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

132 PAGES OF MAKING

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | July 2019 | Issue #20

BUILD A WORD CLOCK

A homemade chic timepiece

DIY VEGAN LEATHER

Animal-friendly cosplay

TOP MAKER TIPS

50 WAYS TO BUILD BETTER PROJECTS

Homemade CNC milling

Hydroponic gardening

The best OLEDs for makers

Laser-cut loom

JEN SCHACHTER

On Rosie the Riveter, hard work, and Barack Obama

July 2019
Issue #20 £6

9 772515 514006 20

Raspberry Pi PRESS

TIN CANS **TAPPING** RADIO CONTROL **SOLDER**

Welcome to HackSpace magazine

Issue 20 feels like something of a landmark issue. We've now produced about ten kilos of hacking, making, and inspiration, so if you've gathered them all on a shelf, now is a good time to check that it's strong enough. Or, if you'd prefer a different comparison, it's about twice as many pages as the complete *Lord of The Rings* trilogy (depending on the version), and our pages are a lot bigger. We're expecting a call from Peter Jackson any day now.

In order to celebrate this issue, we've looked for the best little tricks and secrets to help make things better. Take a look at page 36 for our 50 favourites, but we'd love to find and share more. Let us know your best tips at **HackSpaceMag** on Twitter or email **hackspace@raspberrypi.org**.

I'm personally excited by Andrew Lewis's hydroponic setup (page 88) this month. The slugs are eviscerating our outdoor salad, so this seems like a great solution for us. I also had great fun building a word clock (page 102) which now has pride of place as my kitchen timepiece – it's a slightly tricky build, but endlessly customisable, depending on your personal style and which tools you have available. That's just me, though. Turn the page, and enjoy this issue while we get started on the next 20.

BEN EVERARD
Editor ben.everard@raspberrypi.org

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

get in touch at
hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.org
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

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
Harriet Knight, Sam Ribbits

Photography
Fiacre Muller

CONTRIBUTORS
Lucy Rogers, Andrew Huang, Jo Hinchliffe, Krista Peryer, Cameron Norris, Will White, Mayank Sharma, Dave Astels, Andrew Lewis, Andy Clark, Sophy Wong, Gareth Branwyn, Les Pounder, Marc de Vinck

PUBLISHING
Publishing Director
Russell Barnes
russell@raspberrypi.org

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.org

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

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

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

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

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

Contents

Tutorial

LED patches

Cover Feature

50 TOP MAKER TIPS

Get better at everything with our guide to superior craftsmanship

06

18

122

Tutorial
Desktop hydroponics

88 Grow tasty fresh veg from the comfort of your workbench

Interview
Jen Schachter

60 Why failing isn't really failing (as long as you learn something)

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

Woody

By Josh Warner

SplinterCycles.com

This build took five weeks from where I started building to the first ride. And then it took me six to eight months to get around to putting finish on the wood. You know, it was one of those things: it was ridable, so I was riding it every day. I didn't want to have it out of commission for a week while I was putting finish on it. I took it to the North American Handmade Bike Show a few weeks ago. I was presenting it, so I had to have finish on it. Some varnish and some paste wax, and I think it looks great.

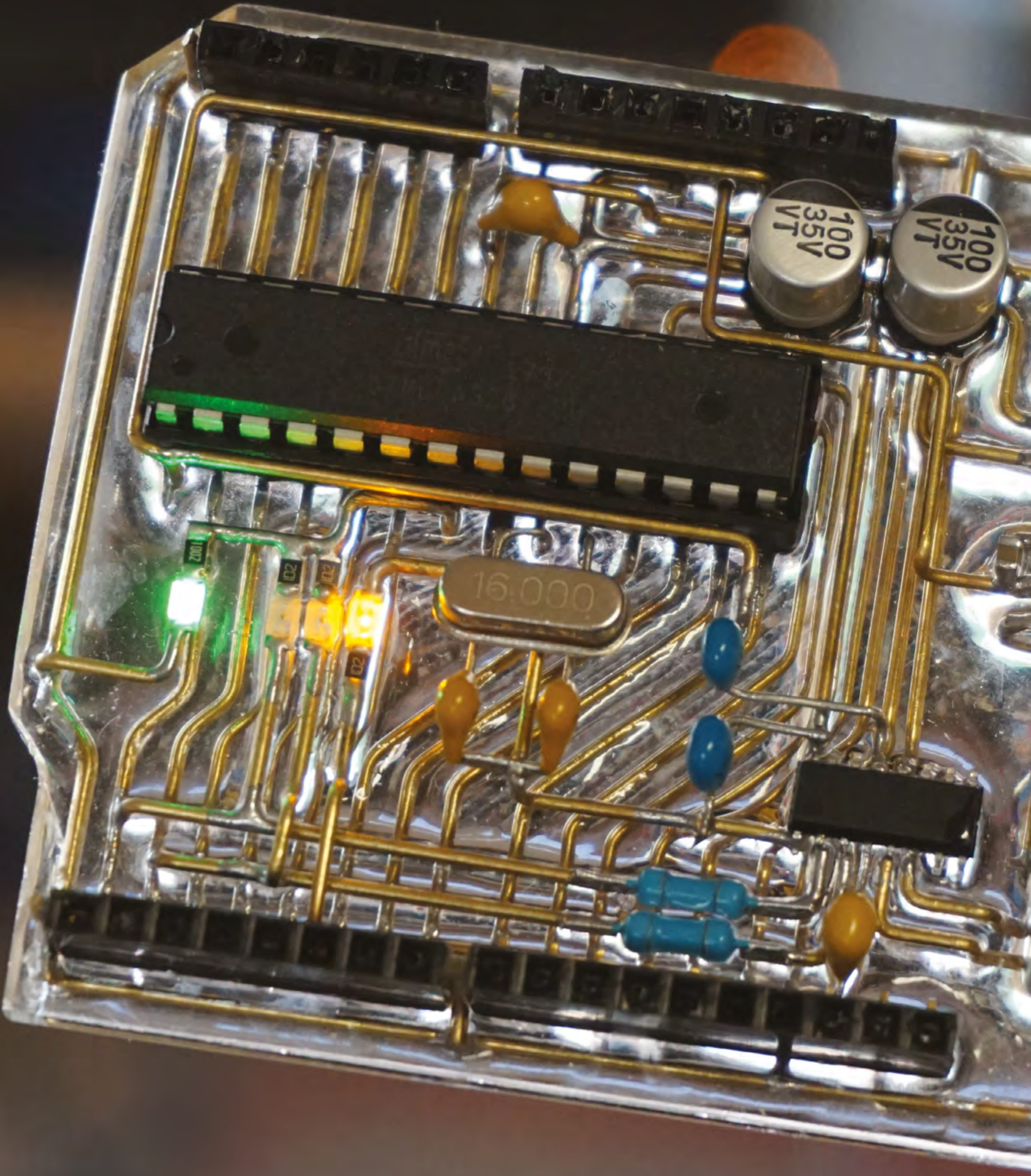
There are four different species of wood: zebrawood, padauk, maple, and wenge. The zebrawood is the stripy one on the outside. Paduak is orange; the maple is the white colour, and then the wenge is that black stripe running down the centre.

I milled the pieces of wood to the right thickness, and then made some templates out of MDF, then match-routed all the shapes for the layers, and laminated them all together. But when I laminated it, I only did the first three layers. Then, I put it on a Bridgeport mill and hogged out 90% of the material on the inside to form a tube. I ran some internal cable routing and then glued the two halves back together. Even when I laminated the layers together, I left them a little bit oversize, and took it down to final dimensions when it was all glued together. I basically made fancy, expensive plywood.

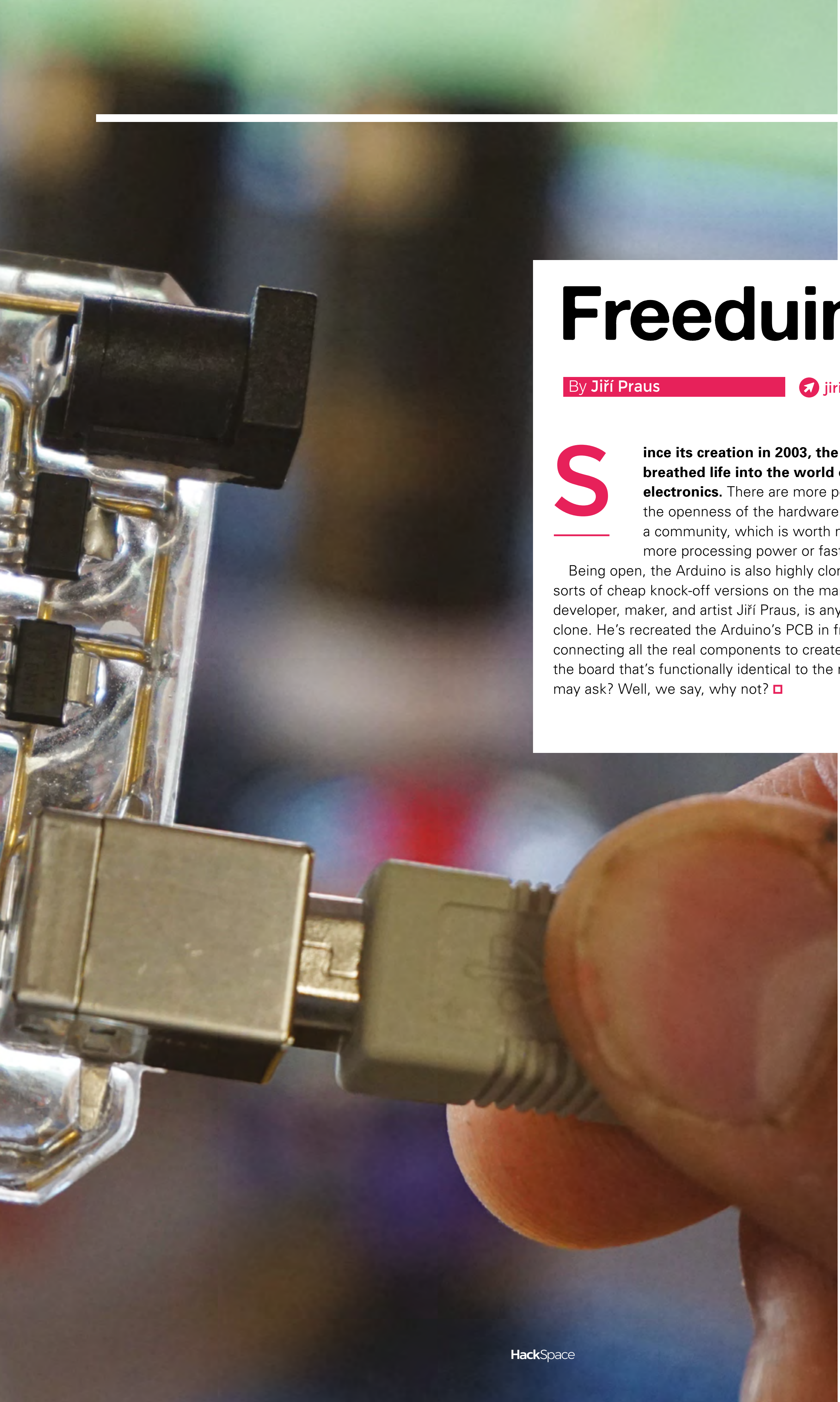
I'm making my next bike out of cherry. I'm doing what's called a tube and lug, which is really classic frame building. Old steel frames, this is how they were all made. They'd have steel tubes and a lug at the intersection where the tubes come together. That has all the complicated geometry in there. The tubes slide into the lugs, and you braze them in place.

