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

132 PAGES OF MAKING

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | **September** 2019 | **Issue** #22

OPEN WORM

A 3D-printed wriggler designed for science

MAKE AN INSTRUMENT, MUSICAL CLOCK, OR ALARM

UGO VALLAURI

We all have the right to repair – if you can't fix it, you don't really own it

Robotic Music

HAPPY TUNES FOR HARDWARE **HACKERS**

\+

CIRCUITPYTHON
CONNECT TFT DISPLAYS

COLOUR SENSING
BUILD A LIGHT THAT CAN EMULATE COLOURS

TOOL STORAGE
LASER-CUT YOUR OWN WORKSHOP SHELF

LORAWAN

Connect sensors to the internet from almost anywhere

Sept.2019
Issue #22 £6

UMBRELLAS **AI** 3D PRINTING **CALLIPERS**

Welcome to HackSpace magazine

Electronic music is a great way of playing with electronics, but what if – like me – you prefer the sound of acoustic music made by plucking strings and hitting instruments? Does that mean that electronic music isn't for you? No! It just means you have to get a little creative with your electronics. This month, Matt Bradshaw is taking us on a journey of robotic music by building a glockenspiel that plays itself with the help of a couple of servos and an Arduino. A very similar approach can be used to robotize almost any instrument, though it's perhaps a little easier with those from the percussion section. You can use this to build your own orchestra, or just add a little musicality to your builds.

If sound isn't your thing, we also take a look at visual output this month as I take a look at hooking up TFT screens to microcontrollers using both in CircuitPython and Arduino (see page 76). Also, Jo Hinchliffe investigates LoRaWAN to send data to the internet from the remote Welsh countryside (page 82), and Dr Andrew Lewis helps you make sure all your cutting tools are kept sharp (page 108). Turn the page and let's get making.

BEN EVERARD
Editor ben.everard@raspberrypi.org

PAGE 52
SUBSCRIBE TODAY

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

get in touch at
hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.org
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store
GET IT ON Google Play

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, Ty Logan

Photography
Brian O'Halloran

CONTRIBUTORS

Lucy Rogers, Andrew Huang, Jo Hinchliffe, Matt Bradshaw, Cameron Norris, Tom Portegys, Mayank Sharma, Jo Hinchliffe, Andrew Lewis, Poppy Mosbacher, Bob Knetzger, Gareth Branwyn, Les Pounder, Glen Horan, Daniel Hollands, 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

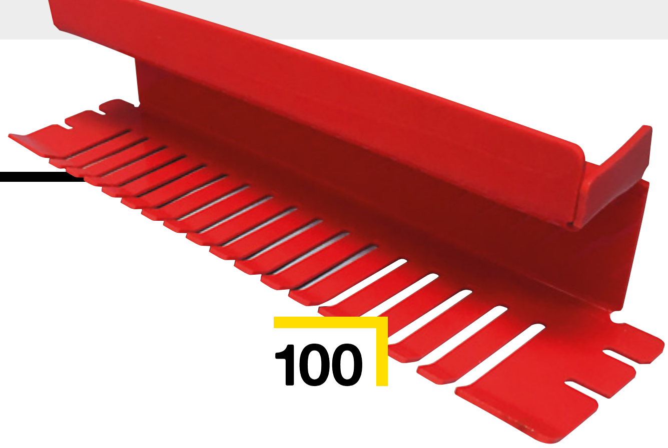
Unit 6, The Enterprise Centre, Kelvin Lane, Manor Royal, Crawley, West Sussex, RH10 9PE

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.



Contents

122

98

Direct from Shenzhen

Third hand

114 Third hands – for when two hands just aren't enough

Interview

Ugo Vallauri

62 Meet the co-founder of the Restart Project – fighting for your right to fix

06

70

75 FORGE

113 FIELD TEST

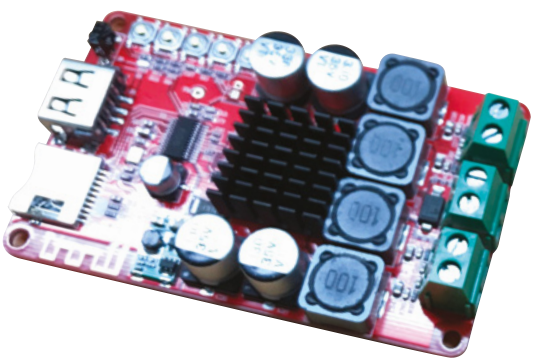
Wine barrel speaker

By **Philipp**

hsmag.cc/GyFYGW

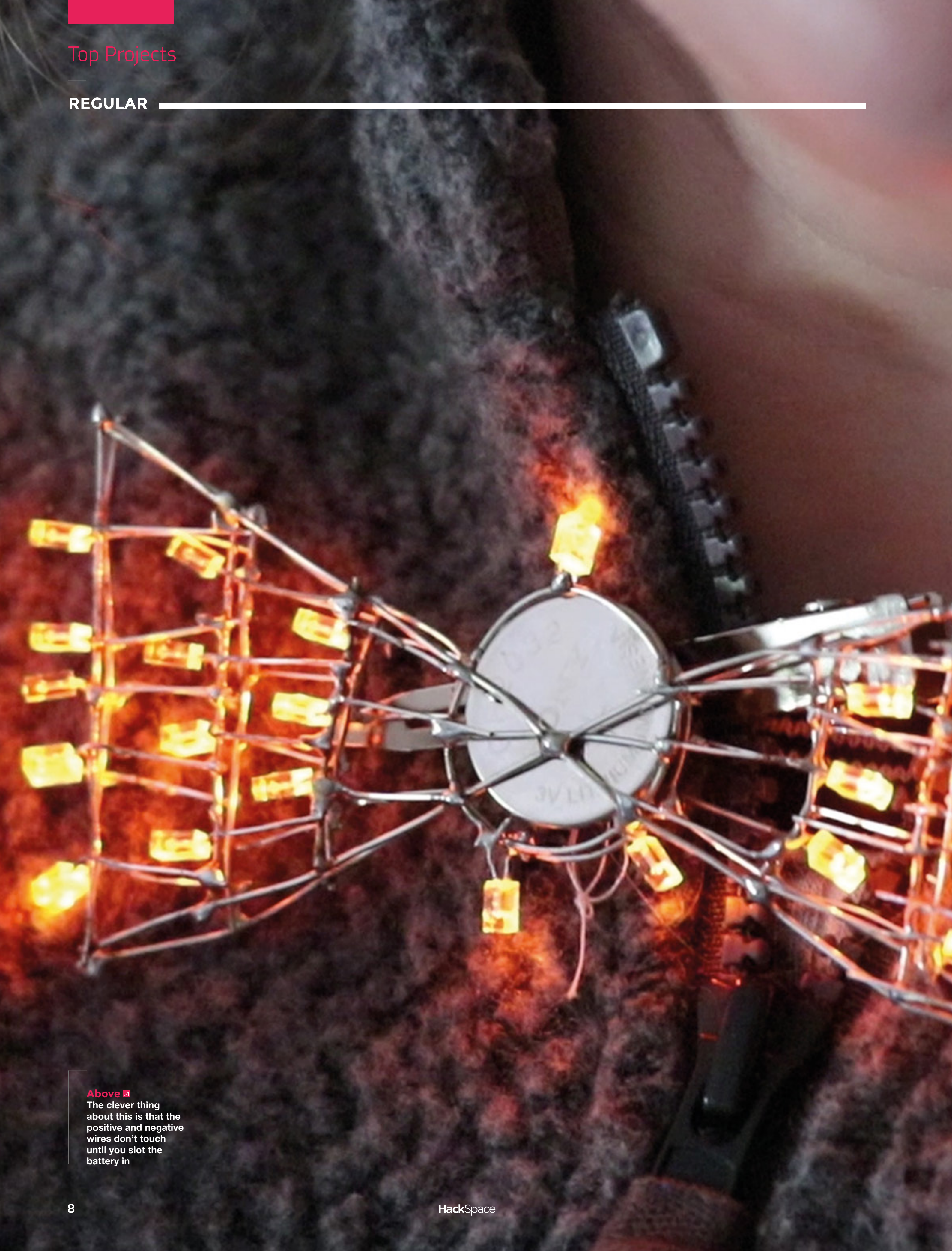
When he came across an empty wine barrel in his late grandfather's house, Philipp knew it was ripe for a project. He removed the rings, sanded the wood, and finished it with linseed oil before putting it back together again. Oh, and he also installed a Bluetooth speaker unit, cutting an extra hole in the base for the speaker.

We like the simplicity of this project. The electronics may be 'off the shelf', but it's a unique object that looks great and does a job.

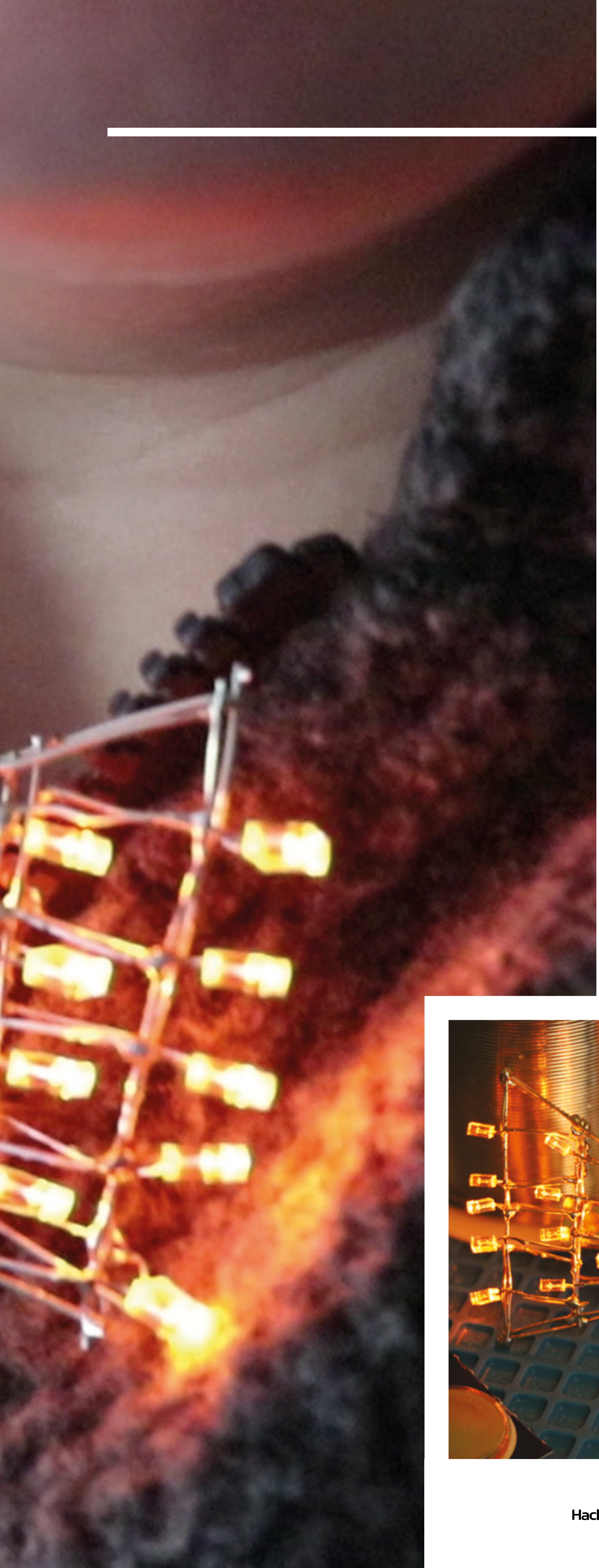


Right
Recycling is good, but reusing is better – turn to page 62 for more on this theme


elho



Above
The clever thing about this is that the positive and negative wires don't touch until you slot the battery in

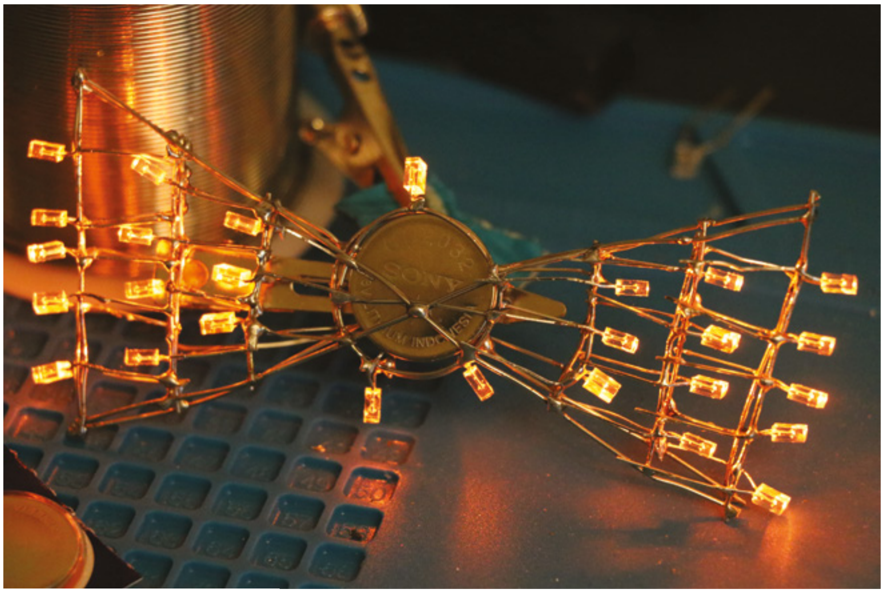


Free-form LED bow-tie

By Clarissa Kleveno

hsmag.cc/MsCABC

As soon as I saw Czech-based artist Jiří Praus' beautiful creations, I knew I wanted to make my own version. While I will never be able to top the Freeduino, I created this LED bow-tie as my own first attempt into exploring free-form circuits. While circuit boards are commonly used to mask complicated wirings, free-form circuits turn it into a feature, and highlight just how beautiful circuitry can be. With some careful planning and soldering, I turned a handful of 3M LEDs and some wire into a fabulous accessory. The subtle orange glow is perfect for proms, weddings, or your next formal event. □



REGULAR

DrumCube

By Franco Molina hsmag.cc/wCgqnt

The DrumCube is an Arduino-based robot drummer. It replicates the sounds a regular drum kit would make, with an arrangement of servos hitting both a small can and a pair of piezo transducers. It also uses a transistor white noise generator that imitates the cymbal sounds.

I'm a Chilean artist/musician and also a maker, self-taught on electronics, Arduino, and in most things I do, really. I developed this project as a way to complement my music gigs without any help from other musicians, big drum kits, or digital tracks. This is a musical machine that you can enjoy watching, as well as hearing.

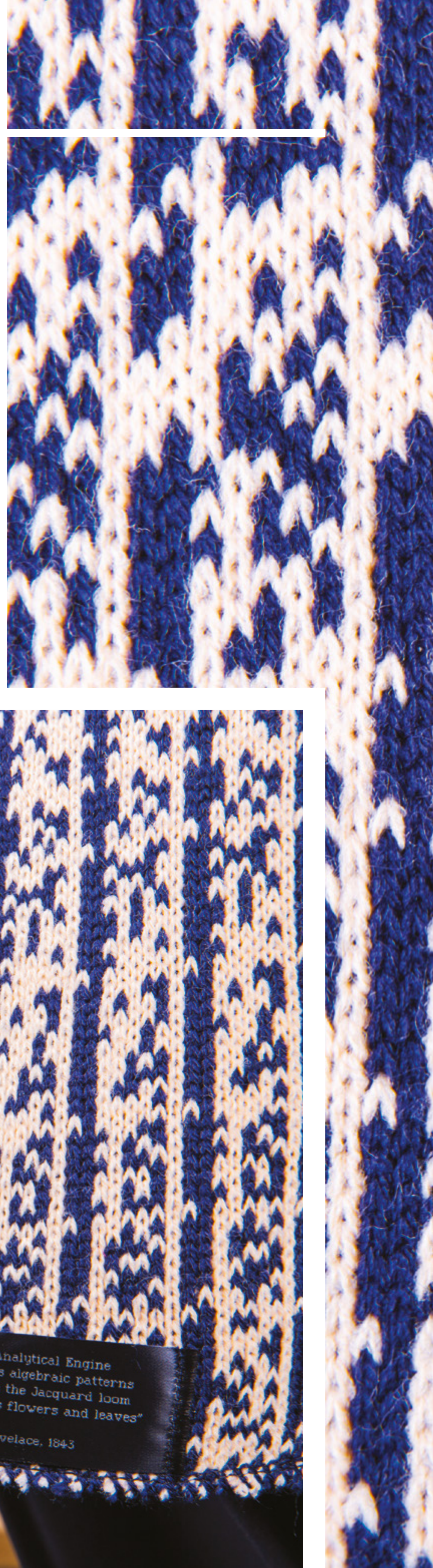
Above For more from Franco, follow him on Instagram, or listen to him on Spotify

REGULAR

Binary scarf

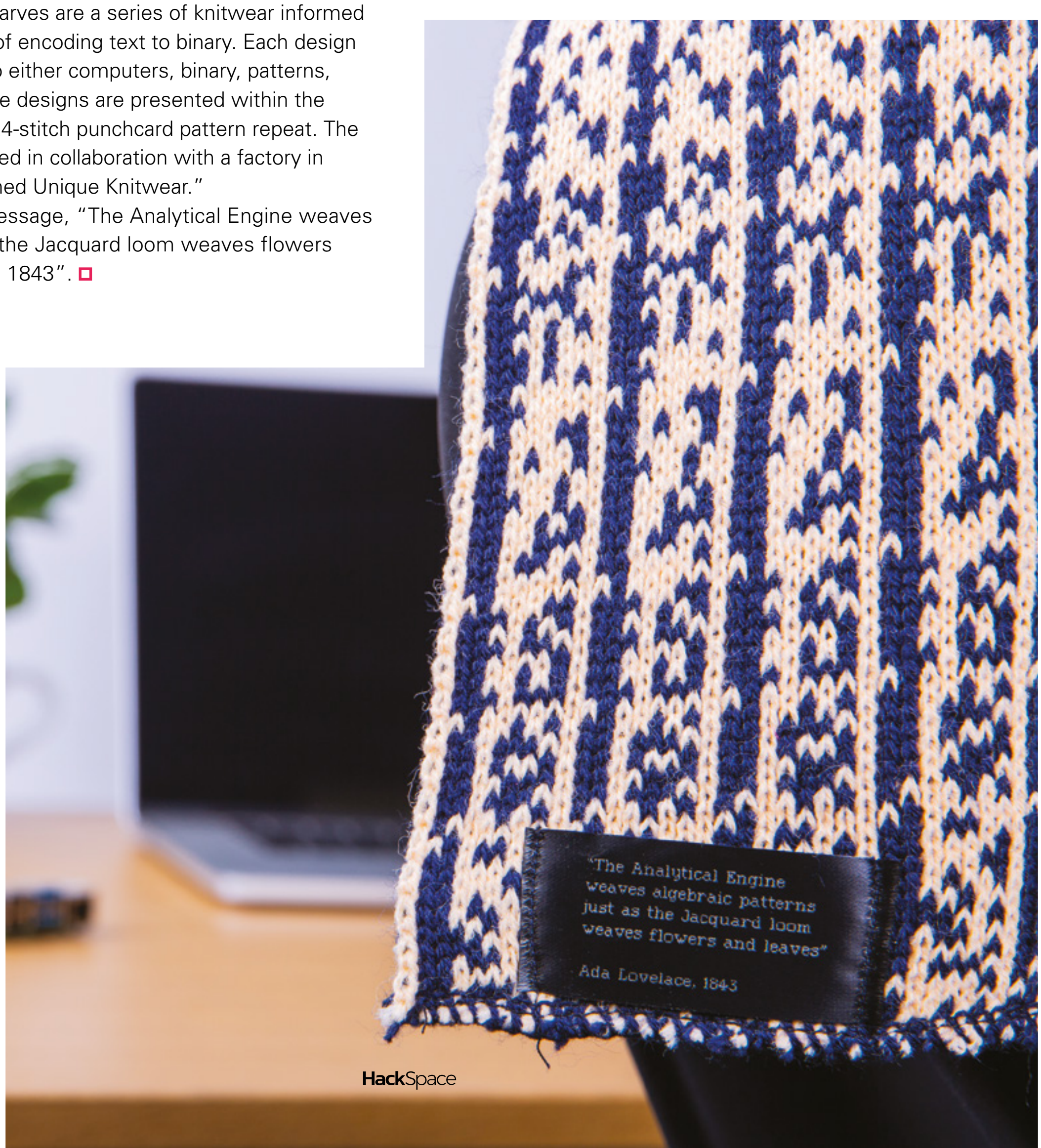
By Sam Meech

smeech.co.uk



On the hottest day ever recorded in Britain, we bought a scarf. Not just any scarf though: this one has a quotation from Ada Lovelace encoded into it. Creator Sam Meech is an artist based in England's creatively fecund North West, who among many other endeavours has made a small range of knitwear with techy messages knitted into them. As Sam puts it, "Binary scarves are a series of knitwear informed by the ASCII text system of encoding text to binary. Each design contains a quote related to either computers, binary, patterns, digital art, or repetition. The designs are presented within the constraints of a standard 24-stitch punchcard pattern repeat. The scarves have been produced in collaboration with a factory in Manchester, ironically named Unique Knitwear."

Our scarf carries the message, "The Analytical Engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves. Ada Lovelace, 1843".



Right
We love the symmetry of the Jacquard loom (which is close enough to a knitting machine) coming back to computing


"The Analytical Engine
weaves algebraic patterns
just as the Jacquard loom
weaves flowers and leaves"
Ada Lovelace, 1843

WW2 radio broadcast time machine

By Adam Clark

hsmag.cc/SvkMVj

My name is Adam Clark (aka Ajax Jones), and I have listened to quite a few of the archive broadcasts from WW2 online, but wanted to listen to them on something more fitting and physical. To this end, I bought a very old, non-working valve radio, and replaced the internals with a Raspberry Pi Zero on a custom 3D-printed chassis. The dial is used to select different years of the war, and the relevant radio broadcasts are then played out. To read the dial, I used an MCP3002 analogue-to-digital converter (ADC), and for the audio, I used a MAX98357 I²S Amplifier. Keeping the original speaker in use was a great stroke of luck, as the reward was an amazingly authentic sound.

Right Like this brilliant Raspberry Pi-based build? We'll feature it in more depth in a forthcoming issue of HackSpace magazine


CALM
1941
1942
1943
1940
1944
1939
1945
600
1200
1000
KCS
650
700
800
900

REGULAR

Objet 3d'art

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

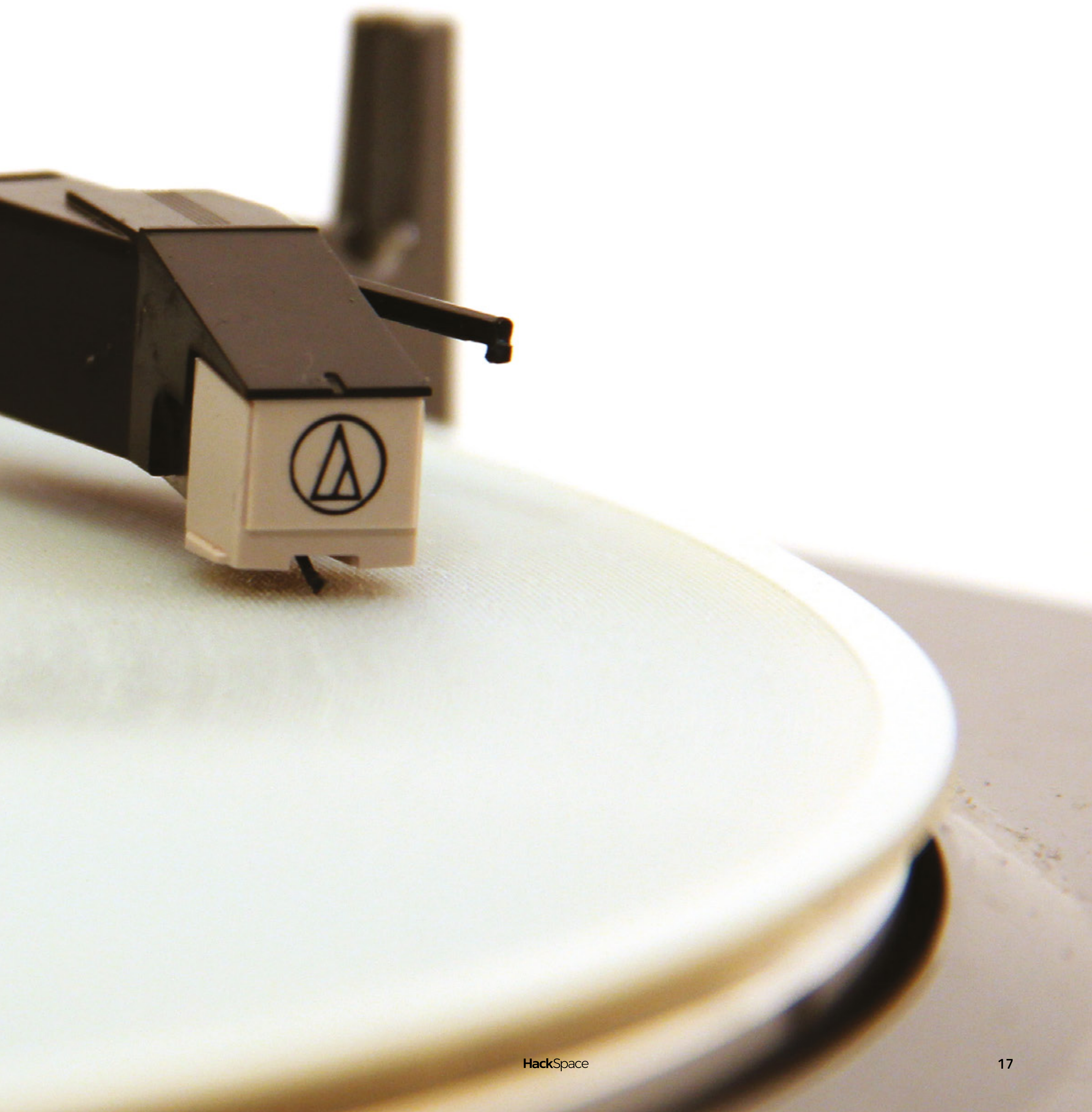
Back in the olden days, people used to listen to music on flat plastic discs covered in tiny grooves. A needle on the end of an arm would bounce around in these tiny grooves as the disc span, and the vibration was amplified electronically to produce music. It was the best of times, it was the worst of times.

Amanda Ghassaei used a UV-cured resin printer called the Objet Connex500, which has a resolution of 600 dpi in the x and y axis, and 16 microns in the z axis to create a playable record (a film of WD-40 is about 17 microns deep). Despite that, the grooves on the 3D-printed record are about ten times deeper and ten times wider than those on a pressed vinyl record, so the 3D-printed record can't hold anywhere near as much audio as a pressed record.

Amanda has published a detailed explanation of using the Python programming language to extract useful data from an audio file, turning the data into an STL file and, from there, into a printed resin object, along with the ups and downs of getting the process right.

She's also released recordings of 3D-printed sample tunes, including Nirvana's *Smells like Teen Spirit*, Daft Punk's *Around the World*, *Debaser* by Pixies, and songs by Joy Division, New Order, and Aphex Twin.

hsmag.cc/0eHpoX



REGULAR

Meet The Maker: DefProc Engineering

Making dreams come true

If you've got an idea for a world-beating product, but you don't quite know how to get it off the ground, you'll need someone to help you out. Someone who can design a PCB for you before it goes to the factory, write software, establish a proof of concept that you can take to investors, or build a prototype that you can use to iron out the creases before you make that first factory run.

That's exactly what DefProc Engineering, run by Jen and Patrick Fenner, does. This pair work with clients big and small to make one-off and small-run electronics-based products – and they started out in their local makerspace using the same equipment as the rest of their local maker community. We spoke to co-founder and managing director Jen Fenner to find out what it takes to go from sharing soldering irons to working on government contracts.

"We started our business in 2010. We didn't have full-time jobs; we had some money to live on, and we just started. We were only 27, so we didn't know what we were doing, but we started it in my mum and dad's spare room. We initially had this idea to design a human-powered vehicle – they were all the rage at the time, but there wasn't a really good, affordable version. So we had this big idea that we were going to produce a human-powered vehicle and design this kit, and we worked on that for about a year or so.

"We found that when it came to building our first prototype, none of the manufacturers would talk to us. They either didn't get back to us with a quote, or the one that did get back to us took eleven weeks to get the quote, and it was astronomically expensive. So, we decided to scale back. We found DoES Liverpool, which had started the year before. Pat went and learned how to use a laser cutter, how to design for laser cutting, how to design for 3D printing, all sorts of little electronic bits. Somebody offered him a piece of work – they asked him about a device they were trying to make, but didn't know how to get started making it. They had the idea, but they just were having difficulty executing it.

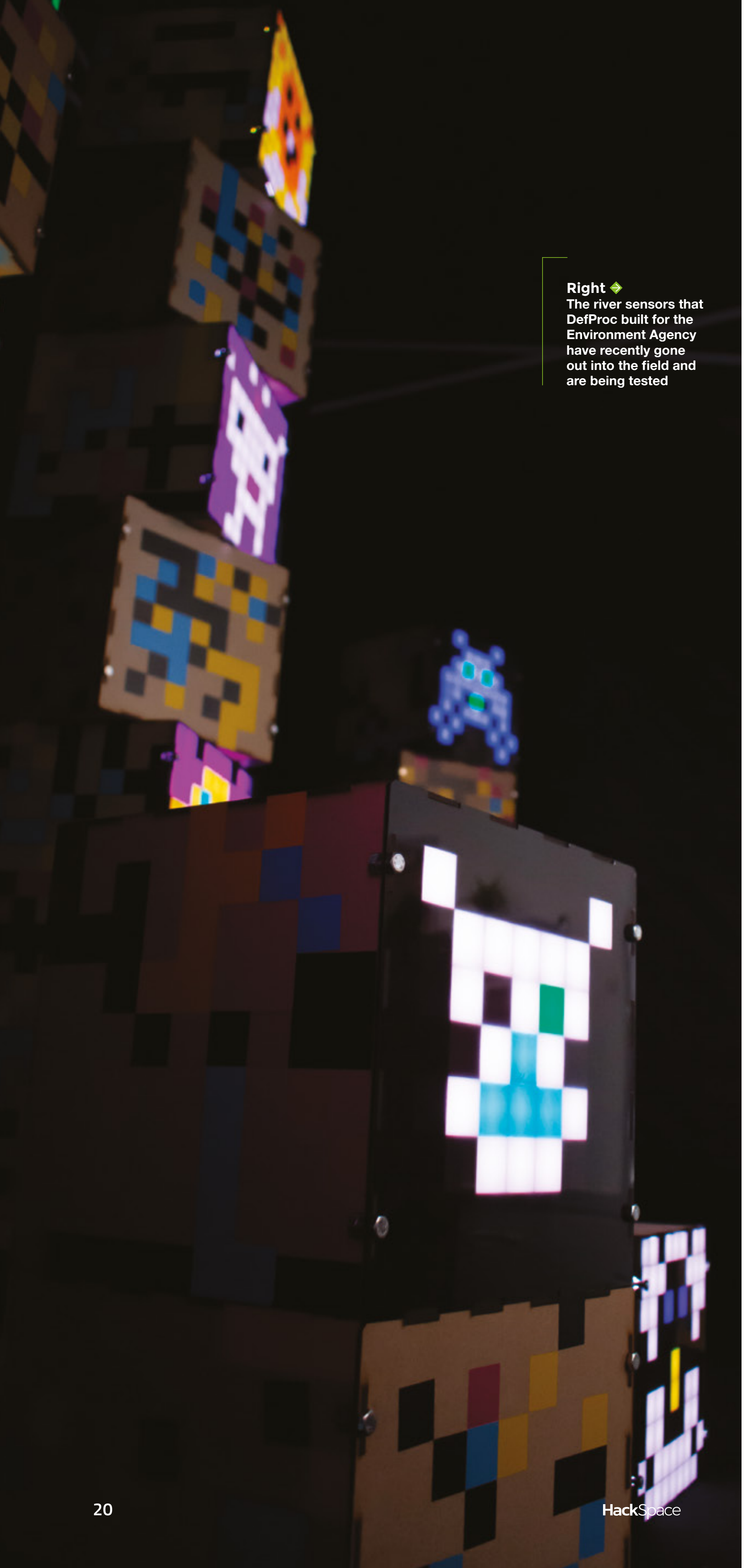
> **We basically set up a business to make things** for people within the makerspace

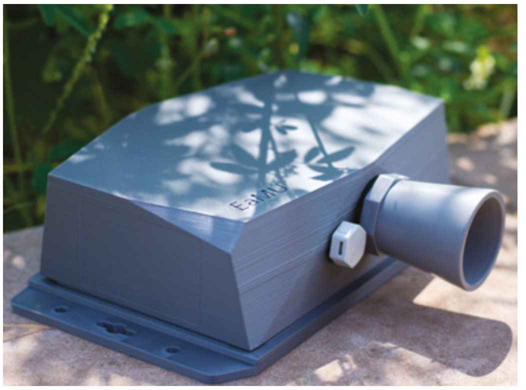
"He started offering advice to people, and eventually they just asked him: 'would you be able to make it for us?'. We basically set up a business to make things for people within the makerspace. People would come to us and ask us to make things. That's how it started.

"Our business has changed over time. We still do fundamentally what we did when we were there; we just do it for slightly higher-paying clients, and we've got our own equipment now. The shared equipment is good, but you want your own professional tools. We've got quite high-end soldering irons now, whereas the space we were in, it was good at the start, having access to all that equipment – I think we actually bought some equipment and put it in the space so everyone could use it. We really loved it there, and we wouldn't have the business we have →

Right Made Invaders is a real, physical version of the classic Space Invaders game. We would have had a go if it hadn't been surrounded by children. Those pesky kids!


NERF





Right
The river sensors that DefProc built for the Environment Agency have recently gone out into the field and are being tested

now if it wasn't for us being at DoES Liverpool. They gave us a break.

"One of the things I like is that we still refer people to the makerspace as well. If a potential client comes to us with an idea, we will sometimes say, 'You don't want to spend loads and loads of money with us, but if you go to the makerspace, there will be someone there who can support you while you start off.' It's a supportive community, and it doesn't cost very much to join.

CLIENT WORK

"Everything we do is tailored to the client. We don't particularly invent things. We have one project, Push to Talk, which is our in-house product that we're developing with Liverpool City Council as part of the Council's 5G project, and that's going out to people's homes.

"The Environment Agency project that we're working on is a live level sensor. Because of the way the Environment Agency monitors its rivers, they have a big base station where you can see at that particular point what the level is, but you don't necessarily know what the water level is like ten miles down the river, because there isn't any data being gathered until the next base station.

"They wanted us to develop some smaller sensors that they could tap into, through which they could monitor the water levels along the length of rivers. We've been using a new technology called NB-IoT (Narrowband IoT), which is a bit like LoRaWAN signal. We started developing that while we were at DoES, and we've done various different versions of it. We've also created some bespoke sensor hardware for them, so that they can train their staff on how to use these small devices so that they can get the data they want out of those sensors. We've done quite a lot of work with the Environment Agency, including a national training scheme for these devices.

"We're currently at a staff level of five. We do a lot of stuff for how small we are. This week we're just finishing off a build for the National Science and Media Museum in Bradford. We're putting together all the

Above "Sometimes people will go to other companies with ideas but aren't able to get exactly what they want; they'll come to us, and we're often able to fulfil what that person wants"

interactive side of four lunar landing exhibits. You touch something and then it lights up and plays video, or you put an RFID tag on something and it plays audio into a specific space. We do all the electronic side of that.

"Patrick's a design engineer; his degree was in automotive engineering and structures, and he's migrated into this field of electronics. My background is in textile design; I do a lot of the creative design. Made Invaders, for example, is my design, and then Patrick works on how to make it function. I don't know much about the electronics, but I do know what they should look like: the design rigour that needs to go into things. We have quite a nice partnership of skills.

"We've got a product designer now who does all the CAD for us and makes everything look good. It's nice to have people to support us. When it was just the two of us, Patrick was doing the electronics and the software, and we've now got a software engineer and a product engineer as well.

"Gradually, we just built up and built up. We're doing much bigger projects now, still working with the Environment Agency. We didn't go to the makerspace to design a product to then start a business out of it; we've formed a business within the makerspace, providing a complementary skill to some of the other people that were there. We've grown from there over the last five years. It's been serendipitous.

"A lot of it's been driven by Patrick and his interest in new problems. One of the things that stands out for us, as a business, is that we're willing to take a punt with a client. Even if we don't know how something works, we know that it must be possible. Whereas we know that a lot of businesses, unless it fits into their formula, they won't be willing to take it on. It comes back to the time when we wanted a prototype – there was no-one who was willing to do a proof of concept, or a prototype. 'No, we don't want to make one of something; if you want 10,000, come back.'

> **We're doing much bigger projects now,** still working with the Environment Agency

"I think our biggest run has been 200 things for a client. If you go to somewhere that's got the big pick-and-place machines, and ask them to make you 200 of something, they'll tell you it's not worth their time to set up the machines to do that. We can manufacture things in small runs that big places just don't want to do. It's almost like the bigger the company, the less likely they are to want to deal with you. We try to fill that void."

Nine squares

Making time for the important things in life

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**

Have you ever used the wrong tool for a job when you have the proper tool elsewhere, but the wrong one just happened to be handy? I often use a junior hacksaw to cut small things, when I have a workshop full of wood-saws, jig-saws, fret-saws, and a band-saw – any of which would make neater cuts.

Having tools is great, but sometimes the right tool is just the one nearest to hand.

There's a time management tool – it's an analogy about rocks, pebbles, sand, and water in a glass jar. Fill the jar with rocks (important projects, e.g. family, health, and income) and there is still room for pebbles (nice projects such as hobbies and friends), and then still room for 'filler projects' (watching TV or social media) – and even then, there's still room for liquid refreshments! But fill it with sand or liquid first and there's no space (time) left for more important things.

So now I am putting the 'rocks' into my calendar – and giving myself permission to do the fun ones too

This is a great 'tool', but one I had filed under 'useful' rather than 'use'.

So, I have taken some time to think about what my 'rocks' and 'pebbles' are – or should be. What areas of my life do I want to be important? In her book, *Feel the Fear and Do It Anyway*, Susan Jeffers mentions that the problem of someone saying, "my job is my life", or "my partner is my life", is that if they lose that job or partner, then their life is empty and meaningless. She suggests making a grid of nine squares and writing part of your life in each. Then, spend equal amounts of your thoughts/efforts on each square.

I have made one of these grids – and I've been using that to help me focus on what 'rocks' I want to put in my jar.

My top three squares are for family, home, and friends. The next three are income, health, and learning. And then I have the things I love to do – making, storytelling, and outdoor activity.

These are not rigid definitions. Some things will tick two or more boxes at once – for example, kayaking with friends, or selling one of my makes.

However, just like knowing there's a better tool in the workshop, knowing that I have large rocks to put into my jar is not the same as doing them. Sometimes it's just easier to procrastinate on social media.

After I went freelance, I spent years telling myself that I couldn't do the 'fun' things until I'd done the 'work' things. Which meant that even if I did do a fun thing, I spent the time guilt-wracked, which killed the enjoyment.

So now I am putting the 'rocks' into my calendar – and giving myself permission to do the fun ones too. And weirdly, I am still finding time for social media and watching a bit of TV!

The quantisation of Moore's Law

It's 2019 – where are our faster chips?

Bunnie Huang

@bunniestudios

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

One of the most remarkable aspects of Moore's Law is how smooth the rate of improvement had been up until a few years ago. For five decades, major industries could count on regular improvements to performance and cost.

In hindsight, this smoothness was a result of dozens of foundries jockeying for pole position. If one foundry stumbled, another would take its place. However, as transistors became much smaller than the size of the photons used to draw them, shrinking dimensions was no longer just a matter of better optics or clever computational optics tricks. At 28 nm, something fundamental had to change: the industry moved to double-patterning and FinFET transistors. This was a difficult transition that fundamentally changed the structure of transistors. Up until 28 nm, things had gotten better and cheaper, so just about every conceivable circuit rode the Moore's Law train for free. However, from 28 nm onwards, transistors got denser and better, but stopped getting cheaper. Thus, only applications that could justify the higher cost continued to ride the Moore's Law train, marking the beginning of the end of Moore's Law.

In recent years, the number of players has continued to dwindle to just three: TSMC, Samsung, and Intel. Progress has been halting; instead of a continuum of performance improvements, the industry is forced to contend with quantised leaps in performance. Intel's absence in the 10 nm and 7 nm nodes has been notable, leaving PC manufacturers struggling to give consumers reasons to upgrade their existing computers. Recently, TSMC has filled the leadership vacuum in the 7 nm node, and everyday consumers have felt this in the form of AMD's Ryzen renaissance.

EUV is hard in so many ways; it's reported to gobble a megawatt of power to produce just 200 watts of EUV light

What's beyond 7 nm? The industry is finally about to embark on a huge paradigm shift from using 193 nm deep ultraviolet light, to 13.5 nm extreme ultraviolet (EUV) light. EUV is hard in so many ways; it's reported to gobble a megawatt of power to produce just 200 watts of EUV light, and it requires landing a bullseye – not once, but twice – with a powerful CO^2 laser on droplets of ultra-pure tin that are 25 microns in diameter, falling at a rate of 50,000 times per second. The big news is that TSMC has recently gone into production with EUV, marking the beginning of a new quantum for Moore's Law. This should hopefully unlock a trickle of incremental improvements over the next couple of years – and then we'll have to wait again, until the next quantum of Moore's Law. □

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**

TETRIS

I'd seen LED animations such as the Tetris matrix in [issue 21's animated clock] but never thought I'd be able to make one myself. It turns out you don't have to – just download the right bit of code, and it's done for you. Truly, this is the lazy man's way to program things.

Frank
Edinburgh

Ben says: You say tomato, I say tomato; you say lazy, I say efficient, intelligent, and leaving plenty of time to get other things done.

