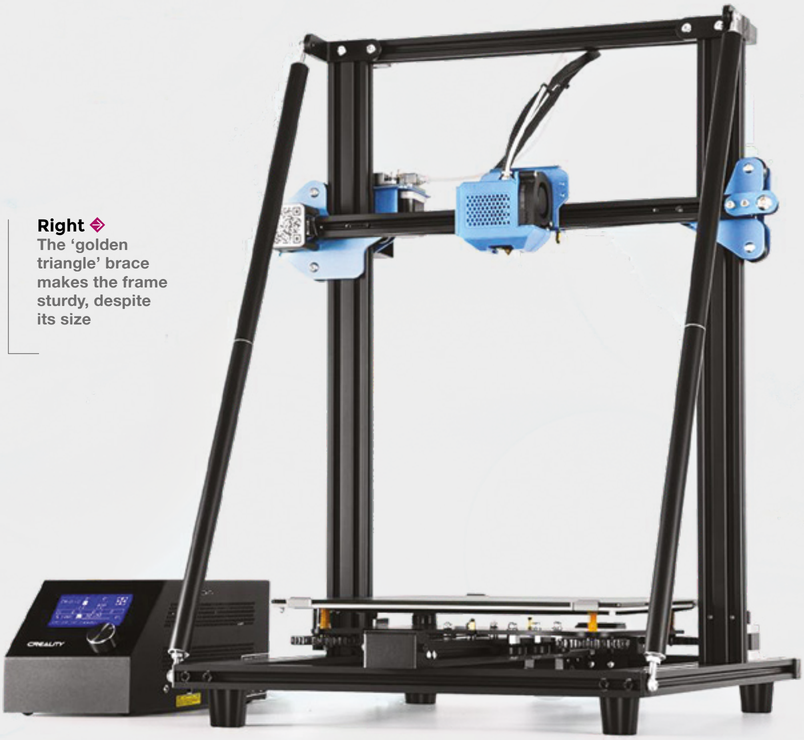


Creality CR-10 V2

A big printer with a small price tag

CREALITY around £380 | creality.com

By Ben Everard @ben_everard



Right The 'golden triangle' brace makes the frame sturdy, despite its size

The CR-10 has been through several iterations now, but this is the first time that Creality has seen fit to label it 'version 2', and it's the first time the firm has tinkered with the frame design.

The stand-out feature of the CR-10 V2 is – like earlier CR-10 machines – the print volume. The print bed is 30 by 30 cm, and you can print up to 40 cm high. This gives a total print volume of a whopping 36,000 cm^3. Large print volumes need stiff frames, as even slight movements are amplified by a gantry 40 cm above the print bed. The CR-10 V2 is made from aluminium extrusions which don't always produce the sturdiest frames, so in order to improve stiffness, this version has a cross-brace that supports the z-axis. Creality calls this the 'golden triangle', though it comes in matt black. Under this frame is a ceramic-coated glass print bed.

The CR-10 V2 arrived well-packaged in protective foam, and required only minimal assembly. The z-axis gantry had to be attached to the base, the cross-brace needed screwing in place, and a few wires needed connecting up. The instructions weren't quite as clear as they could have been, but despite this, we had a working printer in about an hour, with only minor frustration.

The control board and power supply are in an external unit, which seems to annoy some people, but we don't have any problem with this. The connecting wires are bundled into three chunks which are well protected enough not to feel vulnerable. The only downside to it, for us, is that we struggled to fit the whole lot in our slightly limited workshop space.

There's a filament outage sensor mounted on the frame, but this is so close to the filament drive that it's nigh on impossible to get the old filament out and replace it, should the filament run out. Perhaps there's a technique to this that we haven't discovered yet.

We followed the setup guide to the letter, but couldn't get a working 3D print on the machine until we wiped the print bed (ceramic-coated glass) down with isopropyl alcohol. This isn't a particularly unusual step in 3D printing, but it is a little unfortunate that this (or some other bed preparation routine) isn't detailed in the 'getting started' guide, as it would be dispiriting for a beginner to struggle to get a working print from their new machine.