I'm doing something similar, but with wood. Wooden lugs, layered and routed like this form, and tubes made from layers of veneer wrapped around a mandrel to form a consistent width. □

Right Wood is flexible, so Josh says that the difference in ride between a steel bike and a wooden bike is similar to the difference between aluminium and steel



100
35V
VT
100
35V
VT
16.000



Freeduino

By Jiří Praus

jiripraus.cz

Since its creation in 2003, the Arduino Uno has breathed life into the world of open-source electronics. There are more powerful boards, but the openness of the hardware means that it's gained a community, which is worth more than a little bit more processing power or faster ins and outs.

Being open, the Arduino is also highly clonable – you'll see all sorts of cheap knock-off versions on the market. This clone, by developer, maker, and artist Jiří Praus, is anything but a simple clone. He's recreated the Arduino's PCB in free-form wiring, connecting all the real components to create a skeleton version of the board that's functionally identical to the real thing. Why, you may ask? Well, we say, why not? □

Right The build took a few days, and yes – it works!

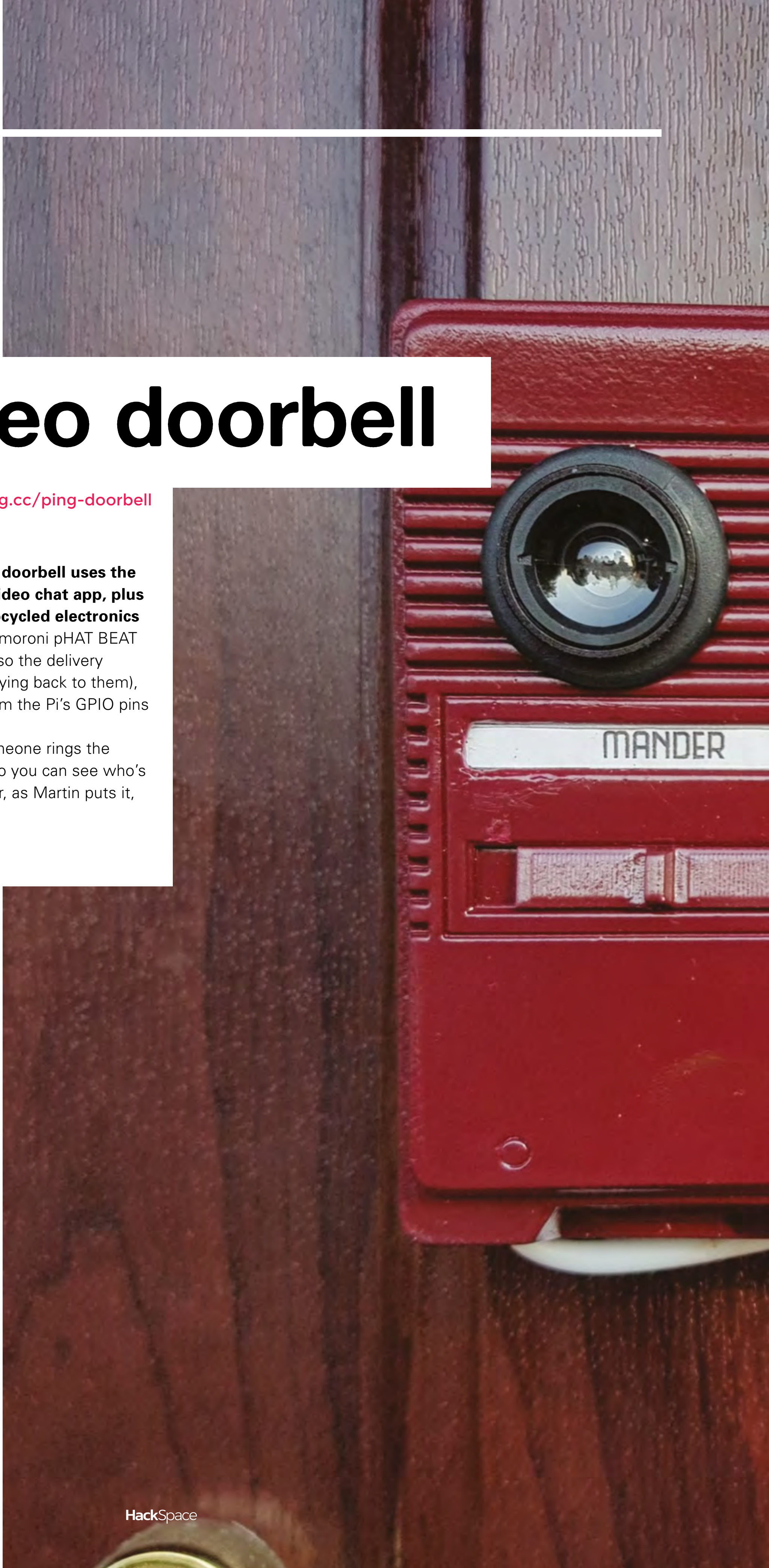
PiNG video doorbell

By Martin Mander

hsmag.cc/ping-doorbell

This WiFi-connected video call doorbell uses the Google Duo browser-based video chat app, plus a Raspberry Pi 3 and some upcycled electronics enclosures. You'll also find a Pimoroni pHAT BEAT board in there to handle audio (so the delivery person can hear what you're saying back to them), plus a Pico HAT Hacker to make the output from the Pi's GPIO pins more accessible.

The Pi runs in headless mode and when someone rings the doorbell, it starts a video chat on your phone so you can see who's at the door, even when you're not at home – or, as Martin puts it, "trapped under the cat". □