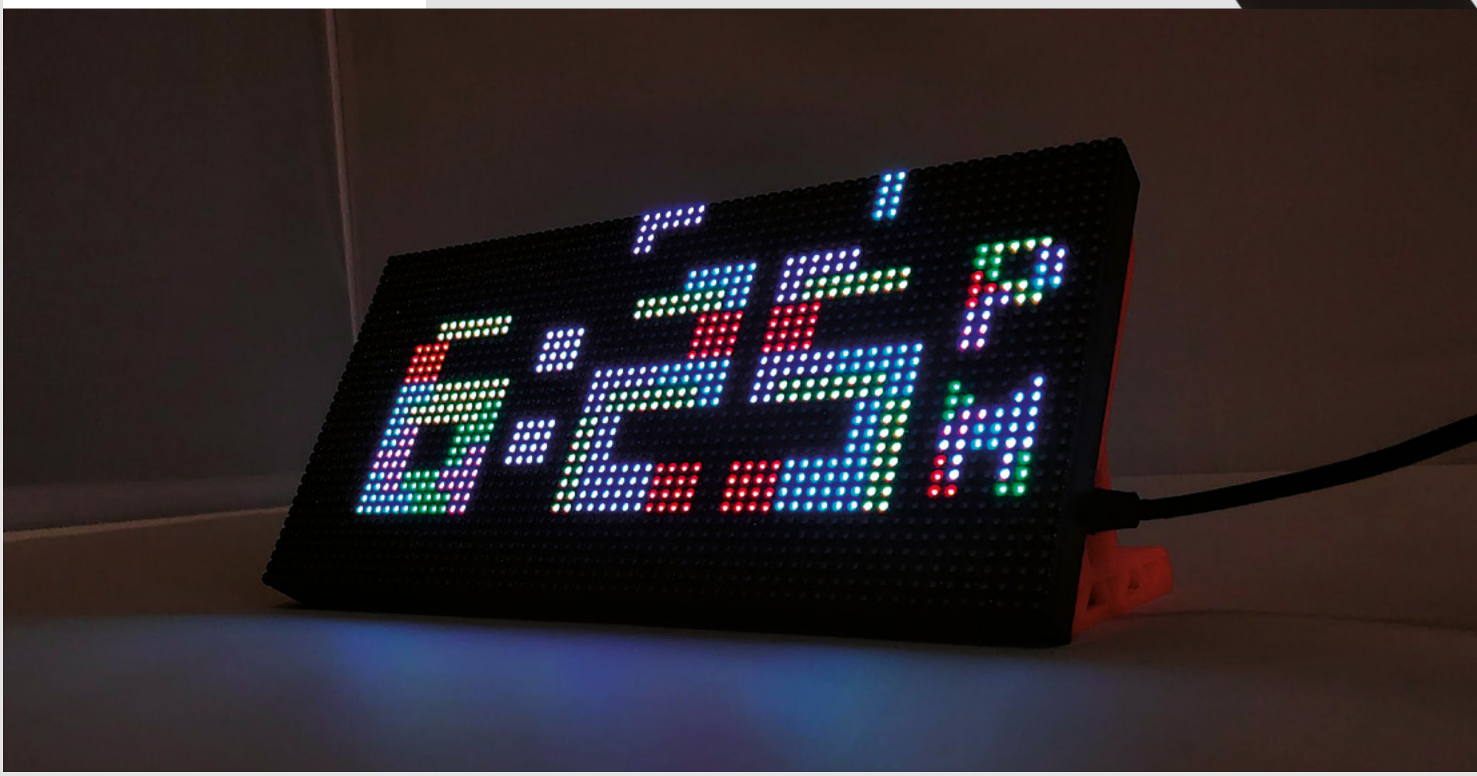
MICROWAVE GLASS

I have to take issue with you. Popty-ping is not Welsh for microwave [as we implied in the contents of issue 21]. The correct word is microdon. 'Don' means wave in Welsh. It's also prototype Celtic for water, which is why there are so many rivers in Europe called the Don.

.

Adam
Caerdydd

Ben says: We have enough trouble with English, so it's crazy that we'd even attempt any Welsh in the mag. Either way, we had good, clean fun with microwaving glass, and it was pretty cheap too.

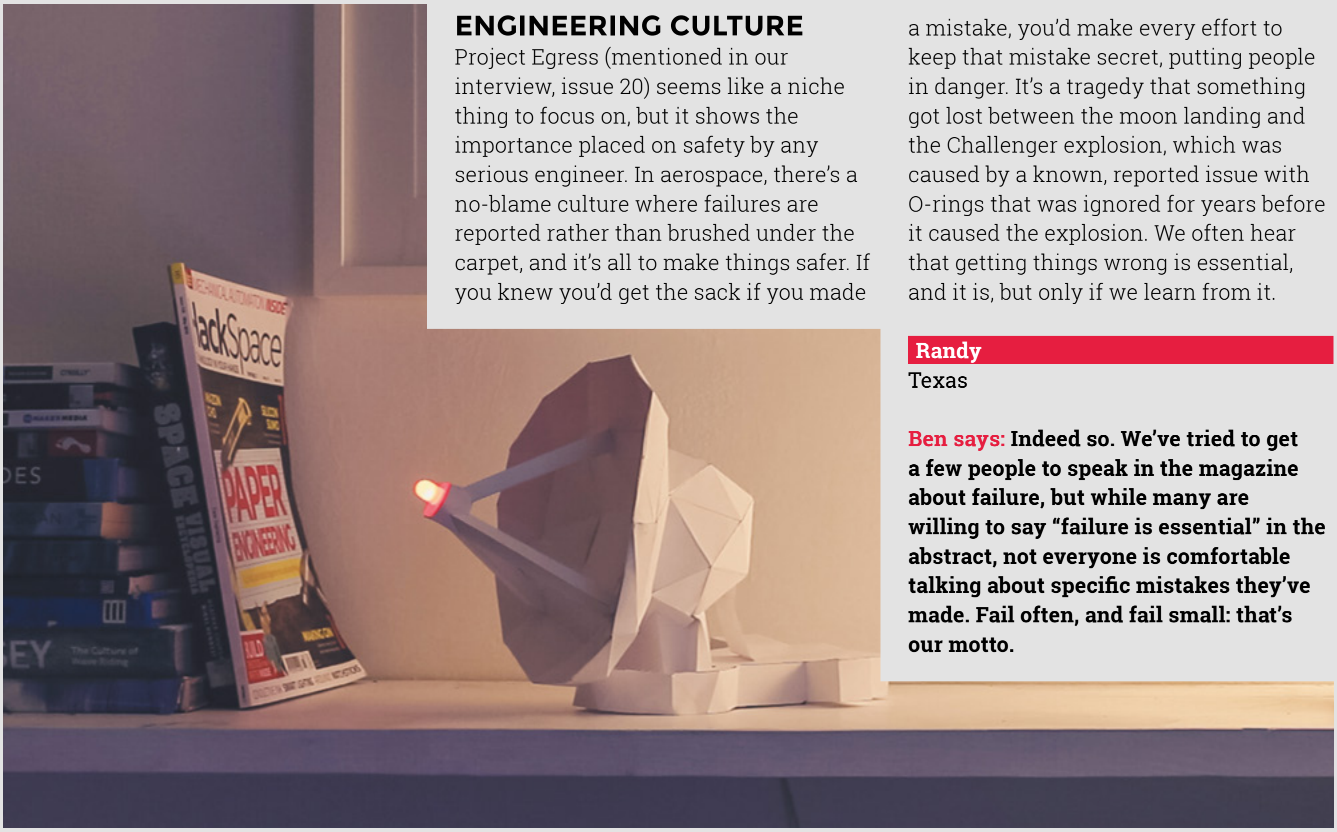


BUZZWORDS

The corporate companion (issue 21, Top Projects) tickled me. I used to work with a chap like that. I wonder if management realised how much money they could have saved if only they'd reproduced him with an empty suitcase, a Raspberry Pi, and a speaker?

Dan
Bath

Ben says: Experts have been telling us for decades that automation is going to turn the world of work upside down, so we shouldn't be too surprised that a middle manager can be replaced by a script and a £30 computer. Check out 8 Bits and a Byte's YouTube channel for more excellent daftery like this.



ENGINEERING CULTURE

Project Egress (mentioned in our interview, issue 20) seems like a niche thing to focus on, but it shows the importance placed on safety by any serious engineer. In aerospace, there's a no-blame culture where failures are reported rather than brushed under the carpet, and it's all to make things safer. If you knew you'd get the sack if you made a mistake, you'd make every effort to keep that mistake secret, putting people in danger. It's a tragedy that something got lost between the moon landing and the Challenger explosion, which was caused by a known, reported issue with O-rings that was ignored for years before it caused the explosion. We often hear that getting things wrong is essential, and it is, but only if we learn from it.

Randy
Texas

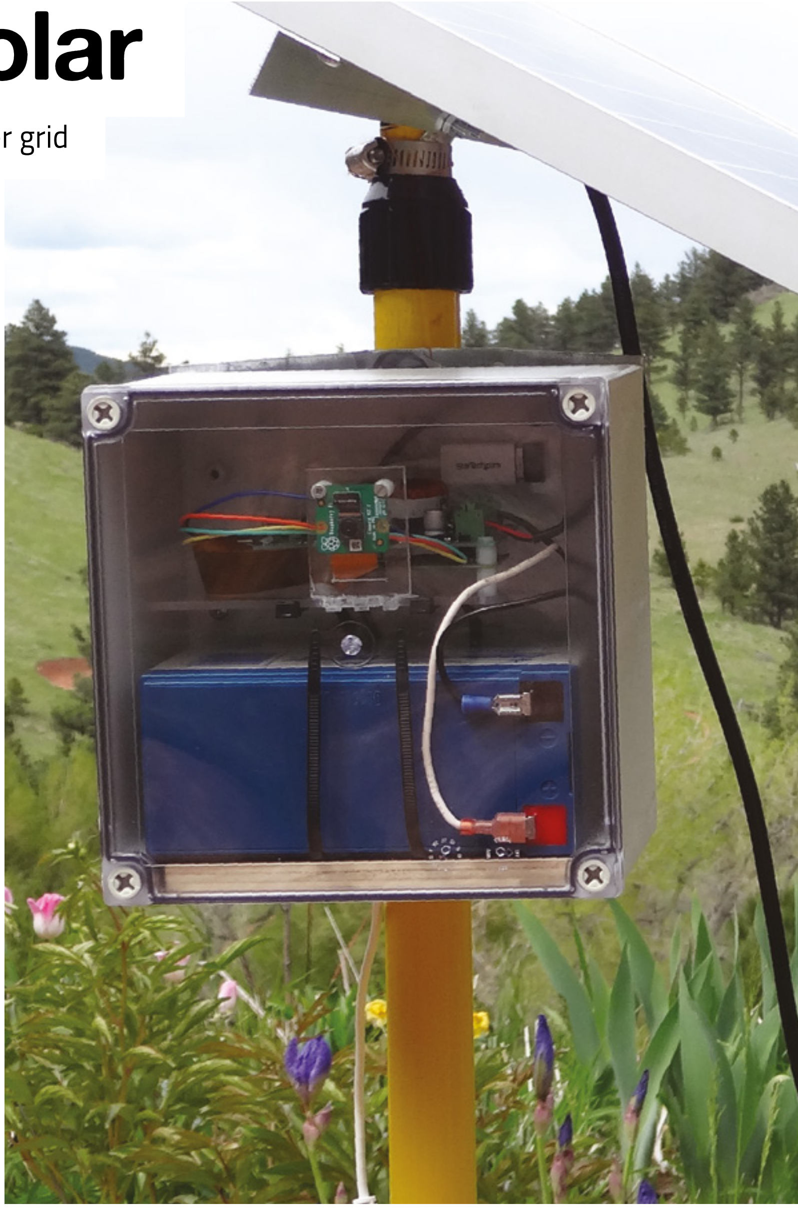
Ben says: Indeed so. We've tried to get a few people to speak in the magazine about failure, but while many are willing to say "failure is essential" in the abstract, not everyone is comfortable talking about specific mistakes they've made. Fail often, and fail small: that's our motto.

CROWDFUNDING NOW

makerPower Solar

Release your projects from the confines of the power grid

From £35 | **crowdsupply.com** | **Delivery:** November 2019



Solar power can seem like a really simple solution to providing electricity for remote projects. Take a panel, hook it up to the power port and you're done, right? Well, not quite. Photovoltaic cells (aka solar panels) provide a varying supply of power to your project that also requires a varying supply of power (unfortunately they aren't varying at the same rate). You therefore need something that you can put between the two to absorb the fluctuating power from the photovoltaic cells and distribute it smoothly to your project – a battery. However, batteries introduce their own complexities. They can't be charged too fast or allowed to drain too low. It might also be useful for your project to understand the power level of the battery so it can ration power appropriately.

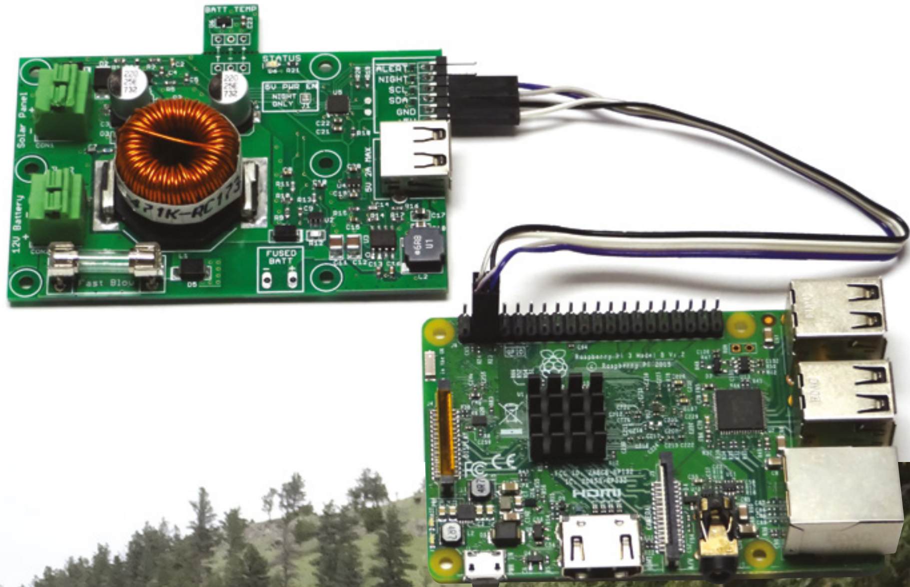
The makerPower Solar board is designed to combat exactly these problems. It takes 12 V solar cells (which don't actually output 12 V) and uses this power to charge a 12 V lead-acid battery. This is then used to power a 5 V regulator that can produce up to 2 A of current through either a USB port or solderable headers. Up to 3 A of 12 V is also accessible via a header.

So far, so normal. However, makerPower Solar also lets you read the current status of the battery and charger via an I²C interface. There are libraries for both Arduino and Raspberry Pi that let you look at the different voltages and currents flowing through the controller board. This can be used to intelligently manage the power you have available. For example, you can stop non-essential activities below a certain power, or perform high-power activities when the battery is fully charged and there's still power arriving from the sun.

While we haven't tested out the makerPower Solar, if it lives up to its claims, it will be a great addition to the toolkit for remote, solar-powered projects.

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.



Above
You can use the makerPower Solar to power a Raspberry Pi, and then use this computer to monitor the makerPower Solar

Left
A camera powered via a makerPower Solar charger

Space of the month: Makerspace Hull

Makerspace Hull

hcandl.co.uk

makerspacehull

@makerspacehull

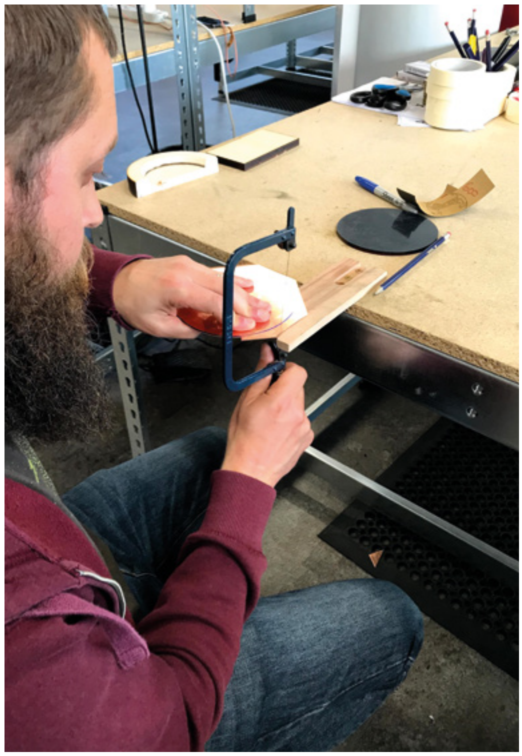
makerspacehull

It's a bit of a cliché to say that you want your space to be for everyone, but being part of a library, it's really true – we're quite publicly visible and accessible, so we can, and do, get all types of people walking through the door.

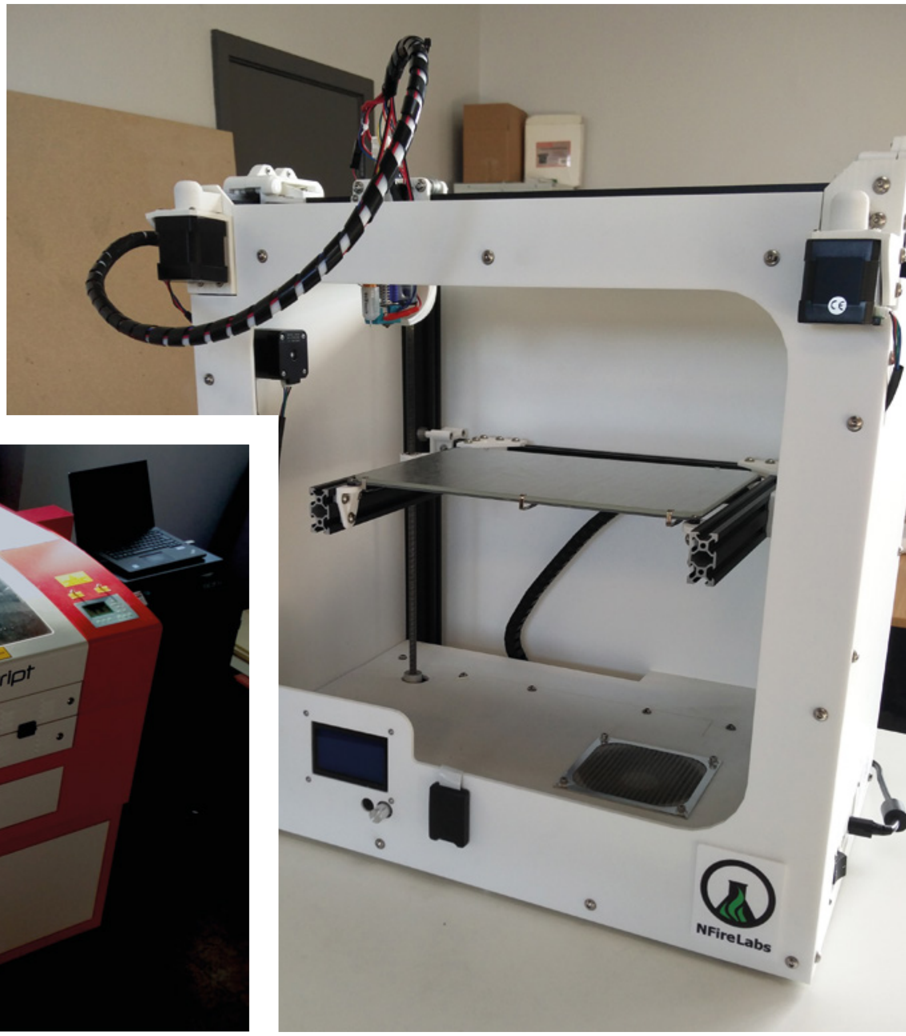
"We want to provide an environment where people can feel comfortable coming to just explore and have a play without necessarily having a 'project' to work on. Again, it's that library thing of wanting it to be a really friendly space which is genuinely open to all.

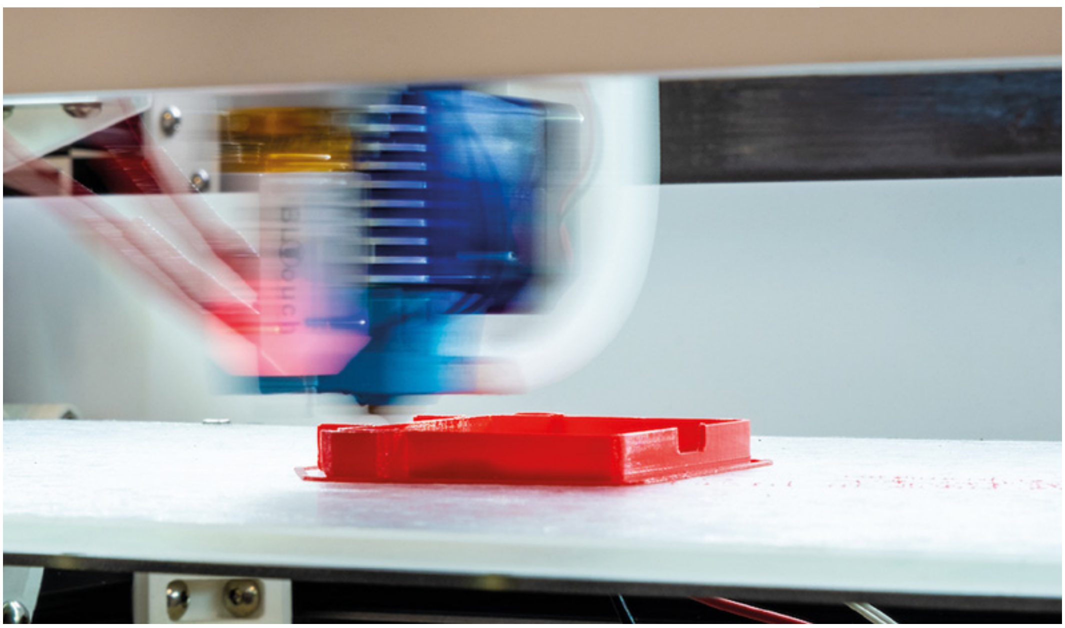
"There's a real mix of members from artists looking into new techniques and new ways of getting their work out there, to your more traditional makerspace hacker, and lots in between. Recently, we've had members laser-cutting things like earrings to sell on their market stall, and lino blocks for their artwork. 3D printing for cosplay and tabletop gaming seem to be really popular uses of the space as well.

"Funding was secured from Arts Council England to set up the space, with additional money from the James Reckitt Library Trust. Part of the original aim was to capitalise on the enthusiasm generated by Hull being the UK City of Culture 2017, and give people the opportunity to develop their skills and have access to new technologies. →



Right Makerspace Hull holds regular metalworking workshops





Yes, it's in a library – but that doesn't mean there aren't laser cutters and 3D printers to play with

Right These lino prints were made by Sean Azzopardi, who's a member of Makerspace Hull (image credit: Sean Azzopardi)

We've been open just over six months now, so it still feels fairly new, and we're learning as we go, but it feels a very exciting thing to be a part of.

"In terms of gear, we've got a laser cutter, two 3D printers, a large format printer, a digital cutter, a dye sublimation printer with heat press, a digital embroidery machine, as well as all the usual hand and power tools, soldering irons, and lots of electronics stuff like Arduino starter kits. There is also a suite of computers with programs such as Fusion 360 and the full Adobe package installed which help people go from idea, through design, to fabrication under one roof.

"The 3D printers are in fact built in the area, and it's really important to us to try and use local companies and suppliers wherever we can. There are plans to expand in the near future with a dedicated room for a CNC machine and also a woodworking room, which is something a lot of the members have been requesting.

"Anyone over the age of 16 can become a member for a small monthly fee – we really try and keep the costs low to reduce barriers to entry as much as possible.

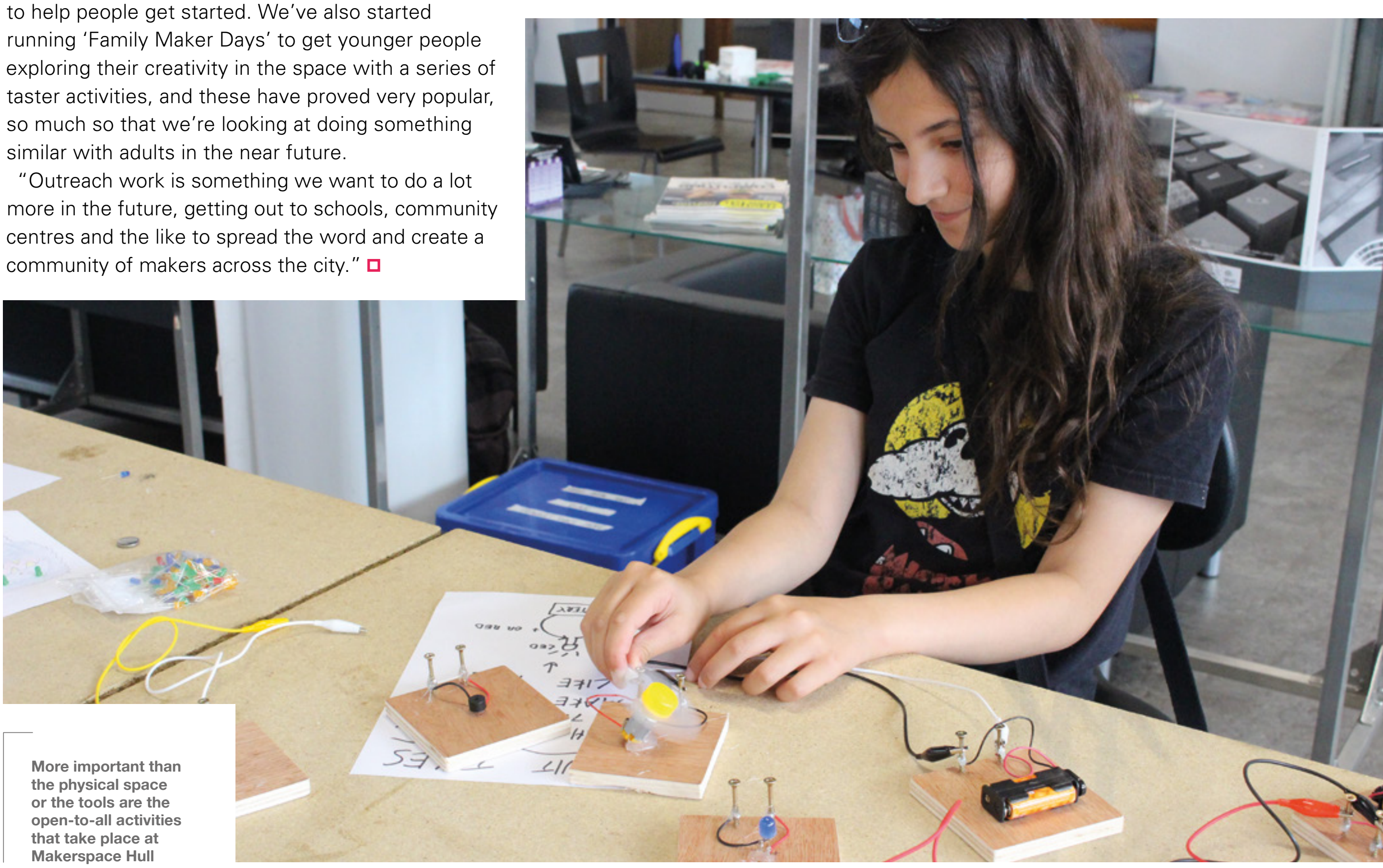
CONTACT US

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.

"The best way to find out more is just to pop in and have a chat when we're open; there'll always be staff on hand to show you around and help get you started on whatever creative journey you want to embark on.

"Workshops are a big part of what we offer, including lots of 'Introduction to…' style ones in subjects like Arduino, 3D printing, and laser cutting to help people get started. We've also started running 'Family Maker Days' to get younger people exploring their creativity in the space with a series of taster activities, and these have proved very popular, so much so that we're looking at doing something similar with adults in the near future.

"Outreach work is something we want to do a lot more in the future, getting out to schools, community centres and the like to spread the word and create a community of makers across the city." ◻



More important than the physical space or the tools are the open-to-all activities that take place at Makerspace Hull

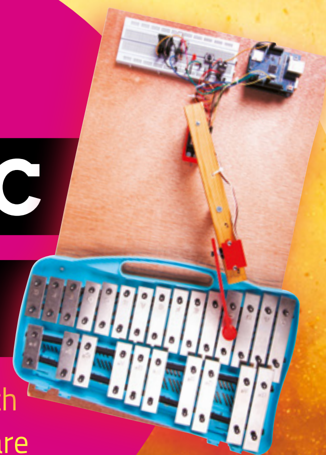

meet
learn
play
Ultimaker

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future



Robotic Music: Glockenspiel

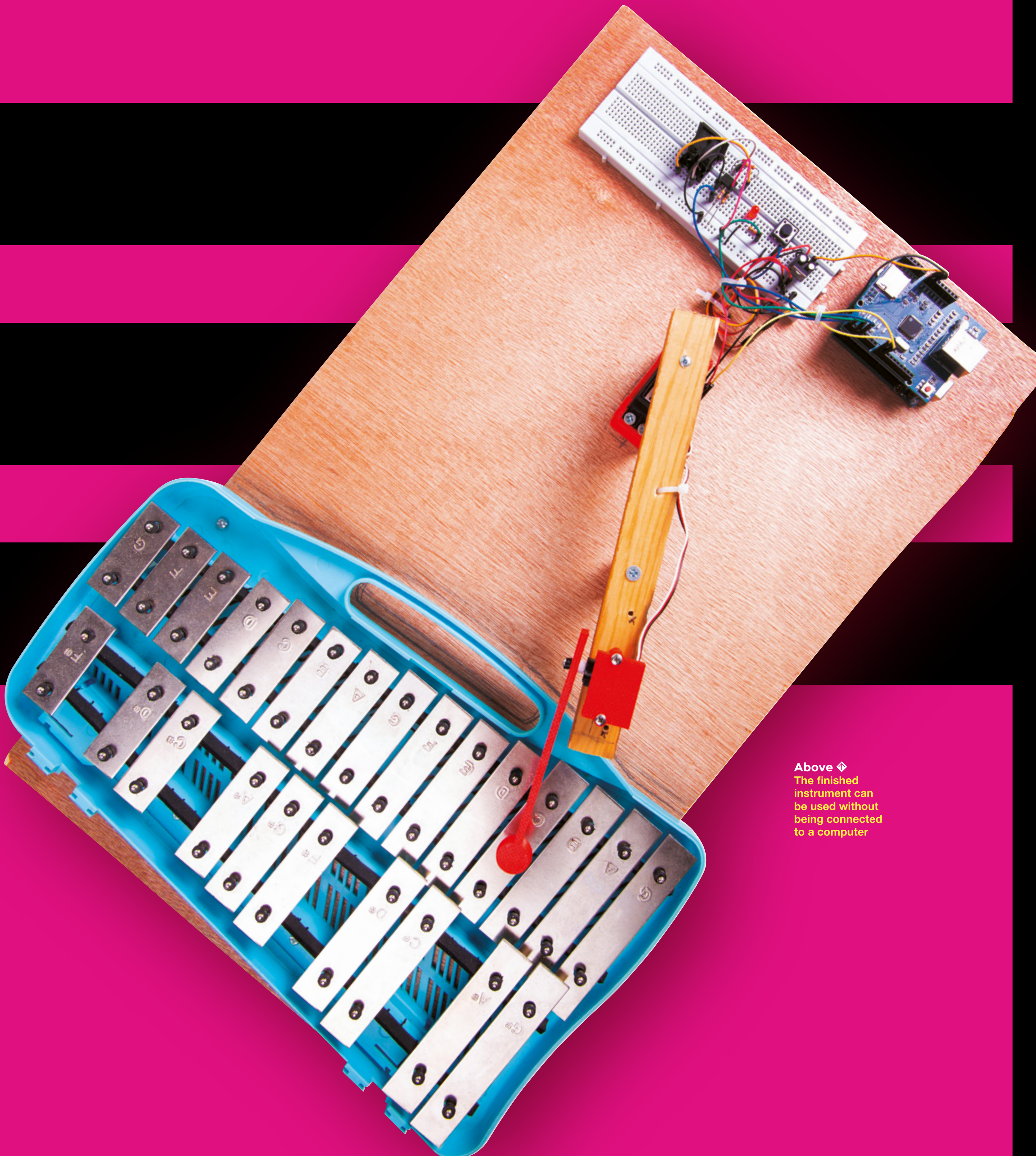
Build a musical instrument that plays itself

When you think about using electronics to create music, you might think of drum machines and synthesizers, with their sound generation technology hidden inside a black box. However, electronics can also be used to control acoustic instruments. In this tutorial, you will learn how to build a robot arm with two motors which can play a tune on a glockenspiel.

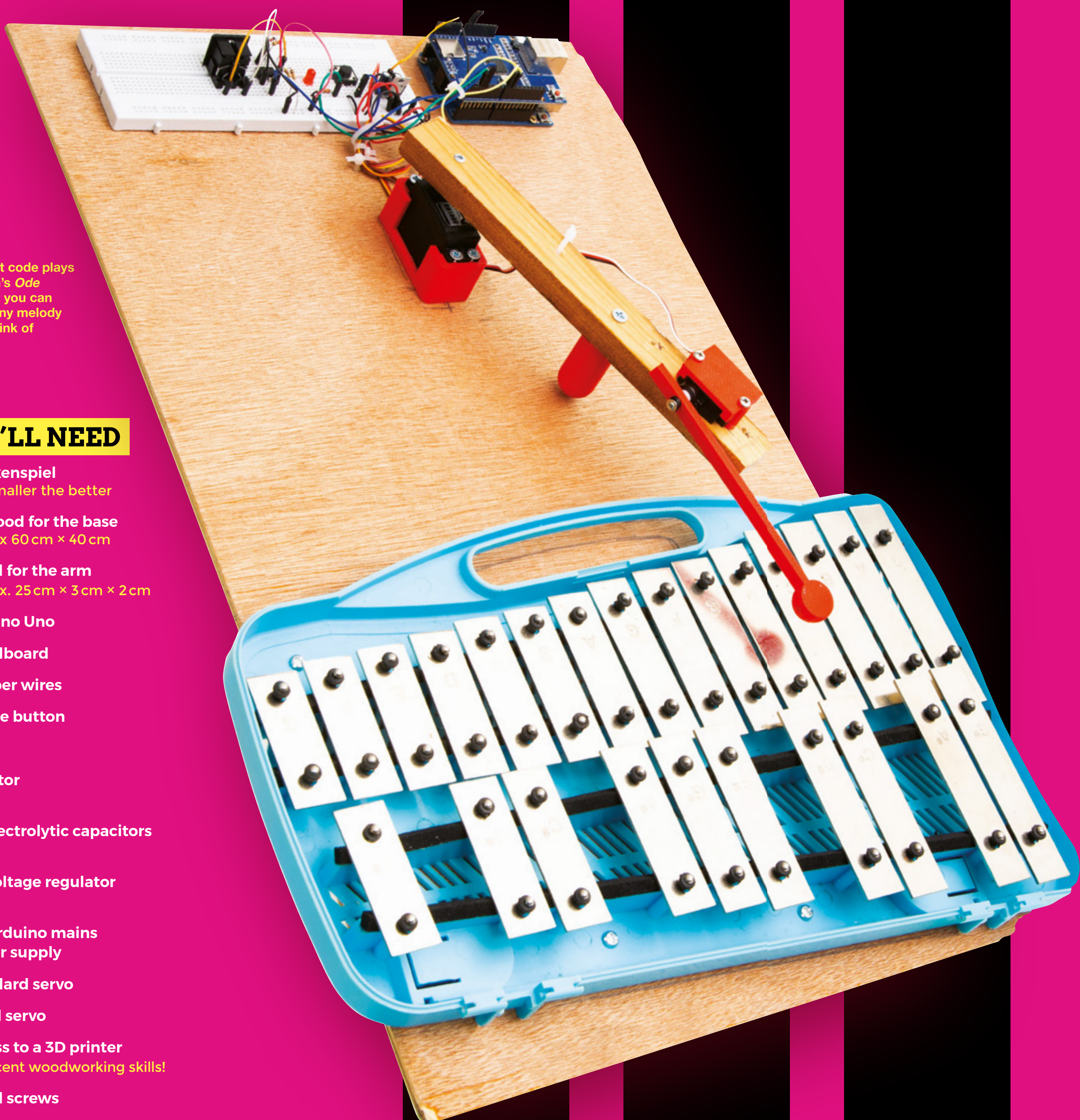
Our aim for this project is to take a glockenspiel and add every possible extra feature we can think of. In the main part of this tutorial, we will build a basic robotic arm to play a preprogrammed melody. Once that works, we will add other features such as controlling the instrument from an external keyboard or via the Internet of Things.

The design will consist of a cheap glockenspiel and an arm with a beater on the end, which will rotate to reach the different notes. The arm's movement is powered by a large servo motor, while the beater is powered by a small servo motor. The servos are controlled by an Arduino Uno, and the circuitry is housed on a breadboard so that we can easily modify it later on. The glockenspiel, arm, Arduino, and breadboard are all mounted to a piece of plywood.

This project can be approached in a number of different ways, depending on what resources are available to you. We have 3D-printed quite a few of the parts for this build, but if you don't have access to a 3D printer, you could achieve the same results using wood, a bucket full of screws and, of course, a roll of duct tape. →



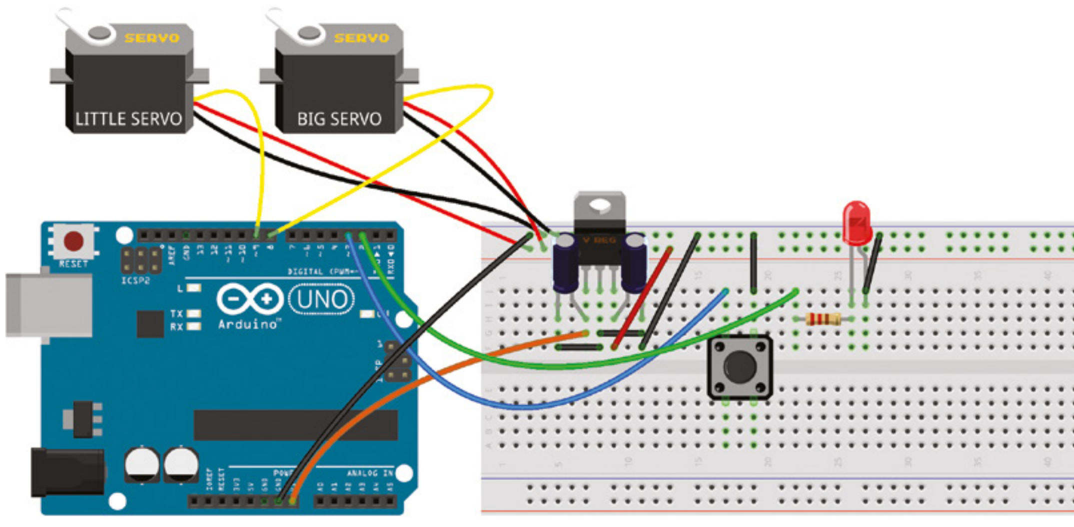
Above ⬗
The finished instrument can be used without being connected to a computer



Right ◈
The default code plays Beethoven's *Ode to Joy*, but you can program any melody you can think of

YOU'LL NEED

- **Glockenspiel**
  the smaller the better
- **Plywood for the base**
  approx 60 cm × 40 cm
- **Wood for the arm**
  approx. 25 cm × 3 cm × 2 cm
- **Arduino Uno**
- **Breadboard**
- **Jumper wires**
- **Tactile button**
- **LED**
- **Resistor**
  220 Ω
- **2 × Electrolytic capacitors**
  10 μF
- **5 V voltage regulator**
  7805
- **9 V Arduino mains power supply**
- **Standard servo**
- **Small servo**
- **Access to a 3D printer**
  or decent woodworking skills!
- **Wood screws**
- **A variety of M3 machine screws and nuts**
- **Basic woodworking tools**



Quick Tip

This project assumes some basic knowledge of the Arduino platform – if this is your first time playing with an Arduino, start here: **hsmag.cc/iSwUtN.**

Above
The 5V rail is fed from a 9V mains transformer, via a 5V voltage regulator

A SOLID BASE

To start, find yourself a piece of plywood or similar to act as a base for your instrument. As you can see in the photos, it needs to be wide enough that you can mount the glockenspiel firmly at one end, and long enough to also contain the arm, breadboard, and Arduino.

Next, mount your Arduino and breadboard next to each other on the base, with the Arduino's USB and power ports nearest the edge of the base. Breadboards often come with a sticky underside, making them very easy to mount. The Arduino can be secured in place by screwing wood screws through its mounting holes, but be careful not to over-tighten them and crack the board.

On your newly secured breadboard, build the circuit shown in the diagram above. The circuit contains a button, which will be used to start and stop the melody, and an LED, which is useful for debugging (and aesthetics!).

The circuit also contains a 5-volt regulator chip. The Arduino can't provide enough current for our somewhat chunky main servo, so we have to think laterally. Our Arduino is going to be powered by a 9V mains transformer, which has plenty of current, and the Arduino lets us access this 9V power feed directly, via the 'Vin' (voltage input) pin. The 5V regulator chip takes the 9V input and steps it down to 5V, giving us enough power (at the correct voltage) to run our servos. The capacitors are there to keep the voltage stable – make sure you get them the right way around.

FIRST MOVEMENT

Plug the 9V power supply into the Arduino and turn it on, before plugging the Arduino into your PC with the USB cable. With a project like this, that requires external power to drive motors, it is important to make sure that there is always external power available to the Arduino. Throughout this project, make sure that you don't turn off the external power unless you have already unplugged the USB cable, otherwise the Arduino will attempt to power the motors via USB, which provides insufficient current and could cause damage.

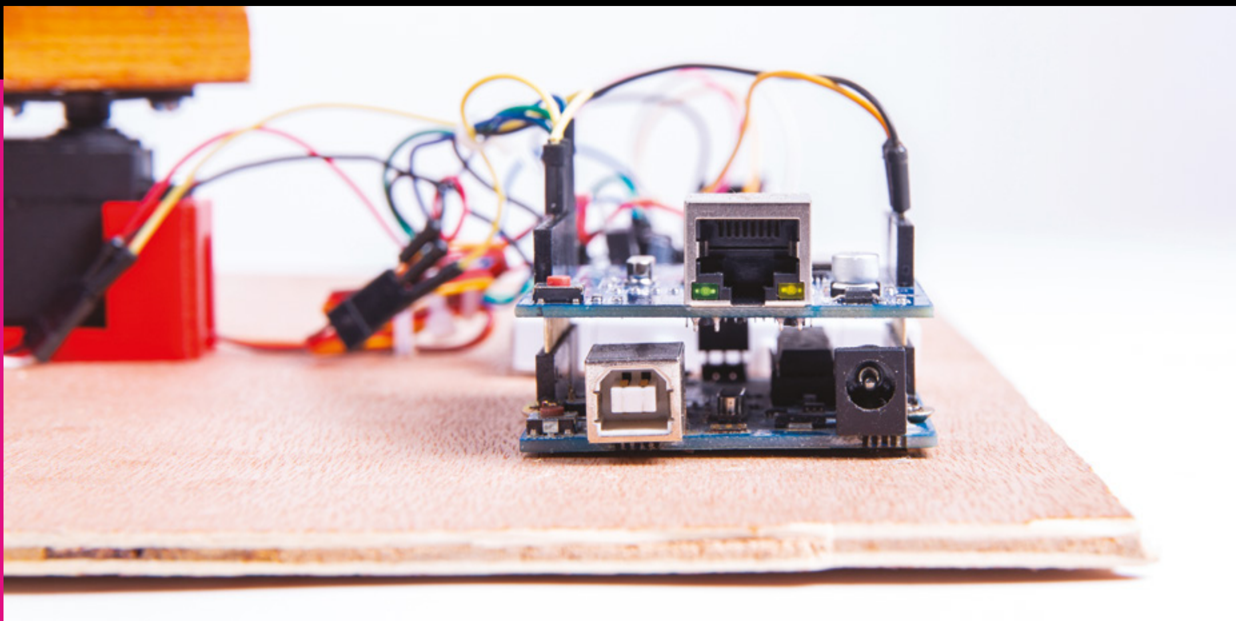
The circuit contains a button, which will be used to start and stop the melody, and an LED

With the servos connected, as shown in the diagram, and the external power supply turned on, try out the Arduino example sketch found in File > Examples > Servo > Sweep. Upload the sketch to →

Controlling servos

A servo is a special type of motor which has a built-in feedback mechanism, meaning that the motor 'knows' its current position (or, sometimes, speed). Our servos read a signal from the Arduino and then move to a particular angle, between 0° and 180°. The Arduino sends a pulse of a specific length, and the servos interpret this as an angle and rapidly move to the correct position.

You don't need to worry about what pulse length to send, because there is a ready-made library called 'Servo' which handles all the calculations. You just need to tell the Arduino which motor is connected to which pin, and what angle you want each motor to move to.



Right ⬗
Shields stack on top of the Arduino to add functionality

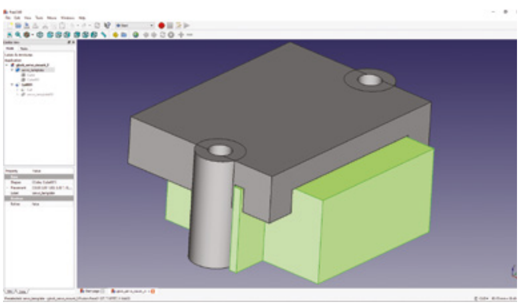
Below ⬊
Creating a basic model of the servo itself in FreeCAD was helpful when designing a servo mount

Timing troubles

The arm takes a finite time to reach a note before the beater can hit it, which needs to be taken into consideration in the code. After some trial and error, we found that a `preHitDelay` of 150 ms was enough time for the arm to reliably reach its destination before triggering the beater, but feel free to adjust this value up (for increased reliability) or down (for a faster maximum speed). There is also a `postHitDelay` value of 50 ms, which can be tweaked to give a nice clean hit of the beater – if this value is too short, the beater won't reach the keys; too long and it will dampen the note, rather than letting it ring.

the board, and you should see one of the servos move back and forth. Take a look at the code, and try editing it. See if you can change the speed of the sweep, or move both servos at once – the code that we will use to play the glockenspiel is similar to this example.