Once we did get everything up and running, we found the printer capable of producing good-quality 3D prints, and running incredibly quietly while it does this. The machine came with an SD card preloaded with two models (a cat and a dog), as well as manuals and the slicing software.

Although you can use just about any 3D printer slicer with the CR-10 V2, the Creality-branded version of Cura that is included (you can also download this from the website) comes with settings that reduce stringing. This is particularly important as Bowden-driver printers such as this are prone to accumulating these little hairs of plastic, created when the nozzle moves without printing. Using a different slicer may leave your prints looking like they've been attacked by a swarm of hungry spiders. The Creality Slicer is a bit stripped back, and wouldn't be our first choice of slicers, but does do all the basic jobs of a slicer. If you'd rather use Cura, pay particular attention to the combing, wiping, and coasting settings too, if you're having problems with stringing.

> **Capable of producing good-quality 3D prints,** and running incredibly quietly while it does this

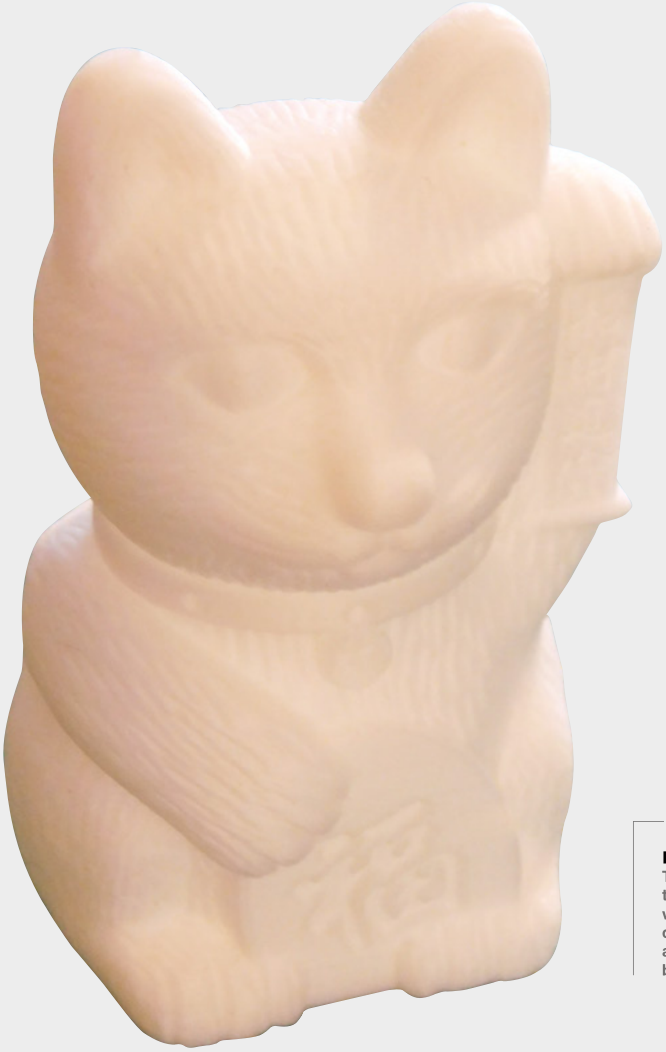
OPTIONAL UPGRADES

While this printer does work well out of the box, there are a few upgrades that you might want to consider.

This printer has a huge print volume, but actually utilising a significant proportion of it would take days (literally). The main reason for this is the 0.4 mm nozzle – this is pretty much the standard nozzle size on 3D printers and while that makes sense for smaller printers, for one this size a larger nozzle will be able to actually use the full print volume in a sensible amount of time. This is a quick and cheap upgrade – new nozzles are only a couple of pounds and not too much effort to swap.