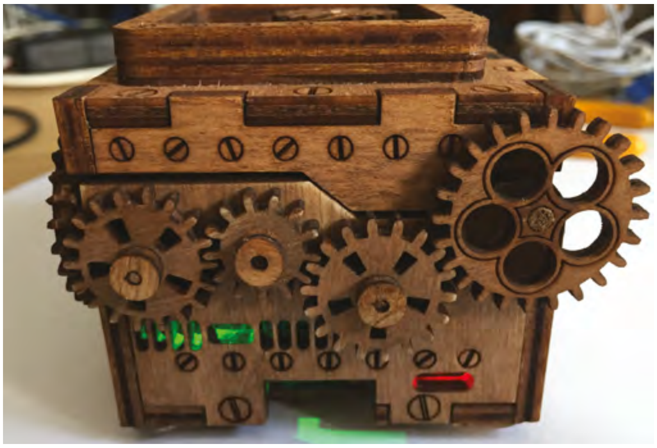


Right PiNG doorbell; freeing felines from disruption since 2019

Steampunk Pi enclosure

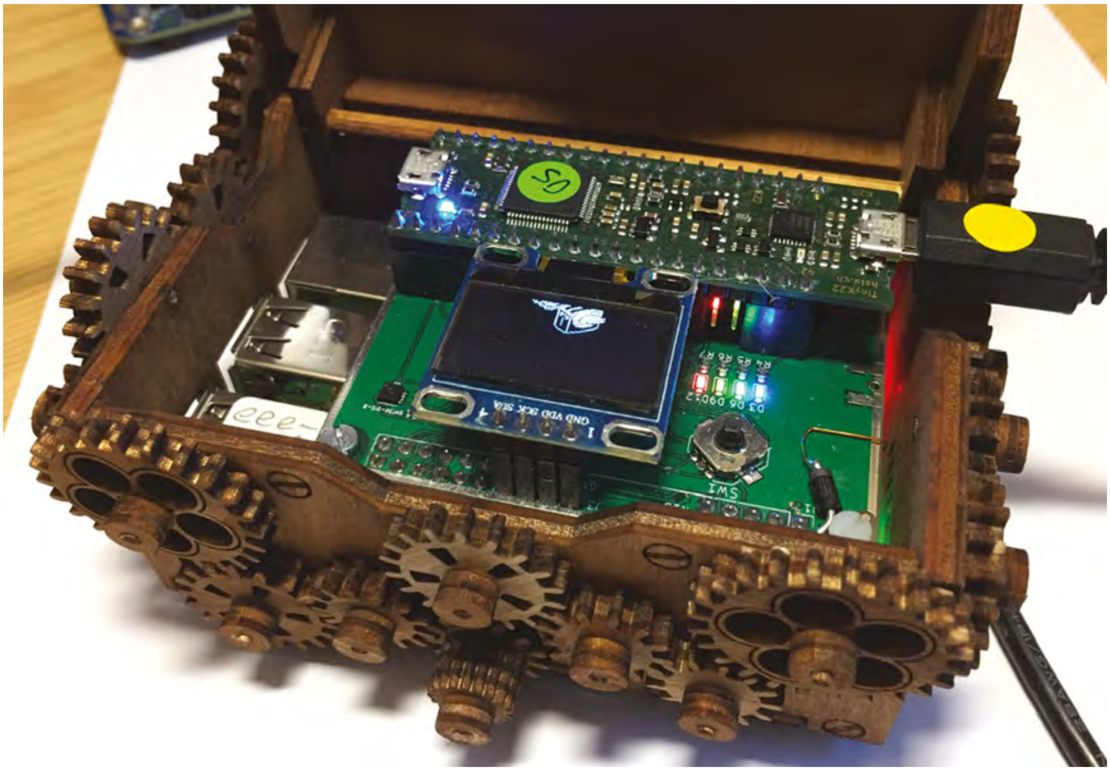
By Erich Styger

hsmag.cc/pi-enclosure

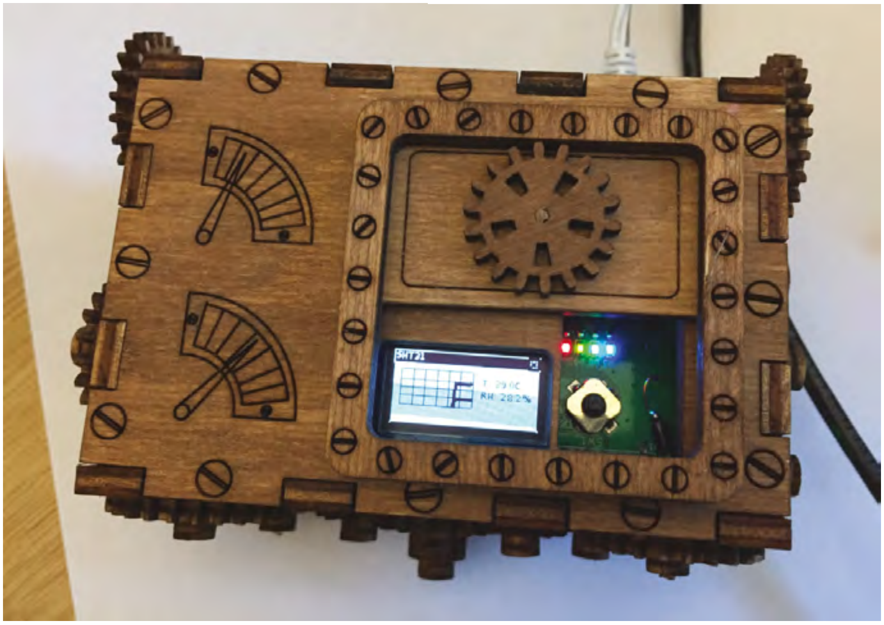


This steampunk enclosure for the Raspberry Pi was laser-cut out of 4 mm birch plywood, and stained to make it look a bit more 1890s. It's built to fit a Raspberry Pi with an NXP tinyK22 board and a battery backup, and there are ports artfully created into it so that the system is fully functional even when the box is closed.

Those gears aren't just for show – turn the central wheel on the front of the box to open the enclosure and get access to the electronics inside. □



Right Erich has put the design files for his enclosure on GitHub at hsmag.cc/styger


SHT21

Trash boat

By Antonio Rizzo

hsmag.cc/trash-boat

I am Antonio, an RC hobbyist and electronics enthusiast! I have been playing with RC toys (and disassembling them!) since I was a five-year-old hyperactive little boy. My first electric toy was a tiny F1 wire-controlled car and, growing up, I kept nurturing my hobbies, becoming an electronic engineer!

I always keep a bunch of old assembled PCBs in a box under my desk, since you never know when they will come in handy to make a new prototype or project. I like the idea of recycling what I can, instead of making a lot of trash that keeps stacking up somewhere on our planet.

Developing uChip (the board that Nicola [Wrachien] and I designed and launched on Kickstarter), I implemented the hardware and software to connect my drone radio to motors/servos. The next step was, naturally, to make good use of the hard work done and build my own RC toy, which is… a boat!

The idea of making an RC boat out of plastic bottles and CD-ROM players is the result of my approach to life: 'Fix it if you can, or recycle whatever is left!'

Furthermore, since I am not a mechanical engineer, I opted for the easiest approach I could imagine and I am proud to say, this time, I exceeded my own expectations! □

Right
Normally we disapprove of plastic bottles in the sea, but we'll make an exception for this

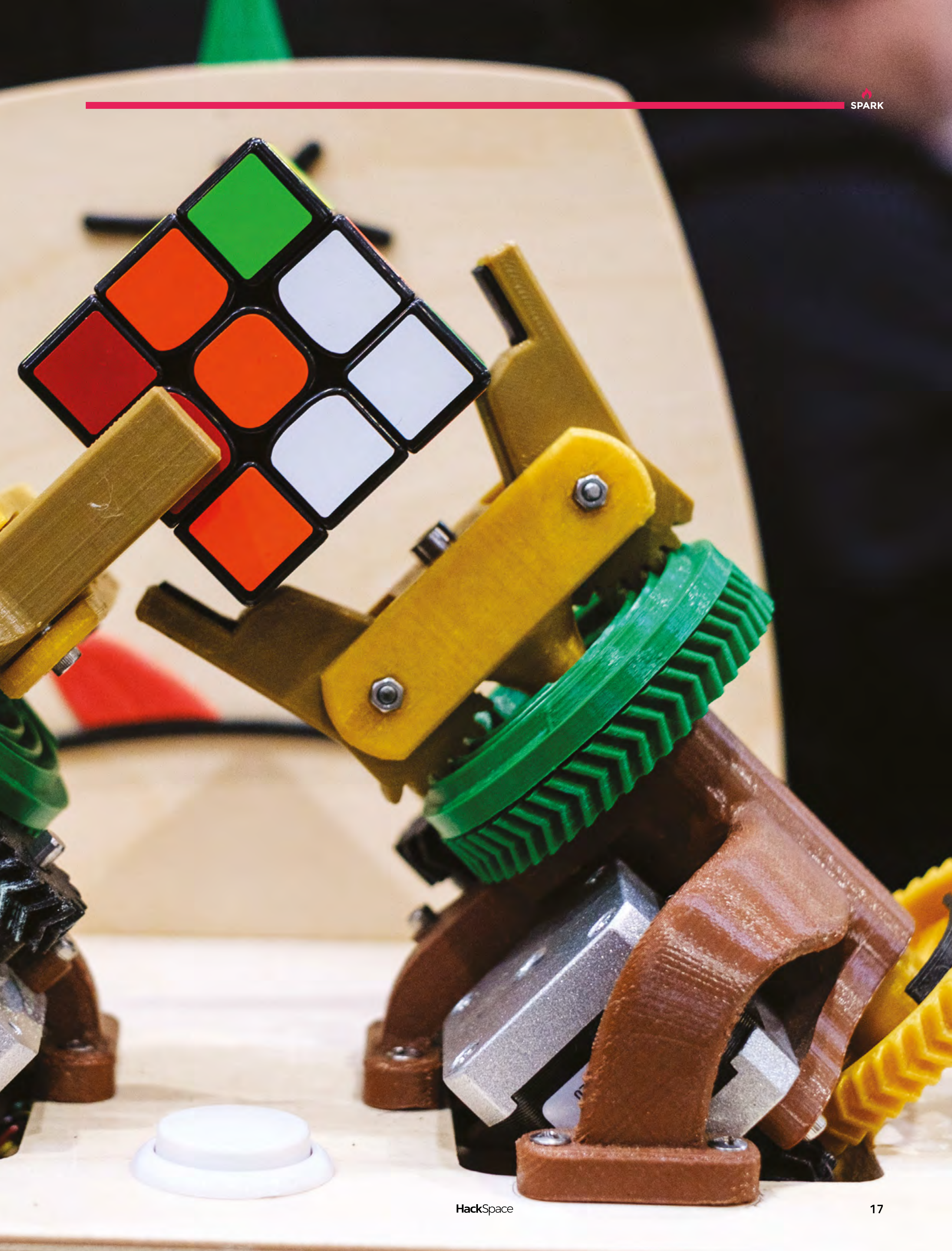
Objet 3d'art

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

We're used to showing off 3D-printed objects for their looks in these pages. Here's one build that uses the unique qualities of 3D printing to achieve an optimal result. Laser-cut plywood gears are high friction, so either break or wear out, and machined metal gears take specialist kit and are a lot more work to make. So, when it came to building the gears for this Rubik's Cube solver, 3D-printed plastic was the best choice.

The robot was built by Matt Keveney of the Ace Monster Toys Makerspace (**acemonstertoys.org**), who used a colour sensor and a Raspberry Pi Zero to control the motors that turn the cube until it's solved. It's a mesmerising thing to see in action, and quite beautifully made. □

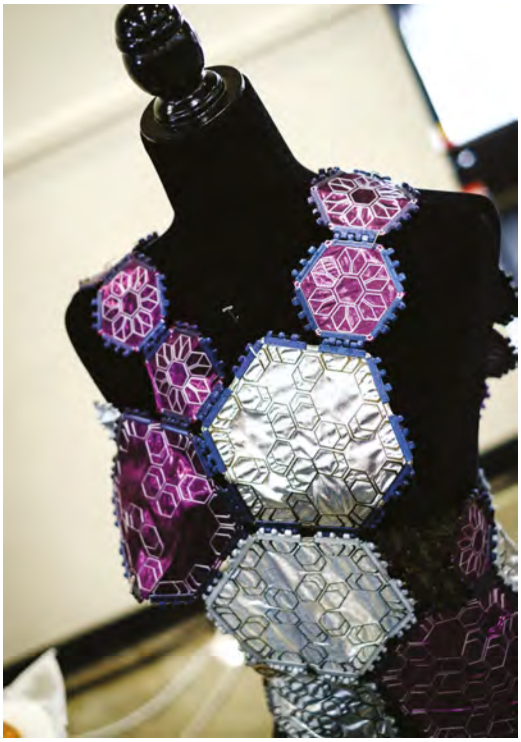
↗ **hsmag.cc/rxFgTU**



Meet The Maker: David Shorey

The master of perfection

There's an extremely high chance you're already aware of David Shorey's work. As a visual artist working in the film industry, he's worked on scenes for the Bond film *Quantum of Solace*, *Mission Impossible: Ghost Protocol*, *The Last Airbender*, *Thor*, and even *Red Dwarf*, among many more.



Right "I print two layers, then I put the fabric down on it, then I print more, and I come up with this. The heat of the 3D printer melts the plastic below so it oozes through and then sets solid"

You might also recognise his 3D-printed work, particularly in cosplay. He is a pioneer of 3D-printing onto fabric, sharing his mistakes and triumphs along the way, so that we can all be a little bit more triumphant.