Although it seems a little counter-intuitive at this early stage of the build, now is a good time to upload the main sketch, since we can see if the circuit is working properly before we start attaching



long, ungainly parts. The sketch plays Beethoven's *Ode to Joy* when the button is pushed once, and stops playing when the button is pushed again. The LED is lit while the melody is playing.

UNDER THE HOOD

Go to **hsmag.cc/issue22** and open the **glock_main** sketch. The code starts by including two libraries: one called 'Servo' to control the servos (obviously), and one called 'Bounce2', which makes it easier to handle button presses. Bounce2 is not included by default in the Arduino software, so go to Tools > Manage Libraries, search for 'Bounce2' and install it – if you don't do this, the compiler will complain and won't let you upload the sketch.

We then declare objects to represent the two servos and the button, before defining various variables which will be used in the code, such as the possible angles of the beater, the amount of time to wait for the arm to reach its position, and the sequence of notes that you want to be played. In the `setup` function, we tell the Arduino which pins correspond to the servos, button, and LED, and we set the delay between notes based on the tempo.

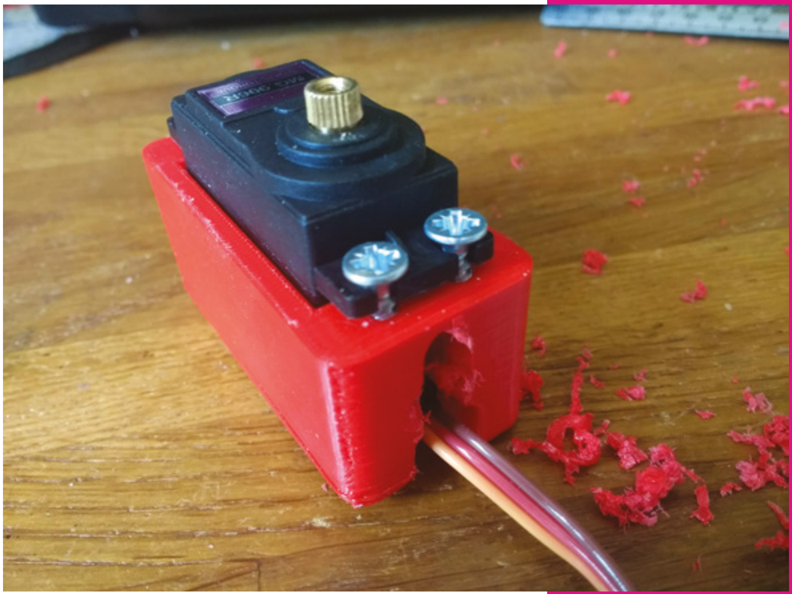
In the main loop, we continuously check whether the button has been pushed. The `updateButton` function listens for a button press and sets `sequencePlaying` to either true or false. When `sequencePlaying` is active, notes are played in order. The `playNote` function (see code extract) handles the process of moving the arm to the correct note,

Quick Tip

Some useful pedantic knowledge: many people will mistakenly call this instrument a xylophone, but a xylophone's bars are made of wood, not metal.

Long arm of the law

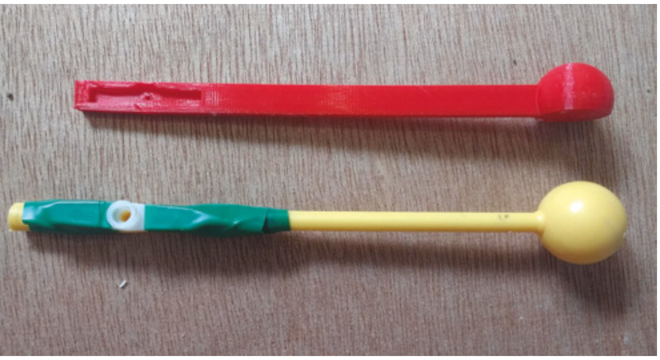
Getting the length of the arm right is important – too short, and its arc of movement will be very small, meaning you won't be able to reach many notes on the glockenspiel (which are laid out in a straight line). If you make it too long, however, it may require too much force for the servo to move the arm, even with the external power supply. For our build, we found that 28 cm from the large servo's shaft to the tip of the beater was about right, with the actual wooden part of the arm measuring 23 cm. This gave us a wide enough arc to reach a whole octave of notes, without appearing to stress the motor unduly.



waiting until it is in position, moving the beater downwards, waiting a few fractions of a second, moving the beater back up, then finally pausing until the next note, according to the desired tempo.

```
void playNote(int thisNote) {
  if(thisNote >= 0 && thisNote <= 7) {
    // note is valid, play note
    bigServo.write(notePositions[thisNote]);
    delayWhileUpdating(preHitDelay);
    littleServo.write(beaterHitAngle);
    delayWhileUpdating(postHitDelay);
    littleServo.write(beaterRestAngle);
  } else {
    // delay instead of playing note
    delayWhileUpdating(preHitDelay +
postHitDelay);
  }
  delayWhileUpdating(tempoDelay);
}
```

Upload this sketch to the Arduino, making sure the external power supply is connected first, and press the button. If all is well, the LED should come on and the two servos should start twitching independently. If this doesn't happen, double-check your wiring.



ARMED AND DANGEROUS

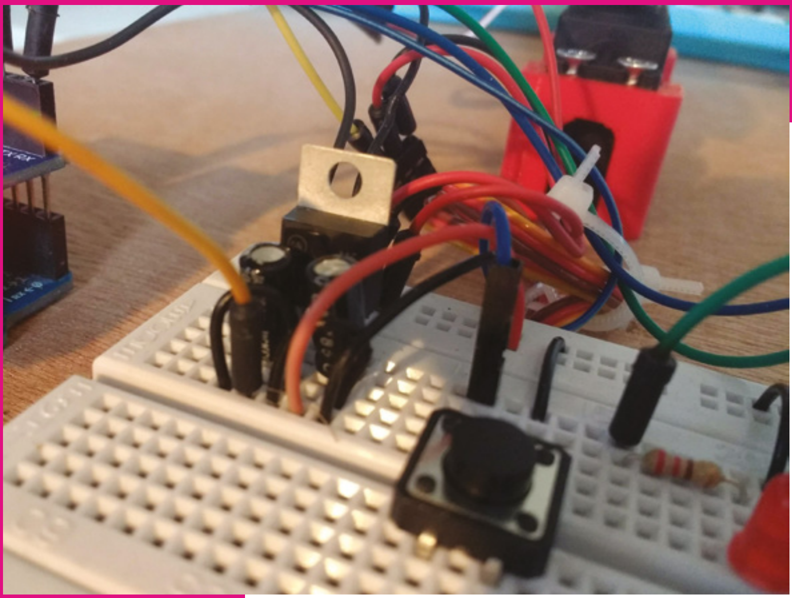
You can now work out where to position your glockenspiel and your large servo. Cut a piece of wood to the desired length for your arm, and attach an appropriate servo horn. Horns are the (usually plastic) pieces that come with a servo and fit snugly onto the servo's shaft. Find an appropriately shaped horn and screw it to the underside of your wooden arm at one end. Mount the wooden arm on the servo so that it is parallel with the long side of the baseboard, aligning with the middle of the glockenspiel. Make sure the servo is at its 90° position when you do this (the sketch defaults to the big servo being at 90° when it starts up).

The arm would work like this, but it's best to secure it down to the servo if possible. Some servos have an M3 thread inside the shaft, allowing you to screw the arm directly to the shaft. Alternatively, you might be →

Top
The beater was printed flat on one side (unlike a regular glockenspiel beater) to make it easier to print

Above
If you find a design flaw after printing a part, it's often possible to bodge it until you get around to printing a better one!

Left
It's possible to bodge an existing glockenspiel beater to work with this build if necessary, but 3D-printing a custom one works better



Quick Tip

Make sure you know what position each servo is in before attaching parts to it, or your machine might try and tear itself apart when you run the sketch.

Above Voltage regulators often have a hole so you can attach a heat sink, if required

Right This previous version of the robot glockenspiel was trickier to build, but has a wider range of notes

Far Right You can use bubble wrap to mute the glockenspiel while testing, if your housemates don't appreciate the noise

able to use a carefully chosen wood screw (be careful not to crack the servo shaft with an oversized screw), or you could resort to glue. If you are going to screw through the arm, you'll want to drill a hole through the wood, starting in the centre of the horn.

Since one end of the arm is hovering a few centimetres above the base, you may want to add a vertical support, as seen in the pictures. You could possibly get away without it if the arm is secured tightly to the servo – it's up to you. We 3D-printed our support, but it's a very simple piece that could certainly be fashioned from wood.

THE THIRD DIMENSION

Speaking of 3D printing, the vertical support is one of several parts of this build that you can either 3D-print or make yourself from scratch – the other parts are a mount for the large servo, a mount for the small servo, and a custom glockenspiel beater. We designed the 3D parts in FreeCAD, a free, open-source 3D design program which is worth downloading. The STL files are available from **hsmag.cc/issue22** – you can either use them as they are, edit them, or create your own.

If you don't have a 3D printer yourself, you could check out your local makerspace, or order the parts from an online 3D printing service. It's certainly possible to build this project without any 3D prints – you could achieve identical functionality with some wood, glue, screws, a saw, and a drill – but we're going to continue the tutorial on the assumption that you're using the printed parts.

TIGHTEN UP

The large servo mount is held in place from under the base with wood screws – you'll need to measure and drill pilot holes quite precisely so you know where to place the screws. Another four screws hold the servo itself in the mount. The small servo mount clamps the motor to the arm using M3 machine screws, although you could also use wood screws.

The custom glockenspiel beater features a recess, into which you can place one of the horns from the small servo. As with the large servo, you need some way of holding everything in place – small servos tend to come with a tiny screw for this purpose, but this will likely be too short to reach through the stem of the beater, so you will require some ingenuity. We used a longer screw with the same diameter, but you could modify the design to allow you to use the original screw, or come up with another solution.

TUNING UP

If you run your sketch at this stage, you will hopefully find that the robot arm functions, but it is very unlikely that it will play the tune correctly. This is because the

The large servo mount is held in place from under the base with wood screws

arm positions have to be dialled in for the individual layout of your instrument. While the low C position on our build was at 109°, yours is likely to be at least a few degrees lower or higher, depending on the size and position of both your glockenspiel and your arm.

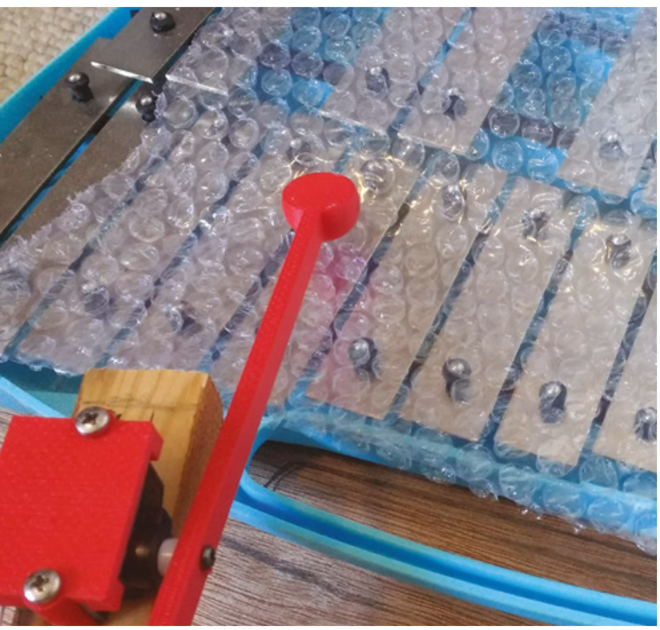
Other approaches

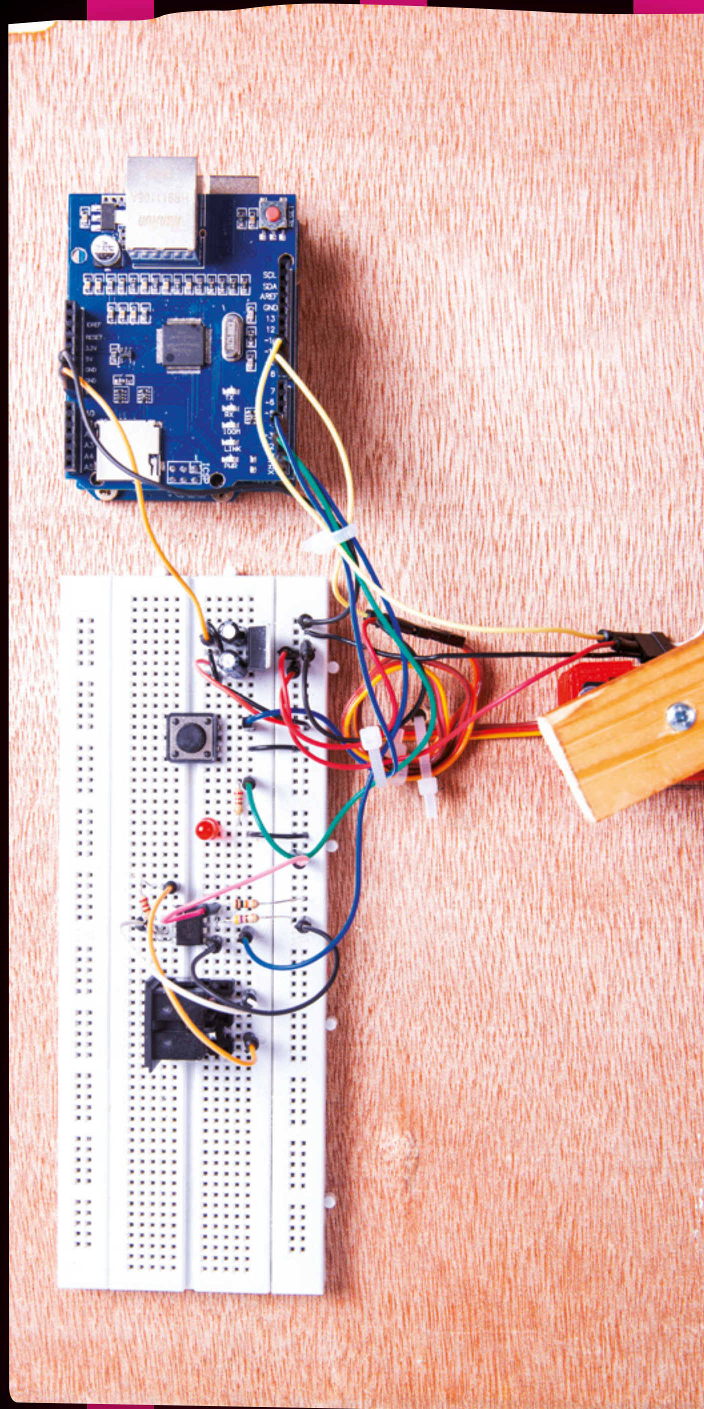
The build detailed here is one relatively simple way to make an electronically controlled glockenspiel, but there are plenty of other methods, each with their own pros and cons. One alternative strategy is to use a solenoid for each key of the glockenspiel, as shown in the 'Glock-O-Bot' made by the Mather family at Fab Lab Manchester – this design allows for multiple notes to be played at once (polyphony) and is more space-efficient than our design, although there is a lot more wiring!

Another approach that we tried previously, with some success, is to remove the metal bars from the glockenspiel and rearrange them in a semicircle, meaning that more of the notes are within range of the beater's arc of movement, including the sharps and flats.

To 'tune' your instrument, make a copy of the current sketch and adapt it to help you figure out which note is at which angle. Try replacing the *Ode to Joy* melody found in the `noteSequence` array with a simple repeating scale, then start tweaking the values in the `notePositions` array until you get a good result. For reference, the number 0 in our build corresponds to the lowest C on the glock, while the number 7 is an octave higher. The number -1 denotes a pause in the melody.

Finally, mount the glockenspiel itself on the baseboard using wood screws. You now have a robot glockenspiel that can play any melody you can think of! But wait, there's more… →





Extending your robot glockenspiel

How to take your instrument to the next level

Hopefully, your glockenspiel can now play a melody at the touch of a button, but there's a lot more we can do! Here are a few extension projects to help you learn more about your robotic instrument, but there are almost endless ways it can be adapted to new uses. The only limit is your imagination – it sounds like a cliché, but it's true!

Big Ben chiming clock

This first modification is a nice one to start with, because we can make it without any extra components. By making some changes to the code in the Arduino sketch, we can tell the instrument to play a melody at a specific time, rather than on the push of a button.

We're going to trigger a melody every 15 minutes, making a chiming clock *à la* Big Ben (although Little Ben might be a more apt name for our tiny clone). Download the **glock_clock** sketch and have a look at it – the code starts off pretty similar to the main build, but there are a few additions and changes.

We're going to trigger a melody every 15 minutes, making a chiming clock

The single, long **`noteSequence`** array in the previous version has been replaced by a two-dimensional array containing four separate lists of notes – these are the four different melodies played by Big Ben. Below this we have also defined variables representing the current time in hours and minutes – you need to set these manually to the current time before uploading your sketch (e.g. if the time is 3:47 pm, set **`startTimeHours`** to 15 and **`startTimeMinutes`** to 48, then upload the code to the Arduino just as the minute hand ticks over from 47 to 48). Once your sketch is running, it should stay approximately in time, and the chimes can be turned on or off by pressing the button. →

Adding a display

The next logical addition for either the chiming clock or the Internet of Things alarm clock would be a display to show the time. Perhaps the most obvious component for this would be a four-digit, seven-segment display, as used in digital clocks since the 1970s, although a more modern OLED display could also be cool, as could Nixie tubes or edge-lit LED displays, sometimes known as 'Lixies'.

YOU'LL NEED

'BIG BEN' CLOCK

- **Nothing extra**

INTERNET OF THINGS ALARM CLOCK

- **Arduino Ethernet shield**
- **Ethernet cable**

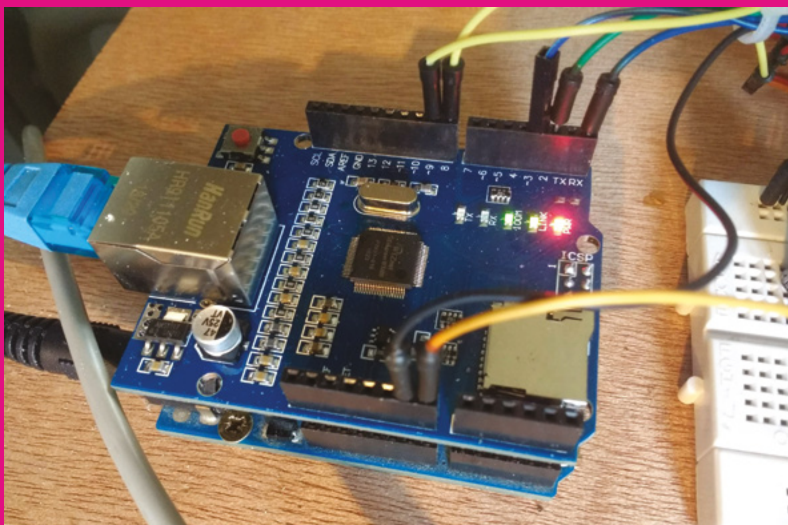
MIDI INPUT

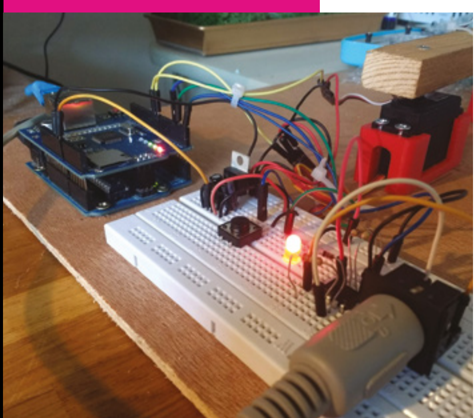
- **Optoisolator** 6N138
- **5-pin DIN (MIDI) socket** breadboard-compatible
- **Resistors** 220 Ω, 470 Ω, 10 kΩ
- **Diode** 1N4148

Left The glockenspiel can be programmed to mimic Big Ben's famous melody

Credit David Liff, CC BY-SA 3.0

Internet-connected alarm clock





Above An Ethernet shield allows your Arduino to talk to the outside world

Below The completed glockenspiel can be controlled via the internet or from a keyboard

Our 'clock' will now chime the same four patterns every hour, which is cool, but nothing that couldn't have been achieved using clockwork in the 19th century. Let's bring things up to date by making our clock read the correct time from the internet, and allow an alarm to be set with a custom melody via a website. By adding an Ethernet shield to the Arduino, we can connect the robot glockenspiel to the Internet of Things.

The Ethernet shield is perhaps one of the simplest and cheapest ways to connect an Arduino to the internet. To add it to your circuit, simply remove the jumper wires from the Arduino, insert the shield into the Arduino's pins, and then reconnect the jumper wires in the same pin sockets as before. Now, plug the shield into your router using an Ethernet cable.

At this stage, both the previous sketches should still work – you should try them out to verify that the circuit is intact.

We're going to use a service called Adafruit IO, which facilitates IoT projects. Go to **adafruit.io** and sign up (it's free). Adafruit IO consists of a series of 'feeds' which can be read or written to. There is an Arduino library which handles the process of either reading or writing data, so we want to install that and try an example.

Go to the Arduino library manager, install the Adafruit MQTT Library, then open the sketch found in File > Examples > Adafruit MQTT Library > mqtt_ethernet. Read through the code – this sketch subscribes to a simple binary feed called 'onoff', reading changes from an on/off button in the Adafruit IO web interface. The sketch also sends dummy data to a feed called 'photocell', and the data sent from the Arduino can then be seen live on a web interface called a 'dashboard'.

Set up these two test feeds, 'onoff' and 'photocell', within your account, and create a dashboard with a button to send data to the 'onoff' feed, and a graph to read data from the 'photocell' feed. This is a little complex, so check out these sites – **hsmag.cc/HRSKql** and **hsmag.cc/GF0brB** – to get an overview of what feeds and dashboards are all about.

Once you think you're up to speed, enter your username and key in the **mqtt_ethernet** example sketch, and upload the code to your Arduino. Open the serial monitor in the Arduino software and, with luck, you should see details of a successful connection to the Adafruit server. Try playing with the dashboard – you should now have two-way communication between the dashboard and your Arduino.

SOUND THE ALARM

If that works, you can download the **glock_iot_alarm** sketch, which is a mash-up of the previous example and our glockenspiel code. As before, enter your Adafruit IO username and key, and make sure that **`timeZoneOffset`** and **`daylightSavingsOffset`** are correct for your region (if you want an advanced project, you could try to obtain these settings automatically somehow). Next, on the Adafruit IO site, create a new dashboard called 'alarm glock' and populate it with three new feeds: 'alarmtoggle' (a toggle button), 'alarmtime' (a text block), and 'alarmmelody' (another text block).

Upload the code to the Arduino and check the serial monitor to see whether everything appears to be running smoothly. If the sketch is working, you should be able to set an alarm from the dashboard. Changing the alarm time or melody on the dashboard should cause the alarm time or melody to be changed on the Arduino (and cause the LED to flash briefly), while the toggle button should activate or deactivate the alarm (with the LED either on or off to show the current state).

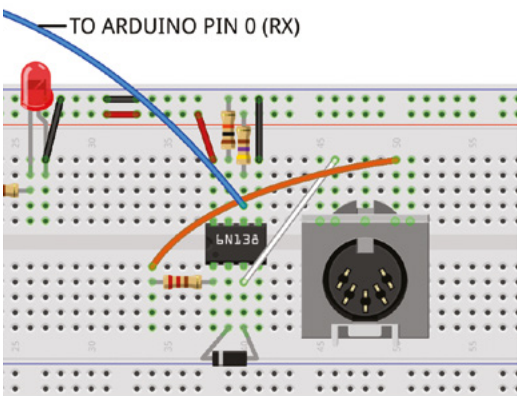
MIDI input

MIDI (which stands for 'Musical Instrument Digital Interface') is a technology used extensively in studios and live gig situations to connect synthesizers, drum machines, sequencers, and more. For our final feature, we're going to add a MIDI input to the glockenspiel so that it can be played digitally by another instrument.

MIDI outputs are relatively simple to wire up, since you're just sending an electrical signal down a wire, but MIDI inputs are a little more complex. Since you can't really trust signals coming from external devices, a MIDI input uses an optocoupler which converts the incoming signal into pulses of LED light, then immediately back into an electrical signal again.

Add the extra components to your existing circuit. In the Arduino software, install the 'MIDI Library' by Forty Seven Effects and upload the sketch. You may actually need to disconnect the jumper wire from pin 0 in order to do the upload, because the Arduino gets confused between the computer's USB signal and the keyboard's MIDI signal – you can replace the jumper after the sketch has been uploaded.

Left
Since the arm has a limited range, only the middle octave of the MIDI keyboard affects the glockenspiel



Below
The rest of the circuit remains in place, but by adding a MIDI port we can control the robot arm from another instrument

Switching brains

Since this project was already based around an Arduino Uno, a plug-and-play Ethernet shield made sense, but there are loads of different ways to add internet connectivity to an electronics project. You could swap your regular Arduino Uno for an Uno WiFi with wireless capability, although it's about twice the price of a regular Uno.

A Raspberry Pi would also be a good choice for this project, since many models come with WiFi. You could either connect a RPi to the Arduino via USB, or run the whole build solely from a Raspberry Pi. There's a huge number of options for small controller boards at the moment, and this project should work on most that can control servos (which is the majority).

GETTING THERE SLOWLY

If everything is working properly, pressing certain keys on the keyboard (middle C, and the octave of keys above it) will now cause the servos to move. The code is written so that holding down a key will move the arm into position, and releasing the key will cause the beater to strike. While this is a little counter-intuitive, it actually makes the instrument easier to play accurately – if the Arduino attempted to move to a note and hit it as soon as a key was pressed, there would be a noticeable delay before each note while the arm moved into position. With this method, as long as the player can get used to playing 'in reverse', the latency is much better, and it's possible to play along in time to a piece of music. →

Signing off

You might notice that some numerical values in these sketches are of the type 'unsigned long', rather than just 'int' (integer). The Arduino usually deals with time in milliseconds, but the number of milliseconds in a day is too large to be stored as an 'int' value, so instead we use 'long' variables which can contain much larger integers.

The 'unsigned' part means that the value can only be positive, and allows for even larger numbers. If you find any strange behaviour while writing or modifying code for your glockenspiel, think about whether any of the variables need to be 'long' or 'unsigned long' instead of 'int'.

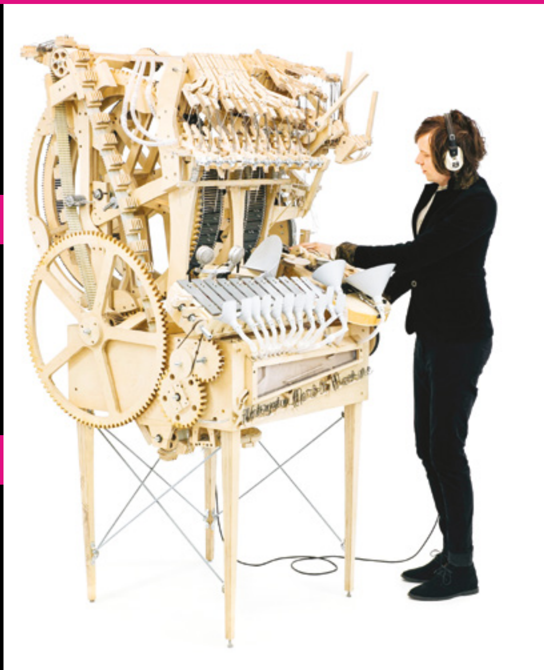
Quick Tip

If you made changes to the note positions or delay times in the main sketch, remember to copy these over to the other sketches.

Awesome musical instrument hacks

Building your own instrument gives you a unique sound

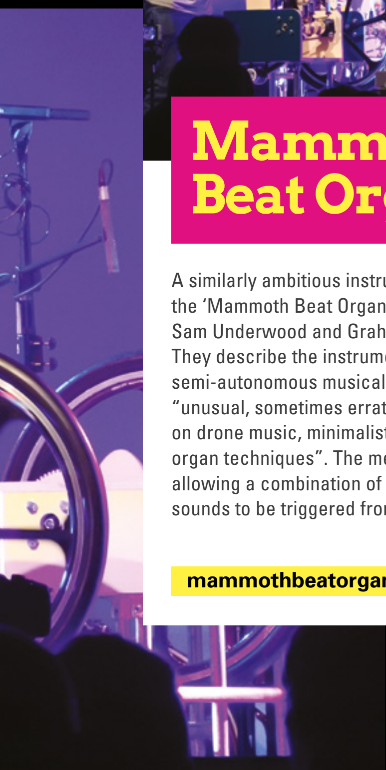
There is a vibrant, worldwide community of makers working on exciting, fun, and often bizarre homebrew musical instruments. By thinking about the underlying design of an instrument, makers often end up creating fascinating, unusual music which would not be possible with 'normal' instruments.



Marble Machine

First up, the terrifyingly complex and impressive 'Marble Machine' by Martin Molin of the Swedish band Wintergatan. This instrument uses a hand-powered mechanism to raise hundreds of marbles to the top of a ramp, before gravity transports them onto a glockenspiel in a preset pattern determined by, what looks like, a system of Lego pegs. On top of this, the machine (which is over six feet tall) has a built-in 'drum machine' with kick, snare, and hi-hat, plus an entire semi-automated bass guitar. It's impossible to adequately describe, so go and add to the Marble Machine's 100 million-plus YouTube views and see for yourself.

wintergatan.net



Mammoth Beat Organ

A similarly ambitious instrument to the Marble Machine, the 'Mammoth Beat Organ' has been created by Sam Underwood and Graham Dunning in England. They describe the instrument as "a two-player, semi-autonomous musical instrument" which plays "unusual, sometimes erratic compositions drawing on drone music, minimalist repetition, and fairground organ techniques". The mechanical design is modular, allowing a combination of melodic and percussive sounds to be triggered from a central drive shaft.

mammothbeatorgan.co.uk

Furbinator

A somewhat more terrifying aesthetic can be found in Psychiceyeclix's circuit-bent instruments. Circuit-bending is the practice of altering an electronic device to generate sounds beyond what was originally intended, such as adding a pitch-shift knob to a children's piano. Psychiceyeclix has an active and fascinating YouTube channel, where he shows off his ability to make almost any piece of electronics sound demonic. He has hacked everything from a Sega Mega Drive to a Thomas the Tank Engine sound book, and sells a lot of his creations through his Etsy store. →

psychiceyeclix.wordpress.com

Above
The 'Marble Machine' is a breathtaking one-man-band of an instrument

Credit
Samuel Westergren

Left
Sam Underwood and Graham Dunning playing the Mammoth Beat Organ live at Supersonic Festival

Below
'The Furbinator' – a circuit-bent Furby by Psychiceyeclix

Right
Thomas Truax's Hornicator instrument is bizarre but versatile

Below
Helen Leigh's MINI.MU glove is designed to be made from scratch by beginners

Hornicator

One musician who is particularly known for his homemade instruments is Thomas Truax. His surreal songs take place in the fictional setting of Wowtown, and are soundtracked by his 'Hornicator' (an unholy union of a gramophone horn, several microphones, and numerous acoustic instruments) and a variety of mechanical drum machines. He shows that DIY instruments can become a signature part of a musician's sound and aesthetic – his creations have a steampunk style, which elevates his stage show to another level.

thomastruax.com

MINI.MU Glove

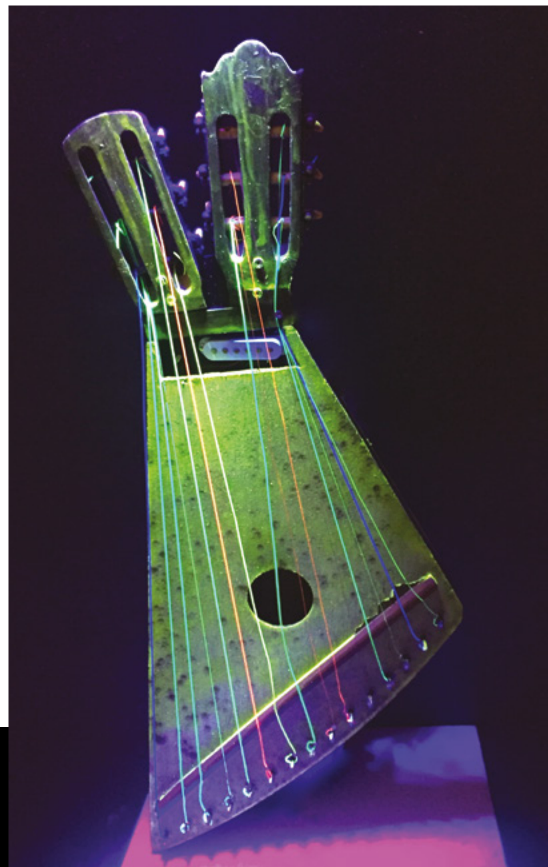
Helen Leigh is another maker who uses microcontrollers to make original instruments. She has created a musical glove called the MINI.MU, which uses a micro:bit. It was inspired by the much more complex MI.MU glove, a gestural MIDI controller which was designed for the musician Imogen Heap. Helen took that idea and created a low-cost wearable instrument with a built-in speaker. The micro:bit's accelerometer can be used to trigger chiptune sounds, and the whole thing is available as a kit which can be sewed and coded by kids. The MINI.MU, along with Helen's other projects such as the beautiful 'sonic circuit sculpture creatures', can be found on her website.

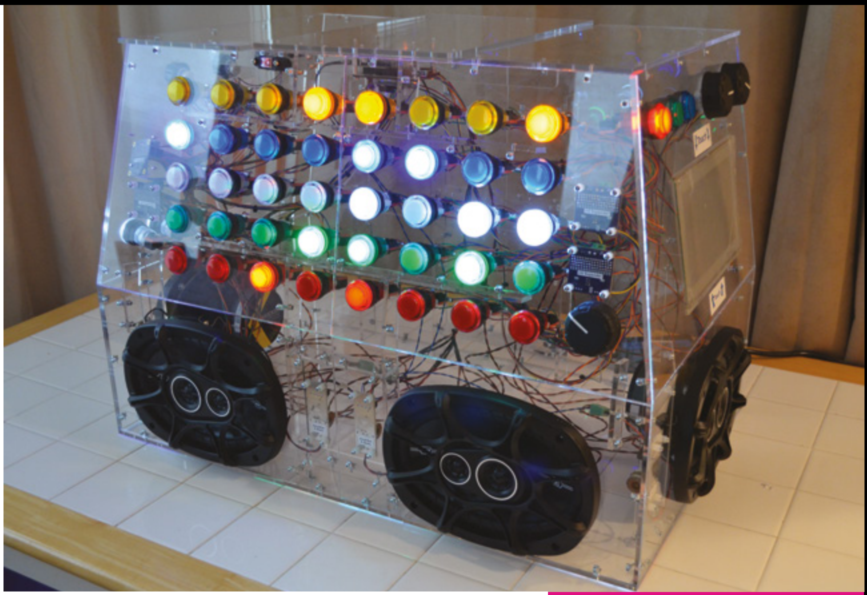
doitkits.com

Instrument Sematary

Lia Mice is another maker who likes to modify existing instruments, and puts them to use both in her live shows and in the studio. Her 'Instrument Sematary' project involves taking old, broken acoustic instruments and combining them into new, Frankensteinian creations. The project was inspired by the zombie animals in Stephen King's story *Pet Sematary*, and its message is that musicians and instrument designers should be aware of the environmental impact of their choices – recycling (or 'upcycling') is a common theme for a lot of the makers mentioned here. The double-headed harp-guitar, pictured, will be exhibited in London at the V&A's Digital Design Weekend, 21–22 September.

liamice.com





Monolith

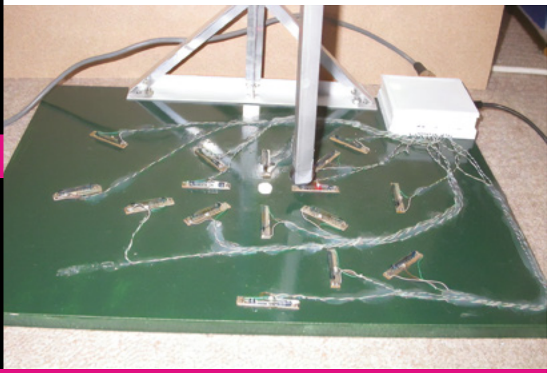
Take a look at the huge and ambitious 'Monolith' synth, created by Ross Fish (Moffenzeef Modular), Darcy Neal (Lady Brain Studios), Ben Davis (Malekko), and Paul Stoffregen (PJRC). The synth is based on a Teensy 3.6 microcontroller (basically a more powerful Arduino, capable of high-quality audio), with a clear acrylic case mainly fabricated on a laser cutter. The 40 arcade buttons act as a drum sequencer, while the two end panels produce bass lines, arpeggios, and textures. The instrument weighs around 40 kg, looks amazing, and was designed as an "interactive sound sculpture" to be exhibited at Maker Faires and other events. If you're looking to make your own synthesizer from scratch, the Teensy's intuitive drag-and-drop interface (which automatically generates Arduino code) is a great place to start.

pjrc.com/monolith-synth

Chaotic Pendulum

Some of the most fascinating homebrew musical instruments combine both mechanical and electronic elements, and a great example of this is Mike Cook's 'Chaotic Pendulum'. The design is based around a pendulum which is free to swing in two dimensions. The pendulum has a magnet on the end, and there are a number of magnetic switches (known as 'reed switches') laid out underneath it. When the pendulum swings close to one of the switches, the magnet on the end briefly closes the switch, triggering a note via the MIDI output port. The tiny amount of energy required to close the reed switch also causes a slight variation in the pendulum's path, creating a chaotic, ever-changing, mesmerising pattern. This project uses an Arduino to generate music, a subject that Mike literally wrote the book on: *Arduino Music and Audio Projects* is available to buy online and is a great reference. Mike's website is also a brilliant resource, as he publishes detailed build instructions for a lot of his instruments.

hsmag.cc/fvMipr



The projects mentioned here are just a tiny sample of all the amazing musical instruments being hacked, modified, or built from scratch right now, so get onto YouTube and find some more inspiration. If you want to start building your own instruments, there are lots of good places to start. If you're interested in circuit-bending, there is an excellent book on the subject by Nicolas Collins called *Handmade Electronic Music: The Art of Hardware Hacking*, and you should also check out Look Mum No Computer online. If you're not yet familiar with modular synthesizers, that's an excellent rabbit hole to go down.

For those who would rather build something digital, there are loads of great platforms: Teensy, Bela, and Raspberry Pi are all great ways to build your own audio inventions, and even an Arduino can be coaxed into producing (lo-fi) sounds using the Mozzi library. Also, don't discount the idea of designing instruments in software on a PC first – it's often a good way to test out your ideas before bringing them into the physical world.

If you're more inspired by mechanical or acoustic projects, you should look into the instruments commissioned by Björk for her *Biophilia* show. It's also worth getting some background knowledge of 'proper' musical instrument building, perhaps by binging on the satisfying guitar build/repair videos on StewMac's YouTube channel.

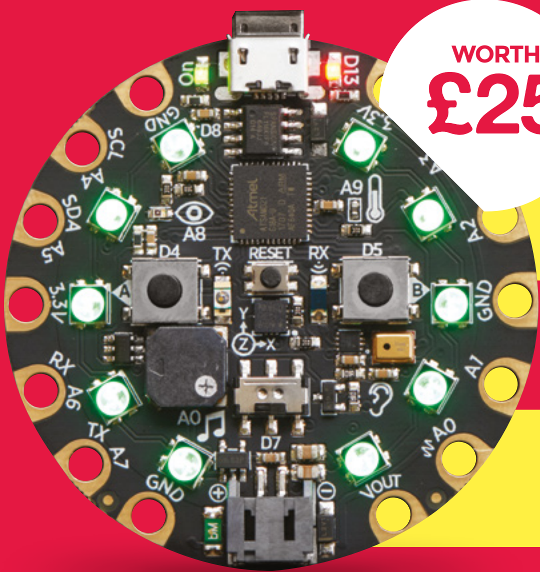
Whatever you do, make sure to have fun, and if you come up with something cool, share the details of your instrument online so that others can build on it in the future. □

Above The Monolith was a collaborative effort between four makers

Below Mike Cook's instrument uses chaos theory to generate MIDI notes

SUBSCRIBE TODAY

Subscribe for 12 months	Rolling monthly subscription
£55 (UK) **£90** (USA)	▸ **Low monthly cost** (from £5)
£80 (EU) **£95** (Rest of World)	▸ **Cancel at any time**
Free Raspberry Pi 3A+ with 12-month upfront subscription only (no Raspberry Pi 3A+ with rolling monthly subscription)	▸ **Free delivery to your door**
	▸ **Available worldwide**



WORTH £25

FREE!

Adafruit Circuit Playground Express

With your 12-month print subscription

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

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

Buy now: hsmag.cc/subscribe

How I Made

A FOOD-FORAGING C. ELEGANS ROBOT

Converting biology into computer science

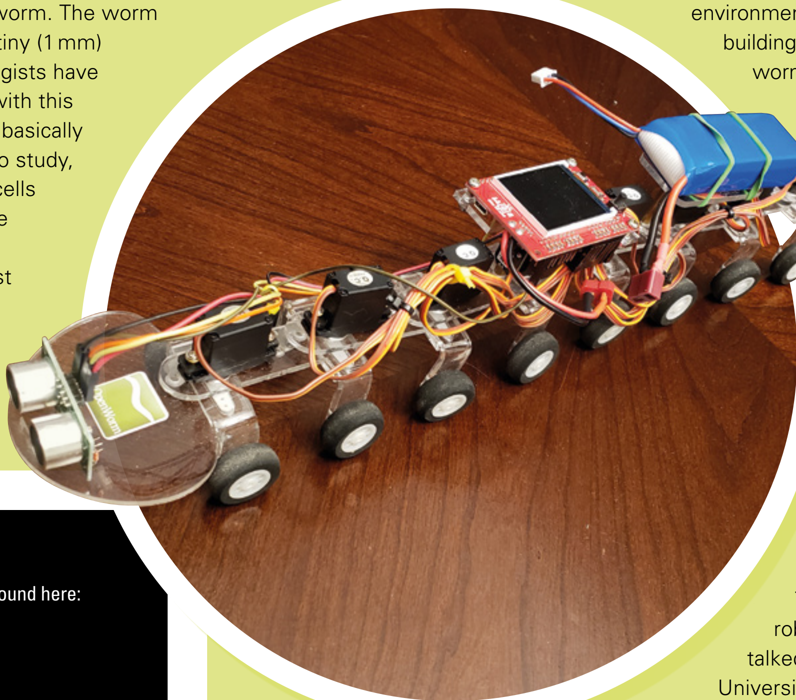
By **Tom Portegys**

I'm a member of the OpenWorm open science organisation (openworm.org). Our aim is to create a virtual worm. The worm is C. elegans, a tiny (1 mm) nematode. Biologists have been very fascinated with this worm for many years, basically because it is so easy to study, with only about 1000 cells – 302 of those cells are neurons (nerve cells). C. elegans was the first organism to have its genome mapped out.