Titan direct-drive print heads are an optional extra (by default, the CR-10 V2 comes with the same Bowden print system as other Creality printers) and if you're sick and tired of bed-levelling, you can add a BLTouch system for auto-levelling.

For a printer this large, the CR-10 V2 is impressively sturdy. During our tests, we found it reliable and capable of producing decent-quality prints. If you're looking for a large-format 3D printer right now, it's hard to argue against this. □



Left
The first print from this printer came out well, with the only defect being a tiny amount of stringing between the ears

VERDICT

A good-value large-format 3D printer, but you might want to consider some of the upgrades.

8/10

Wireframe
£3
ALL FORMATS
LIFTING THE LID ON VIDEO GAMES
CONQUERED
HOW INDIE GAMES DOMINATED THE DECADE
CONKER
The making of Rare's Bad Fur Day
LATENCY
What it is, and how to minimise it
OMNO
A solo developer's big action adventure
Issue 30 £3 | wfmag.cc
BURNOUT
The arcade racer's continuing legacy
BACK TO BASICS
The power of abstract art in video games
TALE SPIN
A beginner's guide to branching narratives
CYBER
207
DISCO ELYSIUM
Why it's the future of RPGs

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
RETRO GAMING WITH RASPBERRY PI
164 PAGES OF VIDEO GAME PROJECTS
+ BUILD AN ARCADE MACHINE

Micro-Cap 12

Keep the magic smoke at bay

SPECTRUM SOFTWARE ◈ free | spectrum-soft.com

By **Ben Everard** @ben_everard

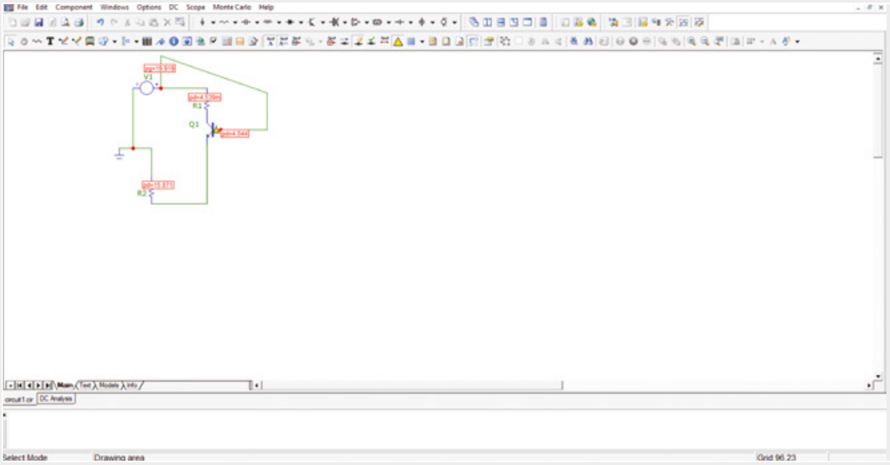
Circuits can be complex, building them can be fiddly, and working out what's going on can be tricky. To make the whole thing a little easier, some organisations have created circuit simulators. These are bits of software that, as the name suggests, attempt to simulate the behaviour of a particular circuit. They range from simple beginner's tools that help you learn how to power LEDs correctly, to hugely powerful tools for professional development that will help you perfect the design of complex circuitry. Micro-Cap is definitely in the latter section. Until very recently, it retailed for thousands of pounds, but in December 2019, Spectrum Software (the company that developed Micro-Cap) closed down. The details of why are still unclear, but as a final act, they made Micro-Cap free to download and use.

> "This is free as in zero cost, meaning that you can use **it without paying, not free as in open source**"

This is free as in zero cost, meaning that you can use it without paying, not free as in open source. There's currently no way for the community to continue to develop this software so, as far as we can tell, this will be the last ever version of Micro-Cap.