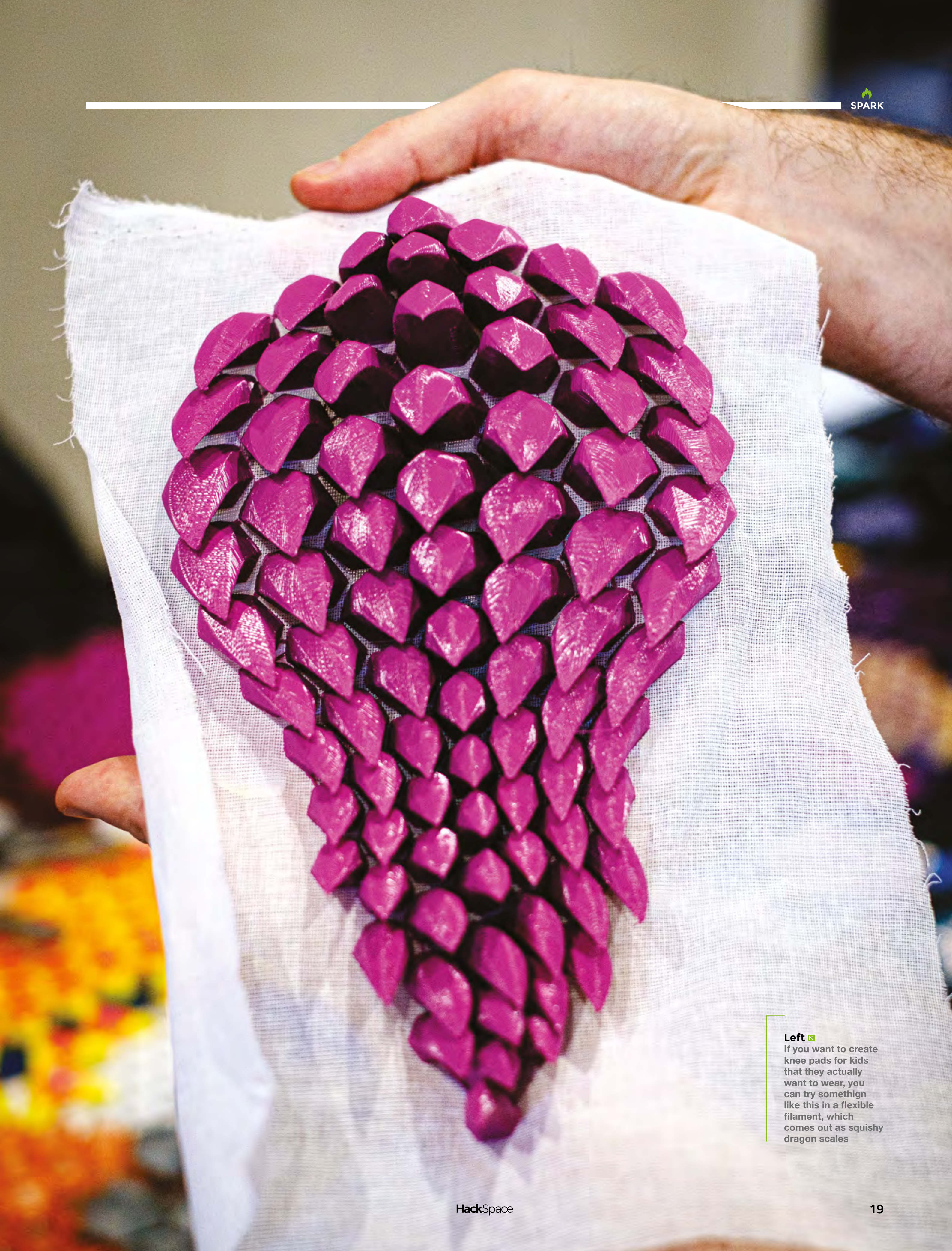
We spoke to him to find out what he's working on, and got a run-through of the techniques he uses to create those cool dragon scales.

"When I first tried to 3D-print onto fabric, I was playing with embedding other things, like popsicle sticks and stuff like that, and my mom does embroidery, so I thought, 'maybe I can try to embed fabric in something 3D-printed'.

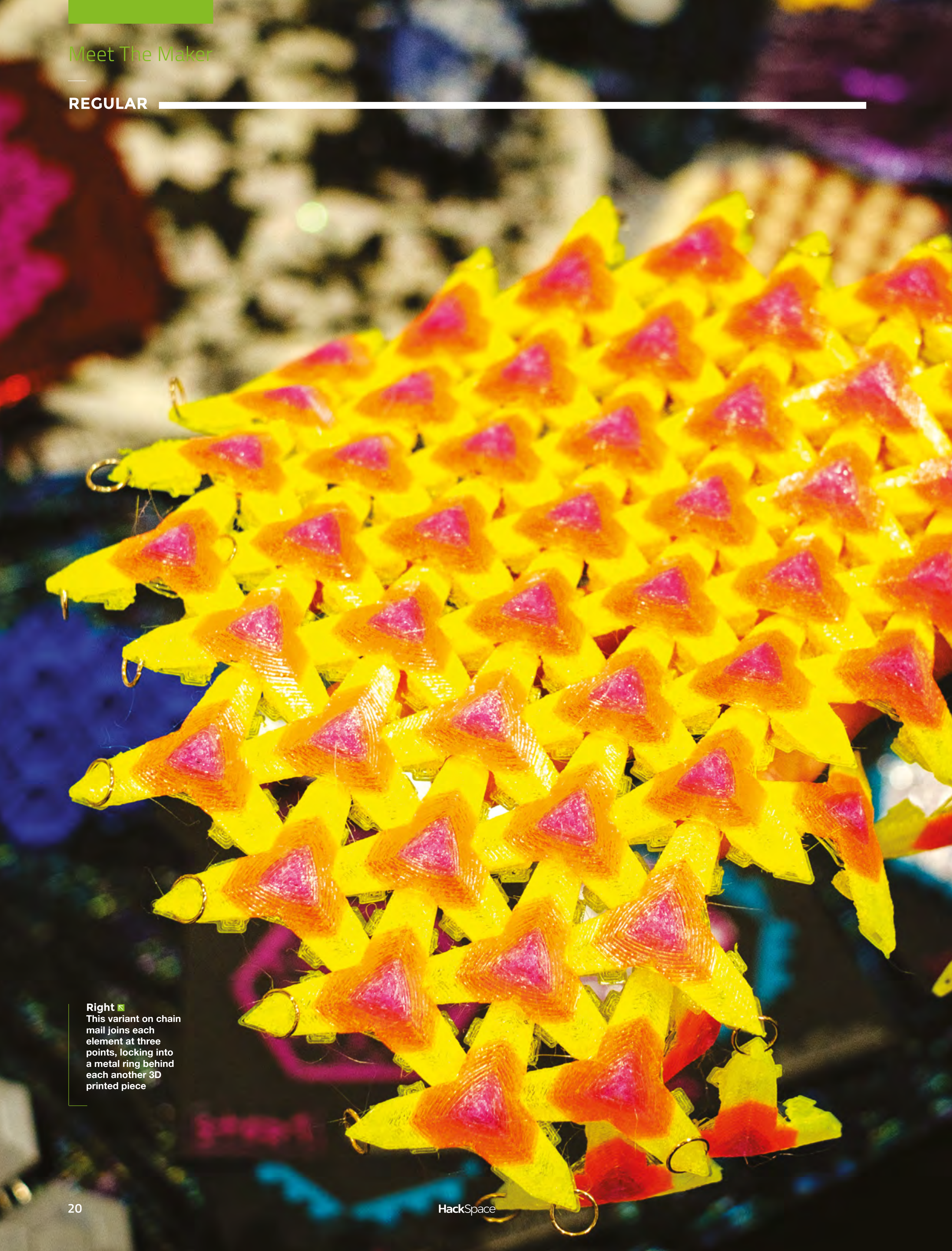
> **With the dragon scales, I modelled up the original** ones here in Fusion 360. And that just took off

"It just made sense to me to print and then encapsulate the fabric. If you take hot fluid and then smoosh it down flat, put the fabric on and embed fabric into it, it's the same thing.

"For example, here's some chain-mail I made. I print the scales with a post [sticking up] on each one – I add a ring to each post, then finish the print with a top layer to hold it in place. I print up the cavity and, just before I cover up the cavity, I drop the rings in, so it comes off the printer as a finished piece.



Left
If you want to create knee pads for kids that they actually want to wear, you can try somethign like this in a flexible filament, which comes out as squishy dragon scales



Right
This variant on chain mail joins each element at three points, locking into a metal ring behind each another 3D printed piece

Above “Some fabrics work better than others; something like 100% cotton works very well”

“With the dragon scales, I modelled up the original ones here in Fusion 360. And that just took off. I put a model of it on Thingiverse of a small patch of it, and people were like ‘what is this?’ I took a small video of it and, even before I was on Twitter, it went viral – people were telling me that I was all over Twitter and I was like, ‘I’m not on Twitter’!

“People saw one of these and saw how it moved. To alter the shapes for cosplay – for example, the scales on a piece designed to go over the shoulders – I take it and bring it into Blender. I took a lattice and deformed it to this shape (with bigger bits at the top, curved to fit around a joint). In the original, the scales were all the same size – this takes it one step further.

“I’ve done other things as well, like lasers, CNC machines, and stuff like that. And then, I’m also working on adding structural integrity to things, like drone bodies with screen door mesh. This is fibreglass screen door mesh. It works like rebar in reinforced concrete – you print the first layer, add the mesh, then print the rest so that the mesh runs through the plastic body. I tested it by swinging a weight off the end of two drone bodies at the same time. The one with the mesh in it has a tiny crack; the one without was completely broken.

“With a Markforged printer, which is $2000 for carbon fibre, I can add fibreglass where I want, or I could add carbon fibre strips for stiffness.” □

Transport for makers

Moving project parts is not always easy

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**

> **After 17 years with my trusty VW T4 Campervan, it is sadly time for a new set of wheels**

What car do you drive? Our culture suggests that the type of car you drive says a lot about you. For many makers I know, it just determines the most efficient and economical way of transporting materials. A sheet of 8'×4' (2440 mm × 1220 mm) ply costs less than £30 – but delivery of that sheet costs another £30. For one of my projects, I needed to transport two wheely bins into the middle of London – not something that could be done in a Mini.

After 17 years with my trusty VW T4 Campervan, it is sadly time for a new set of wheels. Being the main vehicle that I have driven, I have got used to the 'old fashioned' things – such as lack of remote locking, a radio cassette player, and having to turn a handle to make the windows open. I have not been paying attention to the other mod cons that newer vehicles have, such as parking sensors, wing mirrors that automatically fold in, and a special plastic clip on the windscreen to hold your parking permit. None of which make you get there any faster.

My main criteria for a vehicle are 'not tarmac-coloured', 'passes the London low emissions criteria' – although I would love an electric vehicle, they do not yet have the range or load capacity I require. Which brings me to the, 'can I lie down in the back'? The last is not so much for the occasional sleeping in the car trips, but more a measure of how much stuff I can get in. One salesperson commented, when I climbed into the back of a car in the showroom, that no one had done that before. I may have suggested that I transport coffins.