Another goal of OpenWorm is to promote scientific awareness through crossover education in biology, robotics, and coding (a popular approach in many educational environments). For OpenWorm, this meant building robots that approximate the real worm. The downside of this is that C. elegans is, of course, not a robot. However, simulating the worm in robotic form manifests a tangible aspect that pure software models do not possess. Also, a worm robot sidesteps working with actual worms, which requires special conditions and equipment, such as lighting and microscopes.

In early 2017, some of us set about how to make a robot that tries to stay true to the biology of C. elegans.

There were other C. elegans robots around. For example, we talked to a researcher at the University of Leeds, who had made a tube-like robot: **hsmag.cc/bjCmdv**. We also were aware of a project by a former OpenWorm member that runs the worm nervous system, called the connectome, in a Lego robot: **hsmag.cc/ehJ0Zn**. We also were guided by a couple of scientific



Above The unpowered wheels let the robot wiggle forward

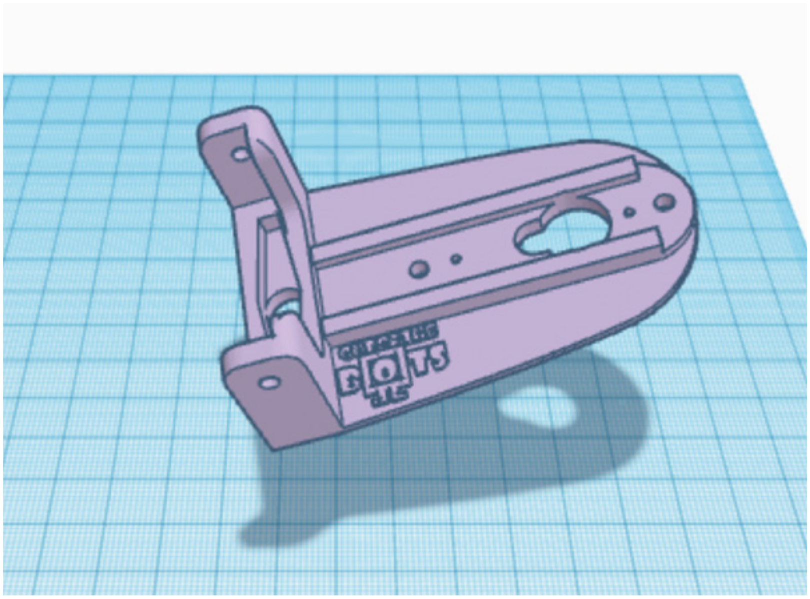
In motion

Videos of the robot can be found here:

Moving without sensor:
hsmag.cc/SdFcjJ

Foraging food to right:
hsmag.cc/gXzGfl

Foraging food to left:
hsmag.cc/mjrTTV



Above The worm body designed and ready to be 3D-printed

Future implementations

These could include:

- Touch response. C. elegans responds to touch on its nose by reversing direction.
- A Jupyter Notebook to provide a programming interface to the robot. This might also support software exercises related to C. elegans, such as how the neural network can be optimised to achieve sensorimotor coordination for touch response.

papers that proposed how the worm moves (locomotion), and forages for food. Since no existing robot incorporated both these ideas, it seemed like a good plan to build a robot that modelled them.

In the spring of 2017, Australian robot-builder Shane Gingell joined the project, and he and I proceeded to forge ahead. We ended up with two versions of the C. elegans robot. The first is a Raspberry Pi processor implementation, and the second is an ESP32 MicroPython system on a chip (SoC) implementation. Both implementations are a combined effort. Shane spearheaded the design and initial build, and I built the versions used for external demos, as well as supplying software to run the simulation.

LOCOMOTION

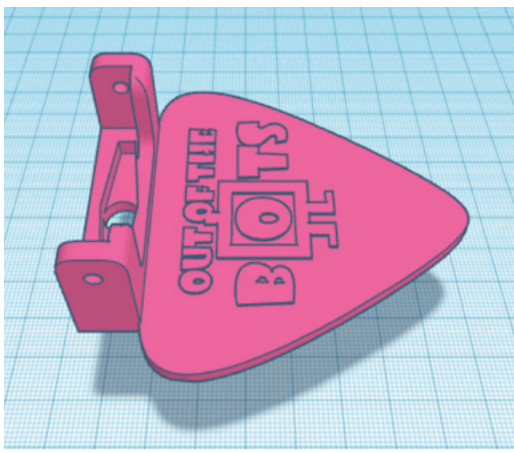
The first version was based on a Raspberry Pi board, and accomplished one of the goals: locomotion.

The robot consists of nine articulated segments, which is a simplification from the twelve segments in the real worm. Each is segment mounted on a pair of wheels. Locomotion is achieved, as it is in C. elegans, by moving in a snake-like manner that relies on surface friction. The wheels are hence not powered and exist to provide a suitable contact surface with the ground.

> Since no existing robot incorporated both these ideas, it seemed like a good plan to build a robot that modelled them

Each segment is a 3D-printed component that articulates with its neighbours via servos. The electronic components are mounted on platforms fastened to several of the segments. We both purchased and assembled Tarantula 3D printers to print the parts. However, considering that 3D printing services are typically easy to obtain, and in hindsight, dealing with the challenges of DIY printing, building, and operating a printer did delay the completion of the robot (personally speaking!).

The Raspberry Pi Zero W is a full-fledged microprocessor that included wireless communication capability. A PWM board →

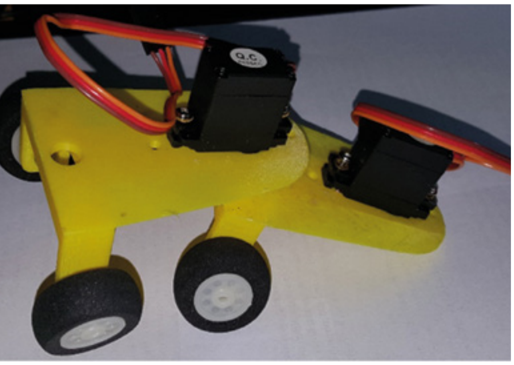


Above The head section of the worm holds the distance sensors

Resources

The 3D models, code, assembly instructions, and parts lists are available at: **github.com/openworm/robots**

Both implementations of the robot can be constructed by someone with moderate technical skills, after obtaining the parts from various vendors.



Above
One segment assembled and ready to be inserted into the worm

distributes power and controls signals from the Raspberry Pi to the servos. Each servo is capable of maintaining a specified angular position that translates to intersegment angular positioning.

We designed the parts and exported them as STL files so they could be printed on most 3D printers. The head is designed to support sensors for food foraging and touch, while segments support wheels and parts of the electronics.

The segments are articulated. A servo is mounted on the front top of the segment, with its geared shaft extending into an aperture in the next forward segment. An arm secured to the gear sits in a recession in the segment that transfers the angular movement of the gear into angular movements between the segments.

As previously noted, the locomotion of the worm was described in a paper that specified a fine-grained model of the worm's neuromuscular system. Unfortunately, the model entails a computation-intensive physics engine that is too demanding for the Raspberry Pi, hence the recording/replay procedure.

The robot replays a recorded sequence of segment angle positions, that are generated by an offline run of the model. Each entry in the recording is processed at a fixed interval of time, and contains a set of angles that are directly distributed to the servos. The anterior (forward) segments receive signals of greater magnitude than the posterior (rear) segments. The overall effect is to achieve an undulating forward motion that simulates actual worm locomotion. Sample recorded angular movement data:

```
[78,-0.63,-0.65,-0.62,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[219,-0.63,-0.65,-0.62,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[375,-0.63,-0.65,-0.62,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[531,0.63,0.00,-0.62,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[672,0.63,0.65,-0.62,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[812,0.63,0.65,0.00,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[937,0.63,0.65,0.62,-0.58,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
[1047,0.63,0.65,0.62,0.00,-0.55,-0.51,-
0.48,-0.44,-0.41,-0.37,-0.34,-0.30],
```

The recording is transferred to the Raspberry Pi via the `scp` (secure copy) command. To run the robot, the user logs in via SSH, and executes a Python script that initialises the servos and executes the recorded file line by line.

A short video of the robot in action can be viewed here: **hsmag.cc/TjaLNi**.

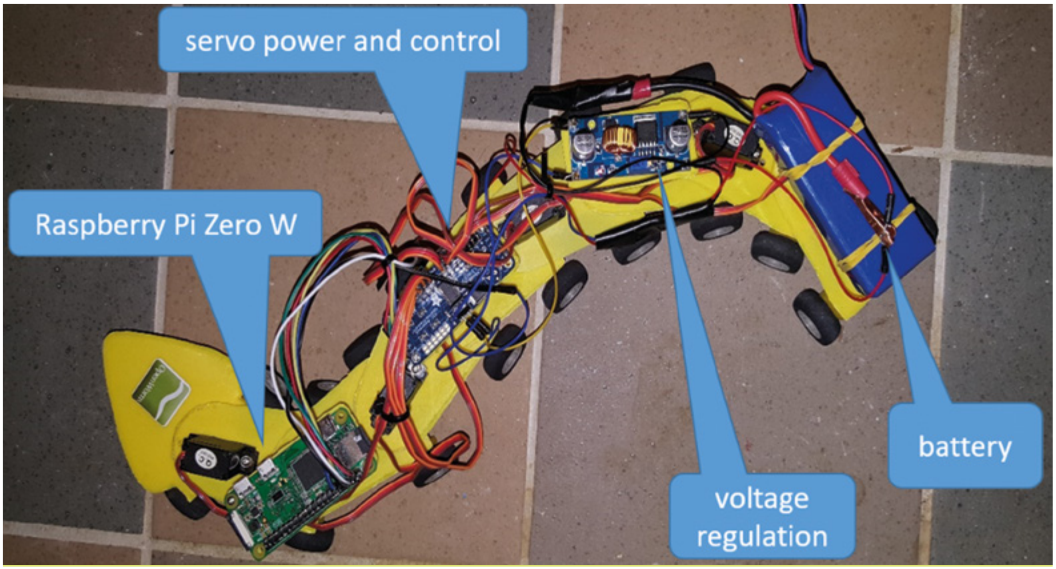
V2: FOOD FORAGING

The RPi version supported locomotion, however, we also wanted to have the robot forage for simulated food. C. elegans forages for bacteria in its environment by sensing and following salt gradients produced by the bacteria. For the robot, a sensor that can provide a proximity signal to a simulated food source was needed as a counterpart. A chemical sensor was not deemed to be feasible. Several other types of sensors were tried: sound, laser distance, and sonic.

> For the robot, a sensor that can provide a proximity signal to a simulated food source was needed as a counterpart

Unfortunately, we found it difficult to use the Raspberry Pi to both read the sensors and drive the servos. So, for the second version, a new processor was selected, the ESP32 system on a chip (SoC).

The ESP32 provides a MicroPython interface that can run numerous general-



Left The first iteration powered by a Raspberry Pi

purpose services, such as WiFi. It was also very beneficial that the board contains both control and power distribution connections for the servos, as well as connections to support other peripherals, such as sensors. This simplified the design by dispensing with the PWM and voltage regulator boards required by the RPi implementation. Another enhancement for the ESP32 version was the use of laser-cut body parts. Although the 3D-printed body parts remained fully compatible, the laser-cut parts made for a sturdier body.

An ultrasonic distance sensor was settled upon (after some stiff competition with a laser rangefinder) to allow the robot to forage for simulated food by seeking it out by proximity. The food source was simply then any object that can reflect sound, such as a cardboard box. This approximates how C. elegans follows chemical gradients in its environment to find the nearest food source. The worm with the ultrasonic sensor mounted on the head is shown at the start of the article.

The ultrasonic sensor has a wide (30+ degrees) view field. To achieve successful navigation toward the nearest object, a distance measurement is taken at 45 degrees from centre as the robot swings its head from side to side. An increment is added to the segment servo angles, depending on which side the object is to force a turn in that direction.

The following algorithm accomplishes locomotion combined with distance sensing:

```
turn_angle_delta = 0
for each step:

    # Set servo angles.
    for each segment:
        servo_angle = angles_array[step]
[segment] + turn_angle_delta
        set_servo(segment, servo_angle)

    # Check right sensor distance.
    if head swing moving right and head
angle is 45 degrees:
     right_distance = activate_sensor()

    # Check left sensor distance.
    if head swing moving left and head angle
is -45 degrees:
     left_distance = activate_sensor()

    # Determine turn direction.
    if right_distance <= goal_distance or
       left_distance <= goal_distance:
         print("food found")
         break
    else if right_distance > max_valid_
distance and
            left_distance > max_valid_
distance:
            turn_angle_delta = 0
            print("food forward")
    else if right_ping_distance < left_ping_
distance:
            turn_angle_delta = right_turn_
angle_delta
            print("food right")
    else if left_ping_distance < right_ping_
distance:
            turn_angle_delta = left_turn_
angle_delta
            print("food left")
    else:
            turn_angle_delta = 0
            print("food forward")
```

With this, we can study the movement of the worm as it seeks out food in its simulated environment.

Worm robot on the move

The C. elegans robot was demonstrated as part of a public program of events associated with the conference that was hosted by OpenWorm at the Sainsbury Wellcome Centre for Neural Circuits and Behaviour in London, UK on 31 January 2018. In May 2019, it went on display at the Ars Electronica Center in Linz, Austria.

FEATURE

Soundscapes

Mapping bird diversity with citizen science and open-source

Cameron Norris

@CameronSNorris

Cameron is a technology and communications specialist, passionate about the use of open-source hardware for social innovation.

Soundscapes to Landscapes is a citizen science project that uses open-source hardware – including AudioMoth, a low-cost, full-spectrum acoustic sensor, developed by Open Acoustic Devices here in the UK – to map bird diversity across Sonoma County, California.

By comparing the research of citizen scientists with environmental data gathered by NASA, the Soundscapes to Landscapes project aims to advance the use of remote sensing satellites for monitoring, predicting, and conserving animal diversity on Earth.

"It is recognised that the data collected by citizen scientists can be much larger and even more accurate than by paying a team to do the work," says Soundscapes to Landscapes project lead, Dr Matthew Clark, a professor at Sonoma State University.

Matthew and his team of volunteers capture 'soundscape' recordings at audible and ultrasonic frequencies, which are then used by experts within the birding community to identify bird species within a particular area. These bird diversity maps make it possible to test the accuracy of models that use algorithms to predict the distribution of bird species based on environmental data gathered by NASA's remote sensors.

351,692 MINUTES OF RECORDINGS

The data collected by these systems can be used to determine the physical structure of vegetation over vast areas, such as the heights of trees and plants, as well as the canopy chemistry, which can be used to assess ecosystem productivity and habitat quality.

"In my PhD research in the early 2000s, I explored these technologies in a wet tropical forest with the goal to map tree species and estimate forest structure and biomass. These were some of the first applications of these two types of sensors in tropical forests," says Matthew.

To complete their prototype phase in 2017, Matthew and his team of local volunteers partnered with the Pepperwood Preserve, a 3200-acre site that acts as a refuge for over 750 species of plants and 150 species of wildlife in North California's Mayacamas Mountains.

While at Pepperwood, they collected AudioMoth sound recordings from over a hundred different locations. The team also completed their species

SENSING SOUND

An acoustic sensor can be any combination of recorder, detector, microphone, or hydrophone, designed to detect and record sounds in the environment. They are often small, increasingly affordable, and can be deployed in the field for extended periods to monitor wildlife and its response to human activities.

Images
Soundscapes to Landscapes

Right
Citizen scientists place audio recorders in the field to capture bird calls

to Landscapes

distribution modelling code in R (an open-source software and programming language commonly used for data analysis) and developed bird call identification models and species distribution probability maps for nine different species of bird.

ANIMAL BIODIVERSITY MONITORING

Initially, the project used Android smartphones to capture audio recordings but quickly switched to

> **"WE WERE THRILLED WITH THE EASE OF USING AUDIOMOTHS"**

AudioMoth due to the hardware's simplicity and low-cost per unit. "We were thrilled with the ease of using AudioMoths," says Matthew. "We have also found the sound fidelity is much better than with the smartphone, which was designed to record human conversations, not environmental sounds."

Following success in Pepperwood and surrounding areas, Soundscapes to Landscapes received funding from NASA's Citizen Science for Earth Systems Program (CSESP), which put the project on track to map all of Sonoma County, a vast area of over 1760 square miles, by 2021.

Since the beginning of March this year, student interns and community volunteers have already collected recordings from more than a hundred sites, totalling over 175,000 minutes of audio. "We could not have had this productivity if it was not for the energy and passion of our new project coordinator, Rose Snyder, based at Point Blue Conservation Science," says Matthew. "Our team worked with Rose to create a whole new process for managing citizen scientists into teams, tracking AudioMoths, organising sound recordings, and identifying properties to visit."

Matthew believes that there is still work to do in regards to providing NASA-funded citizen science projects with more detailed guidelines on how to promote and run projects with the general public. "We worked with Sieve Analytics to create a new Citizen Science interface to ARBIMON," he says. "This will allow us to find soundscapes with potential

bird calls. Citizen scientists can then quickly validate if the bird call is really present or not."

Once that data is validated and labelled, it is passed to a research team at the University of California that is using deep learning to identify bird calls. "We are currently working on having a single convolutional neural network classifier that will identify 32 bird species in all of our recordings… currently 350,000 minutes in total," says Matthew.

Despite the project's growing support, there are still many areas of Sonoma County that are difficult for the team to access. "In the next field season, we will be stepping up our effort to mail out flyers to targeted property owners in these areas, and hope to send recorders out to these landowners for deployments on their properties," Matthew explains.

To find out more about the Soundscapes to Landscapes project and get involved in next year's field season, visit **soundscapes2landscapes.org**. →

WIDE REACH

'ARBIMON' stands for Automated Remote Biodiversity Monitoring Network. ARBIMON is a web-based bioacoustics analysis platform that enables anyone with an internet connection to access soundscape recordings. So far, the research platform has collected over one million recordings from monitoring stations in Puerto Rico, Hawaii, Arizona, Costa Rica, Argentina, and Brazil.

Above
Capturing bird calls in the fields

Below
Dr Matthew Clark and a team of volunteers conduct a survey in Sonoma County

AudioMoth

AudioMoth is a low-cost, open-source, full-spectrum acoustic sensor capable of recording uncompressed audio to a microSD card for environmental and biodiversity monitoring in remote habitats. It was designed by Andrew Hill and Peter Prince, two computer science PhD students at the University of Southampton, along with Alex Rogers, a computer science professor at the University of Oxford.

These palm-sized devices are currently being used to help researchers "move away from passive continuous recording by individual, expensive devices, towards using multiple low-cost, smart devices that flood large and inaccessible ecosystems," says Andrew.

In addition to monitoring wildlife populations by recording sounds made by target species at a sampling rate of up to 384kHz, AudioMoth can also be used to detect potential threats to protected habitats by listening for poachers' gunshots or the sound of chainsaws used in illegal logging.

FLYING MAMMALS

The Bat Conservation Trust is currently using AudioMoth technology to develop an automated end-to-end system for monitoring bats in Britain. The system includes an app that enables volunteers to send audio recordings to a central server for processing, algorithms to automatically detect and classify bat calls to species, and an online portal to share the survey results with volunteers and members of the public. On top of that, as much of the technology as possible will be open-source.

GUNSHOTS AND CHAINSAWS

"Until now, professional ecologists typically had been surveying wildlife with commercial equipment. The cost advantage of AudioMoth completely changes the science people can do," says Alex. "Previously, if an ecologist had a small budget, they could maybe only deploy three or four recorders. Now they can potentially deploy a hundred recorders, meaning different types of wildlife surveys can be conducted."

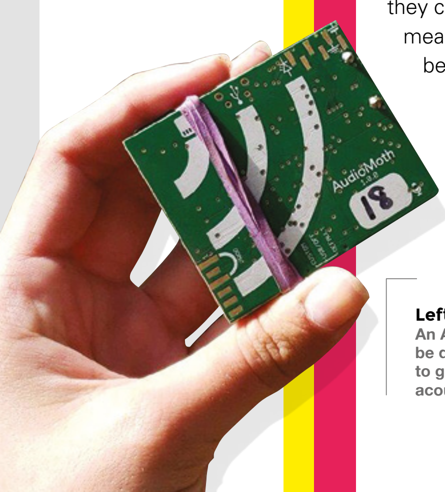
The system is built around an energy-efficient EFM32 Gecko 32-bit MCU, which includes up to 128kB flash and 16kB RAM. It also features an analogue MEMS microphone (similar to the microphone inside your smartphone) and preamplifier with adjustable gain. Power is supplied by three AA-cell batteries or an external 6V lantern battery, while an RTC (real-time clock) is used to keep an accurate track of time for recordings scheduled using the companion smartphone app.

Above The AudioMoth companion app can be used to schedule recordings, customise the gain and sample rates, as well as calculate the device's approximate remaining battery life

The companion app also enables users to customise the gain and sample rates, as well as calculate the device's approximate battery life with any given configuration. The size of a fully assembled AudioMoth is just 58 mm long, 48 mm wide, and 18 mm thick, which is slightly bigger than a box of safety matches. This means that over a hundred devices can fit into a standard 25 l capacity backpack.

"From the beginning, we were looking to create a minimal device we could run smart algorithms on to only record when hearing a sound of interest," explains Alex. "With the Gecko MCU, we can do nearly all the listening while the processor sleeps, then it can wake up to run the detection algorithms across a four-second sound buffer." This feature makes it possible to deploy AudioMoths in the field for up to twelve months, with each device listening continuously for twelve hours per day.

"We have a lot of deployments in remote jungles and forests with extremely limited internet access, but we are still planning to add low-power wireless connectivity to new versions of the device for alerting, streaming, and research purposes," adds Alex.



Left An AudioMoth ready to be deployed in the field to gather important acoustic data

Right The main mission of Soundscapes to Landscapes is to advance the science of global biodiversity monitoring from sensors in space

"THEIR TASK WAS TO SURVEY A SINGLE 1 KM SQUARE"

LAST SEEN OVER 20 YEARS AGO

The first field study using AudioMoth aimed to identify the only cicada species native to the UK, the Cicadetta montana, which was last seen in the New Forest National Park over 20 years ago. Cicadettas spend most of their lives underground as nymphs, emerging as adults every seven years or so, and the high-pitched call of the adult, at 14kHz, is out of the hearing range of most adults, making it difficult to search for in manual surveys. As a result, 87 AudioMoths were deployed in four New Forest locations for up to three months at a time from spring 2016 until early summer 2017.

Although no native cicada was identified during the study, the detection capabilities of the device were tested by playing cicada recordings captured in Slovenia (where the Cicadetta montana is commonly found) along with background noise captured in the New Forest. Impressively, the detection algorithm achieved a true positive rate of 0.98 and a false positive rate of only 0.01.

As AudioMoth is capable of recording the highest frequency bat echolocation calls in the UK, which come from the lesser horseshoe bat at over 100kHz, they also make ideal tools for organisations working to conserve bat populations. Since launching the British Bat Survey in 2017, the Bat Conservation Trust has freely provided AudioMoths to volunteers participating in the annual study each year.

This year's survey focused on discovering how much time is needed to collect the amount of data required to estimate population trends for rarer bat species in South West Britain. Volunteers received a survey kit containing an AudioMoth, a waterproof case, and memory cards for as many non-consecutive nights as possible between 1st July and 11th August 2019. At the end of the survey, participants are provided with a list of all the bat species that they detected and the data is used to determine the best survey methods for a large-scale roll-out of the British Bat Survey next year.

If you would like to get your hands on an AudioMoth for the 2020 British Bat Survey, visit **bats.org.uk** for more details. Or, if you'd like to build your own AudioMoth from scratch, visit **openacousticdevices.info** to find all of the open-source documentation you need to get started.

PRICE DROP

The bill of materials cost for the AudioMoth is around £35 per unit, but researchers can get their hands on working devices just as cheaply by joining online collective purchasing groups to bulk order devices from web-based PCB assemblers, such as CircuitHub. The most popular site so far is GroupGets, which has run several successful 'group buy' campaigns for the device, priced around £40 per unit.

Left Be part of a team of volunteers and hike to predetermined sites on public and private land to deploy sound recorders, and retrieve them after a few days

Below Volunteers are busy deploying AudioMoths throughout Sonoma County

HackSpace magazine meets...

Ugo Vallauri

Focusing on openness in the repair economy

Most people who like making things also like to take things apart. We love to see how things work, and the best way to do that is to break them, then put them back together again. It's like a puzzle game, but one in which you win a prize of an 'extra' nut or bolt.

However, this simple joy is at risk of being extinguished. Appliance manufacturers all too often use hard-to-find screws, or glue things together, meaning that it's getting harder to fix the things that you own. And in some cases, you're not even legally allowed to fix your own stuff.

Thankfully, there are a number of organisations fighting against this rising tide of waste, ignorance, and restriction, including the Restart Project, a London-based organisation campaigning in the UK and Europe to protect our right to repair. We spoke to Restart's co-founder, Ugo Vallauri, about the green economy, recycling, and why it's essential that we have the freedom to repair the stuff we own. →

Right ↗
Two advocates for re-use and recycling, at the Restart office in Brixton, London


restart
WALL·E

HackSpace The United Nations says that the UK is the second highest producer of e-waste per capita. What are we doing wrong?

Ugo Vallauri It's obvious that a country with the kind of economy of the UK would be one of the highest producers of e-waste. Yes, we are consuming too much and too fast, but we're also not very efficient, compared to some other countries, at collecting the products we no longer use for recycling. At the Restart Project, we first believe that we should consume more slowly than we do now, and that when products are no longer used by the first user, we should try to reuse them, by reselling them, giving them away. When certain products reach the end of their useful lifespan, we are extremely inefficient – around the world generally, but in the UK in particular – at collecting all kinds of electronic products.

This is particularly true for smaller products that a lot of people are still adding to their non-recycling waste, so they end up in landfill because they're small. Obviously, it wouldn't be possible to do this with a washing machine, but it's perfectly possible to do with a kettle or a toaster.

On average, around the world, we are still very inefficient at recycling electronics, and a high proportion of them simply don't get recycled at all – more than half of all electronics are not recycled.

On top of that, there is a misconception that recycling is always the best thing to do. Many critical raw materials used in smartphones aren't recycled at all, such as beryllium (used in connectors), gallium (in integrated circuits), indium (in touchscreens) and silicon metal (in microchips).

For other materials, the amount of recycling is extremely low, including graphite (used in batteries), neodymium (in microphones) and tantalum (in microcapacitors).

It gets worse than that. In the life cycle of a smartphone, for example, the iconic product of this generation, approximately 80% of the overall greenhouse gas emissions linked to the whole life cycle of the product have to do with the manufacturing stage. It becomes very obvious that the only thing we can do to reduce the environmental impact of using all these products is to make fewer of them. Obviously when you're talking about a fridge or an oven, there's a significant amount of energy used in the use phase, but when you compare it with small battery-powered products, during the use phase you're going to be using maybe 15% of the overall carbon emissions of the overall life cycle of that product.

Which means that the only environmentally sound option is to keep using, repairing, reusing, and extending the lifespan of that product as much as possible, before it reaches the recycling phase.

> "At the Restart Project, we first believe that we should **consume more slowly than we do now**"

HS I've seen the argument that modern appliances are more energy-efficient, and therefore it's more environmentally responsible to upgrade your laptop, for example, for something that uses less energy.

UV That is absolutely wrong. This only applies to things like your fridge and other larger appliances. All evidence points to exactly the opposite. If you were to look at the energy consumption in use for a laptop, in order to make up for the proportion of environmental damage caused by that product during the manufacturing phase, you would have to keep a laptop in use for ten years to make up for that.

Of course, when you're buying a new product, it makes sense to consider energy efficiency as one of the criteria. It is a useful way to understand how much money you might save during the use phase, but this does not take into consideration how much energy and materials are employed during the manufacturing phase.

This is why we are campaigning for the right to repair, which among other things involves pushing for better regulations that prevent manufacturers from making products that do not meet minimum requirements of repairability, such as being near-impossible to disassemble for repair; providing as many spare parts as possible for as long as possible; and providing access to repair manuals and instructions to facilitate repair for everyone, and not just for professional repairers, which for a lot of products no longer exists widely in our communities.

I don't mean smartphones, tablets, and other IT products – they've generated a thriving repair community. But for many other products that people tend to bring to repair events like our Restart Parties or to Repair Cafés, such as kettles, toasters, printers, headphones, hair straighteners, hair dryers, and many other products.

HS Yes! We visited a Repair Café earlier this year. It's amazing how many simple appliances are designed to stop you from fixing them when they break.

UV That's exactly why we're campaigning, with our partners across Europe, to push for a European-wide movement for right to repair. We know that for many products there simply is no regulation whatsoever limiting the power of manufacturers to just do whatever makes the most sense to them, irrespective of the environment.

For some products, some regulation exists, and we've actually pushed the option of initial minimal right to repair and repair provisions in European legislations. From 2021, new televisions, electronic screens, dishwashers, and fridges that →

Right

On average we in the UK each produce 24.9kg of e-waste per year (the worst offenders are the Norwegians) come to the EU market will have to have a list of spare parts available for at least seven years after the product is taken out of the market, as well as access to repair manuals and design that allows accessing those spare parts without further damaging the appliance that you're trying to fix, which unfortunately is very frequent. You need to be able to fix the things you own: if you can't fix it, you don't really own it.

Unfortunately, the manufacturers have been able to water down our proposals. Access to the wider set of spare parts and to repairer information will be restricted to what are called 'professional' repairers, which would leave a lot to be desired. For example, this means that if you need to replace the light bulb within your fridge, you as an ordinary citizen might not be able to buy the light bulb for your fridge, but a professional repairer might be able to do that for you. Which doesn't make any sense to us.

Obviously there will be manufacturers that provide you with better access, but we need regulations to be more ambitious. This is a good start, but we need it to be more universal in terms of what you can access and in terms of what kind of products are included in these measures. The biggest target for our work in campaigning on the right to repair is to have smartphones be regulated as well.

HS If access to spare parts is limited to professionals, what constitutes a professional? Who decides who's a professional person?

UV The regulations state that member states will have to have a register of professional repairers, which includes repairers that are insured for the work that they do. Where such a register doesn't exist, it will be up to the manufacturer to accept or reject a request for spare parts or access to manuals for professional repairs.

So, it will be up to the manufacturers to determine whether someone is a professional or not, which is fairly worrying, because it could potentially create the conditions for only authorised repairers performing certain types of repairs agreed by the manufacturers, which is pretty much what's happening right now.

HS Still, a start though, to have the need recognised?

UV Absolutely, it's a start, and that's why we want to use this precedent to push for much wider adoption of measures that translate this concept to all kinds of other products. The case of smartphones is quite shocking – we're talking about globally over 1.5 billion new devices sold per year. We see that there's a very recent trend of people wanting to keep products in use for longer, and we're pretty sure that if conditions were simplified for people to extend the useful life of a product by making it easier to access repair when a button goes wrong – or a speaker, or a camera, or a charging port, which we see quite frequently, or more reliable battery replacement – this trend will be more prevalent, and we will see a longer lifetime for this product, which will be win-win for the environment, and also for providing a more vibrant local economy for repair.

> "You need to be able to fix the things you own: **if you can't fix it, you don't really own it**"

Repair jobs are the original green jobs. They provide jobs for individuals in local communities, as opposed to continuing to buy and upgrade to new products that rely on poor working conditions elsewhere in the world. At a time of climate emergency, we have a chance to put things right by promoting more repair and reuse of products and less of a throwaway economy.

HS That's European lobbying that you've been doing – have you done much with the UK government? I know that you've won the backing of the Green Party and the Liberal Democrats. How have you been getting on with Labour and the Conservatives?

UV Last October, in Manchester, we gathered 59 repair activists from across the UK who took part in Fixfest, our national gathering. We co-wrote the Manchester Declaration, which has been signed by over 40 community repair groups and 15 organisations supporting it, asking for all barriers to repair to be taken down. [The Greens and the Liberal Democrats already support this.] We received endorsement for the Declaration from our first two Labour MPs: Helen Hayes in Brixton and Tulip Siddiq from Hampstead and Kilburn.

We have groups around the country that are actively reaching out to their MPs from all parties to invite them to community repair events and ask them to endorse the Declaration, so we foresee a lot more cross-party support on this front.

If it is true that the UK government is aiming to do better than what Europe is trying to achieve, then this is a very good opportunity to ask any manufacturer of any product that's sold in the UK market to make spare parts and repair manuals available to everyone, right away. There's an opportunity to lead in their area, if only there were the willingness to do it. That's what we're working towards.

The UK, Germany, and Italy have historically had large manufacturing sectors, and perhaps it shouldn't come as a surprise that when the regulations for refrigerators, washing machines, dishwashers and the like were discussed, there were three countries that were seen to be blocking the regulations: the UK, Germany, and Italy. And so we put →

Left ↙
Repair activists from around the world will convene at the global Fixfest happening in Berlin from 20–22 September

Above ↑
The Restart Project practices what it preaches – there's an old printer in the office that would have been thrown away years ago anywhere else; here it's being kept alive by a Raspberry Pi

pressure specifically on these countries, to prevent further watering down.

And by the way, this is not just in the interest of consumers; it's also in the interests of repairers of all kinds, because repairs will be faster, better, and easier. And it's also in the interests of the work done at the end of the product's life, by the recyclers: if a product is designed for disassembly, it's designed so that more of the components that still work can be removed. It's in everyone's interest.

Proper regulation will mean a level playing field, so manufacturers can compete without undermining consumer rights and the environment. Until this happens, they will always try to defend the status quo. Also, they use the argument that repairs shouldn't be allowed because it is not safe for other people to access the product they have made. We think that, actually, lack of documentation access to the official precise repair documentation might create cases where repair becomes less safe, because people might use alternative sources that might be less reliable.

HS I remember that argument from John Deere [US maker of tractors, combine harvesters, and other agricultural equipment], that allowing non-authorised repairs would be dangerous.

UV The John Deere case is interesting because it opens up the other big issue, which is that of software. In the United States, a big part of the fight for the right to repair is intellectual property. In the case of John Deere, the question was: 'can a company lock access to a product by installing a piece of software that will make it impossible for a product repaired by a third party to be used again unless it's verified again by the manufacturer?' This is a really big issue and, more broadly, extended firmware support and security updates for products is one growing cause of concern that's not yet tackled by regulation either. At a time when people already want to keep their smartphones

Left Toasters are often built to be thrown away, even when they can be made whole with a tiny fix

for longer – or TVs for that matter – how can we justify manufacturers making two or three years of security updates and then abandoning products? It's contributing to premature obsolescence.

Why is it that Google doesn't provide ten years' worth of security updates to the phones that it provides software updates to? It has the capacity to do that, so why doesn't it?

Could we think of devices with a kill switch, so the moment the smart element of the device is no longer secure, we can turn it off and use it as a non-smart device, the way we used to use devices before, without compromising safety or security depending on the product?

> Use your amazing skills to contribute to a future **where people and planet are a lot happier**

These are some of the trends we have seen. Software is increasingly the reason people give up on a product, and it's not repairable unless it can be substituted with a free and open-source alternative, which is the case for quite a lot of products.

It's a real decision by a manufacturer to stop supporting a product. This is not about conspiracy theories; it is a real decision and there is a plan to say no more support after a certain day. There are consequences to this. However, that said, the trends we've seen in computers preventing any hardware upgrade – because RAM is soldered on the motherboard, SSD drives are soldered as well, or they are provided for some models with proprietary connectors so they cannot be found in the market, and a lack of ports – is certainly something that we see as extremely worrying.

At the end of the day, we're talking about creating an economy that makes more sense for people and the planet. To repair and reuse can be an essential part of this. No data would contradict what I'm about to say that it is environmentally more efficient to extend the use case of the current product, rather than recuperating some materials and some components to make a brand new product to be sold again in the market. You can create a lot more jobs by having proper maintenance and reuse economies to prevent unnecessary recycling and unnecessary throwing away of products that are already manufactured.

Make use of whatever resource you already have; keep using and making the most of the component that you already have – this should be at the heart of every tinkerer. Use your amazing skills to contribute to a future where people and planet are a lot happier, by preventing waste and ensuring that everything is used to the fullest that is possible. □

FEATURE

UMBRELLA

Looking for inspiration? The brolly's got you covered

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.

One of the oldest pieces of handheld accessories, the umbrella was invented over 4000 years ago in Mesopotamia. Archaeologists have found evidence of umbrellas in ancient digs all over the world, including Egypt, Greece, India, and China. The word 'umbrella' is Latin in origin, and means 'shade' – the contraptions were originally designed to provide protection from the sun. The rainproof versions, made with treated paper, popped up in China only about 1500 years ago. The brolly made its way into Europe thanks to the Silk Road – it became popular with the Romans as a guard against the heat.

Back in the day, umbrellas used to denote wealth and status. History is littered with text and drawings of kings, priests, and high-ranking officials being accompanied by umbrella-bearers. Not only do we carry our own umbrellas now, but it's also no longer regarded as an accessory just for women, like it was for several centuries.

Frenchman Jean Marius is often credited for introducing the folding umbrella to Europe in 1710. Manufacturing advancements and access to newer, and relatively cheaper, raw materials made the umbrella affordable to everyone later in that century. Leather made way for lighter textiles such as silk, which became quite popular. These days you can find many different types of umbrellas; some are easier to operate, and others can double up as a walking stick. Several hundred million of these are made in the Chinese provinces of Guangdong, Fujian, and Zhejiang which house over a thousand umbrella factories.

The use of the umbrella was rather unpopular in England in the 1700s. It was thought of as a tool for a poor man who couldn't afford the luxury of a carriage. It was serial philanthropist, traveller, and writer Jonas

> "The rainproof versions, made with treated paper, popped up in China only about 1500 years ago"

Hanway who helped establish the umbrella as part of everyday Londoner's attire around 1750, after enduring much ridicule. The prejudice didn't evaporate until the 1850s. But gradually, umbrellas gained acceptance, and stands for them became a common item of hallway furniture.

Over time, umbrellas have unfurled their way into many aspects of our lives, and are used for all sorts of creative purposes. They are a popular form of decorating exotic drinks and cocktails; the security detail of former French President, Nicolas Sarkozy, has also used a Kevlar-coated one as a defensive device. Here are some less exotic, but equally creative hacks for your portable hand canopy.

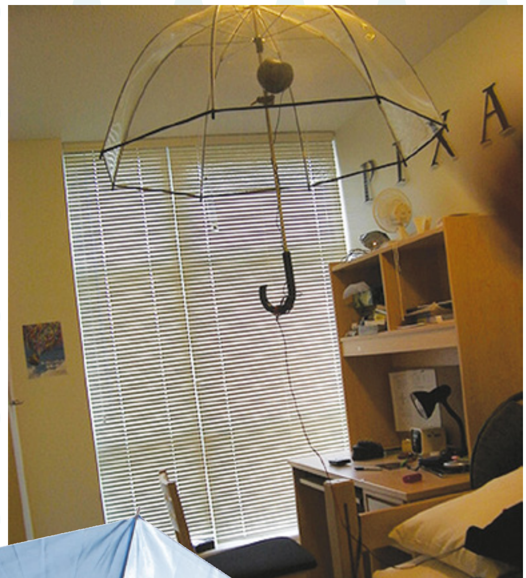
SOUND DOME

Project Maker
Matthew Ebisu

Project Link
hsmag.cc/ZRhOU0

Matthew is fascinated by umbrellas. On the lookout for ways to use them to their full creative potential, he came across the umbrelAudio unidirectional speaker, and decided to create one of his own using an umbrella. Instead of a plastic bowl, Matthew uses a dome umbrella as a parabolic unidirectional speaker that's able to direct audio, such as music, to one specific location. You'll need several bits and bobs to replicate his build. The laborious process involves creating and attaching a mount for a battery-powered speaker, and running wires from the speaker to your audio source. Matthew used a couple of hooks to hang his umbrella speaker on his dorm's light bowl fixture to get the true *Cone of Silence* feel. He suggests you place the speaker at least 5" from the top of the umbrella to maximise the sound space. The original hack is almost a decade old, and Matthew suggests using Bluetooth speakers now to save a lot of hassles soldering wires and keep the build clean, adding that he'll now "probably go with a programmable field sound array to disperse the sound across the entire umbrella, as opposed to just one location."

> "The original hack is almost a decade old, and Matthew suggests using Bluetooth speakers now"



Above
Besides detailed build instructions and videos of audio tests, Matthew's also shared details about the physics of the sound domes

SMARTPHONE CAMERA EXTENDER

Project Maker
LISA RICHARDS

Project Link
hsmag.cc/pyZAFk

Left
One weird trick to spam social media with pictures of your face

Lisa runs a DIY blog – she was out of ideas to make something for her husband for Father's Day when she saw a picture of a camera extender.

"I needed something that telescoped in on itself, but was strong enough to support the camera," writes Lisa. A little research later, she decided to use an old umbrella. She first removed the canopy from the umbrella and then used pliers to strip away all the metal support ribs to leave just the main pole. She then drilled a hole in the top to fit in a screw. Lisa made sure the hole was long enough to leave some threads of the screw exposed. Here she screwed a mini tripod camera holder. After shaving off some of the plastic cover of the umbrella, she was done. Remember though, the DIY selfie stick doesn't have a mechanism to trigger the camera's shutter. Lisa suggests using a camera app with a long enough timer to allow you to set up the shot. If you can live with this minor inconvenience, Lisa says you can save yourself quite a lot of money, even if you had to buy everything for the hack.