Micro-Cap 12 is a powerful bit of software and can simulate a wide range of conditions for both making sure a circuit will behave as intended, and looking at conditions that push particular components beyond their rated capacities. All this power means that it's a complex bit of software that takes some time to learn. There's a comprehensive user guide to help you, but this is over 1000 pages long.

We're torn about this software – it's powerful, free, and comes with a good library of tools. On the other hand, it will take some time to learn to use, and it won't get any bug fixes or new features. How you balance those two factors will decide whether or not this is the software for you. If you're willing to invest the time in learning this, then it should serve you well for a few years at least, before inevitable bit rot sets in, and it becomes harder and harder to keep it running. At which point, you'll have to go through the learning curve again on a new bit of software. For now, we're enjoying exploring Micro-Cap and finding it useful but, given its uncertain future, we're not going to invest too much time learning the full range of features. □



Above ◈
Circuit simulation can monitor the behaviour of your circuits before they're built

VERDICT

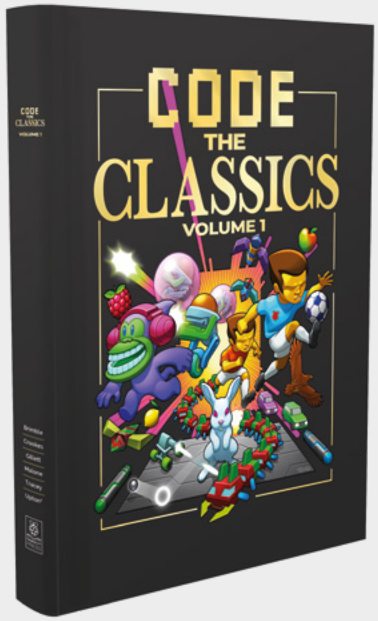
Powerful circuit simulator software, but how long will it keep running?

8/10

Code the Classics

David Crookes, Andrew Gillett, and Liz Upton ◈ £12.99 | Raspberry Pi Press

By **Ben Everard** @ben_everard



The history of computing is sometimes told as a history of serious analytical progress where scientists, businessmen, and engineers worked together to push electrons through silicon to aid societal advancement. This is total and utter tosh. The history of computing is the history of computer games. Just as today, the most powerful computer systems ever devised are being used to share pictures of cats, previously, the majority of computer time was spent running, jumping, firing, and punching animated creatures around virtual worlds.

Code the Classics tells this real history of computing, or at least a part of it. We start with a look at Pong, the first computer game to achieve any sort of widespread success, and progress through the years to Sensible Soccer, a classic of the early 1990s. In this 30-year timespan, computers went from being obscure machines in universities and the back rooms of larger businesses, to a common sight in homes, and it's thanks to games like the ones covered in *Code the Classics* that this happened.

With each game, the authors (which include luminaries of the computing world past and present) look at what made the game special and recreate it using a more modern approach. Each game includes a full listing, so you can get it up and running and relive (or experience for the first time) these classic games on modern computing hardware.