Car manufacturers boast about the capacity of the boot in litres. I may be unusual, but it's not often I want to fill the back of my car with a fluid (apart from air). Most things I transport are not squishy enough to flow into nooks and crannies. They are in sturdy, un-deformable boxes that protect the stuff I need to move about. I use the 'Really Useful Box' brand – they come in standard sizes, and can stack. Being told that the boot space is 2.5 boxes across, and 2.9 boxes deep would still only mean I can get four boxes in per layer. And no one mentions that bit between the folded-down back seats and the boot, which has a bar across it for headrests, and the roller blind thing that hides the boot-full of goodies.

What I'd really like is a chassis with different 'bodies' I can put on it; one for people transportation, one for goods transportation, one for camping etc. There have been a few concept cars like this, but there's nothing on the market yet. So, I'll have to make do with a 'not quite ideal for anything' vehicle. □

Avalanche alert

What happens when you plug a diode in backwards?

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.

Diodes are supposed to conduct current in only one direction. But what if you try really hard to get a diode to conduct current the wrong way? It turns out the answer to this unveils a little secret about Zener diodes, while also revealing a mechanism we can use to help protect our own secrets.

When you plug a diode in backwards, the thing that makes a diode work, the PN junction, effectively turns into a capacitor, known as the 'depletion capacitance'. If you keep cranking up the voltage, the depletion capacitance develops an electric field high enough to become a miniature particle accelerator: any stray electron that happens to fall into it (due to the jostling of thermal noise) will be accelerated until it smashes into an atom liberating another electron, thus unleashing a chain reaction that allows current to flow. This process is known as 'avalanche breakdown' and, as the name implies, if the current isn't limited, the diode will ultimately overheat and destroy itself in an avalanche of electrons.

When the current is limited – for example with a series resistor – avalanche breakdown has a number of useful properties. The voltage at which avalanche breakdown initiates is a parameter that can be engineered, which means it can be used as a fixed voltage reference. If this sounds similar in function to a Zener diode, you're right. The Zener effect (which is based on quantum tunnelling) is employed for low voltage references, while avalanche breakdown is used for high voltage references, and a blend of the two for voltages in between. Because of this ambiguity, we've come to call all voltage reference diodes 'Zener diodes', even though the higher voltage varieties don't use the Zener effect.

When you plug a diode in backwards, the thing that makes a diode work, the PN junction, effectively turns into a capacitor

Significantly, the avalanche effect is much noisier than the Zener effect, so much that it can be used as a source of randomness for cryptographic security systems. Unfortunately, the naming ambiguity makes it difficult to know whether you've got an actual avalanche diode, or a Zener diode. The good news is that the phenomenon is rooted in solid-state physics, so if a junction breaks down north of 10 volts at room temperature, it's likely due to the avalanche effect. The same truth applies to any PN junction – even those found in bipolar transistors. Keep in mind that avalanche noise goes down with higher temperatures, so if you're using an avalanche diode for cryptographic purposes, it's good to have some margin, and go for a junction that breaks down around 15 volts or higher. □

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**

FEELING BLUE

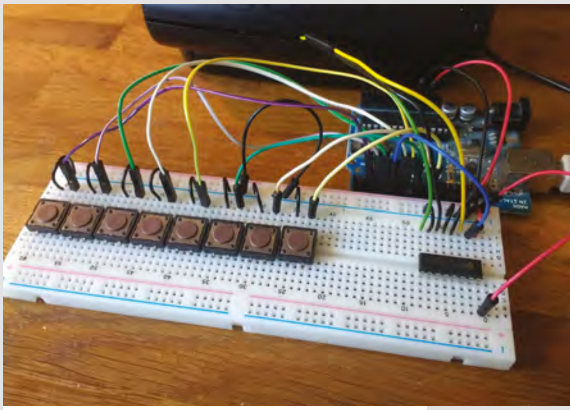
Sorry to be picky, but there is an error in Jo Hinchliffe's article on marking out. The stuff in the blue tin in Figure 2 is marking blue, and is used when scraping a surface to make it flat. It does not really dry out to leave a finish suitable for scribing lines into. In use, a very thin film of marking blue is applied to a surface, and a second surface brought to rest on the blued surface. The colour will be transferred to the high spots that can then be scraped away using an engineer's scraper. The work is then cleaned up, and the process repeated. If all is going well, more blue spots will be visible on the mating part. It takes many cycles to achieve a good fit judged by the appearance of lots of small blue dots evenly distributed over the whole of the mating surface. Bluing in is not restricted to flat surfaces; it is commonly used when fitting soft bearings (white metal or bronze) to shafts.

The blue that Jo really needs is called layout blue. It is a clear blue lacquer that is dissolved in either cellulose thinners or methylated spirits. It is painted onto the surface to be marked out and allowed to dry. The most commonly encountered brand is Spectracolour (**hsmag.cc/cromwell**). He is quite correct in his assertion that spirit-based felt pens (Sharpies) are a very handy substitute for layout blue. If something more durable is needed for setting out work on steel, acidified saturated copper sulphate solution works well. In use, the parts are degreased and then wiped over with the copper sulphate solution. The steel is immediately plated with a thin coating of bright copper. Once coated, wash away all traces of the acidified copper sulphate to stop everything in your workshop from going rusty.

Bob

Jo says: Thanks for reading the article and for sending your feedback through. The Stuart and Sons 'Micrometer' brand marking paste is sold as either for use as you describe in comparative work on surface plates to find high spots for scraping (Scraping… now there's a lost art!), but it's also sold with instructions that you can let it down with meths and use it as layout blue.

I omitted the stuff about that from the article as, to be honest, I felt our readership will probably just go for a Sharpie these days. These machining/fitting/benchwork articles are always a compromise as we are aiming to get people interested, but not bog down with too much fine detail.



I AM THE MUSIC MAN

As someone who loves music and electronics, issue 19 was a great issue for me. The tutorials on building a synth from a cassette player, and on using a 555 timer chip were really interesting. Together they show two different sides to electronic music – 555's with their beep boop square waves, and tapes with their more complex waveforms. The only trouble is, I don't know what to try first.

Doug
San Francisco

Ben says: Music is such a rich seam of making, and there's a long history of musicians building their own instruments. You can hear the detail of your make, and you can tweak and twist it to your particular musical tastes. Be sure to let us know what you make.

CROWDFUNDING NOW

BUYER BEWARE

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

Programmable-Air

Breathe in ... and out

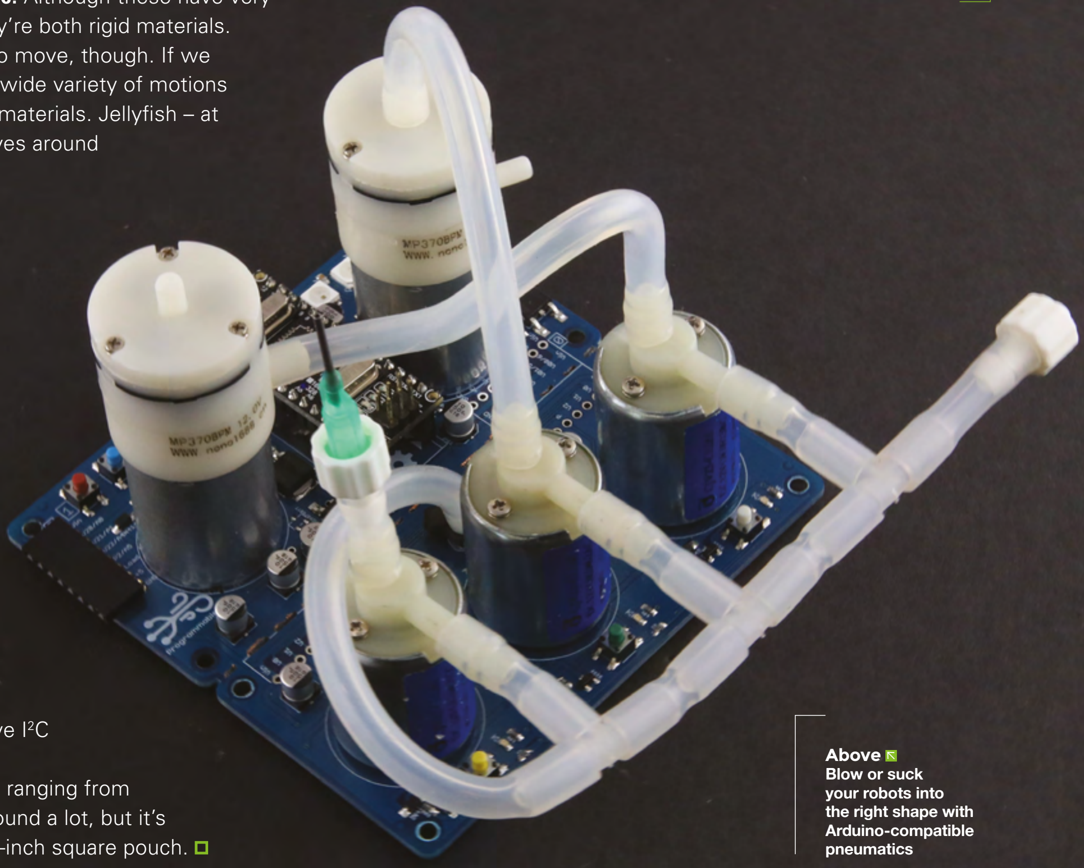
From $150 | crowdsupply.com | **Delivery:** October 2019

> "One of the easiest ways to manipulate soft objects is through pneumatics"

Robotics, as we usually think about it, is about moving metal or plastic. Although these have very different properties, they're both rigid materials. This isn't the only way to move, though. If we look at nature, there's a wide variety of motions made up of hard, soft, and flexible materials. Jellyfish – at one extreme – can propel themselves around the ocean with nothing but, well, some jelly.

The field of 'Soft Robotics' explores these types of movement, and one of the easiest ways to manipulate soft objects is through pneumatics. Programmable-Air is a board designed to help you automate the blowing and sucking. Based on an Arduino Nano (the maker is planning on porting them to the newly announced Nano Every), the Programmable-Air has two compressor/vacuum pumps, three pneumatic valves, and a pressure sensor. It's programmable, and includes a Grove I²C adapter for adding extra hardware.