> "THE DIY SELFIE STICK DOESN'T HAVE A MECHANISM TO TRIGGER THE CAMERA'S SHUTTER"

BACKPACK COVER

Project Maker
Agatha Lee

Project Link
hsmag.cc/JwhsKp

Agatha's backpack bore the brunt of the rainfall as she and her son walked to the Handmade Movement Fair in the pouring rain with a small umbrella. That's when she decided to upcycle an old unused umbrella into a waterproof cover for her backpack. The first step is the trickiest, and involves removing the canopy from an umbrella without any damage to the fabric. Agatha used a kids' umbrella that had a thread that connected the canopy to the ribs and was relatively easy to dismantle, but your mileage may vary, depending on the type of umbrella you use. It was fairly simple from here on. Agatha is a textile artist – she details the entire process of sewing a tube around the canopy with a needle and thread, and some chalk and measuring tape. She then uses a safety pin to pass a length of elastic through the tube, before sewing the ends to complete the cover. □

Above You can even use the umbrella's original fastener to fold the cover when it's not in use

UMBRELLA PARACHUTES

Project Maker
Brian Wente

Project Link
hsmag.cc/eHzixx

Brian loves making parachutes from old damaged umbrellas. He starts by cutting out the canopy from the umbrella using a box cutter – in a similar fashion to Agatha's (above). If you do it carefully enough, you can get it done without cutting off the tip, which then comes in handy to tie a length of nylon cord. Brian's canopy has eight tips, so he takes a cord that's about 1.5 times the radius of the umbrella, and doubles it into eight strands before tying off one end into a knot. He then cuts the other end and attaches each of the eight strands to the tips of the canopy. If you don't have tips, you can knot each end, and stitch them to the canopy with a sewing machine. Finally, attach a piece of weight. Brian suggests using a heavy weight for a good show. Now gather your chute, roll it up, and throw it as far up in the air as you can. □

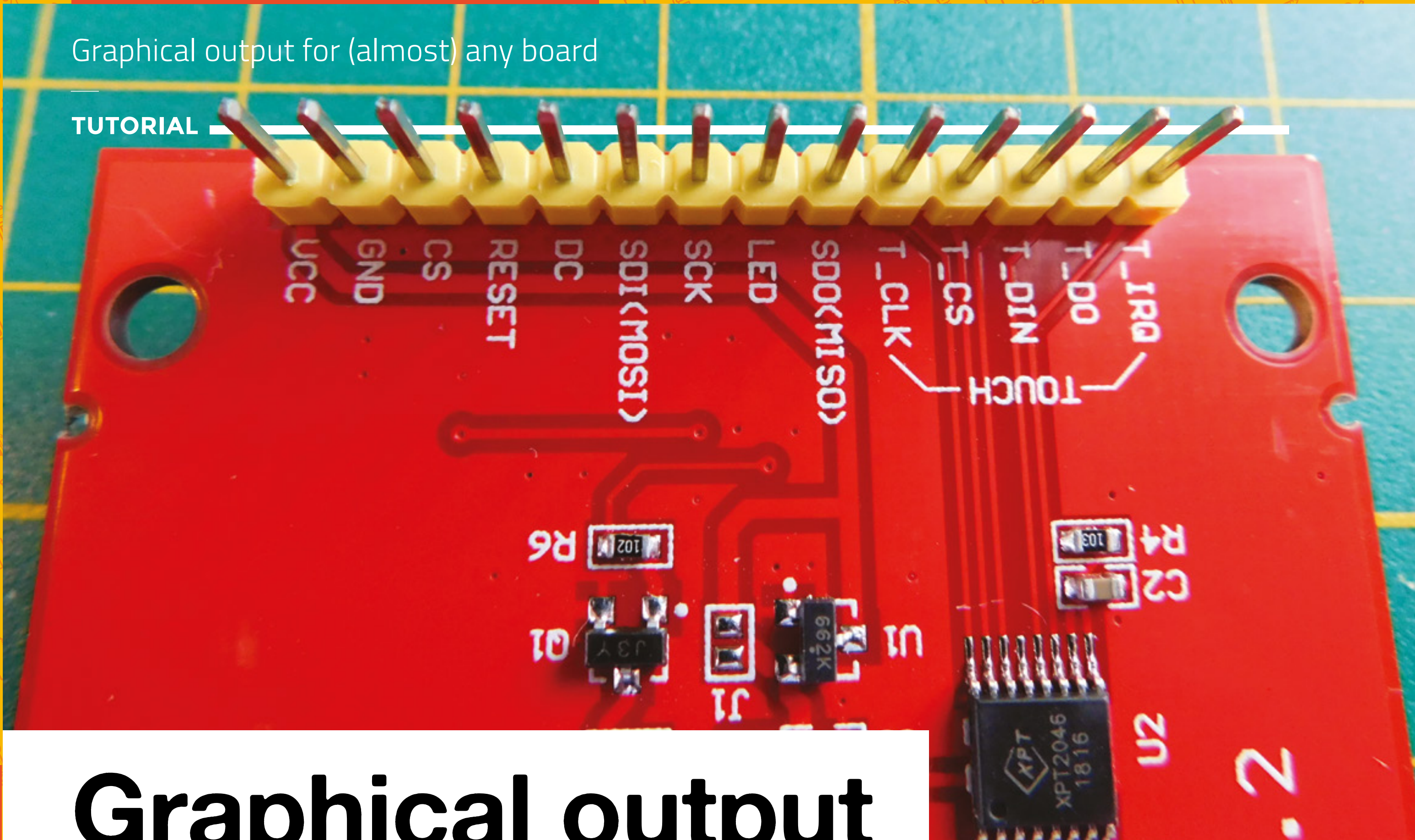
Left Brian created one of these for his daughter's *Hunger Games*-themed birthday party

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



Graphical output for (almost) any board

Add an ILI9341-powered screen in Arduino and CircuitPython

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 Your TFT board should have the pins labelled to make it easy to connect up to your microcontroller

Sometimes you need a little bit of graphics or text output for your projects and, for this, TFTs are a great solution – they're cheap, available in a range of sizes, and well supported by maker hardware. TFT stands for thin-film transistor – essentially, TFTs are liquid crystal display (LCD) screens with fast refresh rates. The main thing you need to know when looking to get one is the driver chip it uses. There are a few that are easy to use; in this tutorial, we'll look at ones driven by the ILI9341 chip. The driver should come on the TFT board, so we don't need to worry about wiring the two together. Instead, we just need to connect the module to our microcontroller.

This connection uses SPI (plus a couple of other wires), so the first thing we need to do is to connect the four SPI pins to an SPI bus – it's quite high speed, so it's best to make sure it's on a hardware SPI. Most microcontrollers have one of these, so look up where yours are. The following connections go from the relevant pins on the microcontroller to the labelled pin on the TFT screen. We did this on an Adafruit Grand Central, so the numbered pins are the ones we used.

- MISO (Grand Central pin 50)
- MOSI (Grand Central pin 51)
- CLK (OR SCK) (Grand Central pin 52)
- CS (or SS) (Grand Central pin 53)

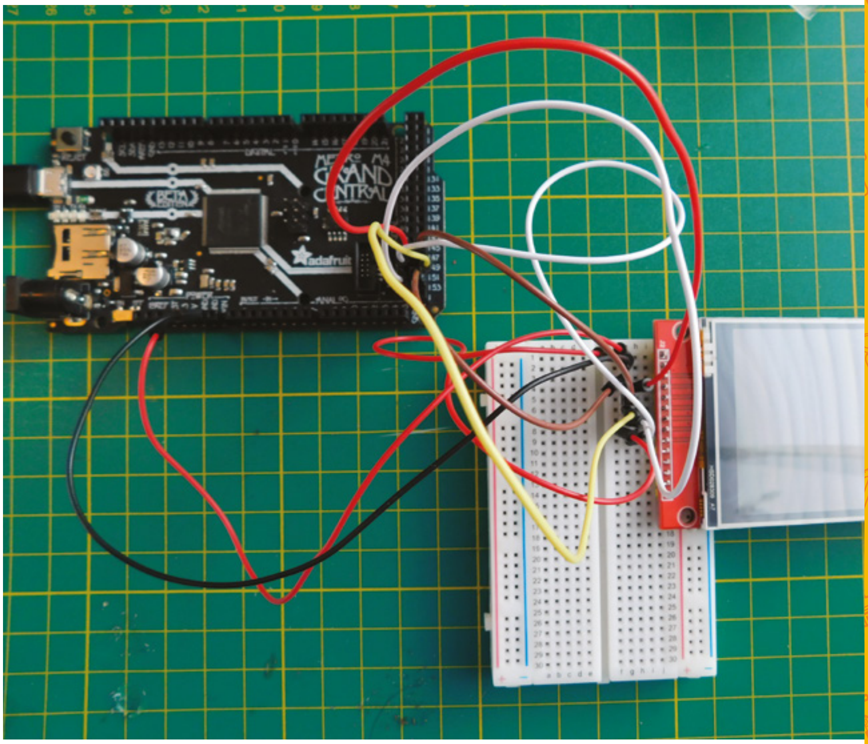
The following aren't SPI pins, but also need to be connected. As they're not part of SPI, it doesn't matter which pins you connect them to, provided they're digital IO. The pin numbers we've used could easily be changed.

- DC (Grand Central pin 49)
- RST (Grand Central pin 48)

That's all there is in terms of physical setup.

Let's now take a look at what code we need. The Grand Central can use either Arduino or CircuitPython, so let's take a look at how to use it in both these languages. First, we'll look at Arduino.

You'll need to install the Adafruit GFX library – that's a high-level graphics library that can work with a range of displays. You'll also need the low-level Adafruit-ILI9341 library that handles this specific driver chip.



In the File > Examples menu, you'll find Adafruit ILI9341 > graphics test. This is, as you may guess, a test for this particular TFT. There are a few configurable options on this. Above the setup method, you'll need the following code:

```
#include "SPI.h"
#include "Adafruit_GFX.h"
#include "Adafruit_ILI9341.h"

// For grand central. Other boards might need
different pins
#define TFT_DC 49
#define TFT_CS 53
#define TFT_MOSI 51
#define TFT_CLK 52
#define TFT_RST 48
#define TFT_MISO 50

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS,
TFT_DC, TFT_MOSI, TFT_CLK, TFT_RST, TFT_MISO);
```

With that saved, you can upload it to your board and you should see various different graphics tests appear on the screen. That's the screen running. What you do with it is up to you. If you have a look through the example code, you should get an understanding of how to use the GFX library to display whatever graphics you want.

GETTING SNAKEY

Now let's try with CircuitPython. First you'll need to flash your board with CircuitPython 4.1.0 or later.

There are two sections to the code. The first part sets up the display. You'll need to use this whenever you're using this screen. It is as follows:

```
import board
import displayio
import adafruit_ili9341
from time import sleep

displayio.release_displays()

display_bus = displayio.FourWire(board.SPI(),
                                 command=board.D49,
                                 chip_select=board.D53,
                                 reset=board.D48)
                                 display = adafruit_
ili9341.ILI9341(display_bus, width=320,
height=240)
```

Above The wiring is the same whichever language is used

This creates a display object that you can draw to. For more information on what to do with this, take a look at the CircuitPython tutorial that builds on the display. However, for the purposes of this article, we just need a simple test to make sure everything's wired up properly and working. For this, you can enter the following code under the above:

```
splash = displayio.Group(max_size=10)
display.show(splash)

color_bitmap = displayio.Bitmap(320, 240, 1)
color_palette = displayio.Palette(1)
color_palette[0] = 0xFF0000
bg_sprite = displayio.TileGrid(color_bitmap,
                   pixel_shader=color_palette,
                   x=0, y=0)
splash.append(bg_sprite)

while True:
      pass
```

If everything's worked properly, the display turns red.

That's how you get your ILI9341 display wired up and running in both languages. There are pros and cons to both. You'll get much better performance from the Arduino language, but it can be quicker to get started with CircuitPython. Your choice may also be determined by the hardware, as almost all hobbyist boards run Arduino, while a smaller (but growing) number run CircuitPython.

Control a screen with CircuitPython

Help otters frolic through the fields on a PyPortal

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.

The displayio framework first appeared in CircuitPython 4.0.0, but the performance was slow and it could only really be used to display static images. However, in version 4.1.0 the performance has improved vastly, and it's now possible to use this language to display animations. Let's take a look at how this works.

We've used a PyPortal for this, but other CircuitPython devices should work. If your device doesn't have a screen built-in, you can attach one (see the page 76).

Displayio is made up of multiple levels:

Display This is what controls how the screen is connected to the microcontroller. It displays a Group.
Group This pulls together the graphical elements into a scene. It's made up of other Groups and TileGrids.
TileGrid This contains one or more images that can be displayed (but aren't necessarily all displayed at the same time). They're used to control Bitmaps.
Bitmap This is a grid of pixels and associated colours. Bitmaps are the lowest-level primitives that your images are built up from.

This all sounds a bit abstract, so let's build up an animation and see them all in action. Firstly, we'll need some images to build up our bitmap. You can draw your own if you like, but we took a look at **opengameart.org** to find some we could use; we came across an adorable otter at **hsmag.cc/UfAfne** – it's the **otters.png** image you need. You can download this as a PNG, but we need a bitmap, so you'll need to open it in a graphics program and save it in BMP format. We need this file as **otters.bmp** in an **images** folder on the CircuitPython device.
If you have a look at this image, it's made up of 16 smaller images in a 4×4 grid. Each image is 96 pixels wide and high. The following code will display a running otter on the screen:

```
import board
import displayio
from time import sleep

display = board.DISPLAY
group = displayio.Group()

with open(“/images/otters.bmp”, “rb”) as otters_
img:
    otter_bitmap = displayio.OnDiskBitmap(otters_
img)
    otter_tile = displayio.TileGrid(otter_
bitmap,pixel_shader=displayio.ColorConverter(),
width=1, height=1, tile_width=96, tile_height=96)
    group.append(otter_tile)
    counter = 0
    direction = 1
    while True:
        for image_num in range(4):
            counter = counter + (3 * direction)
            otter_tile[0] = image_num + 8
            otter_tile.x = counter
            display.show(group)
            sleep(0.2)
        if counter > 320 or counter < -96:
            direction = direction * - 1
            otter_tile.flip_x = not otter_tile.flip_x
```

As you can see, adding a display is simply a case of pulling it from the `board` module. If not running on a board with a built-in screen, you'll need to set this up more fully. See the page 76 for details.

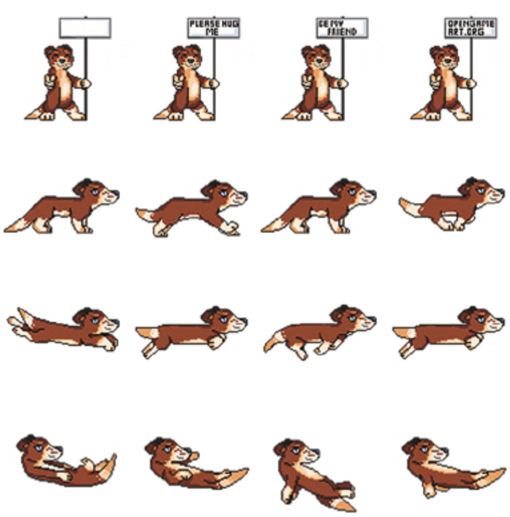
A bare group can also be created with a simple line. Most of the graphics setup is done with the lines:

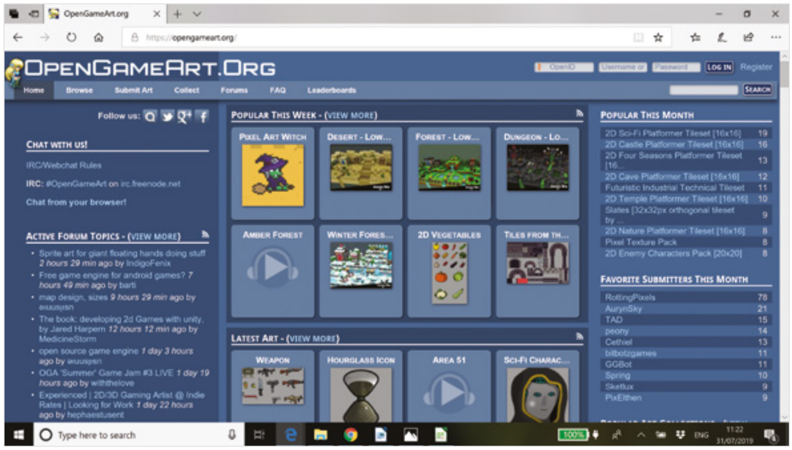
```
with open("/images/otters.bmp", "rb") as otters_
img:
    otter_bitmap = displayio.
OnDiskBitmap(otters_img)
    otter_tile = displayio.TileGrid(otter_
bitmap, pixel_shader=displayio.
ColorConverter(),width=1, height=1, tile_
width=96, tile_height=96)
    group.append(otter_tile)
```

This opens the **otters** bitmap and uses the `file` object to create a `bitmap` object. The `TileGrid` object takes a few more parameters:

otter_bitmap is the bitmap that the grid is taken from.
pixel_shader: Bitmaps contain a grid of colours; as we'll see later, these aren't necessarily actual RGB colours, so we need a way of converting between the bitmap and the display colour. This is the `pixel_shader`, and in this case we can use the built-in one.
width and height define how many tiles will be displayed in the TileGrid when it's on the screen. As we only want one of the otter images on a screen at any one time, this is '`1`' in both cases.
tile_width and tile_height are the size of each tile in the source bitmap.

At this point, we've got all our images loaded into memory, and sliced up, but we haven't actually put any of them on the screen. We do that in the `while` loop, particularly in these three lines:






```
otter_tile[0] = image_num + 8
otter_tile.x = counter
display.show(group)
```

The first line is probably the oddest. We've got our TileGrid called `otter_tile` that only displays one tile (at index '`0`'), but contains 16 source tiles from our bitmap. We have to decide which of these source tiles to put on the screen. This is in a `for` loop that iterates between the numbers 0–3, and we want the displayed image to rotate between images 8–11. We also have the variable `counter` that we increment and decrement to move the x position of the TileGrid along the screen. Finally, we have to display the group in order to make our changes visible.

If you upload this code, you should see an otter running backwards and forwards across your screen. However, it will be in a white square on a black background. Otters don't run in white squares, so the next step is to get ours running joyfully across a natural background.

ENTERING THE FOREST

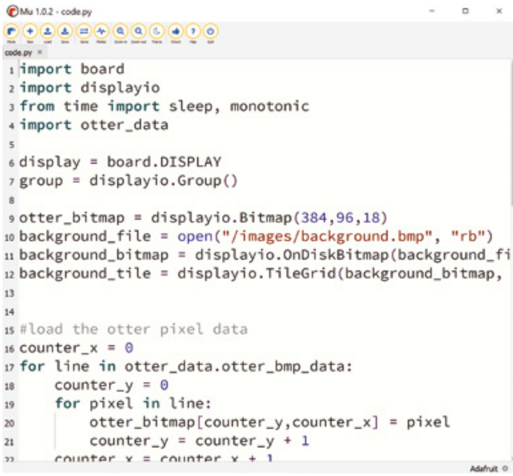
Our source image has a transparent background, but this gets lost when we convert it to a bitmap image. We need a way of importing our image which preserves this. The alternative way of bringing images into displayio is by setting the data points in the bitmap directly and creating a palette.

Each pixel in a bitmap is stored as a single number. The palette then links these individual numbers to an actual RGB value. As such, it uses far less memory because there's only one number per pixel rather than three (one for each RGB value).

We need to process our image in the right format. It's a bit CPU-intensive to do this on the CircuitPython device, so we'll do it on our main computer using Python. You'll need Python 3 with NumPy and Python Imaging Library (PIL) installed – we recommend using Anaconda, but you can also →

Above
OpenGameArt is a fantastic resource for graphics for embedding in your projects (whether game-related or not)

Left
Our lo-res otter gives us a good hit of nostalgia for 16-bit consoles



Right The Mu editor is the easiest way of programming CircuitPython – some other text editors cause problems

install them manually (or via your package manager if you're on Linux).

The code for the conversion is:

```
import numpy
from PIL import Image

#NumPy magic to convert image to Python list
img = numpy.asarray(Image.open('otters.png')).
tolist()

pallette = []
outbitmap = []
strippedbitmap = []
outpallette = []

#convert array of RGBA values to bitmap and pallet
for i in img:
  line=[]
  for j in i:
    try:
      colour = pallette.index(j)
      line.append(colour)
    except ValueError:
      pallette.append(j)
      colour = pallette.index(j)
      line.append(colour)
  outbitmap.append(line)

#strip transparency & convert to hex
for colour in pallette:
   outpallette.append('0x%02x%02x%02x' %
(colour[0],colour[1], colour[2]))

#strip down to just the running otter
for x in range(96):
   strippedbitmap.append(outbitmap[2*96+x])

print("otter_bmp_data =", strippedbitmap)
```

```
#need to print the pallette data differently as the
hex values are stored in strings
#and will be outputted with quote marks unless we
do this
print("otter_pallette_data = [",)
for colour in outpallette:
   print( colour, ",")
print("]")
```

This opens a file called **otters.png** (located in the same directory that you're running this script from) and outputs, to the terminal, the Python data structures holding the bitmap and palette. We pushed this straight into a Python file by running it with the following:

```
python convert.py > otter_data.py
```

This will create a PY file that we can copy to our CircuitPython device and import – but before getting to that point, let's take a look at what the script does.

We won't dwell on the line that does some magic from NumPy and PIL to create a 2D list containing the RGBA (the final A is the Alpha or transparency) values for each pixel. It just uses some built-in methods to do the hard work.

We then have to loop through this and, for each pixel, see if that colour is already in the palette and, if it is, link that pixel to the index of the colour. If it isn't, add the colour to the palette, then link it to the index of the colour.

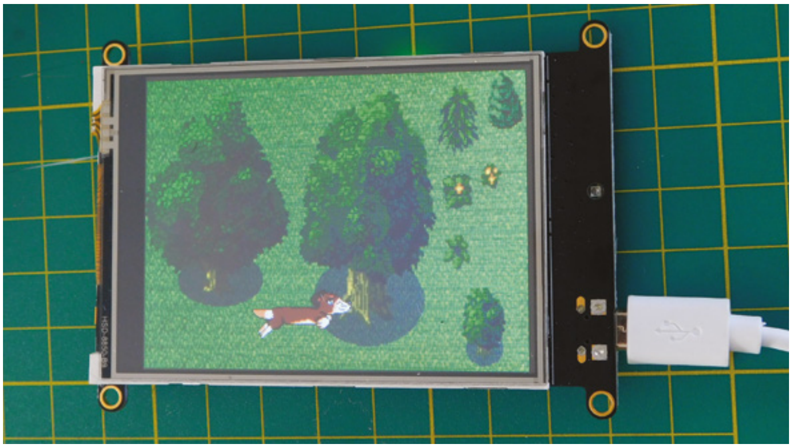
> This gives us our transparent image, **but we need something for it to be transparent against**

We finally have to convert the RGB values, which are stored in lists to a hex string, and strip out the parts of the image that we don't want (as we're pushing the memory limits of the PyPortal as it is).

This gives us our transparent image, but we need something for it to be transparent against. We used the trees and bushes images from **hsmag.cc/rtasZA**, but scaled to be 320×240 pixels to match the dimensions of the screen. We saved this as **background.bmp** in the **images** folder on the PyPortal.

The final image for our running otter is as follows:

```
import board
import displayio
from time import sleep
import otter_data
display = board.DISPLAY
group = displayio.Group()

otter_bitmap = displayio.Bitmap(384,96,18)
background_file = open("/images/background.bmp",
"rb")
background_bitmap = displayio
OnDiskBitmap(background_file)
background_tile = displayio.TileGrid(background_
bitmap, pixel_shader=displayio.ColorConverter())

#load the otter pixel data
counter_x = 0
for line in otter_data.otter_bmp_data:

    counter_y = 0
    for pixel in line:
        otter_bitmap[counter_y,counter_x] = pixel
        counter_y = counter_y + 1
    counter_x = counter_x + 1

#load the otter pallette data
otter_pallette = displayio.Palette(18)
counter_colour = 0
for colour in otter_data.otter_pallette_data:
    otter_pallette[counter_colour] = colour
    counter_colour = counter_colour + 1
otter_pallette.make_transparent(0)
otter_tile = displayio.TileGrid(otter_bitmap,pixel_
shader=otter_pallette, width=1, height=1, tile_
width=96, tile_height=96)

group.append(background_tile)
group.append(otter_tile)

counter = 0
direction = 1

otter_tile.y=144
while True:
    for image_num in range(4):
        counter = counter + (9 * direction)
        otter_tile[0] = image_num
        otter_tile.x = counter
        display.show(group)
        sleep(0.2)
```



Above The otter running across our PyPortal

```
    if counter > 320 or counter < -96:
        direction = direction * - 1
        otter_tile.flip_x = not otter_tile.flip_x
```

As you can see, this loops through the data structures our script created to add the pixels to the bitmap and the colours to the palette. The only bit the script didn't do is extract which colour on the palette relates to the transparency. We can tell from looking at the data that it's the first entry in the palette, so we use the following code to mark that as a transparent colour:

```
otter_pallette.make_transparent(0)
```

Once the transparent bitmap is created, we can use it in a TileGrid as usual.

As we're now using two TileGrids in our group (one for the background and one for the image), the only thing we need to be careful of is the order they're in. Displayio will put them on the screen in the order they are in the group, and we want our otter to appear on top of the background, so we append that to the group after we add the background. You should recognise the rest of the code from the initial test.

That's all there is to it. Displayio is a powerful way of controlling the graphics on your screen, and now runs fast enough to control animations. This otter should run at about ten frames per second on a SAMD51 device. That's just about enough to make animations look like animations rather than stuttering images. If you reduce the size of the animation, you should be able to get it to run even faster.

You now know enough to take this and embed it in your own projects. You could use this to display information relevent to your project, add a little graphical spice or create an added bit of interest in a prop or costume.

Whatever you choose to do, let us know about your makes via Twitter **@HackSpaceMag** or email **hackspace@raspberrypi.org**.

Let's learn LoRa!

Explore LoRa and LoRaWAN and transmit temperature and humidity to an online dashboard

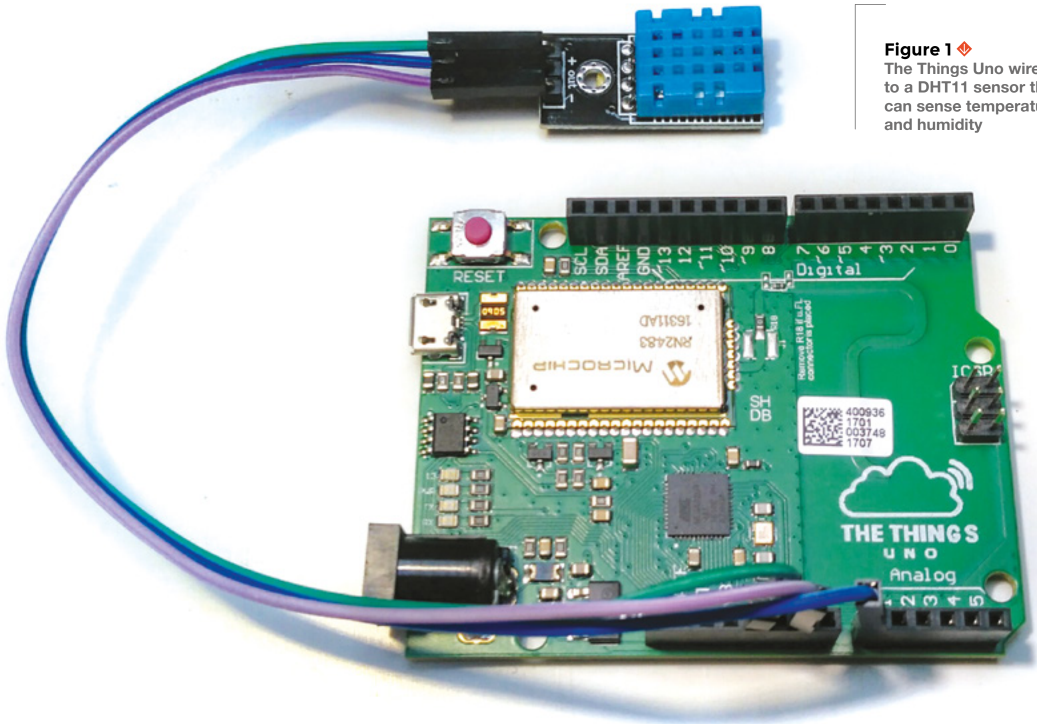


Figure 1 The Things Uno wired up to a DHT11 sensor that can sense temperature and humidity

Jo Hinchliffe

@concreted0g

Jo is a contributor to the Libre Space Foundation, 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, milling machines, and CNC kit!

It seems that the terms LoRa and LoRaWAN are everywhere at the moment, but what are they? LoRa is a platform for sensors to communicate wirelessly over long range; LoRaWAN is essentially the same, but when the receiver receives something from a LoRa sensor device, commonly called a 'node', it acts as a 'gateway', sending the information onto the internet. In this tutorial, we're going to work through some simple LoRaWAN activities and connect a LoRa node to 'The Things Network', a crowdsourced network of gateways. This enables us to receive data from a node and transmit some data across the internet to a nice dashboard displaying our data.

We are going to work with The Things Uno, which is essentially an Arduino-shaped board that has the LoRa communications chip built into it. We can also program The Things Uno using the Arduino IDE, so the first thing is to download and install the latest Arduino IDE software from **hsmag.cc/APNJVV**.

To test that The Things Uno board is working, let's upload a simple program to check the board. Connect your The Things Uno to your computer using the micro USB cable. In the Arduino IDE, click Tools > Board, and then check it's set to 'Arduino Leonardo'. Next, click Tools > Port, and select the port that includes the label 'Arduino Leonardo' to ensure the Arduino IDE is communicating with the correct port.

Next, click File > Examples > 01.Basics > Blink, and then click the verify button (looks like a tick on the top left of the screen), and then click the upload button (the right-pointing arrow button next to the verify button). All being well, after a few seconds your The Things Uno should have a flashing LED that is connected to pin 13 on the board (one of the four green LEDs next to the micro USB port).

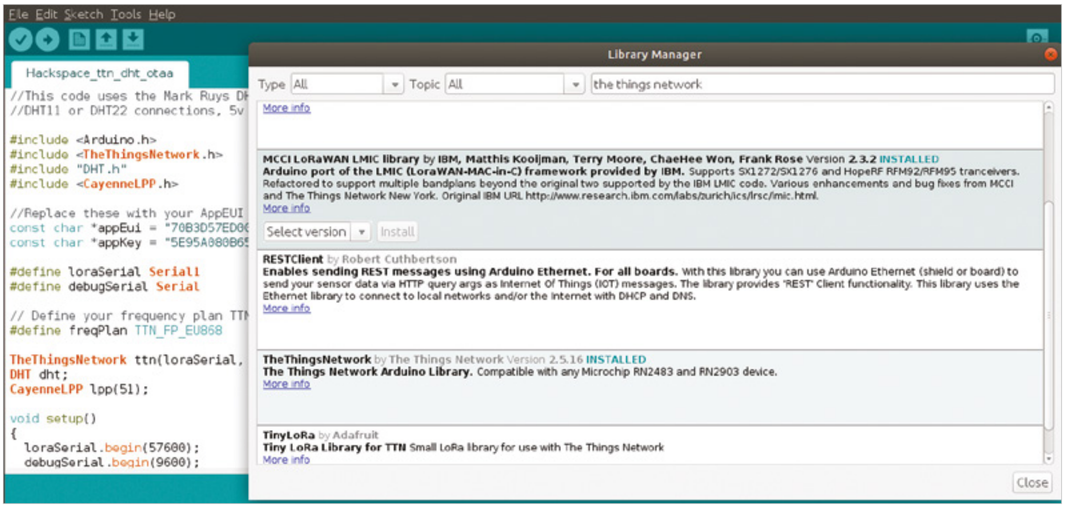


Figure 2 Using the Arduino IDE library manager to install the libraries we need for our project

PUBLIC NETWORK

The Things Network is a community-hosted network that consists of gateways connected to the internet. LoRaWAN devices, in our case a The Things Uno, can be received by any gateway, and their data packets are then forwarded to an account registered by the device owner on The Things Network website. The website application can be set up to integrate or forward those pieces of information to other systems, allowing the user to create a visual web dashboard, a phone application, an SMS alert, an email, or other options triggered or populated by the data from the device or devices in the field. The Things Network website has a map of gateways – check and see if you have one locally that you may be able to connect to.

Next, we need to install some libraries we are going to use in this tutorial. We'll install two of them using the Arduino IDE libraries manager, and download and manually install one from the internet. Open the Arduino IDE and then click Tools > Manage libraries (**Figure 2**). The first library we are going to install is called 'The Things Network', so type that into the 'filter your search' bar at the top of the library manager. You should find a library whose description begins 'The Things Network by Johan Stokking, Ludo Teirlinck…' – select this library and click Install. Repeat the above process, searching for 'cayenne LPP' to install a library called 'CayenneLPP by Electronic Cats'. Finally, to install the third library, we need to download it from **hsmag.cc/pEDXUY**. Click the large green 'Clone or download' button and then click the Download ZIP option. Once downloaded in the Arduino IDE, click Sketch > Include Library > Add .ZIP Library, and then navigate to where you downloaded the zip file, and select it.

> "We are going to work with The Things Uno, **which is essentially an Arduino-shaped board that has the LoRa communications chip built into it**"

GOING LOCAL

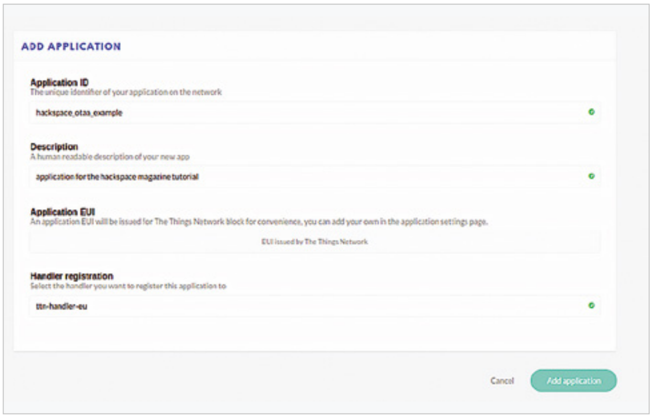
The next job is to upload an example sketch from The Things Network library we just installed. Click File > Examples > TheThingsNetwork > Device info in the sketch that is open – we need to make one small change before we can use this sketch. The Things Uno frequency for Europe is 868MHz, so we need to replace some text. Edit the sketch so that the 'REPLACE_ME' is replaced with 'TTN_FP_EU868'. Readers in other regions will need to replace it with the example that matches the frequency available in your region – it can be found on a sticker on the reverse of your The Things Uno.

Double-check your board is still connected and set to Arduino Leonardo, and the port is correct. Verify and upload the device info sketch to your The Things Uno. Once uploaded, you need to open the serial monitor in the Arduino IDE, this can either be opened by clicking the Magnifying Glass icon at the top right-hand side of the screen or clicking Tools > Serial monitor. In the serial monitor after a few seconds, you should see some details appear that are unique to your The Things Uno – copy and paste these details into a text document somewhere on your computer for later use.

We've now got the hardware set up and configured, it's time to take a look at the networking side of things. This is what gives us some where to send our data to. →

YOU'LL NEED

- **The Things Uno**
- **DHT11 or DHT22 temperature and humidity sensor**
- **Some breadboard connector wires**
- **Micro USB cable**
- **Access to a LoRa gateway** (see tutorial for details)



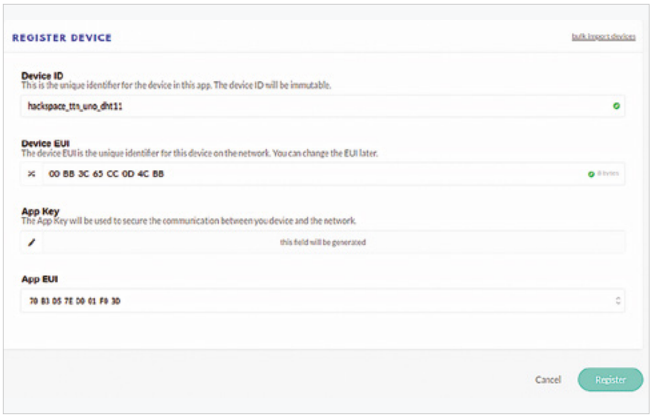


Figure 3 Adding an application to our The Things Network account

Figure 4 Registering a device into an application on The Things Network is essentially introducing The Things Network to our The Things Uno so that they are connected and enabled to communicate with each other

TO THE THINGS NETWORK

We'll use The Things Network as the glue to hold our sensor together with dashabord (which we'll look at in a bit). Navigate to **hsmag.cc/BtGluJ** and register an account. Once registered and logged in, you should see a link for 'console' in a drop-down list when you click your username. Navigate to the console and you should see two large icons: one that says applications and one that says gateways. We are (hopefully) going to rely on you being in range of a gateway, so we are interested in setting up an application: click the Application icon.

An application, in terms of The Things Network, can be thought of as the area to which your devices or nodes (in this case, your The Things Uno) will send their data. It is here that the The Things Network will choose where to send and what to do with the data it receives. An application can receive data from multiple nodes or devices and can also be integrated into other online services that allow you to do things with the data (for example, send a text message when a temperature gets too high, populate a dashboard, send key information to an online spreadsheet).

For now, we are going to create one simple application to receive data from our The Things Uno, which will be the humidity and temperature from our DHT11 sensor. Click the 'Add application' button in the top right-hand corner and give it an application name – note that these have to be in lower case and also have to be unique, so if you try 'test' for example, you will probably find when you try to add the application, it has already been used. As instructed in the second section, add some human-readable text to remind you what this application is; for example, 'HackSpace tutorial temperature and humidity example'.

The last two input boxes should be as we want them, with 'Application EUI' set to 'EUI issued by The Things Network', and 'Handler registration' set to 'ttn-handler-eu'. Leave these as they are and click the turquoise 'Add application' (**Figure 3**) button in the lower right-hand side of the page.

The application should now be created and you will be pushed on to the Application Overview page. If you scroll down this page, you should find a section called 'Devices' which will show there are no registered devices. So let's add a device, which will be our The Things Uno, so that our hardware can connect to this application. On the upper right-hand side of the devices box, click 'Register device'. In the resulting Device Registration page, give the device a device ID and then copy the 'Dev EUI' from the text document we made earlier when we got the device information off our The Things Uno via the device info sketch. Leave the App Key field on this page as it is (set to be generated by The Things Network) and click the turquoise 'Register' button in the lower right-hand corner (**Figure 4**).

> For now, we are going to create one simple application **to receive data from our The Things Uno**

You should now end up on the 'Device Overview' page for the device you just registered. There is a lot of information on this page, including the activation method (which should be OTAA) and the various keys that the device has or needs to communicate with the application. If we scroll down to the

QUICK TIP

Use The Things Network website map to see if you are close to any LoRa gateways.

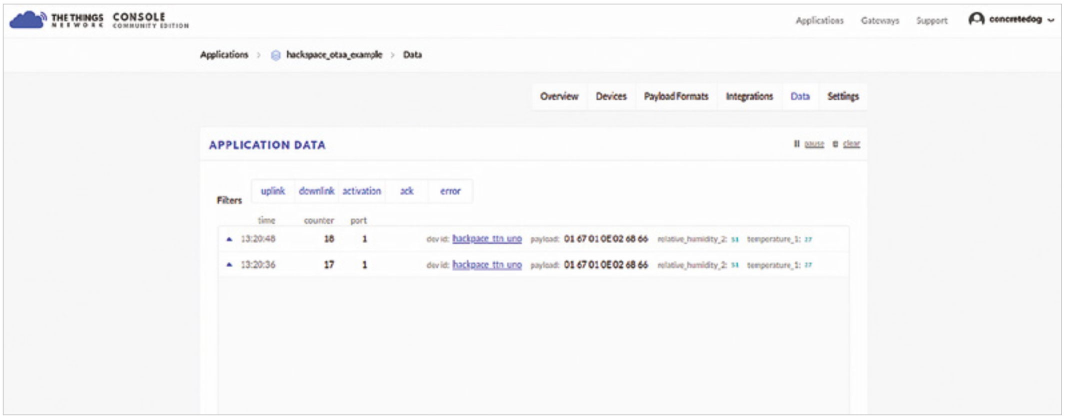


Figure 5 Success! Data from our device successfully being received by our application on The Things Network

bottom of this page, we should see a box called 'Example Code'.

IT'S ALL IN THE CODE

Rather wonderfully, this is a snippet of code containing the two key pieces of information an Arduino sketch on our The Things Uno needs to connect it to our application on The Things Network. Copy these (either select and right-click and select 'copy', or press the copy button in the upper right-hand side of the box) and paste them into a text document or a blank Arduino sketch. Before we move on to the next part of the tutorial, we are going to make one last change to the application we have made on The Things Network. Return to the Application Overview page – navigate here by clicking 'Applications' in the upper right-hand side of the page near your profile name – then select the application we just created. Once back in the Application Overview, click the 'Payload Formats' tab on the upper right-hand side. On the resulting page, you should see a box called Payload Format, and it should show 'Custom' in it. Click on this box. In the drop-down menu, there should only be one other option, which is CayenneLPP; select this and then make sure to click the Save button in the lower right-hand corner of the page.

LET'S GET CONNECTED

Connecting our DHT11/22 sensor board to The Things Uno is pretty straightforward. Connect breadboard wires between the DHT11/22 and The Things Uno 5 V and GND pin sockets. The data pin on the DHT11 sensor needs to be connected to pin A0 on The Things Uno (as seen in **Figure 1**).

Returning to the Arduino IDE, we will now upload the sketch for our sensor to The Things Uno; having made some changes and added the keys, we need to allow it to communicate with the application on The Things Network. Download the sketch from **hsmag.cc/issue22** and open it in the Arduino IDE. There are only a couple of changes we need to make. The first is to check the frequency plan is correct for our The Things Uno; this is the same bit of code we replaced earlier in the device info sketch. In our code, it is set as the 'TTN_FP_EU868' European version and will only need changing if you are using the US frequency plan.

The second change is that you will see a section in the code which is similar to the code we copied from the 'Example Code' box on the Device Overview page on The Things Network earlier. (It's →

> The data pin on the DHT11 sensor needs to be connected **to pin A0 on The Things Uno**

SPICY MESSAGES

Cayenne is an IoT platform by a company called myDevices. CayenneLPP (Cayenne Low Power Payload) is a format for data packages over LoRa that allows for some key types of sensors to be integrated into the Cayenne IoT platform simply via The Things Network. Put simply, if we can send sensor data in a CayenneLPP format, a lot of the work to unpack this data and present it in a straightforward and readable way is done for us in The Things Network and the Cayenne myDevices environment.

QUICK TIP

Remember, an application on The Things Network can support multiple devices – perfect for large, remote sensor array projects!

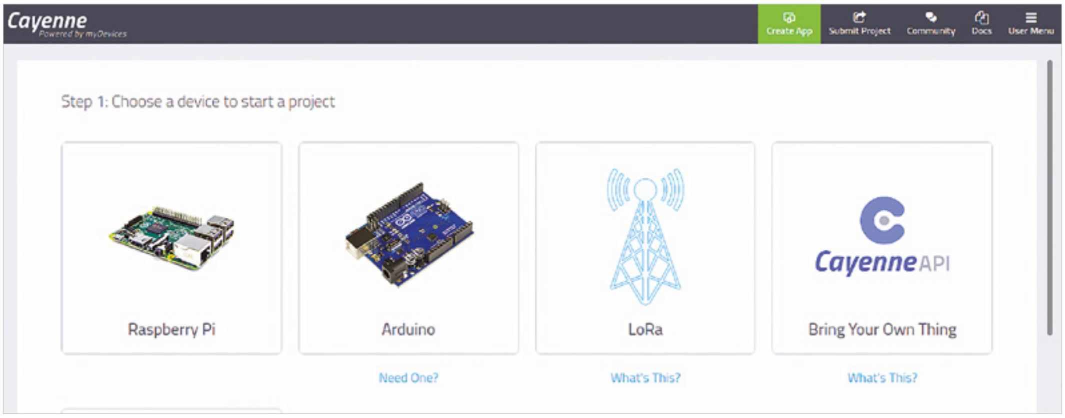


Figure 6 Beginning to set up a dashboard on the myDevices site

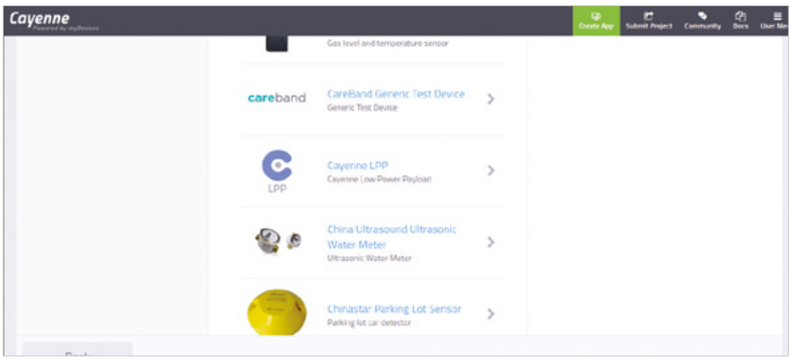


Figure 8 Selecting the CayenneLPP options for our dashboard

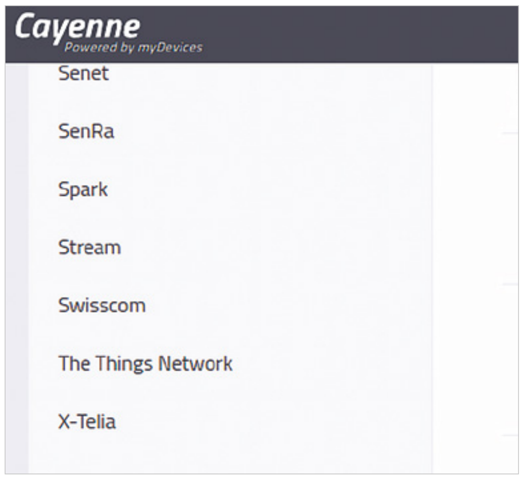
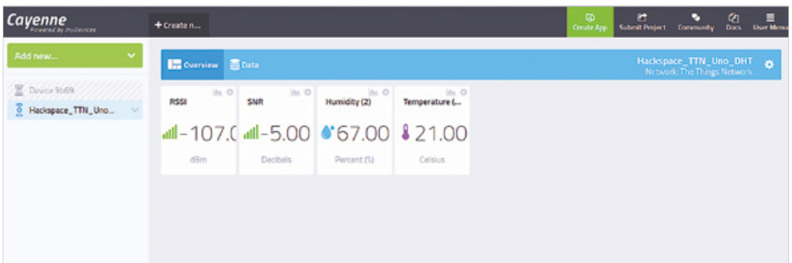


Figure 7 Select The Things Network from the left-hand menu



Above The standard dashboard showing our data

> Our payload is decoded and is nicely displayed, **labelled correctly 'temperature' and 'humidity', instead of just a raw collection of bytes**

the two lines under the comment '//Replace these with your AppEUI and AppKey') So, of course, copy and paste those entire two lines from your Device Overview Example Code box to replace the similar ones in the Arduino sketch.