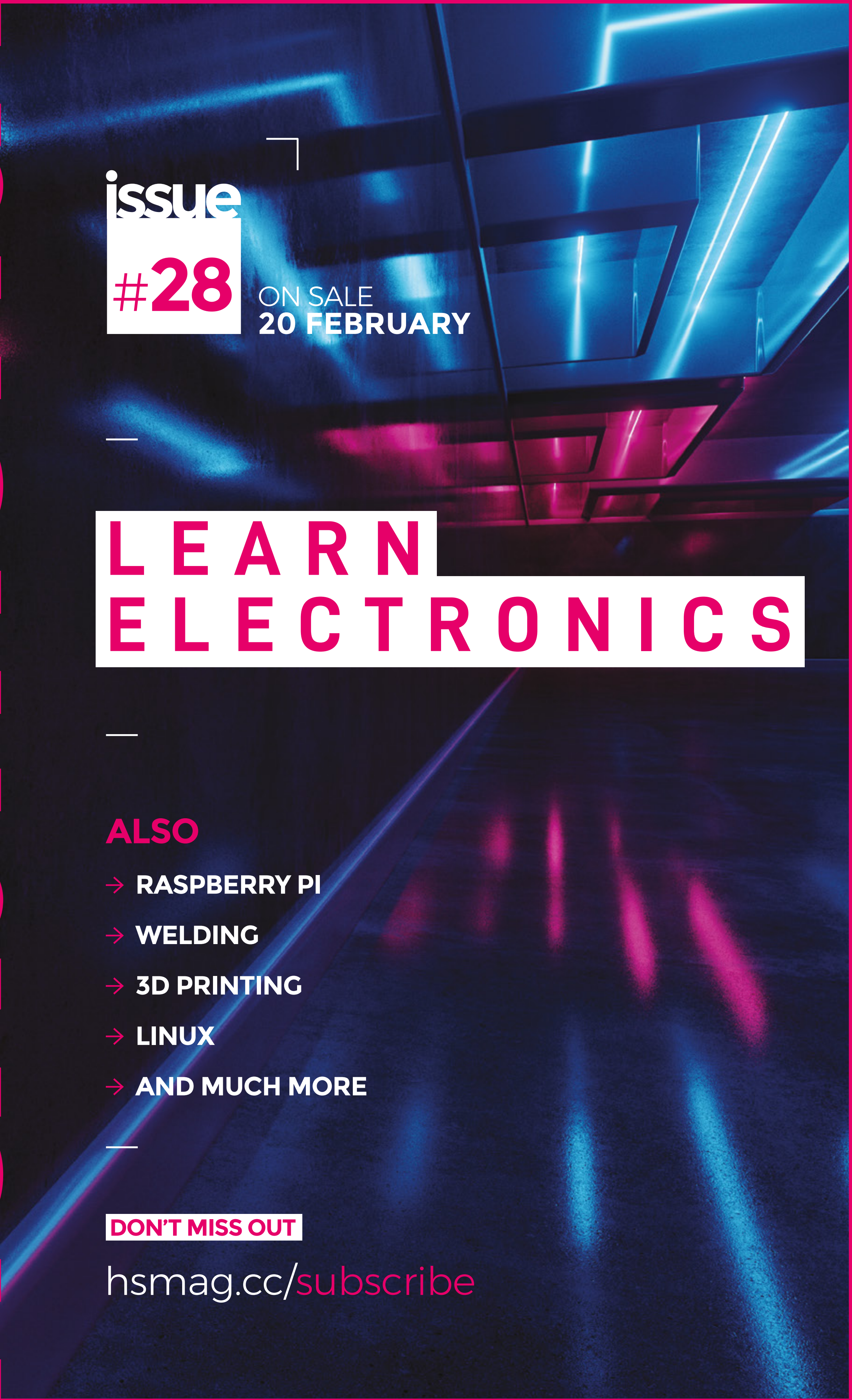
Code the Classics is essential reading for anyone interested in game design, computing history, or learning to program. □

VERDICT

Learn the history of computing by recreating it.