The pumps can create pressures ranging from -7.5 psi to 7.5 psi, which may not sound a lot, but it's enough to lift over 100 kg using a 6-inch square pouch. □



Above
Blow or suck your robots into the right shape with Arduino-compatible pneumatics

Space of the month: Tinker Kitchen

tinkerkitchen.org

Tinker Kitchen is a makerspace for those interested in food hacking. Rather than laser cutters, soldering irons, and big boxes of components, it's full of spices, ovens, and kitchen gadgets, and the people. That's what's different to the average makerspace; what's the same is that the people who go there are interested in messing about, learning, sharing, and seeing what's possible. We spoke to Tinker Kitchen's founder, Dan Mills, and dribbled slightly over all his cool toys.

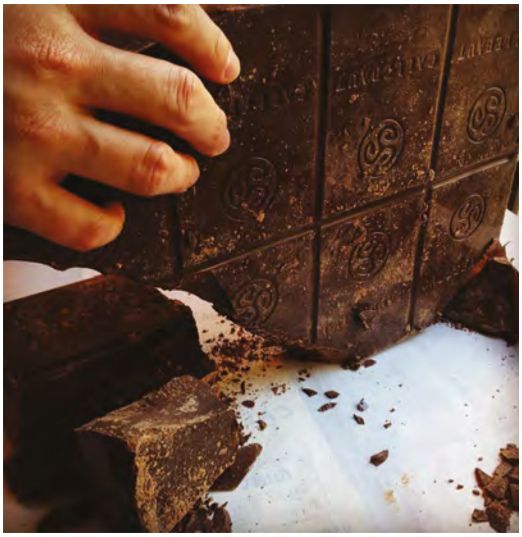
"I used to be in software – I was a programmer and product manager, but I like food, I like cooking, and I like cooking gadgets. I decided that I wanted more and bigger tools. I had stuff that was overflowing from our apartment kitchen, and thought that there must be other people like me who wanted access to a bunch of tools. So, I thought about doing a makerspace, and it took me four years to put it together. It's been hard – especially with zoning and food regulations – but it's a natural fit, a kitchen makerspace.

"We have lots of machines; smaller things on the one side, things like a grain mill, chocolate machines, coffee roasters, blenders, things like that. Professional kitchen stuff.

"On the other side is the kitchen proper. There's a fridge, a hot line right here, a six-burner stove with a conventional oven, a convection stack (that's two convection ovens), then the last oven is a combi oven. It can inject steam into the cabinet, so you can change the humidity, the temperature, and the fan speed. It's a German-made oven, and it's one of the nicest appliances we have.

"We have a soft serve machine. The thing that looks like a washing machine is a freeze-dryer. We have a centrifuge, there's an ice cream batch freezer, a commercial meat grinder, and a Pacojet, which is another kind of ice cream machine that lets you experiment in small batches very quickly.

There are also a lot of chocolate moulds, pasta tools, and an ice-maker. Then, we have a storage room at the back where members can rent extra storage space if they need. →



Above
Members come to the space, they clean the table, they can put stuff in the fridge during the day, make whatever they want, then they can eat here in our dining area, or take it home

"Our members are mostly two kinds: R&D, they're working on the development of food products for their clients. They're food scientists. Developing beverages, or frozen yoghurt, protein bars, all kinds of food products. They get clients, and there's some product that needs help, whether it's in very early development or there's some problem with it. Like maybe, it's changing colour while it's on the shelves, or becoming gummy, or they just don't know what they want yet so they need to experiment.

"The other set of members is hobbyists: people who aren't in the food industry, they just want to have fun exploring all these machines.

FOOD SCIENCE

"That's the split in terms of members, but then the other big revenue source for us is team-building activities. If you're a team looking for something fun to do together, we can put something together for you to do. Chocolate-making is really popular. We also do a pasta-making one with our extruder, and we do an ice cream-making one that's also popular.

"We have these machines that enable you to melt and then temper the chocolate so that it will be nice and smooth, shiny, and snappy. If you melt a bar of chocolate in a bain-marie at home, and then let it cool down, it gets a satiny finish. That's called bloom, it's a recrystallisation of the fat and sugar molecules into shapes that we do not like. So, it's not an inherent defect in the product, it's just that chocolate is mostly fat and sugar, and those things crystallise. The crystals can take different shapes. There's a particular shape that we want. These machines play around with the temperature to kill off the bad shapes, the bad crystals. The different crystals happen to have different melting points, so the machines exploit those different temperatures to leave the current ones.

"It's like making steel – that's why it's called tempering. The difference is that, in the case of steel, you have a structure and you're looking to insert other elements to add strength, which you're not with chocolate. But, you are looking to create crystals that line up tightly. When you melt it and reform without temperature, you can see that looks ugly; that's because there are different crystal shapes that don't fit tightly together. It's waxy, and it's mushy. You put it in your mouth, and the melting temperature is now slightly higher than body temperature, so it doesn't melt in your mouth.

"Between the food scientists and the hobbyists, everyone is really nice. People feed each other. Everyone has their own projects, but they're curious about what people are working on, and they share. There's this one guy who makes chocolate, bean to bar. He comes in with raw cacao beans. He roasts the cacao here, cracks it with a rolling pin, winnows the beans with a hair-dryer – that's to remove the shells. He does a pre-grind, a dry grind, then a wet grind in this machine that grinds stone on stone. It keeps on turning, and eventually, over one or two days, it gives you very smooth chocolate. He makes little chocolate bars that he takes to his office. He's not in the business at all.

"Members can always bring one guest, and he just keeps bringing different people to taste things". □

CONTACT US

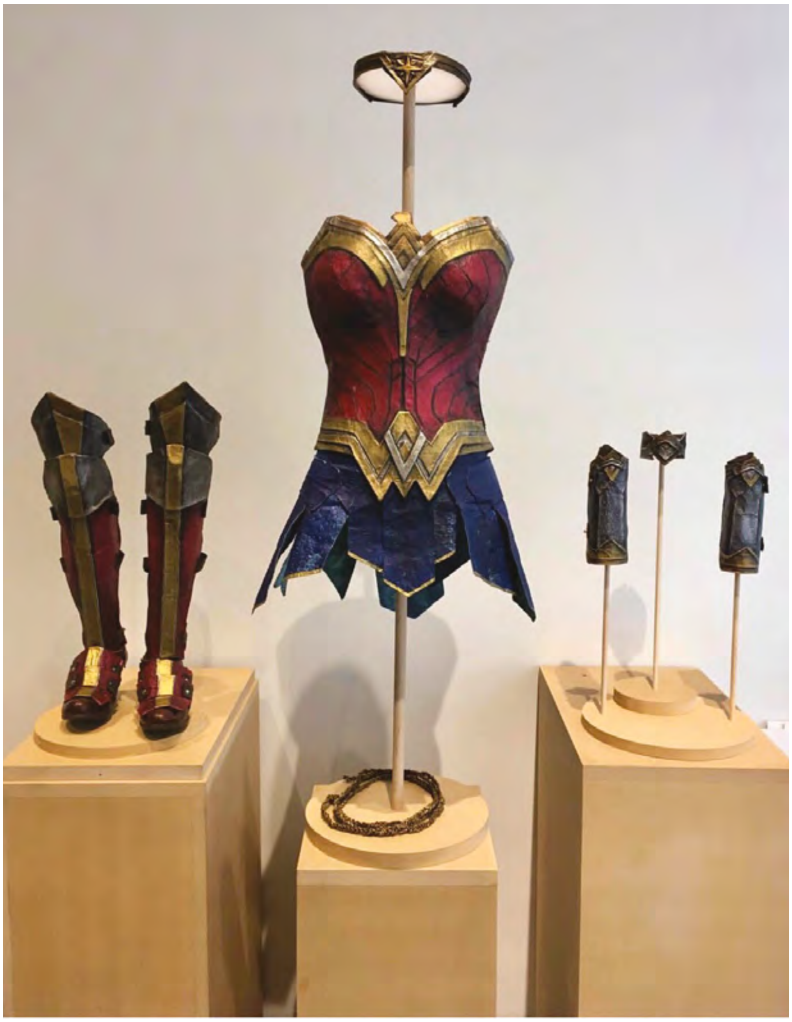
We'd love you to get in touch to showcase your makerspace and the things you're making. Drop us a line on Twitter **@HackSpaceMag,** or email us at **hackspace@raspberrypi.org** with an outline of what makes your hackspace special, and we'll take it from there.

Left
Tinker Kitchen opened late in 2018, but the idea was in development for a few years before that