DECODING THE PAYLOAD

Save your sketch and then click Verify. If the code compiles correctly, then double-check that your The Things Uno is still attached correctly as an Arduino Leonardo, and the correct port is selected, and then upload the sketch to The Things Uno. Leave The Things Uno plugged into your laptop for power after the sketch has uploaded.

You now have a LoRaWAN node with a sensor hopefully transmitting its payload of the sensor data on humidity and temperature, if you are within range of a gateway (it's worth taking your laptop and The Things Uno outside to increase chances!) Returning to The Things Network website, click the applications tab and select the application you created, and then in the Application Overview page, select the 'Data' tab from the upper right-hand side. Wait for a short while, and you should start to see data packets appear with some information about the data, and most importantly, the payload in the end columns stating the temperature and humidity readings from your sensor (**Figure 5**). As we set the application to read the payload as being of the CayenneLPP type, our payload is decoded and is nicely displayed, labelled correctly 'temperature' and 'humidity', instead of just a raw collection of bytes. If you click on a particular data packet, you get a drop-down with more information, such as the signal strengths and which gateways the device sent the data through.

As things stand, we have our sensor data going to The Things Network, but you might notice that if you refresh the Applications Data page or close it and reopen it, it doesn't keep the data there. Simple applications on The Things Network don't retain data; they act as a holding area that can send and forward data to other places. We are now going to create a simple dashboard for our device to which our application will send the data, and the dashboard will keep our data more permanently so we can review it when we need to.

> "The myDevices dashboard elements can all be edited and customised **so you can swap the icons or the type of graph**"

LET THERE BE DATA

Apart from it making it simple to get a payload in a readable form on The Things Network, we used the CayenneLPP library and payload format as it makes it very trivial to create a dashboard for our device online that will collect and display all the data from our The Things Uno device. To set this up, we first need to register a free account on the Cayenne myDevices website: **hsmag.cc/YlgAGf**.

Once logged in, select the large LoRa icon (**Figure 6**) and then select 'The Things Network' from the lower end of the menu bar on the left (**Figure 7**). Then scroll down and click the CayenneLPP option (**Figure 8**); in the settings window that should appear, you need to give the dashboard/device a name, and then add the Device EUI in the DevEUI box – leave the Activation mode set to 'already registered', and the tracking box locations as 'this device moves'. Save these settings and leave this tab open in your browser.

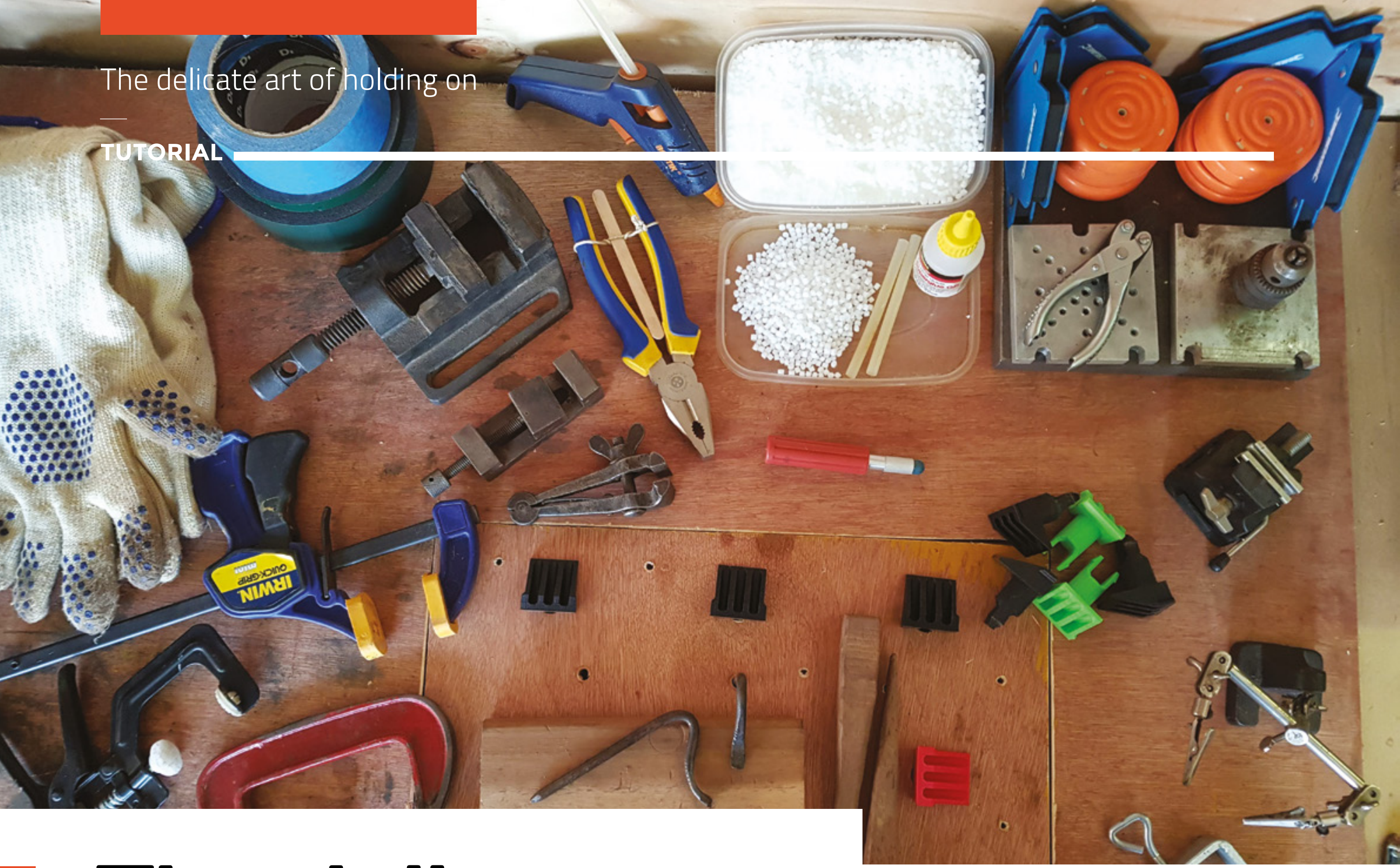
Finally, we need to go back to The Things Network site, and in our Applications Overview we need to select the 'Integrations' tab and click 'Add integration'. Scroll down and click the myDevices icon; in the Process ID box, give this a name such as 'hackspacedashboard', and in the 'Access Key' drop-down menu, when you click on the empty box, it should reveal only one option 'Default key' next to two buttons that say 'Devices' and 'Messages'. Click on 'Default key' to select this into the 'Access Key' box, then click the blue 'Add integration' button in the lower right-hand corner.

If you now switch back to your myDevices page we left open in another tab, as soon as myDevices receives some data from your The Things Uno, it should automatically make a dashboard for you and display the data. It should create a dashboard with RSSI (received signal strength indicator), SNR (signal to noise ratio), and, of course, our sensor humidity and temperature data. This dashboard will update with the latest data and will store the data it receives, meaning you can come back and check it anytime – or, if you take your The Things Uno offline, it won't lose all the existing logged data. The myDevices dashboard elements can all be edited and customised, so you can swap the icons or the type of gauge or graph by clicking the settings menu for each widget.

TIME FOR ANOTHER PROJECT

Congratulations on setting up your first LoRaWAN device and application. There are dozens of different platforms for devices, and innumerable sensors that can be developed and added to them. In addition, as a rapidly growing community, there are lots of tutorials to explore online to help you develop your next projects. ◻

Below
Our altered dashboard showing data in visual form



The delicate art of holding on

Learn to get a grip on your project and make your workshop a safer place

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.

Almost everyone is familiar with the function of clamps and vices. They're a staple of every workshop, shed, garage, and factory. However, not everything fits neatly into a bench vice, or lends itself to easy clamping. Knowing how to hold your work securely is an important skill, especially if you are using power tools or sharp hand ools that could do damage if you slip. In this tutorial, you'll learn some basic work-holding tools and techniques, and some sneaky tips for getting a grip on more awkwardly shaped objects.

BARS AND VICES

If you want to hold onto something, you probably need to put it in a vice or a clamp of some kind. The secret to successful work-holding is knowing what type of vice of clamp you need to use. Most vices have sensible names, like 'engineer's vice' or 'woodworking vice', that should give you a good idea about which vice you need to choose. For basic forms like bar and sheet, the process of clamping is very simple, and it doesn't really need to be described in detail for this tutorial. There are some simple tips that can save you from problems with any material, and these tips apply to work-holding in general.

Firstly, check the vice jaws and the threaded bar are clean. Clamping onto a workpiece when there are pieces of grit or swarf on the jaws will almost certainly leave a mark on your work. Grit or bits of swarf on the threaded bar will wear the vice mechanism and make it more difficult to clamp. Grease on the jaws will make it more difficult to hold your work securely without slipping.

If you are worried that clamping might damage your work, you can protect it with a layer of masking tape and fit a pair of soft jaws onto the vice. Soft jaws are made from soft metal or plastic and should be softer than the item you are trying to clamp.

SPLIT NUT

Threaded bar, bolts, and machine screws are very difficult to clamp effectively without damaging the threads, even if you are using soft jaws. Last month we looked at how to make a stud clamp for just this purpose, but another option is to use a split nut. To make a split nut, you will need an ordinary nut with the same thread as the workpiece that you need to clamp. Hold the nut in the vice with one of the high points at the top, and cut down to the middle with a hacksaw, parallel to the direction that the bolt fits. Clean up any rough edges, then screw the nut onto your workpiece. Tightening one or more of these nuts into the vice with the cut part between the jaws will deform it slightly, gripping the threaded bar firmly without damage.

Above A pipe vice typically has different-sized inserts that let you hold different sizes of pipe. They're relatively cheap and are very useful if you work with lots of hollow stock

You can make your own soft jaws, or buy them ready-made for popular brands of vice. You can also buy 'universal' soft jaws that fit onto existing vice jaws using magnets or tape, although these are less secure than replacement jaws that bolt onto the vice body.

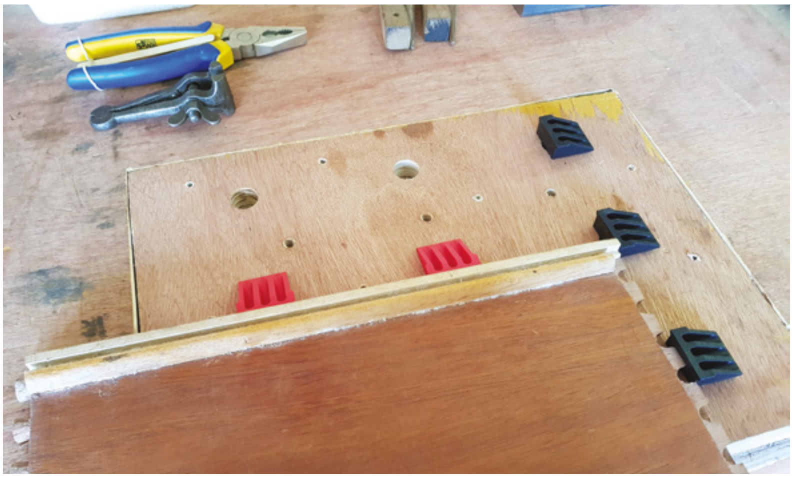
Clamping tube or hollow stock is tricky, and you really only have two good options to avoid deforming the part. You can either use shaped jaws that match the profile of the object you're working with (a dedicated pipe vice is an example of this), or you can increase the rigidity of the part before you clamp it by filling the hollow part with something solid. Musical instrument repairers sometimes use low-temperature plastic as a packing material in brass instruments. The plastic is heated, pushed into the desired section of the instrument, and then allowed to cool. When the repair is finished, they warm up the plastic again and remove it.

DOGS AND COOKIES

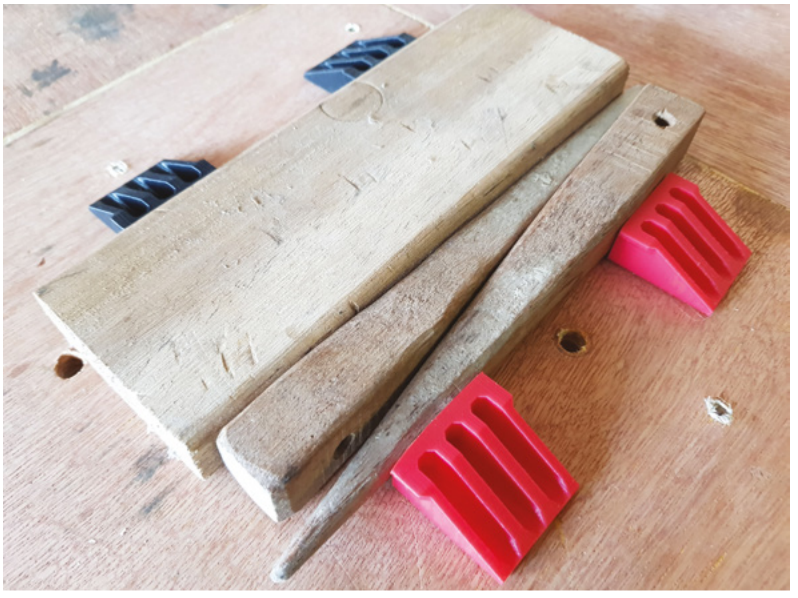
Holding work flat onto a surface poses a different set of problems, particularly if you need to work all the way along the edge of the material. Clamps are an effective strategy in some cases, but the jaws of a clamp can get in the way of tools you're using. If you have a workbench, it might be worth investing in some bench dogs. Although the name might not be familiar, you've probably seen a bench dog before on a portable workbench. Bench dogs are the small stoppers that fit through regularly positioned holes in a workbench and butt up against the edge of a piece of wood to stop it sliding around. You can make these quite easily with a 3D printer, or by gluing a piece of dowel into a wooden block. Bench dogs won't stop your work from flipping upwards, but they will keep it in position on the table when you apply a lateral force. →

QUICK TIP

Silicone mats or baking grips are useful for holding and moving hot metal objects.



Left This work area is drilled with holes that accept bench dogs and holdfasts. It can be removed from the workbench and replaced with a blank piece of wood when it isn't needed



NORTH AND SOUTH

You can use magnets to hold non-magnetic, thin, flat materials together. This is particularly useful in cases where you need to glue smaller items into place on a large sheet of material. Simply place a magnet on each side of the material and use them to hold the pieces in contact while they dry.

Above
You can fix your workpiece more firmly onto the bench by fitting dogs on every side, and then hammering wedges between the edge of the workpiece and the dogs

Right
A homemade holdfast made from a large nail can be very effective and only takes a few minutes to make. Larger cast holdfasts are expensive to buy, but can be a valuable workshop tool

Alternatively, holdfasts can be used instead of bench dogs. Holdfasts are steel bars with a hooked foot at one end, very much like the end of a walking stick or shepherd's crook. Like a bench dog, a holdfast fits into a hole in a workbench and the hooked part rests on top of the material you want to keep in place. Tapping the top of a holdfast with a mallet causes it to sit crookedly in the hole and wedges the workpiece in place. Although holdfasts are quite expensive, you can make a small version very quickly using a large nail. Heat the head of the nail and flatten it into a spatula, then bend a curve near the spatula to make the familiar shepherd's crook shape. You might need to experiment with the size of hole you need to drill in the workbench.

QUICK TIP

If you need to apply pressure on an uneven surface, a sock or bag filled with ball bearings makes a useful deformable weight.

Standard holdfasts are around 18 mm in diameter, but a nail will be considerably smaller. If you find the holdfast slips too easily, try roughing the sides with some coarse sandpaper.

If that all sounds like too much effort, or you don't have access to a workbench that you can drill holes into, bench grippers make an excellent alternative. Bench grippers (also commonly called cookies) are plastic discs coated in a non-slip rubber material, and they are great for holding flat materials in place for planing, sanding, or carving.

If you need to drill into the end of a bolt or a short piece of bar, you can use a spare drill-chuck to hold the work perpendicular to the bed of your drill-press. This is also a great way to keep hold of short pieces that need to be polished or worked on a grinder. The drill-chuck makes a comfortable, weighty handle that makes it less likely you'll inadvertently launch the part you're working on across the room.

For light machining, you can hold small parts in wax to make them more manageable. The wax won't last very long if the part starts to heat up, but for light engraving or scribing, the wax (or low-temperature plastic) should give even the most irregularly shaped object enough support to work with.

It might sound strange if you've never tried it, but masking tape and superglue can hold flat sheets like brass or PCB blanks in place for light milling or engraving. Apply the masking tape to the bed of the machine and to the back of the sheet material, then apply superglue to the tape, and join both tape surfaces together. As an aside to this, double-sided tape on cardboard or plywood can be very useful for holding small parts in place for spray painting or fine detail work. Having the parts fixed on an easily movable surface really speeds up the process of painting.

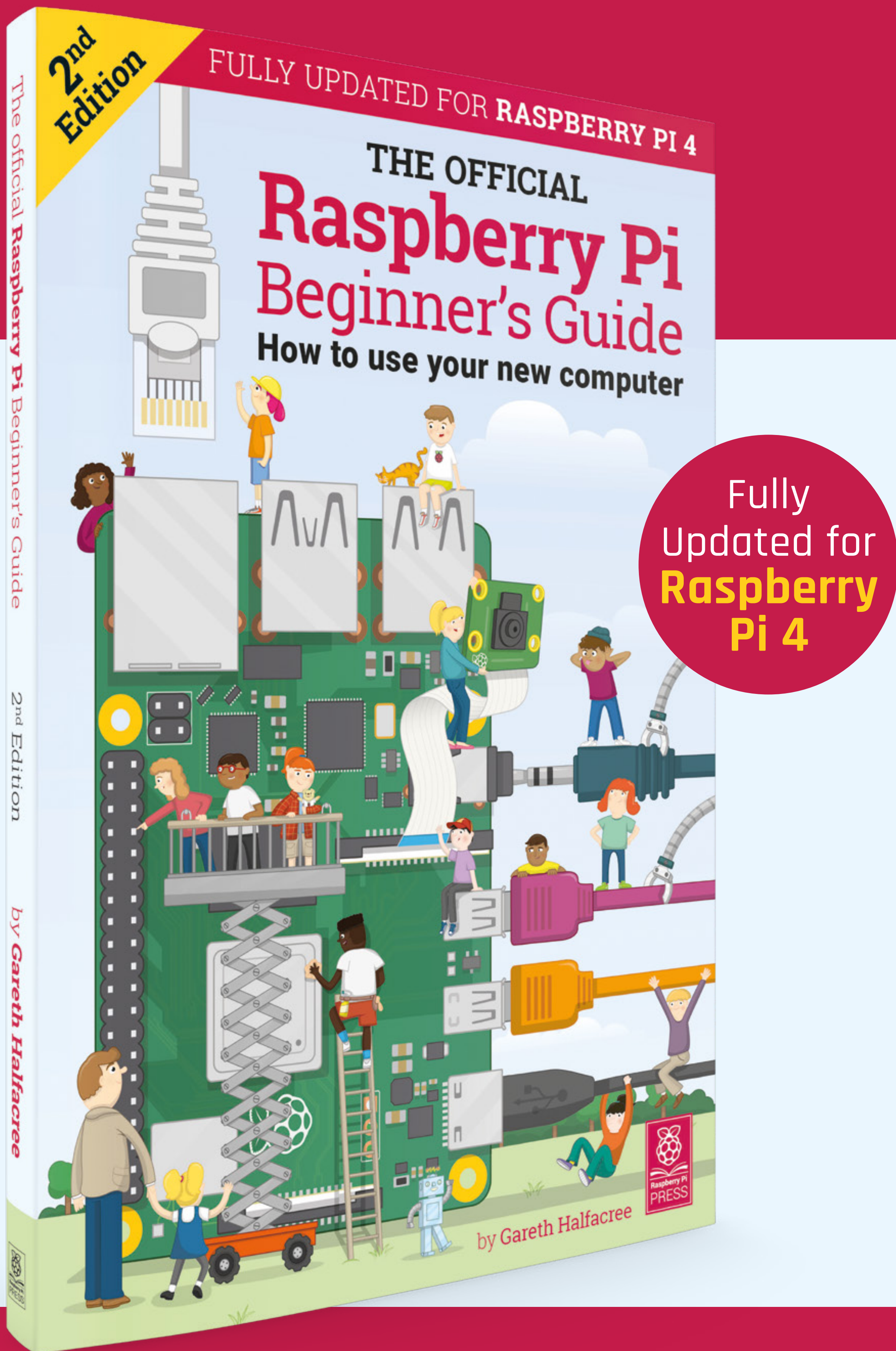
IF ALL ELSE FAILS, GLUE IT

The topic of work-holding is huge, and there simply isn't room to write about all of the techniques out there for clamping and fixing objects to work on. For some projects, the vast majority of the work can be creating an appropriate tool to hold the thing you need to cut, drill, mill, or turn. Whether it's bulldog clips or custom-made jigs, there is always a tool or method that will solve the problem. Hopefully, you will be able to use some of the tips in this article to save some time, and also make your workshop a safer place. □

Above
A hand vice makes it easy to hold small parts safely on a drill-press. You can use a pair of pliers, some string or wire, and a lollipop stick or long nail to do the same job, but make sure it's secure enough

QUICK TIP

Parallel jaw pliers close equally, leaving a parallel gap along the entire jaw. This makes them good for holding delicate parts without leaving marks.


2nd Edition
FULLY UPDATED FOR RASPBERRY PI 4
THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer
by Gareth Halfacree
Raspberry Pi PRESS
The official Raspberry Pi Beginner's Guide
2nd Edition
by Gareth Halfacree
Fully
Updated for
Raspberry
Pi 4

Rustic lamp with colour-sensing lights

Create glowing mushrooms that change colour to match an object

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.

Colour sensors are a novel way to create mood lighting. Place an object over the sensor and this lamp will shine with the matching colour. The sensors are easy to use with an Arduino, and are widely available.

In this tutorial, we'll embed a colour sensor in a log and use biodegradable plastic and multicoloured LEDs to make mushroom lights. The instructions use an Arduino Uno for the electronics because it's a popular board for beginners, but it can also be made with a Nano, ESP, or Flora, which are smaller and easier to enclose.

A BIT OF WOODWORK

Find a piece of wood that you could imagine seeing mushrooms growing on, with enough bulk to hide the electronics. A section approximately 20×12×12 cm works well. If necessary, use a band-saw to adjust the size and create a level base.

On the underside, choose where to hollow out a space for the electronics. It needs to be close to where your mushrooms will stand so that a drill bit will reach through from the top, and at least 8×6×3 cm to fit an Arduino Uno. Use a chisel and hammer, or power tools, to create the recess. Sand any rough edges to avoid splinters.

Put the Arduino in position to see where the power cord needs to go, and chisel a channel through the wood for the cord to come out the back of the lamp.

Decide where the mushrooms will stand and drill a hole for each one. Also, choose a position for the colour sensor and drill two holes close together for the wires to go through.

Above Seasoned wood, such as firewood, is ideal because it is dry and any cracking will have already happened

CUT AND STRIP

Cut 4×10 cm wires of each colour, except black. These are for the LEDs. Also cut 1×10 cm strip of each of the black, orange, blue, and green-coloured wires for the sensor. Using wire strippers, expose approximately 5 mm of wire from all the ends.

Each of the RGB LEDs have a tiny red, blue, and green light inside, and a corresponding wire, called a leg, to control them. The fourth leg provides a common anode, which will be the positive wire in the circuit. The legs for each colour are different lengths, so before you start soldering, check which is which from the LED data sheet. Solder matching colour wires to the tips of each leg on the LEDs and orange wires to the anodes. Then cover the whole of each leg with separate bits of yellow heat-shrink to prevent the wires from touching. A darker colour could show through the mushroom stalks.

MOULDING THE MUSHROOMS

To make the mushrooms, drop about six teaspoons of Polymorph into a cup of hot water, above 60°C. The granules are ready to use after a few seconds, when they turn clear and clump together. Use a spoon to remove a blob of Polymorph, and wait until it's just cool enough to touch.

Mould the Polymorph into a flat rectangle and wrap it around the legs of one of the LEDs to create a stalk. As it sets, the clear plastic will turn white, hiding the LED legs inside. If the Polymorph becomes stiff before you've finished moulding it, just drop it back in the water to soften. It doesn't matter how wet the LED gets.

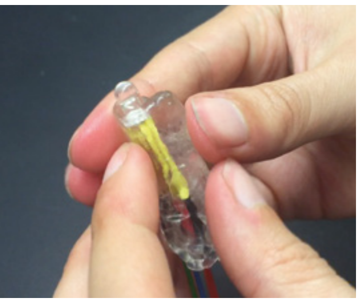
Mould another lump of Polymorph into a ball. Push your thumb into it to make a cone and then place it on top of the LED, squeezing gently to join it to the stalk. Continue moulding until it resembles a mushroom.

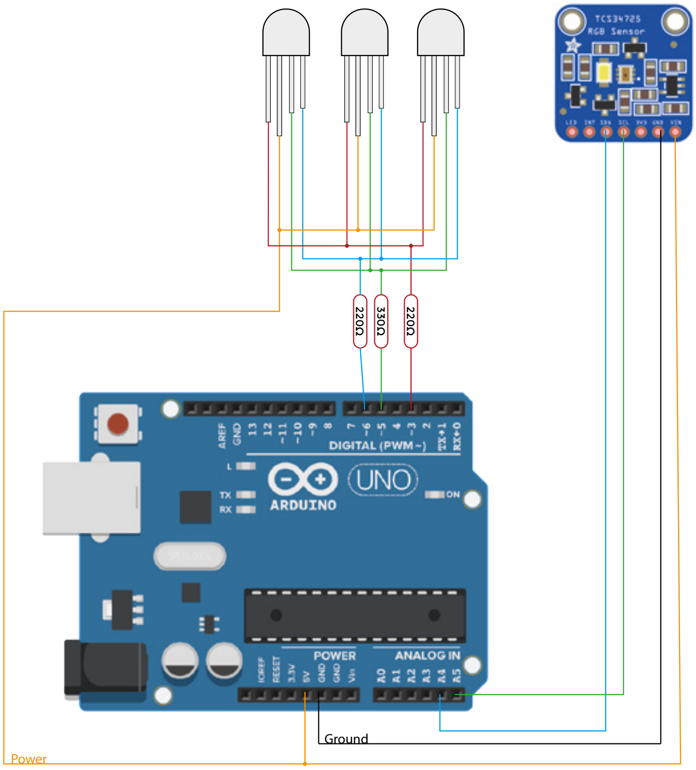
Top up the hot water as it cools down, and add extra Polymorph granules to make the rest of the mushrooms. →

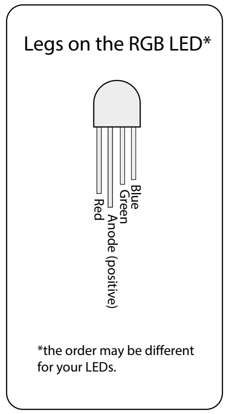
Above If you choose a log that has been chopped into quarters lengthwise, you can use a band-saw to make grooves along the intended opening. Then tap the thin wafers of wood out with a hammer

YOU'LL NEED

- **Arduino Uno** preferably a version with a row of holes alongside the pins
- **RGB colour sensor** (TCS34725)
- **3 × RGB LEDs with a common anode**
- **2 × 220 Ω resistors**
- **330 Ω resistor**
- **Small perfboard**
- **Red, green, blue, orange, and black insulated wires**
- **24 cm yellow heat-shrink**
- **100 g Polymorph plastic granules**
- **Piece of wood** e.g. firewood, part of fallen branch or driftwood
- **USB A-Male to B-Male lead with on/off switch** (printer cable)
- **Lithium-ion battery with USB connector** such as a power bank or 5 V mains adapter
- **Insulating tape**
- **Two small screws** (optional)
- **Dry moss from twigs** (optional)
- **Cup and tray of hot water**
- **Teaspoon and spatula**
- **Band-saw**
- **Chisel hammer**
- **Drill with 5 mm drill bit**
- **Soldering iron and solder**
- **Hot glue gun and glue stick**
- **Rolling pin**
- **Hair-dryer**
- **Wire cutters**
- **Wire strippers**
- **Sandpaper**
- **Scissors**







Above This wiring diagram shows how the components connect together to create the circuit

Top Polymorph looks a bit like frogspawn the first time you melt it

Above When Polymorph is heated, it becomes malleable like clay

Position the mushrooms on the log and poke the wires through the drilled holes. Remelt the mushrooms slightly with a hair-dryer to mould them to the contours of the log. That may be enough to stick them to the bark. If not, use a hot glue gun.

CREATE THE CIRCUIT

Perfboard has predrilled holes and copper pads on one side, which make it easy to arrange components and attach solder. For this tutorial, we will refer to the side with the copper pads as the top.

> Many clones of Arduino Uno boards (available on eBay) **have a line of holes as well as pins, which makes it easy to solder wires to**

With the top of the perfboard facing down, poke the three resistors into any holes. Turn the board over, solder them into place, and cut the legs off with wire clippers. All the green wires share the 330 Ω resistor, so push the ends of the wires through the perfboard close to the resistor and solder from the underside. All the red wires go to one of the 220 Ω resistors, and the blue wires go to the other 220 Ω resistor. Use solder to create a bridge from each of the wires to the relevant resistor.

Solder the orange wires attached to the LEDs to an unused area of the perfboard and join them together using solder. This will become a shared power connection. Red wires are often used for power, but that colour has been used, so orange has been chosen to avoid confusion.

From the underside of the perfboard, poke the remaining green wire next to the 330 Ω resistor and secure in place with solder, making sure the solder touches the resistor too. Do the same with a red and blue wire for the other resistors. These wires will attach to the Arduino later.

ADDING MAGIC TO THE MUSHROOMS

Although there are seven pins on the colour sensor, you only need to attach wires to four of them. Solder one end of the remaining orange strip of wire to Vin (voltage in) for power and do the same with black to GND (ground). Connect the blue wire to SDA (serial data) and the green to SCL (serial clock), both of which will communicate with the Arduino.

Connect the other end of the orange Vin wire to the shared power area on the perfboard, created in a previous step. The rest of the wires from the colour sensor will go straight to the Arduino board.

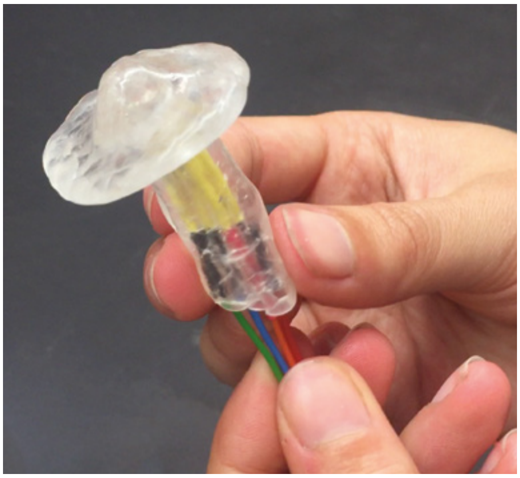
Connect the black GND wire from the colour sensor to one of the GND pins on the power section of the Uno. Also connect the blue SDA wire to Analog pin 4 and the green SCL wire to Analog pin 5.

QUICK TIP

Glue dried moss from twigs onto the log for extra texture, or to hide the sensor's circuit board. Make sure the tiny sensor and on-board LED are left uncovered.

PRINT YOUR OWN

Instead of using Polymorph, you could 3D-print the mushrooms. A search for 'Mushroom LED' at **thingiverse.com** brings up free designs, ready to resize and print. Buy diffused RGB LEDs, or cover clear ones with masking tape or a blob of Polymorph, to make the colours blend evenly.



Left Colour-coding the wires makes it easier if you need to do any trouble-shooting

Above Play around with coloured objects to explore the different effects

Many clones of Arduino Uno boards (available on eBay) have a line of holes as well as pins, which makes it easy to solder wires to. If yours only has header pins, you could remove them or use jumper wires instead of regular wire.

Connect the red wire beneath the perfboard to Digital pin 3 on the Uno, the green wire to Digital pin 5, and the blue wire to Digital pin 6. Make sure that the components are dry before connecting to power.

HIDE THE WIRES

Fill a tray with hot water, and sprinkle a layer of Polymorph granules on top to create a thin sheet. Remove with a spatula to keep it flat and use a rolling pin to create a roughly rectangular shape, a little larger than the gap you hollowed out of the log. Return it to the water when necessary to keep it flexible.

Hold the Polymorph over the electronics to check the size, then cut and bend into shape. While it's pliable, thin pieces of Polymorph can be cut with scissors. If you can't cut it, reheat and roll a bit more. Use the glue gun or screws to keep it in place.

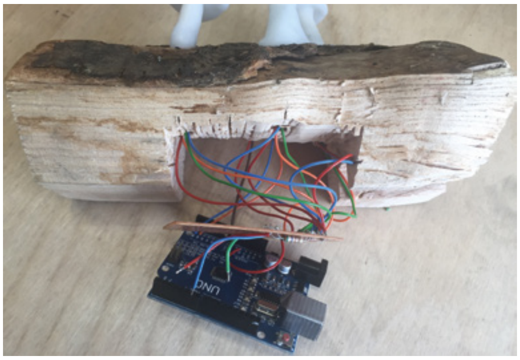
LET THERE BE LIGHT

Open the Arduino IDE software and connect the Arduino to your computer. If you are new to Arduino, check out the guide at **hsmag.cc/shYSMW**, which walks you through how to select the board and port settings.

Install the colour sensor library by going to Sketch > Include Library > Manage Libraries. Search for Adafruit TCS34725 and click Install. For help on installing libraries, go to **hsmag.cc/zmXjEI**.

Open the ColorView example code that comes with the sensor library by going to File > Examples > Adafruit TCS34725 > ColorView.

Upload the code to the Arduino, by going to Sketch > Upload. Remove the power cord from the computer and attach it to a USB power supply, either mains or battery powered. Then hold a colourful object over the sensor. After a few seconds, the LEDs should shine the same colour as the object. Place a different object on the colour sensor and the LEDs will change colour too. □



Left Perfboard is put between the log and the Arduino. Cover the USB port with insulating tape if there is a risk of components short-circuiting

How to use digital callipers

Learn to get a grip on your project and make your workshop a safer place

Gareth Branwyn

@garethb2

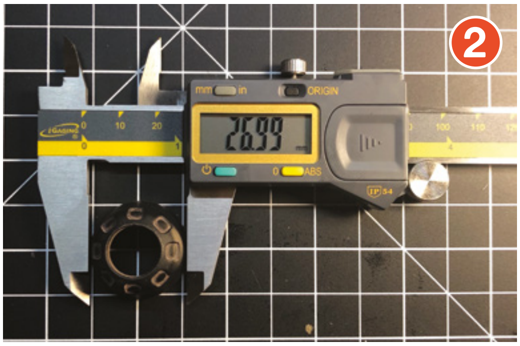
Gareth has been a lifelong practitioner (and chronicler) of DIY tech, media, and culture. He is the author of ten books, including *Tips and Tales from the Workshop*, and is a former editor for Boing Boing, Wired, and Make:.

There is a reason why 'measure twice, cut once' is a golden rule of making. Getting accurate measurements is critical to the success of many projects. While the measuring abilities of a common imperial/metric ruler are sufficient for many measuring situations, sometimes you need much higher precision (like when measuring parts to feed into a computer design for 3D printing). That's the time that you need a set of digital callipers.

You might think that getting a set of digital callipers in your toolbox is an expensive proposition, and not worth it for the few times most of us might need such an instrument. Not exactly. For under US$10, at online stores and discount tool markets, you can get a worthy set of callipers that can reliably measure at a 0.0005″/0.01 mm resolution. Many of these cheap callipers work just fine, once you learn to work around their eccentricities. If you want higher confidence in your tool, for under US$40 you can get a surprisingly high-quality set. We recommend the iGaging Absolute Origins. They deliver everything you need in such a tool at a really reasonable price. They even have a data-out port. With an additional special cable, you can send your measurements directly into a CAD program. Unfortunately, the cable will cost you twice as much as the callipers (but you can sometimes find them cheaper on eBay).

So, what can digital callipers do for you? All models of such callipers are designed to provide four basic types of measurement:

Outside Diameter The main jaws of the callipers (called the 'outside measuring faces') are designed to measure the outside diameter of objects (usually up to 6″/150 mm). Many models of callipers have a zeroing function. To do a measurement, you close the jaws, press the Zero button, and then spread the jaws to take your measurement.



The results appear on the LCD. Most calliper sets have imperial and metric selections. Some also include a fractional measurements mode.

Inside Diameter The smaller 'faces' at the top of the callipers are for measuring the inside diameter of an object. The jaws here are inverted from the main ones, but work in the same fashion. You open them to the walls of whatever inside area you are measuring and find the reading on the LCD.

Depth On the back end of callipers' ruler/scale is found a depth gauge (or 'depth bar'). This is used to measure, for instance, the depth of a hole. Simply extend the bar into the distance to be measured, and read the results on the display.

Step You can measure the difference between two surfaces using the outside measuring face (main jaws). You simply place the outside edge of the sliding jaws against one of the edges you are measuring and the outside edge of the inner, fixed jaw against the other edge. The display will reveal this distance.

Some models of callipers have a fine adjustment screw that allows you to move the jaws more precisely. Some also have a set screw so that you can tighten to hold a measurement. Knowing how much pressure to apply when taking a measurement takes some practice. It's often a good idea to take several measurements along an object and average them out to get closer to a reliable number.

Other uses and measuring tricks:

Using callipers like a funky slide rule You can use callipers as a conversion calculator. Simply move the fine adjustment wheel (if you have one) on your callipers to the number along the scale that you wish to convert, then press the Mode button to switch between millimetres, inches, and fractions (if your set has this mode). Digital slide-rule!

Measuring the balance thickness of a 'blind hole' Let's say you have a blind hole (a hole that does not go all of the way through) in a workpiece, and you want to know the difference between the depth of that hole and bottom edge of your piece. Simply use the depth bar to measure the depth of the hole, zero out your callipers, and then measure the overall height of the piece. The result shown will be the balance of the blind hole.

Measuring centre-to-centre distance When measuring the centre-to-centre distance between two holes, first measure the hole diameter. Then zero out your callipers and measure from near-edge to near-edge of your two holes. (The hole diameter, which you zeroed out, will be automatically subtracted). This gives you your centre-to-centre distance. This is particularly useful when trying to figure out where to place PCB mounting posts when designing project enclosures.

Callipers are also known as a vernier, which is a short scale made to slide along a graduated instrument for indicating divisions. □

Thanks to
Lenore Edman, Jeff Worley, Richard Gould, Jake Hildebrandt

Figure 1 The top faces are used for inside diameter measurements

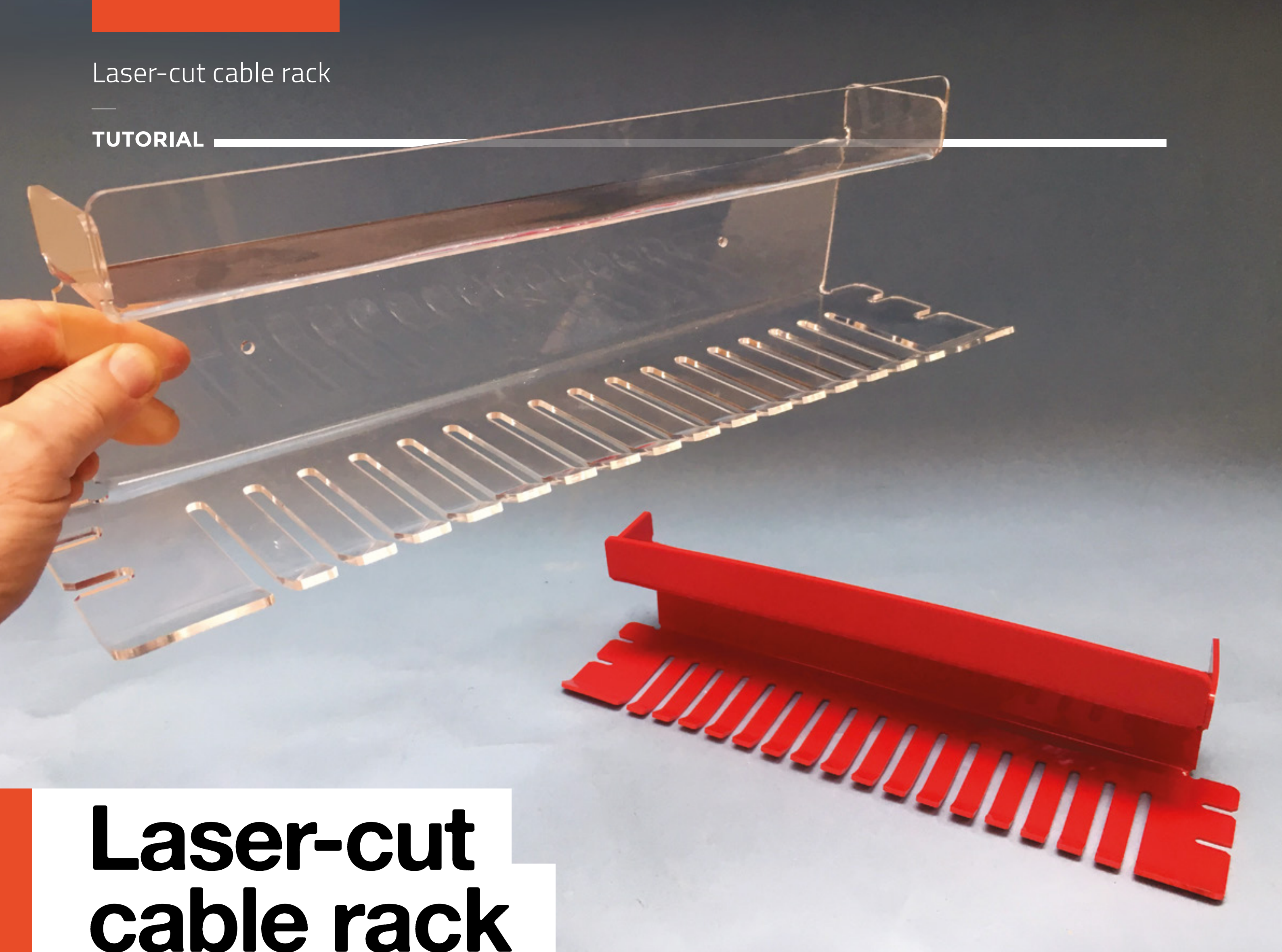
Figure 2 The bottom 'jaws', known as 'faces', are used to measure outside diameters

Figure 3 You can use the distance between the edge of the inside face and the outer edge of the outside face to measure the distance between two edges

Figure 4 The 'depth bar' found on the opposite end from the faces is used to take depth measurements

YOU'LL NEED

- A set of digital callipers
- Something to measure



Laser-cut cable rack

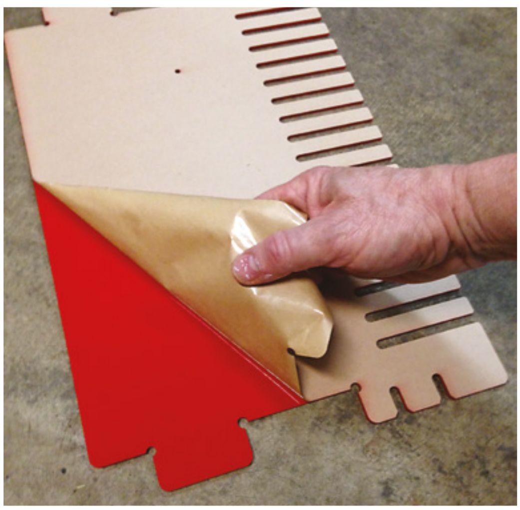
Cut and thermoform a one-piece cable organiser

Bob Knetzger

Bob Knetzger is a professional toy designer/inventor and has created dozens of products for Mattel, Hasbro, Spin Master, and many others. He is the author of *Make: Fun! Create Your Own Toys, Games, and Amusements.*
Makerfunbook.com

If your shop or workspace is like ours, you probably have a snarl of assorted cables on hand: audio patch leads, proprietary camera cables, various vintages of USB and FireWire, ¼" guitar leads, HDMI, MIDI, AC power cords... and what is that odd, mystery cord from?! Here's a simple design for cable storage to organise such a mess. It's made from a single piece of laser-cut acrylic, thermoformed into shape. When the rack is mounted on the wall, the cords hang down for easy identification of plug types and cord lengths. It also has a shelf to hold other little bits and bobs.