10/10

next month



issue #28

ON SALE 20 FEBRUARY

LEARN ELECTRONICS

ALSO

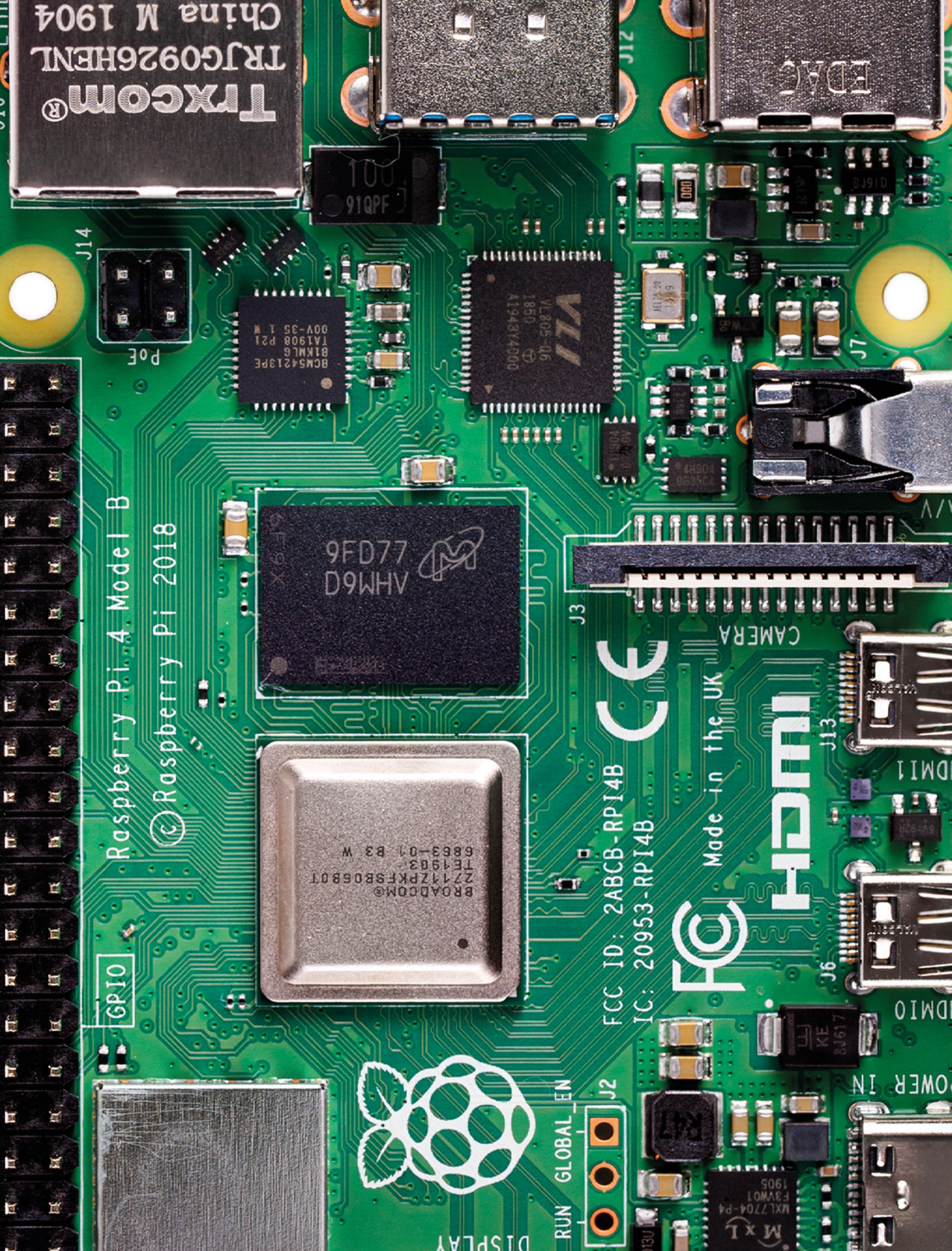
- → RASPBERRY PI
- → WELDING
- → 3D PRINTING
- → LINUX
- → AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

Making machines

Sewing machine pioneers Elias Howe and Isaac M. Singer are thought to be the first people to become millionaires from inventions. In doing so, they revolutionised the worlds of textiles, manufacturing, and even patent licensing.


Raspberry Pi 4 Model B
©Raspberry Pi 2018
GPIO
PoE
J14
J3
CAMERA
Made in the UK
FCC ID: 2ABCB-RPI4B
IC: 20953-RPI4B
HDMI1
HDMI0
POWER IN
GLOBAL_EN
RUN
J2
J13
J6
J7
A/V
9FD77
D9WHV
BROADCOM®
VL805-Q6
BCM54213PE
Trxcom®
TRJG0926HENL
China M 1904