Wonder Woman cosplay
Made from kombucha

Ferment your next outfit

By Krista Peryer @missladygeek_



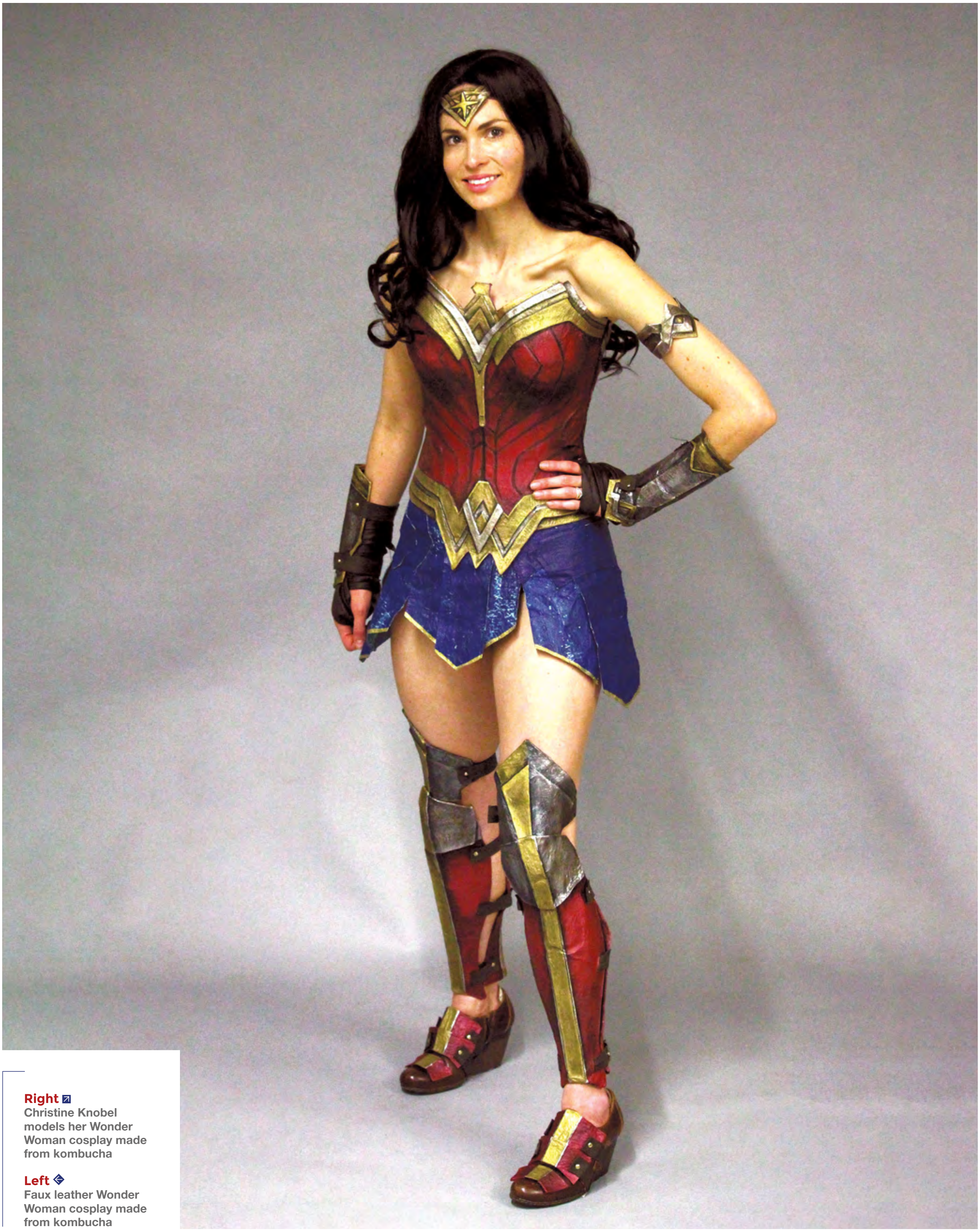
Kombucha may be that funny-sounding word often associated with words like 'yoga', 'hipsters', and 'hippies' but, jokes aside, there are other uses for the fermented, fizzy bottle of goodness besides its known characteristics of promoting digestive health and wellness.

Generally made from green or black tea, kombucha is the term used to describe the drink, as well as the healthy organisms it contains that are made from a mixed culture called SCOBY (symbiotic colony of bacteria and yeast). The SCOBY – a mat of bacterial cellulose that can grow into any shape it's contained in – grows on top of the tea during fermentation, creating a slightly alcoholic, effervescent, sweet drink.

During the growth process, you can make it thicker and have more layers by adding more sweet tea, and make it as large as any container you grow it in. Once its growth fills the growing space, it is then known as the 'mother', and can be used to make multiple new tea fermentations by dividing it into smaller pieces.

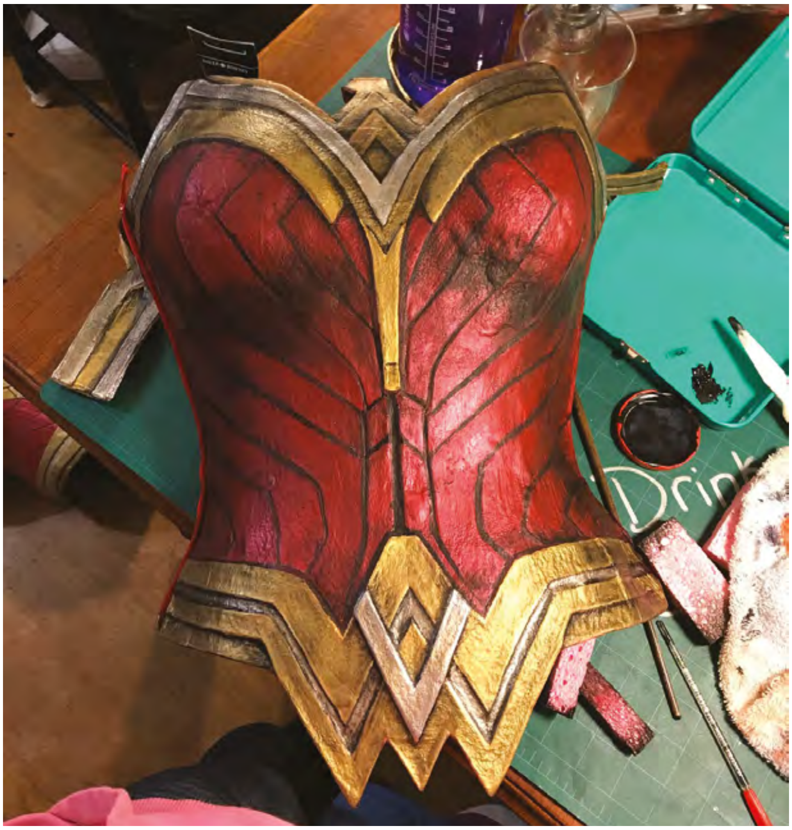
Because of kombucha's natural properties and ability to be easily moulded into many shapes and sizes, fashion design graduate, educator, and cosplayer Christine Knobel decided to study its practicality for use in her grad school research on sustainable and biodegradable fashion in cosplay and zero-waste fashion.

Knobel has been an avid cosplayer since 2002, cosplaying as characters like the McVengers team member Wendy's Widow (Black Widow and fast-food giant Wendy) and Marvel's Galactus (which earned her a feature on the cover of Marvel Comics). After graduating from the Fashion Institute of Design and



Right
Christine Knobel models her Wonder Woman cosplay made from kombucha

Left
Faux leather Wonder Woman cosplay made from kombucha



Above The kombucha can be made to look like several different materials

Below Parts of the 'mother', treated and waiting to be assembled

Merchandising (FIDM), she decided to apply for grad school at UC Davis, and got accepted.

While brainstorming textiles for her thesis research, she explored ideas like wearables, but ultimately went in the direction of her main interest: creating sustainable options in the fashion and textile industries. Kombucha's leather-like texture, mouldable capabilities during growth, and biodegradable properties made it the perfect choice for her thesis project: a biodegradable Wonder Woman costume.

The use of kombucha in the textile industry had only been reported about in research papers, up until the point Knobel decided to try it out, so there weren't resources like YouTube videos to watch or blog posts to read, with tips and tricks for DIY kombucha leather projects. This made it one of the most difficult and interesting projects she had taken on, and a whole lot of trial and error was involved. Knobel states that, while it took her about two months to complete the Wonder Woman cosplay, she spent the better part of a year researching and testing the kombucha's effectiveness for use. She spent six months of that time in the lab every day.

During her time in the lab, she tested multiple types of dyes, as well as applications of the dyes, tested the strength and durability, sampled various amounts of layers, and tested various drying techniques. She learned that four layers of kombucha, grown each one week at a time (four weeks total for the complete costume), made the layers thick enough to make the costume durable enough to last for a year of wear. Knobel states that, since she enjoys making and wearing multiple cosplays and doesn't want to keep her costumes around forever, the one-year time-frame felt like a perfect length of time for her before she moves on to creating a new cosplay with more kombucha.

HOMEGROWN MATERIAL

From an outside perspective, the process for growing kombucha for textile use appears to be quite tedious, but Knobel explains after the initial trial and error, she finds it easy to use and make. She can grow it in the comfort of her own home in the kitchen; mould, shape, and sew it as it grows; and use it immediately once it's finished growing.

> "She frequently massages the kombucha throughout the growth process"

She grows the kombucha in multiple containers the length and shape she wants the sheets of kombucha to reach. After mixing the ingredients, and pouring the ingredients into the containers, the watching process begins. Knobel states that the warmer the room temperature, the better chance the kombucha has for reaching its potential, but it is temperamental. She says that a temperature of around 80–95 °F (27–35 °C) is what will make the kombucha really thrive and grow well, and that this step can also be achieved quicker with heat sources like a portable heater or heat applied throughout the process.

Above
Kombucha will not make you into a superhero, but it can make you look the part

Once it's time to dry the faux leather, you have to be a little bit careful about the environment. If the air quality where you live is too humid, it will make for a much longer dry time (generally about a week per layer), and if it is too dry, it will cause the kombucha to dry brittle and crack.

To achieve the smooth, soft leather-like texture, she frequently massages the kombucha sheet throughout the growth process. During these steps, she also adds more ingredients to increase growth size if needed, and applies the dyes in various colours for her project.

THE SUIT OF A SUPERHERO

For her Wonder Woman cosplay thesis project, she found inspiration in the Gal Gadot DC Cinematic Universe rough and tumble version of the famous Amazonian warrior costume. The leather composition, rich colours, and bold design were an instant hit with Knobel. She's watched the 2017 *Wonder Woman* movie many times over for costume study, just because she truly enjoys the film that much.

Once the seed had been planted, Knobel scoured the internet for Gal Gadot version Wonder Woman patterns and came across Beverly Downer of Creative Studios. She purchased her pattern, utilising all the pieces for her cosplay, with the exception of the bodice. Knobel states that, while there are multiple sizes available, the bodice pieces were too large for her, and she also wanted to make some changes like adjusting the seam allowance to add a zip for ease of use. To achieve this, she used Worbla, which she moulded around a body block that she made to her exact proportions to create the bodice needed for her perfect fit.

Knobel states that, while this experience has been rewarding and fun, since the cosplay only lasts about a year, she's ready to embark on a new make. She plans to make another Wonder Woman cosplay, this time experimenting with adding more layers for strength and durability, and using different colouring methods that are more environmentally friendly, like cake paints. She says that the benefits of kombucha cosplay – like biodegradability, comfortability, and ease of use – far outweigh any cons. Really, the only negative would be that the kombucha is not waterproof. Those are things that Knobel plans to experiment more on in the future as her builds get more extensive, and as she continues to learn through the process. When asked what her dream build would be, if she could make anything she wanted with kombucha in the future, she was quick to respond and had no doubt in her tone: Klingon battle armour from *Star Trek: The Next Generation.* □

meet
learn
play
Ultimaker
Ultimaker

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 36

50 TOP MAKER TIPS

A half-century of ways to be a better maker today

PG 50

HOW I MADE A CNC PCB MILL

Before you send off that order to the factory, fail faster with a DIY in-house PCB maker

PG 56

OPEN-SOURCE HARDWARE CERTIFICATION

We all use it, but what does open-source hardware actually mean? Here's the official answer

PG 60

INTERVIEW: JEN SCHACHTER

Multimedia boss of enormous projects – as seen at the White House

PG 68

IMPROVISER'S TOOLBOX: DRINKS CANS

Bend, cut, and tape these cheap aluminium shapes to your will

FEATURE

TOP MAKER TIPS

UPGRADE YOUR HACKER SKILLS

By Gareth Branwyn

ELECTRONICS

PROGRAMMING

It's the little tips and tricks that make life easier. You won't find these hard-learned snippets of information in text-books or datasheets. They're the lessons of workshops, makerspaces, and sheds, and invaluable for beginners and experts alike. We've gathered up 50 of our favourite little maker secrets of the trade to help you create better things, whether you make in woodworking, electronics, desktop fabrication, metalwork, programming, or all five.