You could cut out this acrylic rack using hand or power tools, but the comb shape means laboriously scoring, drilling, and scraping – and fearing cracks and chipping at each operation. Laser cutting makes for easy and worry-free fabrication. Unlike laser-cut wood and fibreboard, with its burnt smell and ugly charred edges, acrylic melts like butter, and the laser leaves a perfectly smooth edge, similar to the flame-polished display cases seen in museums. You may even come to like the 'sweet' smell of melted acrylic (but of course, always use proper ventilation when laser cutting).



01 GET READY

We used a Glowforge laser cutter, so the basic size of the flat shape fits nicely onto a single, full-size sheet of Glowforge's own Medium Proofgrade Acrylic. For any other laser cutter, use a 12″ × 20″ sheet of 1/8″ thick acrylic. Choose the speed and power setting for a clean through-cut on your laser cutter; there are no engraves in this design. The Glowforge's built-in camera reads the Proofgrade's QR coded stickers to automatically set the power and speed for the material to be cut, but on your machine, you may have to set it up manually.

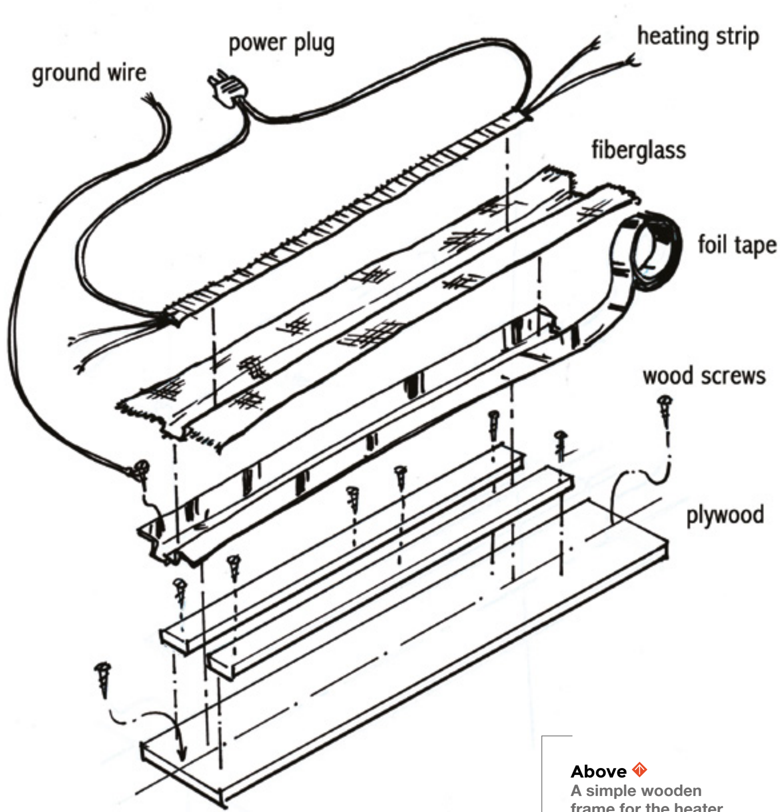
02 MAKE ADJUSTMENTS

Because you'll be bending the corners, there are no worries about kerf allowances and finger joints. In fact, none of the dimensions is critical, so you can adjust the size or change the proportions to fit your particular needs (or to optimise for a scrap piece of material). And you can freely change the sizes and locations of the holes for mounting fasteners to match your installation.

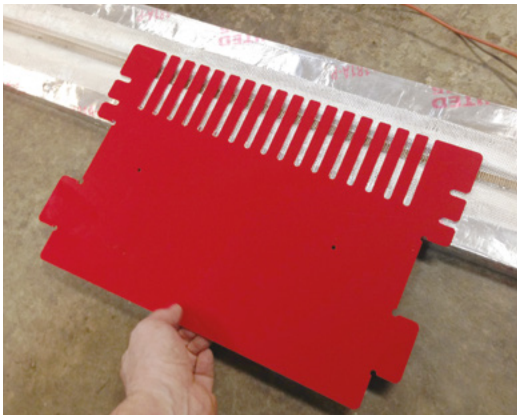
Leave the protective paper label on both sides of the acrylic when laser cutting to avoid 'flare' and smoke marks. Remember to remove the label before strip-heating! →

YOU'LL NEED

- ⅛" thick acrylic sheet, 12" × 20"
- Access to a laser cutter
- SVG file for the design (go online at hsmag.cc/issue22)
- Strip heater
- Mounting screws or hollow wall fasteners

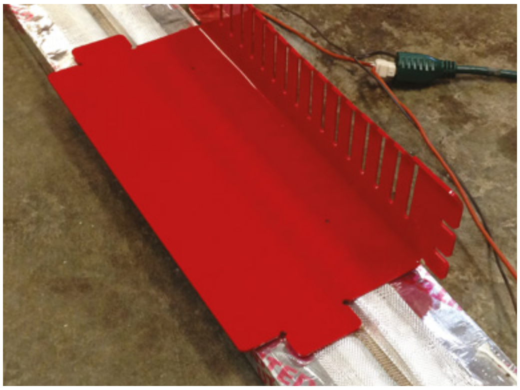


Above A simple wooden frame for the heater to sit in



Above Making the first bend

Below Think about the order you need to perform your bends in



> **A strip heater is ideal. If you don't have access to a commercial heater,** there are inexpensive units available as kits

03 THE STRIP HEATER

Once the basic flat shape is cut out, you'll next heat and bend the corners. For that, a strip heater is ideal. If you don't have access to a commercial heater, there are inexpensive units available as kits. We like the BriskHeat strip heater element. It comes pre-wired and ready to use. You build a simple wooden frame for the heater to lie in. There are more details in the kit's instructions, but here's a useful sketch (above).

04 GETTING BENDY

The strip heater is easy to use. After the heater has warmed up, place the material to be formed carefully in position over the strip. The strip very gently and slowly warms the plastic. You'll see a slight deformation in the surface when it's nearly ready. Test by gently bending: when ready, the material will bend easily. Remove the material and make the desired bend, then hold in position as the plastic quickly cools into the new shape. Unlike slump moulding, where the material is heated all over, the strip heater softens the plastic only where you'll actually bend it, so it's easy to get nice-looking, accurate corners.

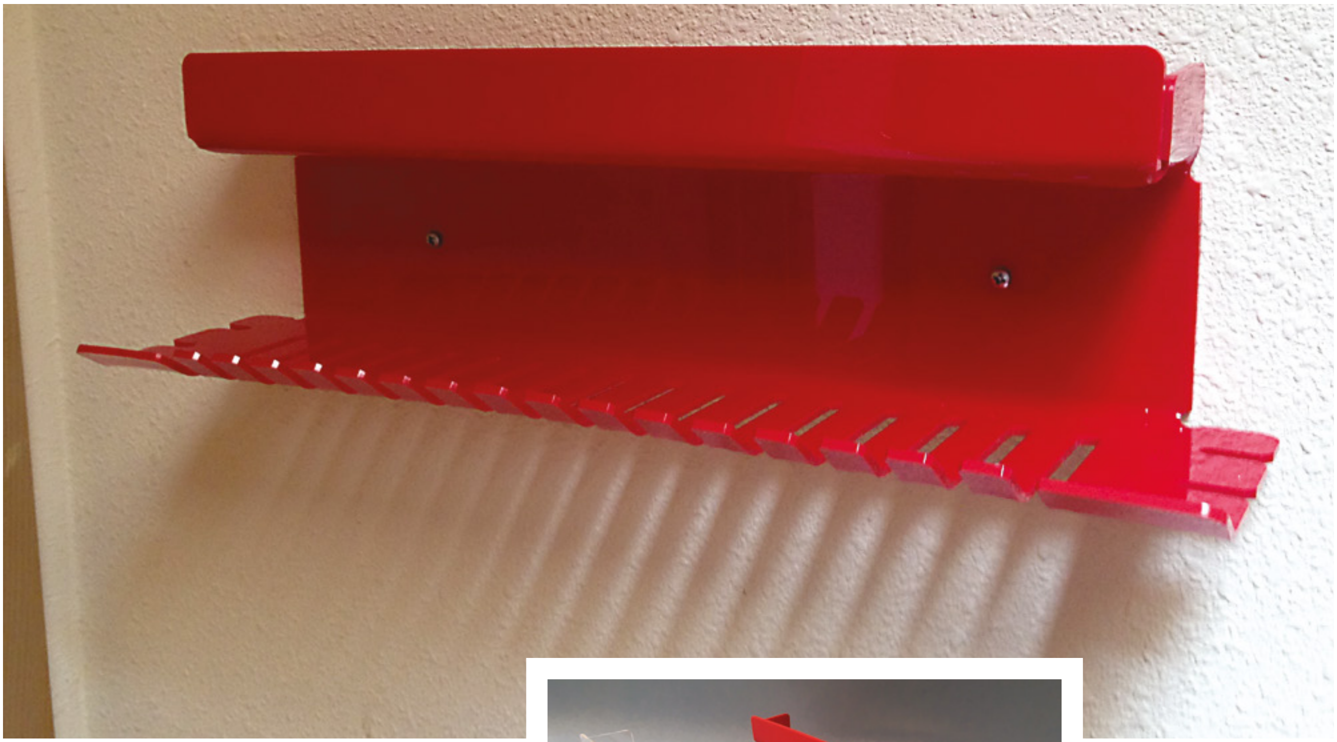
05 MAKING BENDS

To make the slight angle on the tips of the comb, first soften the plastic and then press down firmly with a piece of wood as you hold the tips to make the bend. Hold in place until the plastic cools.

Consider which side of the material to place on the strip heater. Place the side of the outside of the corner face down, directly on the heater. The plastic there has to be softer, to stretch more to make the

QUICK TIP

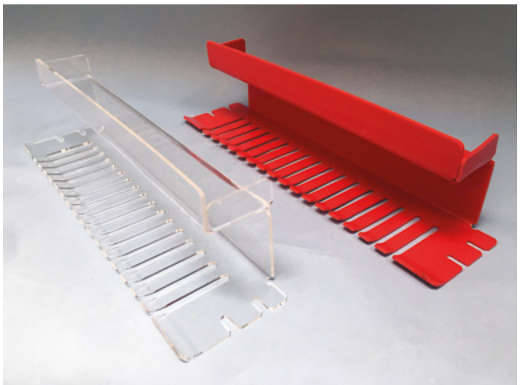
I used self-adhesive aluminium foil tape to line the channel and to tape down the fraying edges of the fibreglass mat.



larger, outside radius of the corner. The cooler side (away from the heater) is the smaller inside radius corner.

Plan your sequence of bends so that the edge to be heated can reach and lay flat on the strip heater. Place the strip heater of a worktable and near the edge to help position and heat multiple bends in a tricky shape.

Sometimes a shape will require a backward bend, where the side facing the strip heater will become an inside corner. In that case, leave the material over the strip for some extra time to soften thoroughly so that the plastic bends without cracking.



Above Mounted and ready for storage

Left You can use any colour of acrylic

06 WALL MOUNTING

To mount the finished rack, use wood screws or hollow wall fasteners. If needed, drill new mounting holes in the rack to match your installation requirements.

You can now store your knick-knacks, cables, and assorted doodads on your wall where they'll be easy to find. Now, go forth and declutter your workshop with your own wall-mounted storage rack. Let us know how yours turns out **@HackSpaceMag** on Twitter, or email **hackspace@raspberrypi.org**.

MAKING MORE

The laser cut-and-bend method of fabrication is quick, easy, and highly versatile. Once you get the hang of making accurate bends, you'll find a wide range of things that you can make.

Controlling temperature in the kitchen

Reuse waste plastic to make a fermenting box

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.

Temperature control is hugely important in working with food. Sometimes we want to heat it to a high temperature, and for this we have ovens and microwaves. Sometimes we want to chill it, and for this we have fridges and freezers. However, sometimes we want to warm it a little, and for this there are fewer options. For example, you might want to warm bread dough to help it rise, or keep milk at the right temperature for yoghurt to form. To help with this, we're going to build an insulated, heat-controllable box.

The mechanics of a heat-controllable box are pretty simple. We need something to heat it, something to sense the temperature, and something to turn the heater on or off, depending on what it reads from the sensor. There are lots of options for all of these and, in many ways, there's not a 'best' option.

THERMAL CUT-OFF

Our temperature controller claims to have a built-in thermal cut-off, but it's still relying on a single unit to ensure that it never overheats, and we don't have a good idea of how reliable this unit is yet. The risk is that the unit somehow jams on and continues to push power into the heat pad, even when it's too hot. Whether or not this is a problem depends on how hot your unit can get with the heating element permanently on. It's a good idea to test this scenario out by putting your heat pad on permanently while you monitor it for temperature to see what the risk is.

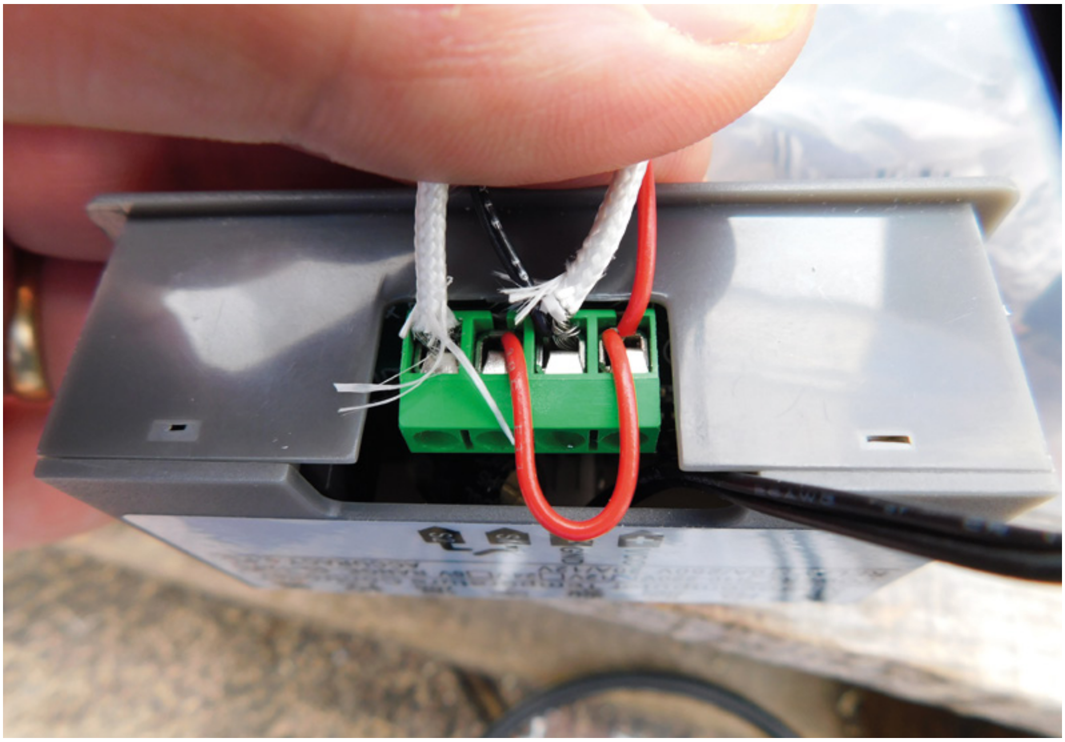
If you're concerned about it getting to a dangerous temperature, you may want to consider putting a separate heat monitoring unit in there that can cut off the supply to the first unit. This could be as simple as another identical temperature controller that's set to the same temperature with both relays rigged in series. That way, as long as one of the two units is working properly, it will cut off the power once it gets above the predefined temperature.

The first real decision is what voltage to run your system at. Many off-the-shelf heating components are designed to run at mains voltages. For example, some people use filament light bulbs. However, we wanted to run ours at a lower voltage, so we didn't have to worry too much about the electrical safety. However, if you go too low in voltages, you need a high current to get enough power to provide enough heat for your box, and that introduces its own problems. We went with a 12V system.

For heating, we went with a 12W silicon heating pad. These are available cheaply from a variety of sources (ours cost £2.78, including delivery).

For temperature and control, we could have used more or less any microcontroller and programmed our own interface, but we wanted to make this project super-simple, so opted for an off-the-shelf temperature control unit. This has a temperature probe, a display, and a relay. You can set a temperature you want it to hold, and it will turn the relay on and off to help you achieve this. You just have to wire the heating pad into the relay.

Above Insulating our box with shredded plastic packaging – just make sure it's all clean



The other item we need for the build is a 12 V power supply (we pulled this from our big box of spare power supplies) – the only requirement is that it can supply just over 1 A of current (1 A for the head pad, and a bit for the controller). A 1.5 A power supply should be plenty. We attached this via a 5.1 mm jack, but you could also wire it directly in if you'd prefer.

That's all the electrical gubbins sorted; we just need a box to house it in. As we're trying to hold a warm temperature, we need insulation to help it stay warm so we don't waste electricity. Insulation also helps keep the temperature even inside the box. There are a few options here. The simplest is to use an insulated box like a cool box, but this is also the most expensive option. The most hacky solution is to use a simple Tupperware box, and wrap it in a blanket when it's running. However, we opted to build our own insulated box using two plastic boxes, one smaller than the other. With the small box held inside the large box (more on this in a bit), we filled the gap between them with insulation. There's loads of stuff you can use for insulation, but we used old plastic packaging. Anything that can be scrunched up and cut to size can be jammed between the two boxes to help keep our heat in, and our insulation consisted of everything from bubble wrap to (clean) crisp packets. We used scissors to cut it all into little bits, and jammed it in place.

Above The wiring for our box puts the relay in line with the power for the heat pad

Left The temperature controller displays both the current temperature and the desired temperature

THE BUILD

The first part of the build is to get the electronic gubbins mounted into the outer box. The temperature controller is panel-mounted with clips to hold it in place, so you need to cut out a rectangle the appropriate size from the outer box. We used a rotary tool with a cutting disc, but a hot knife would also do it if you don't have a rotary tool. You'll also need a hole for the power to go through (we used a panel-mounted 5.1 mm jack port; you could equally just have a hole for the wires to go through, but you may want to consider using hot glue as strain relief if you do this (this will also help with insulation). →

Right While it may not look pretty, our temperature-controlled box is both insulated and helps reuse waste plastic

You'll also need a hole through to the smaller box for the wires for the heater.

The second step is to mount the smaller box so that it's held in place while the insulation is packed around it. We did this with some 18mm-wide wood that fitted in perfectly; however, you will have to improvise as your gaps are unlikely to be the same as ours. We screwed the wood to the smaller plastic box, then positioned this box at the appropriate place inside the larger one, and screwed it to that as well. This held the smaller box off the bottom of the larger box and with gaps all around it. We filled these gaps with cut-up bits of plastic. We packed it in quite densely as this should provide more insulation.

> **You'll also need insulation on the lid. We just happened to have** an Amazon Prime padded envelope the same size as the lid

The insulation should also cover the top half of the box, so we need to extend the sides of the smaller box upwards. We happened to have some modelling foam left over from a previous project, so in the spirit of reducing waste, we used this. Really, you can use any moderately stiff plastic sheet. We tried a few options for holding them in place, but found that T-Rex tape worked best. It doesn't have the most aesthetically pleasing look, but we're building with rubbish, so aesthetics aren't our top priority. We built the walls up as high as we could without blocking the lid.

You'll also need insulation on the lid. We just happened to have an Amazon Prime padded envelope the same size as the lid. We filled this with more cut-up plastic to add extra insulation and taped it to the inside of the lid. This padding on the lid should push in and fit snugly with the padding around the side of the main box.

It's worth noting that we're not worrying about if any of this is food-safe – this is because whatever food we put in this heat box will be in its own food-safe container, and this contraption shouldn't come into contact with any foodstuffs.

That's everything in the right place; it's time to wire up the electronics. Your 12V input should go into the temperature controller, but it also needs to go to the heat pad – the temperature controller doesn't power the heat pad directly, it just turns a relay on and off. The wiring goes like this: 12V goes into one side of the relay. One heat pad connection goes into the other side of the relay (it's not polarised so can go in either way around), and the other heat pad connection goes to ground. You can do this using the screw connectors in the back of the temperature control unit, so no other soldering or wire joints are necessary.

REDUCING WASTE

We could have built this temperature-controlled box in many ways, but we chose to use otherwise waste plastic as the insulator. Humanity is currently producing huge amounts of single-use plastic that then ends up littering the planet for generations to come. As makers, we can find ways of using this resource rather than just consigning it to landfill. This build was one attempt to do that.

If you've got other ideas for how we can reuse rather than waste plastic, we'd love to hear them. Drop us an email at **hackspace@raspberrypi.org** or message us on social media.

The only bit left is to mount the temperature sensor in the box and have something to rest our food on, as we don't want it sitting directly on the heat pad. We've used three bits of 18 mm wood – one with a hole drilled in it to hold the temperature sensor. These are in the bottom of the small box, with the temperature sensor configured so it's between the food and the heat pad. This means the food should never go above the set temperature.

That's the box built and ready to power on. When you first turn it on, be aware that there may be a few different modes that the controller can be in – for example, ours came configured to work with a chiller rather than a heater, so it would have left the heater switched on permanently. Follow the instructions on your temperature controller to make sure you've got it set up properly, and make sure you keep it supervised so you're sure it's working properly the first few times.

That's your temperature-controlled box built. You're now ready to help your bread rise and make yoghurt, tempeh, or all manner of other fermented goodness.

YOGHURT

Yoghurt is a form of cultured milk. To make it, you inoculate some milk with the appropriate bacteria, and then keep the milk at the right temperature for the bacteria to thrive. What temperature this is depends on the particular cultures you're using, but most common cultures want between 40 and 44 °C.

The traditional method of creating yoghurt has the rather unappetising name of 'back slopping'. This is where you take a couple of spoonfuls of live yoghurt and add it to milk before putting it in your temperature-controlled box. You can start this with any commercial live yoghurt; however, the bacterial colonies in commercial yoghurt aren't very virile, and you'll probably find that after a generation or two of yoghurts, it stops working. We've had the same experience with commercially produced 'yoghurt starters'.

If you want an ongoing supply of yoghurt, the best bet is to acquire an 'heirloom' yoghurt starter. These cultures should continue to grow indefinitely if you look after them well.

Above
Internet deliveries can lead to huge amounts of plastic waste, so make sure you find ways to reuse it

Staying keen

Learn how to stay at the cutting edge with your workshop tools

Dr Andrew Lewis

@monkeysailor

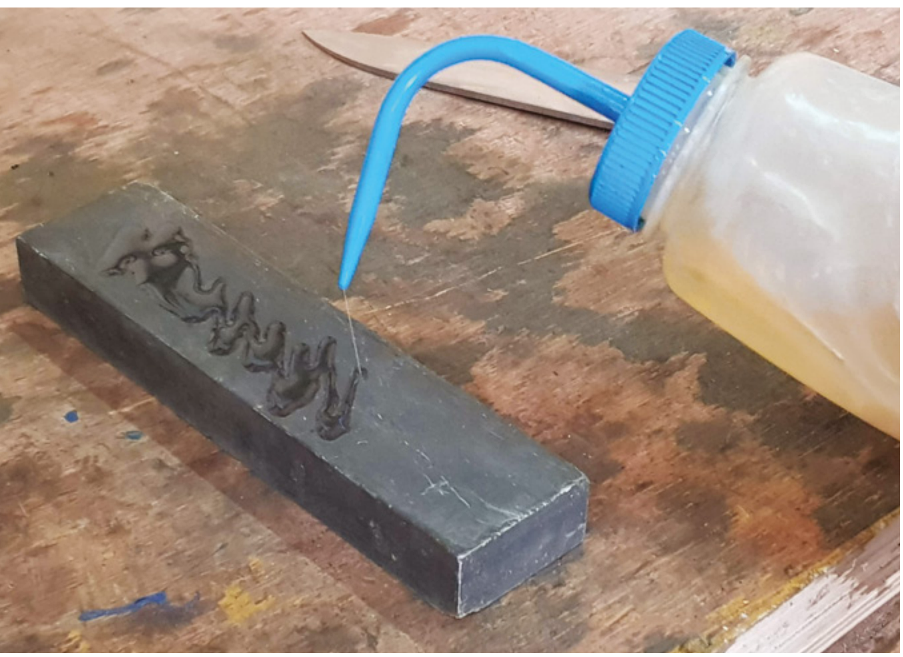
Dr Andrew Lewis is the owner of **Shedlandia.com**, a restorer of old tools, a fabricator for hire, a research scientist, and a founder member of the Guild of Makers.

There's an old saying that a blunt knife is more dangerous than a sharp one. The reasoning behind the saying is that you need to apply more force with a blunt knife to make it bite into the surface, so it's more likely that you will slip while you're using it. The same thing can be said about most workshop tools, and keeping the edge on your tools is important if you want to keep using them effectively. Blunt tools can overheat, slip, shatter, and can delay or even completely ruin the project you're working on. In this article, you'll be taking some of the most common workshop tools and finding out the best way to keep them in working condition, with the minimum amount of effort.

Sharpening a blade is something that's easy to do by hand with a little practice, and there are a few different tools out there that use automation to let you get the job done quickly. The simplest method of sharpening a blade is with a whetstone. A whetstone is an abrasive stone made wet with oil. Modern whetstones tend to be double-sided, with a coarse and a fine abrasive side. The coarse side is used to grind out more noticeable nicks and blemishes from a blade, while the fine side is used to hone the edge to a sharper point. Whetstones are graded, just like sandpaper. The finer a whetstone, the higher its number will be, with a coarse stone being 400 or 1000 grit, and finer polishing stones going up above 30,000 grit.

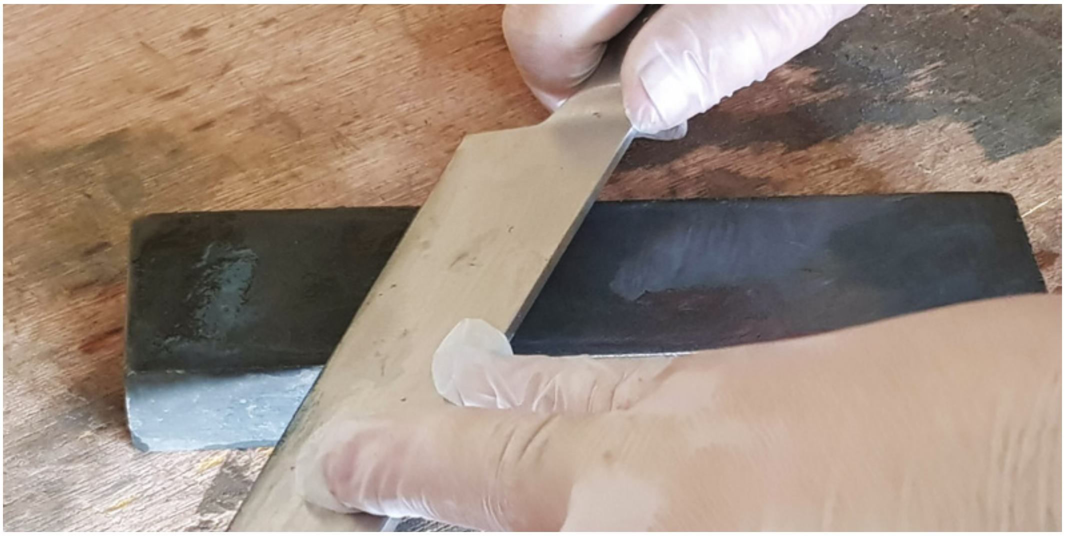
To use a whetstone, apply a little bit of oil to the surface, and move the blade across the surface so that the edge of the blade is touching the stone evenly, with light pressure. For the best results, always try to move the blade so that the sharp edge is being pulled along the stone, rather than pushed onto it. The blade of a knife is usually a V-shape when viewed end on, so you'll need to flip the knife over every couple of strokes so that you sharpen each side equally, and keep a steady angle between the stone and the blade. You want to match the existing angle that the blade makes, so keep an eye on the edge as you work to see whether you have the knife tilted too far up or down.

If you have nicks or gouges on the blade, start with the coarse side of the whetstone and keep working the blade steadily until you've got them out. Once you have a smooth edge, switch to the fine side of the stone and continue honing the blade until it's as sharp as you need. Some tools like planes, chisels, and scissors don't sharpen to an equal V in the centre of the blade; the sharpest point of these blades is on the very edge of the metal. For these blades, you



HOW SHARP?

The internet is full of videos where people sharpen their knives to a razor edge, and then use them to cut through tissue paper or a single human hair. This is definitely something that you can do if you have the time and equipment, but it's entirely unnecessary for general workshop tools. A razor-sharp blade is easy to nick and will blunt quickly on harder materials. You'll probably find yourself stopping to sharpen it every few minutes.



YOU'LL NEED

- Whetstone
- Scythe-stone
- Mill file
- Three-point file
- Angle guide or protractor

only really need to sharpen the angled side of the blade, unless there are any burrs that stick up from the flat side of the blade. For planes and chisels, in particular, you might find it easier to sharpen the blades if you use a wedge to help you keep the correct angle between the stone and the blade.

The trickiest types of tool to sharpen with whetstones are scythes, sickles, and bill-hooks. Unlike knives that might have a convex curve near the end of the blade, these tools have a concave curve that makes it hard to get good contact between a normal whetstone and the edge of the blade. For this type of tool, you can buy conical barrel-shaped whetstones that will better match the curve of the blade.

> "You should carefully consider the type of grinding wheel you use"

The technique for these shaped stones is the reverse of a normal whetstone. You hold the blade stationary while working the stone along the length of the blade.

AVOID THE DAILY GRIND

If you want to sharpen your tools more quickly, you might consider using a bench grinder, wet and dry grinder, or a linisher (a belt sander) instead of a whetstone. Aside from the obvious ferocity of grinders and linishers, you need to take a lot of care when using them to sharpen a blade. You should carefully consider the type of grinding wheel you use.

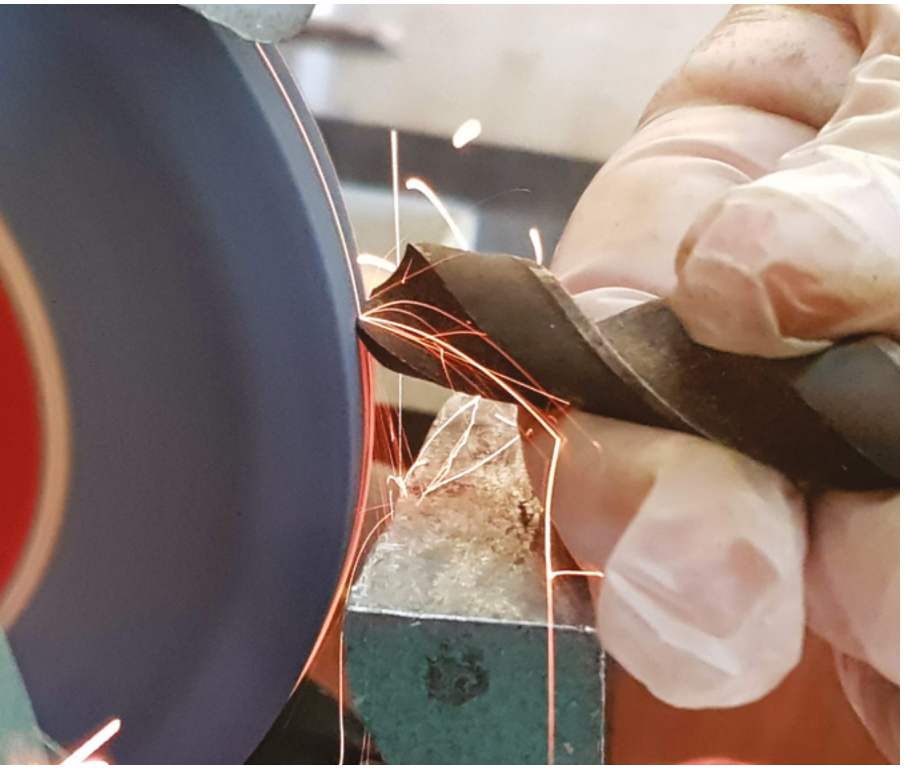
Grinders and linishers move an abrasive belt or disc at very high speed, and that will remove a lot of material from your blade in a very short time. Finer belts or discs are much better for sharpening tools on a grinder. The high speed of the grinder also results in more friction, and that heats up the part that you're working on. If your blade gets too hot during grinding, it can ruin the tempering and hardening of a blade. To help stop this from happening, you can apply some light oil to the blade before you start grinding. Move the blade evenly across the abrasive surface applying a minimal amount of pressure, and keeping a fixed angle between the edge of the blade and the wheel or belt. Remember that even a small bench grinder has enough power to pull a blade from your hands and fire it at high speed across the room. Safety is very important with these types of machine. →

QUICK TIP

A 400/1000 grit double-sided whetstone is a good choice in a general workshop.

Above
The right amount of oil and light finger pressure will help keep the blade moving smoothly without dripping all over the table

Left
Finding a comfortable position to sharpen a curved blade is half of the battle. You should always have a firm grip on the tool handle and the stone



Above Use the rest on the grinder to support your finger, not the drill. You'll get more control that way

Right It feels very unnatural to run a file across the top of a saw, but it's the only way to get the top of the blade level

MAKE A WHOLE HOLE

If you have a bench grinder, you can use it to sharpen your drill bits. Sharpening a drill bit isn't as daunting as it looks, as you only really need to set the point angle of the drill (the angle of the cone at the end of the drill), and make sure the centre of the drill is even. Most general-purpose drills have a point angle of about 118 degrees, which is an angle of 59 degrees per side. This angle is fine for most wood, metal, and plastic drilling needs. Use an angle guide to set 59 degrees, and then begin grinding the point of the drill evenly on one side, checking against your angle guide regularly. If you're struggling to set your angle, you can either cut a template out of card or plastic, or even just get the angle from a drill that's already sharp. When you're happy with the angle, switch to the opposite side and grind that in the same way. Keep watching the point of the drill, and make sure that you don't grind it off-centre.

A SAW POINT

Saw blades are probably the most time-consuming type of blade to sharpen, and the sharpening process has several different steps. To sharpen a saw blade, you'll need a mill file (a flat file) and a three-square (triangular) file. You'll also need a special tool called a saw-set, which you'll use to set the position of the teeth to either side of the saw. Once you have those things, you need to know whether you're sharpening a cross-cut saw or a ripsaw.

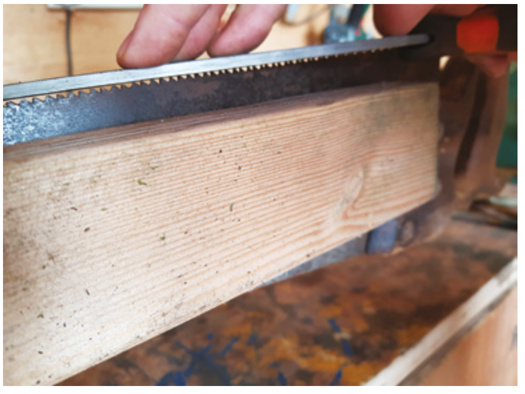
QUICK TIP

Ripsaw teeth have an angle of 95 degrees from the blade, while cross-cut saws have an angle of about 75 degrees.

> **Sharpening a drill bit** isn't as daunting as it looks

Begin the sharpening process by putting your saw into a vice with the teeth facing upwards, and use the mill file to flatten the tops of the teeth so that they are all the same height. If you find that the blade is too flexible to file, use some pieces of scrap wood to support it on either side of the vice. You'll know you're done when you can see shiny metal on the top of every tooth. It might seem counter-intuitive to do this, but saw blades do not always wear evenly, and flatting the tops (also called jointing) makes sure that you'll be sharpening the teeth evenly in the next step.

With the tops of the teeth level, run some masking tape a couple of millimetres below the bottom of the existing teeth (the bottom of the saw tooth is sometimes called the gulley or valley). You're doing this so that you have a straight edge to work to, so pay attention and keep the measurement as close as you can. Use your triangular file to recut the teeth. Start at one end of the saw, and keep filing the gulley until you reach your tape line. Looking at the top of the teeth, the flat spots you made with the mill file should now be spikes again.



SHARPENING **STATION**

As an alternative to a bench grinder, you can get small desktop sharpening stations from most DIY stores. Typically, these sharpening stations will come with a small diamond grinding wheel on a low powered motor, and they have several jigs and guides to help you hold different tools in the right position while you sharpen them.

Left You don't always have to reshape the teeth on your saw. If the blade isn't too worn, you can just go ahead and set the teeth, then sharpen them

The teeth are now shaped correctly, but they're not sharp yet, and they're not in the right position relative to the blade. You'll need to use the saw-set to push the teeth to the left and right of the blade. The easiest way to set the blades is to work all the way down the saw, setting the position of every other tooth, and then flip the saw around to do the other teeth. Make sure you don't skip a tooth, as it can be very frustrating to go back and correct your mistake later on.

You can fine-tune how much you bend the teeth to suit different tasks. Wider spacing will increase the width of cut, but the saw will be less likely to bind. A general rule of thumb is to set the teeth about 1/3 of the width of the blade on either side.

The final step in the process is sharpening the teeth using your three-point file. If you're sharpening a ripsaw, you'll file the teeth perpendicular to the blade. Cross-cut saws usually have a slight angle to the blade, so they will be about 75–80 degrees (with the point of the blade cutting when the saw pushes), rather than the 90 degrees of a ripsaw. Again, it's easiest to work down one side of the blade first, sharpening all of the teeth on the left side before flipping the saw over and sharpening the right side. If you're having trouble judging the angle of the file, you can create a simple guide by part-cutting through a scrap of wood at the appropriate angle. Just place the wood onto the blade and use this to guide you.

You can also use this technique to sharpen an electric reciprocating or oscillating saw. The replacement blades for these saws are quite expensive, so it can be worth the time it takes to set and sharpen a blade in this way.

Hopefully, you'll find that sharpening your tools using these techniques will get your projects running much more smoothly. Blunt tools are very frustrating to use, and just a little extra effort with a whetstone can save you from hours of extra work, and allow you to breathe extra life into expensive consumables.

QUICK TIP

You can sharpen old files by soaking them in vinegar for 24 hours, then rinsing and oiling them.

Left The numbers on a saw-set represent the number of teeth per inch of the saw you're setting. It's just a guide though, so you can experiment with different settings

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

DIRECT FROM
SHENZHEN

Third hand

Make soldering easier and avoid gassing yourself

By Ben Everard

@ben_everard

Above Five extra hands and a fan make soldering easier

Third hands are wonderful. They're devices for holding your work while you solder. Typically, they include two or more crocodile clips on movable arms that can hold not only the PCB, but also a component in place. They're so useful, we'd be tempted to call them an essential part of an electronics tool-kit.

There are a few basic differences between different models – how they attach to the work surface, how many arms they have, how flexible they are, and what extra attachments they have. Let's explore these a bit.

We've never had much luck with third hands with suction cups to stick them down. These never seem reliable enough in our experience. Hands with screw clamps (similar to G-clamps) can work, but only if you have a work surface they can clamp to, and even then, they can only be on the edge of the surface. Our favourite option, by a significant margin, is those with a large, heavy base that doesn't need additional clamping.

Two arms work well: that gives you one to hold the PCB and one to hold what you're soldering, provided you're only soldering one thing at a time and the PCB is light enough to be held by one arm. For more complex soldering, three or four arms are much better. Above about four arms, there seem to be diminishing returns – we very seldom find ourselves not able to hold our work in place with four.

There are two common types of arms: those made out of metal rod with two or three joints, and those that are almost endlessly flexible. There are advantages and disadvantages to both. The rod-and-joint style can be a bit tricky to get into position, but usually can be locked with screws and are then fixed in position. The more flexible arms are easy to get into position, but can then wobble and move.

Left The plastic pads on the handles are prone to coming off, but they're not functional anyway

DIRECT FROM **SHENZHEN**

There are a few extra attachments you can get – magnifying glass, fan, and light are all common. Magnifying glasses can be useful for fine work, but we find that only the rod-and-joint-style third hands are capable of holding work still enough to be useful for this level of fine work. Of course, depending on your eyesight, you may find magnifying lenses more or less useful.

Whether or not a light is useful to you will depend on your workshop lighting.

Since switching to lead-free solder, we've found soldering fans to be essential parts of our desktop. Even with leaded solder, there are quite a few fumes given off, and that stuff is not good for your health.

With all that in mind, we set off to find our perfect third hand on a direct-from-China website to see what it would cost.

We got a NEWACALOX Multi Soldering Helping Hand Third Hand Tool from the NEWACALOX official store on AliExpress for £20.93, including delivery to the UK, and set out to do some soldering.

The arms do have a certain wobble in them, but no more than we'd expect for fully flexible arms, and the number of arms means you can double up to minimise this.

The fan is a computer fan with a USB connection that has an M3 bolt through one corner, attaching it to one of the arms. This has the advantage of making it easy to position in the most effective place for your particular piece of work, but it has the downside of feeling quite flimsy. The USB cable isn't particularly long, so you'll either need a power source very close by, or a USB extension lead. We put our laptop on our desk and used that, but for more regular use we'll splice in a bit more cable to make it long enough to be useful.

The crocodile clips are covered in heat-shrink tubing to protect the component being held, and plastic covers on the arms. The latter are prone to falling off, but this doesn't matter as they don't do anything anyway. Overall, the clips are big enough to hold most things we want to solder, and firm enough to hold them securely.

> "The arms do have a certain wobble in them, **but no more than we'd expect for fully flexible arms**"

The base is just heavy enough to hold itself in place. Any lighter and we'd worry about it moving about too much. We would feel a bit more confident if it were heavier, but we haven't had any problems with it in use.

The base includes recesses that can be used to hold components, but we find it a bit fiddly to use them, as the arms are quite close together and you risk knocking them when trying to pick something up. Having external component bins works far better.

Overall, this third hand is a great tool for soldering with, especially if you hack the USB cable on the fan to be a bit longer. Fortunately, we've got just the thing to hold it in place while we solder up the new cable. □

BEST OF BREED

ONLY THE BEST

Tactile inputs

A surprising variety of keyboards for your next electronics project

By **Marc de Vinck** @devinck

Below If you'd rather go the full DIY route, you can build your own keypads from scratch

Sometimes old school is cool. And you can't get more old school than keyboard inputs. And that's what we'll look at in this Best of Breed. There is something enduring about a simple key press. Just ask any gamer about their keyboard and you'll most likely get into a lively conversation about the best layouts and specific brand of keys they like to use. Even in the DIY electronics world, we know it's important. Just look at Pimoroni's Keybow: it's available in different 'feeling' keys, both clicky and non-clicky.

Keyboards are also incredibly useful and extensible. And you don't need to simply input characters. They can be used for entering code patterns, movement, and – this author's favourite use – programmable displays for creating sequenced music. Another nice thing about having a keyboard input with your project is it's very natural for humans. OK, maybe not natural, but at least it's very familiar. If your project requires user input or feedback, you won't need to explain how to use a keyboard.

Keybow 12-key Kit vs Adafruit Trellis Monochrome Driver

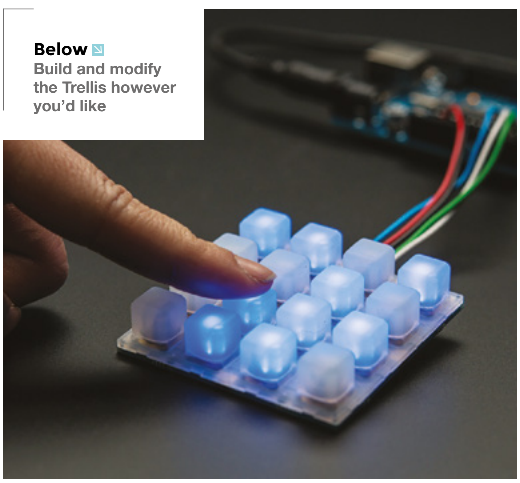
Hackable mini keypads

PIMORONI $60 | shop.pimoroni.com

ADAFRUIT $ 9.95 | adafruit.com

The Pimoroni Keybow is an easy-to-build mini mechanical keyboard kit that doesn't require any soldering. It's powered by Raspberry Pi and includes twelve beautifully illuminated keys. You can customise the inputs with a variety of macros, or just use as a simple keyboard.

Clicky or non-clicky is really the only decision you'll need to make when you pick up the Keybow kit. And unlike many other kits or shields, this really includes everything – OK, almost everything – you need to get up and running, even a Raspberry Pi Zero WH. The only additional item you will need is an SD card, which you most likely have already. The quality of the keys and PCB is really good, which is what we expect from any Pimoroni kit. If you want a self-contained mini keyboard, this is it. And if you are looking for something a bit smaller, check out Pimoroni's 3-key Keybow kit. It's just as good as the 12-key, but a bit less expensive and a lot smaller. It's a great way to get user input in a small form factor. □



Below
Build and modify the Trellis however you'd like

Below
A really well-made and complete kit

The Trellis driver PCB assembly from Adafruit is an open-source backlight keypad driver system. It's really easy to use, thanks to an extensive tutorial from Adafruit, and highly customisable. This Trellis PCB is made to match the Adafruit 4×4 elastomer keypad and has areas in the centre of each button for customising with 3 mm LEDs.

You will need a microcontroller to control the Trellis, as the circuitry on the board only handles the background key presses and LED lighting of the tile. Any Arduino, or similar microcontroller, can handle reading of the key inputs and configuring the LEDs.

Our favourite part about the Trellis is its ability to be expanded and be customised. Each Trellis PCB has an I²C-controlled LED sequencer and keypad reader already on it. The on-board chip can control all of its 16 LEDs individually. The same circuitry reads any key presses. You can connect up to eight PCBs together via their edge connectors. Each board will share the same power, ground, interrupt, and I²C clock and data pins. This makes building a huge 128-button matrix really fun and easy! Just remember to pick up the required matching membrane and LEDs. □

VERDICT

Pimoroni Keybow

A great solution for adding keyed input to your Raspberry Pi.

10/10

Adafruit Trellis

Customisable, affordable, and extensible.

10/10

BEST OF BREED

Adafruit 1×4 Membrane Keypad

ADAFRUIT $2.95 | adafruit.com

Above You can easily wipe these keys to keep them clean

This Adafruit 1×4 Membrane Keypad has four input buttons, each with its own wire for hooking up to your project. And since all of the buttons are wired individually, you don't even need a microcontroller to use it. Use it in your next project just like any simple button or switch. The membrane has a soft touch, and it can easily be mounted by removing the paper backing, exposing a strong adhesive. Adafruit includes a 5-pin, extra-long header strip so you can easily plug it into a breadboard.

> Use it in your next project just like **any simple button or switch**

VERDICT

Easy to clean and affordable.

8 /10

ADAFRUIT 3×4 MEMBRANE MATRIX KEYPAD

ADAFRUIT $3.95 | adafruit.com

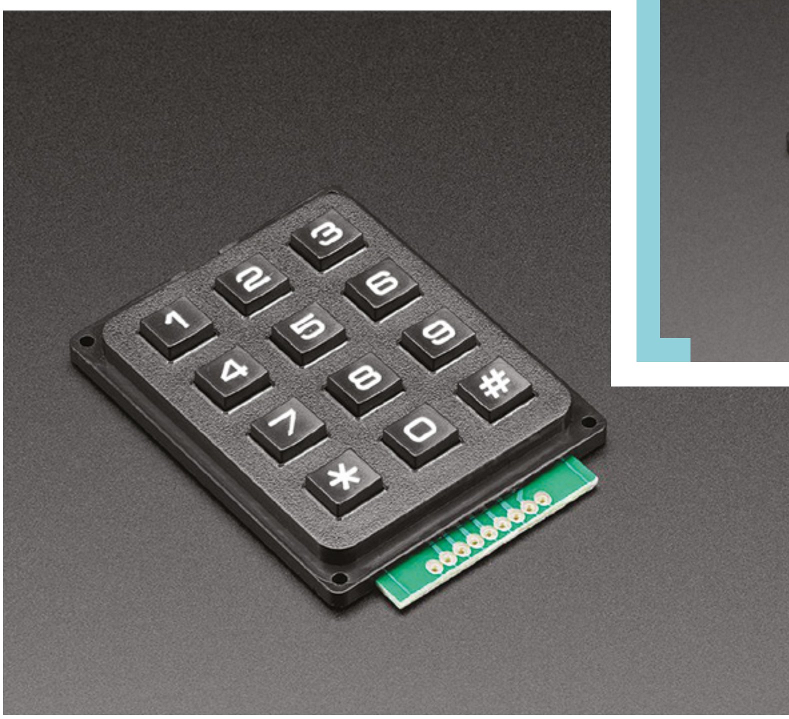
Adafruit also sells a larger version of its membrane keypads if you need extra inputs. It's a similar hook-up process, and is just as easy to get up and running thanks to Adafruit's online tutorials and projects. Just keep in mind, this version will require a microcontroller.

3×4 Matrix Keypad

ADAFRUIT $6.50 | adafruit.com

> Adafruit includes a library to get your **Arduino project up and running**

Got a project that requires a stealthy code to be input? If so, this is the keypad for you! The Adafruit 3×4 Matrix Keypad has twelve buttons, arranged in a traditional telephone keypad grid. They are wired in a matrix, which can easily be read by a microcontroller using only seven pins. Adafruit includes a library to get your Arduino project up and running. A row of header pins is also included so you can easily prototype with it on a breadboard.



Left: A great way to get fast and reliable keypad input

4×4 MATRIX KEYPAD

ADAFRUIT $5.95 | adafruit.com

Just like Adafruit's membrane keypads, the mechanical matrix keypad is available in multiple sizes. The larger 4×4 matrix is also arranged in a telephone-style layout and will work just as easily as the smaller version with your microcontroller. With this keypad, your code can include a few alphanumeric inputs, making it extra secure.

Left: Need more keys? Try the larger 4×4 matrix keypad

VERDICT

Easy to use and reliable.

9 /10

Button SHIM

PIMORONI $30 | shop.pimoroni.com

Right
A very handy addition for your Raspberry Pi project

The Pimoroni Button SHIM may be the smallest, and most discreet, keyboard-like input available for Raspberry Pi. OK, it's a button bar, but isn't that what all keyboards really are anyway? The Button SHIM gives you five physical buttons, along with a handy RGB status LED.

An interesting aspect of this SHIM is the ability to solder it into the GPIO pins and still allow you to use a HAT or pHAT at the same time. Alternatively, you can solder on the included female header and use it as a simple standalone board. Installation of the required software couldn't be easier: just one line of code typed into the terminal and the package will be installed. Head over to the Pimoroni website to learn more.

VERDICT

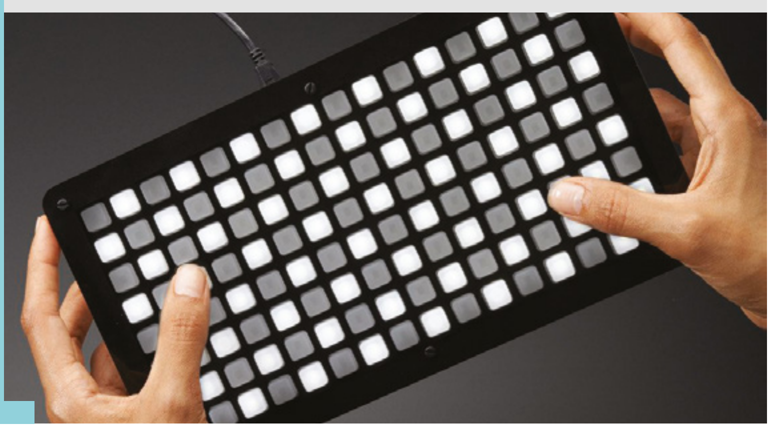
A discreet way to add button input to your Raspberry Pi.

8/10

ADAFRUIT HELLA UNTZTRUMENT!

ADAFRUIT $199.99 | adafruit.com

Is one Trellis just not enough for you? You can have more Trellis with the Adafruit HELLA UNTZtrument! This kit includes eight Trellis control PCBs (16×8 total) and everything else you need to make one cool instrument. And if this is just more fun than you can handle, you could always pick up a smaller 8×8 kit, which is still pretty big!

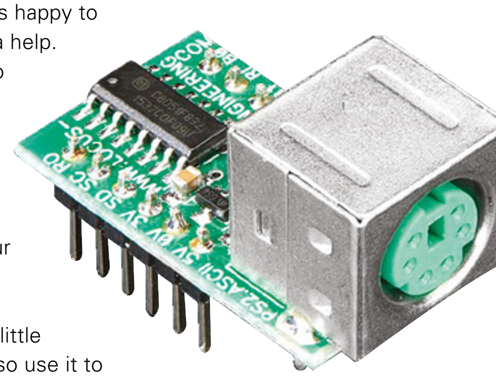


PS/2 Keyboard to TTL Serial Converter

ADAFRUIT $17.95 | adafruit.com

The PS/2 Keyboard to ASCII Converter from Adafruit is a simple solution for producing a single TTL ASCII character on the 'press' of a keystroke from a standard, and ubiquitous, PS/2 keyboard. We've made a lot of different projects using old PS/2 keyboards, and this breakout board is a lifesaver. It's designed to offload the complex keyboard decoding of PS/2 keyboards. This allows your microcontroller, like an Arduino, to tackle other tasks. And as we all know, your Arduino is always happy to have a little extra help.

All you have to do is plug in a standard PS/2 keyboard in one end and start typing, while your Arduino receives the specific keystrokes via a little code. You can also use it to break out magstripe and PS/2 bar code readers. It comes ready-to-go and requires no soldering.



Above
Adding a PS/2 keyboard – made simple

VERDICT

A simple solution for adding a full-size keyboard to your project.

8/10

Can I Hack It?
A PlayStation Classic?

Can we hack this mini retro nostalgia console?

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**

In the late 1990s, Sony released the original PlayStation upon a generation nurtured by Sega and Nintendo. CD-based games with sprawling levels, full-motion video, and high-quality audio became the norm. But, fast-forward to 2018 and Sony released its own mini retro nostalgia console, following the success of the NES and SNES consoles. Alas, supplies of PlayStation Classic exceeded demand, and in 2019 it was put on offer as clearance. And that begs the question, "Can we hack it?"

GENERAL CONSTRUCTION

The all-ABS plastic chassis is made from a firm plastic that can be worked with hand/power tools. The top and bottom of the case are held together with five cross-head screws that tap directly into plastic struts. With the bottom of the case exposed, we can see that there are a further four cross-head screws which need to be removed to enable the circuit board to be released. These screws also hold a large metal cover to the board. The cover is our heat-sink, connecting directly to the CPU via a 2 mm-thick thermal pad. Take care to remove the heat sink without damaging the pad, as it is needed to keep the CPU cool.

ELECTRONICS

The single circuit board is powered by a MediaTek ARM (MT8167A) system on a chip (SoC), with a quad-core ARM Cortex-A35 CPU running at 1.5GHz. This is partnered with 1GB of DDR3 RAM and a PowerVR GPU. Storage is provided by 16GB of flash memory, enough to house the games and the operating system. This is a typical configuration for tablet devices, something this chipset is designed to work with. Power comes in the form of a micro USB port, so 5 V is the norm, and we can solder connections to the port for powering projects. This also means that the board can be powered from a USB battery for portable hacks. In our tests, the unit used 180mA on average at idle; when gaming, this rose to 320mA during a brief bout of Tekken 3 for testing… yes, testing.

YOU'LL NEED

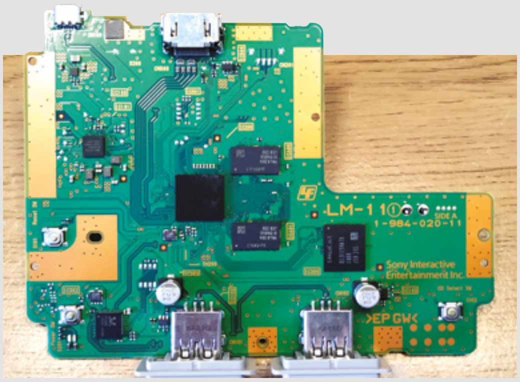
Sony PlayStation Classic

COST

£44

WHERE

hsmag.cc/ymXZkK



Above The main circuit board for the PlayStation Classic is well-made and provides lots of power for emulation

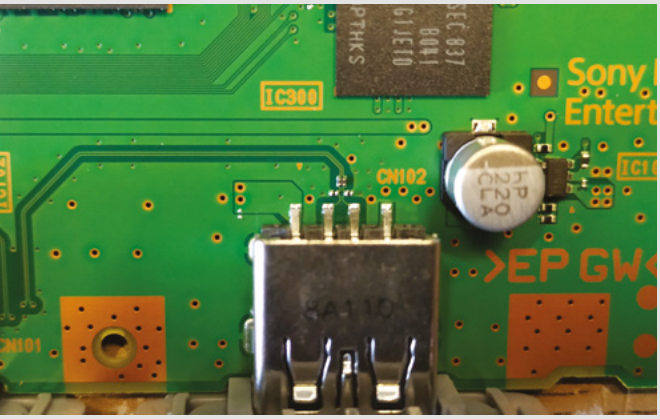
Around the perimeter of the board are large ground planes that will enable us to find good spots to connect our projects to ground. The two USB ports have easy access for hacks, and can be used for the included joypads, or for USB drives – more later.

HACKABILITY

But can we hack this? Well, yes! Cosmetically, we can remove the board from the PlayStation chassis and insert it into a 3D-printed case, with a small HDMI screen, and break out the USB ports. Adding a USB battery would enable the kit to be used on the move.

But, electronically, this is really just a single-board computer similar to the Raspberry Pi. It's running a version of Linux, and so it is just a computer. The two game controller USB ports are the gateway to fun. Using some software called BleemSync, it is possible to adapt the operating system on the device and run your own games from a USB drive plugged into port two. BleemSync also provides a neat interface via a web browser and a direct USB connection to your computer.

Below The USB ports can be used to connect flash storage, externally or internally via soldering for a clean look



ROM PROBLEMS

Game emulation is a legal grey area, but the common points to note are that if you are downloading the games, collectively known as 'ROMs', from the internet, then you are breaking the law, and opening yourself up to possible viruses and malware. If you are using your own game CDs/cartridges to create backups, known as 'dumps', then this is still illegal. The whole ROMs debate has raged for many years and across many systems, and it will continue to rage for many years to come.

Some games are 'abandonware', and the creators/publishers have disappeared into the mists of time. But the majority of games have a publisher or owner who can claim copyright theft. Nintendo recently issued take-down notices to a number of websites hosting games for Nintendo consoles as old as the NES.

To blur the lines of this issue even further, in recent months there have even been legally dumped ROMs on **archive.org** which can be downloaded for use or played directly in the browser. Before engaging in this activity, it would be prudent to review the laws for your country.

We at HackSpace magazine do not condone software piracy or copyright theft.

If you would like to add some extra tech, then the 5 V and handy GND planes mean that we can add an Arduino, or other microcontroller, to control NeoPixels and many other types of LEDs.

When the PlayStation Classic came out, it was overpriced and featured an underwhelming game selection. But, as something that we can hack, the PlayStation Classic offers a conveniently sized chassis into which we can add our tech, hack the existing software, and enhance the product to meet our needs. Right now, this is being sold off with drastic price cuts – it was £15.99 in Amazon's recent sale. At this price, it's a great option for a hacking platform, whether you're looking to build a DIY handheld gaming rig or something more adventurous.

Above Attaching a USB drive to the internal USB port connectors means that we can hide USB drives full of games!

SparkFun Edge

Artificial intelligence stripped bare

SPARKFUN $14.95 | sparkfun.com

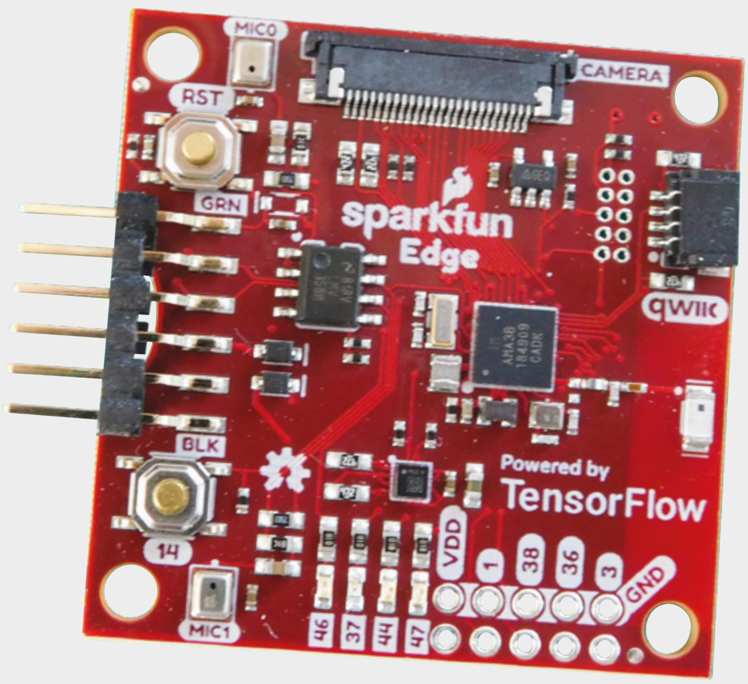
By **Ben Everard** @ben_everard

At the moment, artificial intelligence is in a transitional phase where words have different meanings to different people. No one's lying; it's just that the AI world hasn't quite decided how to describe the various aspects of embedded AI. For starters, what is the Edge? Very briefly, it refers to AI that happens on a device itself. The easiest way to understand this is to look at the alternative: for example, the way most popular voice assistants work. When you speak to them, they upload the audio to some cloud service and that cloud service works out what you're saying, what this means, and sends the commands back to the device. This has a few disadvantages – it requires a permanent connection to the internet, it sends quite a lot of data back and forth, and it requires cloud servers to be kept running in order for it to operate.

Edge computing is the opposite of this. It means that some level of AI is running directly on the device doing the sensing. So far, it all seems fairly straightforward. Where it all seems to break down is what people expect these edge devices to be. Last month we looked at Raspberry Pi 4 as an edge device – and it is. It can do some powerful AI while still being cost-effective and low power (at least when compared to a traditional computer). The SparkFun Edge, however, is an order of magnitude smaller than this. It features an ultra-low-power ARM M4 core, a couple of microphones, and just enough other components to keep everything running.

People will argue about which of these is the 'true' edge, but semantics need not bother us. They both have niches in AI that they fill well, and we can just look at what is the best device for our needs.

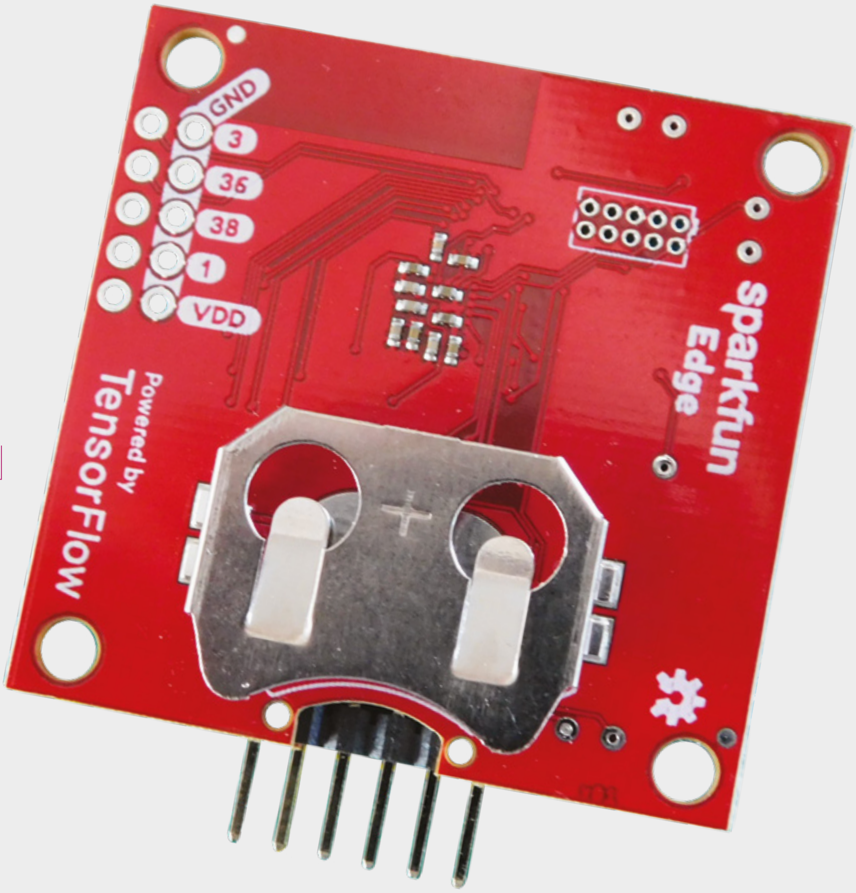
Below At 4 cm by 4 cm, this should fit into even space-constrained projects



LIVIN' ON THE EDGE

We'll start with the issues with the SparkFun Edge. First is that it's not particularly easy to program. You'll need a USB to UART cable, and it's programmed via a downloadable SDK. This is pretty much par for the course with many commercial dev boards, but anyone used to the Arduino IDE, or similar things, will find it a bit of a shock. That said, it's well documented; so as long as you're comfortable using the command line, you should make it through unscathed.

> The hardware is very good for a very specific set of circumstances – **AI without mains power and needing Bluetooth connectivity**



Above There's a battery holder for off-grid running

This refers only to programming the device. Programs take models (which are the matching engines of neural networks) and tell the board what to do depending on how the model reacts to the input. Creating models is a separate task from programming – it exists somewhere between art and science. Essentially, it comes down to throwing a lot of example data at a neural network and attempting to 'train' it to understand particular patterns. In many cases, it's possible to bypass this, and use pre-created models similar to how you might use a library in programming.

Once you've got your code into the device, you need a way to interact with it. This board is designed by SparkFun, and the team at TensorFlow, to run neural networks. These models are first trained to recognise if an input falls into a particular category or not; for example, if a particular sound is a word. On the Edge, there are two microphones, four GPIO pins (that can run SPI or I^2C), a Qwiic connector, and a camera connector. As yet, there's no camera module available for the camera connector.

This is all controlled by an Ambiq ARM Cortex M4F processor with Bluetooth. The Cortex-M4F core is one of the more powerful microcontrollers around, and this one runs at 48MHz (with a 96MHz burst mode). What makes this stand out is that it does this while drawing under 2 mA of current at 3.3 V – great if you're running on battery (there's a CR2023 holder on the back), solar, or other limited power supply.

CONNECTION CONUNDRUMS

Perhaps the biggest limitation for this board as an edge device is the connectivity. Bluetooth makes sense from a power and cost perspective, but it makes your overall system setup a bit more complex as it'll need something to pair with to send data into the world.

This can make sense, particularly if you're running a number of devices in a small area and power consumption is an issue, or if your devices only need occasional connectivity.

You need to be realistic with your expectations with the SparkFun Edge. If you're looking for high-precision matching of complex input – such as recognising people from images – then you're probably going to struggle. However, if you're looking for something to run on very low power and react when it recognises one of a small number of conditions, then this might well be the board for you. At the moment, the lack of pre-prepared models means that it's not really suitable for casual uses; however, given that it's an official board by TensorFlow, we expect that there's likely to be more in the future, so keep an eye out to see what models currently exist before making a purchase.

The hardware on this board is very good for a very specific set of circumstances – AI without mains power and needing Bluetooth connectivity. While these are quite specific, they're not all that rare. This lets you bolt on neural networks to remote sensor deployments. For those conditions, this board stands alone.

VERDICT

For low-power AI with Bluetooth connectivity, the hardware's here, but you'll need to see if there are the models you need.

8/10

Creality Ender 3

Hot plastic fabrication on a budget

CREALITY ◈ around £180 | creality3d.cn

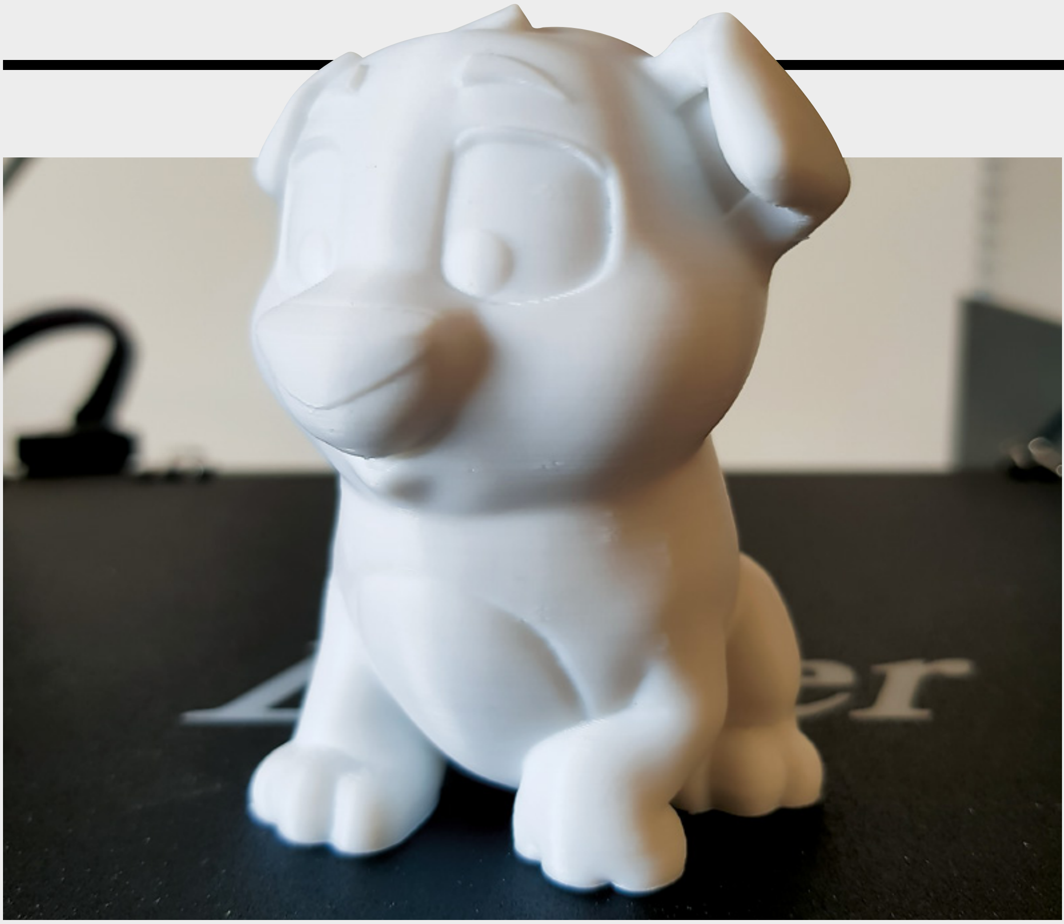
By **Glenn Horan** @BatGlenn13

Right ◈
The minimal and solid-looking Ender 3 printer

Only a few short years ago, the thought of getting a consumer-grade 3D printer, that works out of the box, for anything under £1000 was a maker pipe dream. Today, thanks to Creality, we've very nearly got just such a printer for under £200.

The Ender 3, like many printers in this price range, arrives as a kit. Opening the box, we are greeted with a printer that's around 80% assembled and includes a full-colour manual detailing the twelve-step building process. There's a selection of essential tools required to build the printer, and some additional tools that are useful for its day-to-day operation (a selection of Allen keys, a scraper, a microSD card, and USB adapter – to name but a few). All of the parts are well labelled, with the screws neatly organised into labelled bags. From box to first print takes between one and two hours, and there is plenty of help available online, via the Ender 3 subreddit and Facebook pages.

Our favourite thing about the Ender 3 is the frame – it's made entirely from aluminium extrusions, rather than wood or acrylic found in its contemporaries, which gives it an incredibly solid feel. The level of stability this provides means that it can achieve very good-quality prints, even at higher speeds. As if that wasn't good enough, the Ender 3 also boasts a Bowden extruder setup. While the debate of Bowden vs direct-drive extruders is a lively topic among the 3D printing community, Bowden systems do offer a few undeniable advantages. A Bowden setup has the stepper motor that feeds the filament into the hot end mounted on to the frame, rather than on the extruder itself. This means there's less weight being swung from side to



Left
The Ender Dog was this author's first print with the Ender 3 – it turned out great!

side as the printer operates, resulting in the ability to print faster while maintaining print accuracy.

WHAT'S IT CAPABLE OF?

This printer has a long list of excellent features that really make it a delight to use. All the wires are neatly hidden away within the frame, or bundled neatly into the stylish black cable tidy. A spool holder, which sits on top of the machine, reduces the footprint that it takes up on your desk. The power unit itself not only comes with a switch and a removable plug, but it operates at 24 volts, meaning that the bed can heat up in a fraction of the time of its competitors, which are usually kitted out with 12 V units. There are small rubber pads on each corner of the base to help dampen the sound it makes, and there are large wheels on the bottom of the bed to allow very quick and precise bed level calibration. The build plate is also removable and flexible, allowing users to remove their prints easily, cleanly, and with little risk of breakage.

So how does it print? Every Ender 3 comes with a pre-prepared file to do a test print of what we've come to call the 'Ender Dog', and we were very impressed with the results with no calibration or tweaking (with the exception of the necessary Z-levelling). We've consistently been impressed with what this printer can do. The quality of prints, particularly when using a higher resolution, is amazing for this price. If functional prints are more your bag, the sturdy frame and Bowden extruder will theoretically allow you to print things quickly (or at least quickly in the world of 3D printing).

THE BOTTOM LINE

As expected, there are a few small downsides to the Ender 3. The springs on the underside of the bed aren't very strong (more tension in the springs means less build plate movement, and less calibration needed between prints), but it's both cheap and easy to replace these. Printers manufactured before 2019 do not have thermal runaway protection, which is a pretty important safety feature, so make sure that it's enabled on your printer. Unfortunately, the board running the Ender 3 does not have a friendly bootloader, which makes it a little tricky to update the firmware. While there are tutorials online that detail the process, it's definitely an inconvenience if you want to do additional tinkering to improve the printer.

> "If functional prints are more your bag, the sturdy frame and Bowden extruder **will theoretically allow you to print things quickly**"

For the price and print quality, though, we can live with these issues. The Ender 3 has raised the bar for budget 3D printers.

VERDICT

You won't find a better printer in the sub £200 range.

9/10

Sonoff Mini two-way smart switch

Create your own Internet of Things

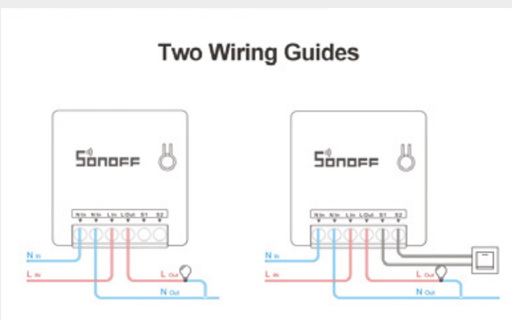
SONOFF £6.40 | itead.cc

By **Daniel Hollands** maker.rip

In issue four we reviewed the Sonoff Basic, a cheap option for people who wanted to make their dumb devices smart by providing power switching via apps and voice control – provided they're prepared to get their hands dirty with a little DIY. Today we're looking at its younger sibling, the Sonoff Mini, and discussing the differences between the two devices.

Much like the Basic, the Mini provides the ability to add smart switching features to dumb devices. What's different is the Mini has the optional ability to attach an on/off switch to it, providing two-way switching control, letting a regular switch work alongside smart control.

The classic use case for this would be adding smart control to the pre-existing lights in each room of your home, while keeping the ability to use your regular light switch, allowing both to provide independent control. It won't matter which state the switch is in, when you flip it, the lights will toggle on or off accordingly, with the same being true of the smart control.



At less than half the size of the Basic, it lives up to its Mini name, which according to Sonoff's marketing materials means it will fit inside a standard EU switch box – but unfortunately, when testing the device we found it didn't fit behind any of our test switches, so if you're planning on using it to upgrade the lighting in your home, you will need to measure the available space inside the switch box first. This is important, not only for aesthetics, but because it needs to sit inside an enclosure due to the live screw terminals which are exposed on the unit.

Much like the Basic, the smart control via the eWeLink app works really well, and we had no issues getting it connected to our WiFi network, or added as a device to our Amazon Echo. Among the features of the app are the ability to set a regular schedule, timers, and interaction with the IFTTT service, letting you do things like, automatically trigger the switch when the ISS is overhead, for example.

One new feature is that of LAN access, meaning smart control can continue to work independently of a cloud service, which solves the previously reported issue of the cloud service occasionally going down, and bypassing potential issues with an unstable internet connection.

Also new is a DIY mode, aimed at developers, which provides a REST API for direct control of the device, which is enabled by opening the casing and attaching the included jumper. The documentation and tools for this exist on a GitHub repo, but still appear in their infancy. Provided you have a suitable enclosure, the Sonoff Mini provides an easy-to-use way of adding IoT control to existing appliances.

Above
A small and compact unit, living up to its name

Left
Diagram showing the two wiring options available

VERDICT

Adds smart control to room lighting in an elegant way, provided it fits behind your switch.

8 /10

Blown Away

By **Ben Everard** @ben_everard

The format for this Netflix show is simple – take a group of glass-blowers and, each episode, give them a challenge and eliminate whoever performs worst. This format is particularly brutal because it doesn't give the competitors a chance to redeem their mistakes – one false move and they're out. When working with glass, it's easy for one false move to result in the utter destruction of the piece you're working on. While this can happen at any time, a particularly risky time is getting the piece off the punty (a metal rod used to hold the glass as it's worked) and into the annealer (an oven that very slowly cools down the glass to room temperature to ensure that it doesn't crack). It doesn't matter how well your glass is worked, if your tap to break it off the punty is a little too hard, or you don't hold it properly, all you've made is a pile of broken glass.

There are parallels to the *Great British Bake Off* in *Blown Away*, and the strongest of these parallels are in the medium, rather than the structure of the competition. Molten glass, like cake batter, is in a constant state of flux. You can influence it, you can work with it, but you can never really control it in the same way you can with, say, metal or wood. It doesn't stay still, so once you've begun the process of creation, you're thrust into a stream of work that it's hard to stop or pause, until you've gone all the way through to a finished product in the annealer.

Blown Away is an entertaining watch in its own right, and we enjoyed getting an insight into the process of glass-making, but we would have preferred a slightly safer competition, where makers were eliminated on the results of two or three makes, rather than having just a single chance each episode.

VERDICT

Great entertainment and some truly impressive builds.

9/10

next month

issue #23

ON SALE 19 SEPTEMBER

Halloween

ALSO

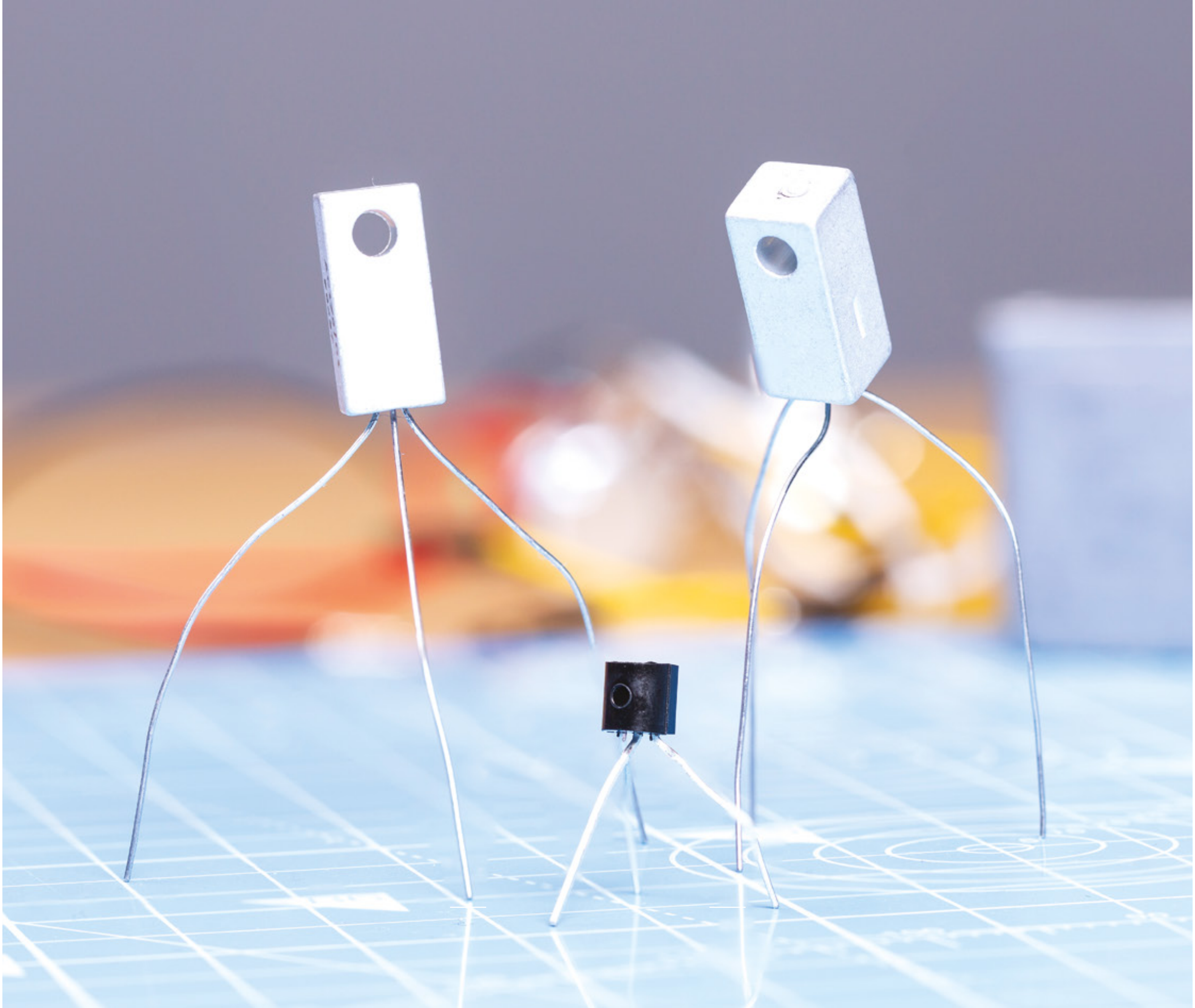
- → ESPCAM
- → WINGS
- → IFTTT
- → 3D PRINTING
- → AND MUCH MORE

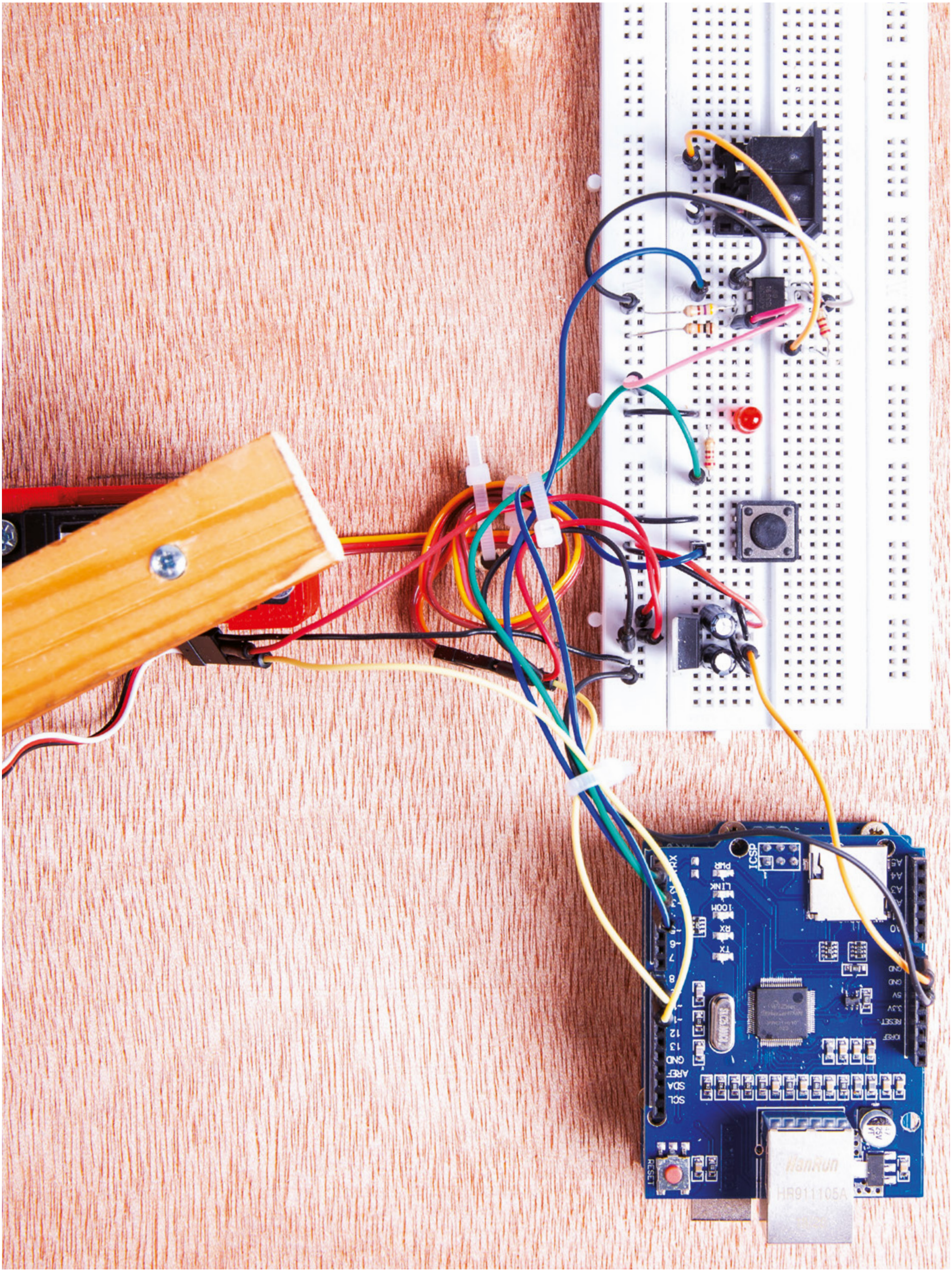
DON'T MISS OUT

hsmag.cc/subscribe

Transistors

Since John Bardeen, Walter Brattain, and William Shockley created the first transistor in 1947, these electrically controlled switches have come out in many different forms. Here, two new old stock germanium transistors look on with pride at a much smaller, cheaper silicon transistor. The smallest transistor created so far was just 2.5 nanometres wide. You could fit approximately 9,273,600,000,000,000 of them on this page.




ICSP
PWR
LINK
100M
RX
TX
GND
AREF
SDA
SCL
RESET
HanRun
HB911105A