We're here to help you keep track of bits as you disassemble, unlock the insulating powers of hot glue, reassemble, and a myriad other things. These won't solve all your problems – alas, no secret sauce is going to help you break the pesky laws of physics – but hopefully, they'll help remove some of the frustration, and perhaps help you enjoy the process a little more.

We claimed to have 50 tips this month, but here's a bonus tip number 0: the only failed makes are the ones you don't make. Even when something doesn't work as expected, you can still learn from it, and it all helps build up your maker experience. Your projects don't all have to look polished or have every conceivable feature to be great products. Get making! Now, here's 50 more to help you as you do.

DESKTOP FABRICATION

METALWORK

WOODWORKING

Thanks to all of the following for helping us find these tips:

Jimmy DiResta, Sean Michael Ragan, Becky Stern, Andrew Lewis, Miguel Valenzuela, John Edgar Park, Saar Drimer, Donald Bell, John Ülaszek, Bob Clagett, Andy Birkey, Tim Slagle, Bob Knetzger, This Old Tony, GreatScott, Steven K. Roberts, mrpete222, Ray's Garage, Sam Freeman, Bill Livolsi, Izzy Swan, Happy Wife Happy Life, Matt Griffin, Linn of DarbinOrvar, Corey Schafer, Michael Toppa

MAKE ARTY SOLDER JOINTS

Custom circuit boards are becoming impressive works of art. If you have a bespoke board that you want to make look extra-pretty, try this method of 'finishing' your solder joints:

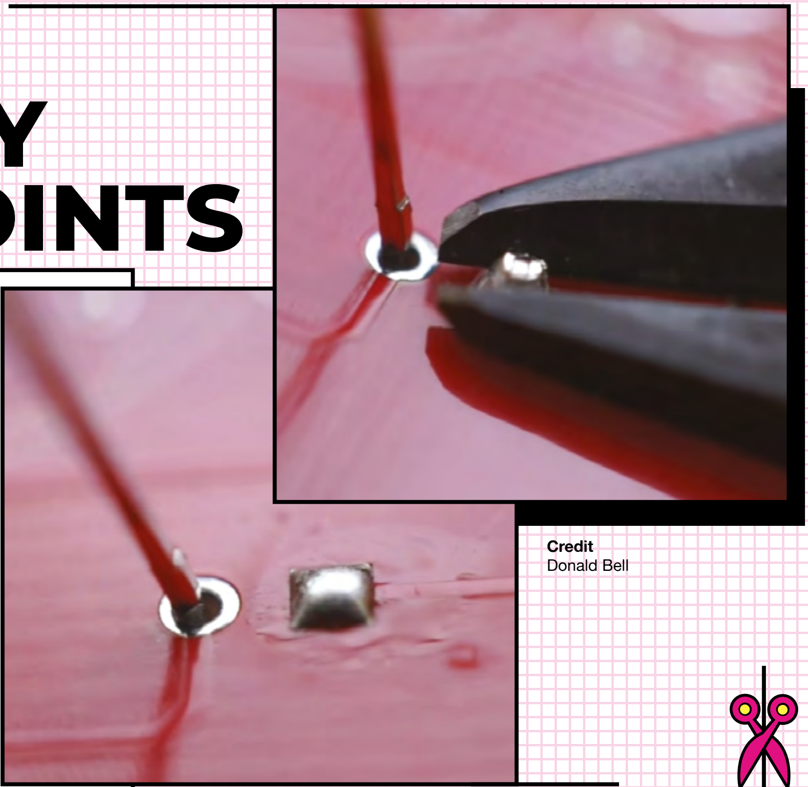
Step 1 Solder your through-hole component leads as normal.

Step 2 Clip off the soldered joint near-flush with the circuit board (using angle cutters).

Step 3 Reheat the joint quickly, and apply a little pillow of solder to it.

Step 4 When all of your solder points have been beautified, clean off the excess flux with acetone, isopropyl alcohol, or a flux remover.

Note that card-carrying engineers don't take kindly to this little 'hack'. Cutting and reheating a joint is not ideal for a strong connection. This means you wouldn't want to use this technique on a critical part, or hardware that you rely on. This tip comes from Saar Drimer, of the boutique kit-maker Boldport, by way of Donald Bell.



Credit
Donald Bell

ELECTRONICS

KEEP SMALL PARTS SORTED AND SECURED

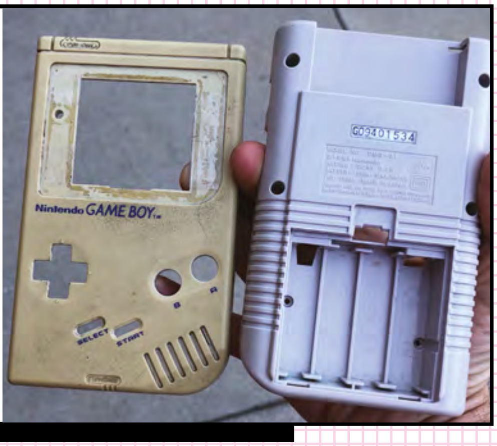
When assembling/disassembling components, use double-sided tape to hold small parts. You can even label the items around the edges to identify what the parts are and where they came from. An alternative is to use a car dashboard sticky mat. You can get these online for under $10.

Credit
Richard Sheppard, Tips and Tales from the Workshop

RESTORING OLD PLASTIC ELECTRONICS ENCLOSURES

Credit
John Edgar Park

If you want to restore old electronics that are housed in filthy and faded cases, you can dramatically bring those cases back to life using hydrogen peroxide and UV light. For this, all you need is some salon-grade hair bleach, some cling film, and sunshine. After thoroughly cleaning the case, paint on the salon bleach/peroxide with a brush, wrap the case in the cling film, and place in direct sunlight for around six hours. You will be amazed by the results.



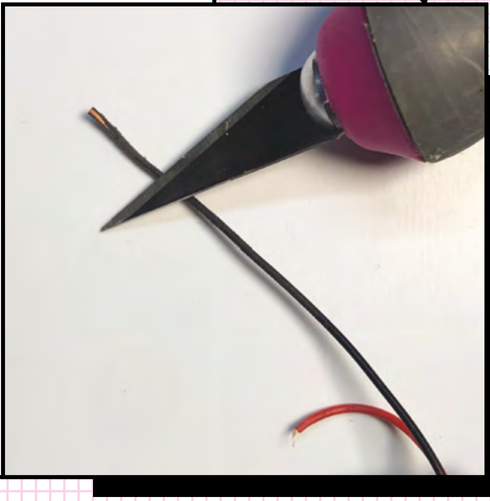
HOT GLUE AS AN INSULATOR

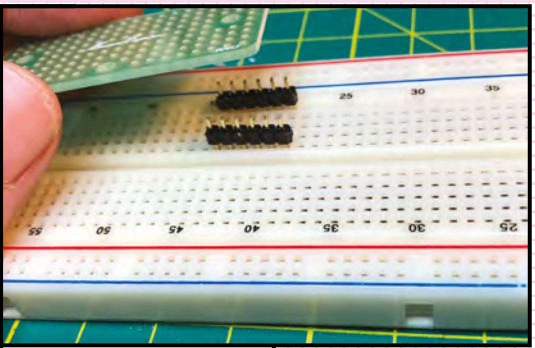
Credit
Linn at DarbinOrvar.com

You can use hot glue to hold wires in headers, insulate wires and components on a PCB, and otherwise encase your electronics in a protective plastic. If you decide you want to remove the glue-insulator, applying denatured alcohol along the edges of the glue blob will break its bond.

HOW TO STRIP VERY THIN WIRE

If you have really fine, insulated wire to use in a project, what's the best way to strip off the insulating jacket without harming the wire? You can carefully strip it by holding a sharp hobby knife across the wire and at a 45-degree angle, carefully pull the wire toward you to peel off the jacket. This works, but you can easily cut the wire if you're not careful. There is also a method that involves getting a bubble of solder on your iron, touching it to the end of the wire, and letting the heat from the iron melt back the jacket. Your wire gets tinned in the process.





HOLDING HEADER PINS IN PLACE FOR SOLDERING

Use a breadboard to hold header pins in place and straight while soldering. Insert the header pins into a breadboard and then align your circuit board on top, slot the pins through the PCB holes, and solder.

BLACK HOT GLUE AS LIGHT SEALANT

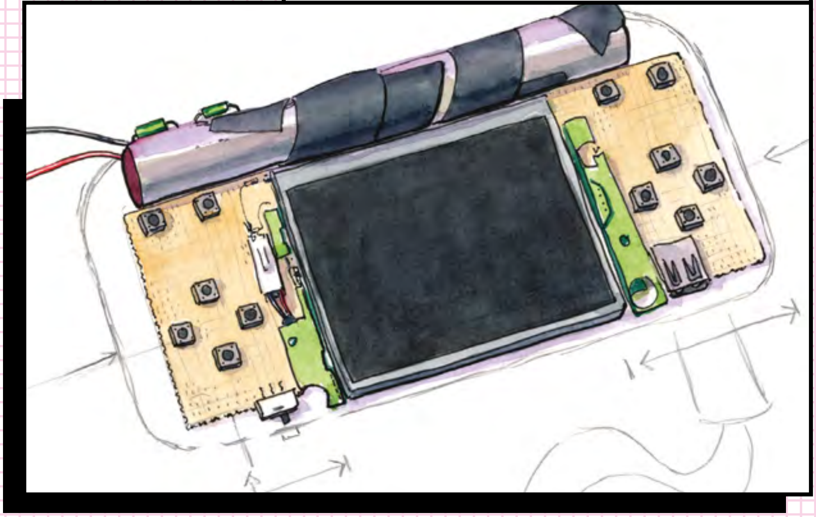
Hot glue sticks come in a number of colours, and at least one of those colours has a practical use. Black glue sticks can be used in electronics to act as a light sealant. In some electronics projects, you might want to seal up or block light-emitting components so that you don't get photonic noise or crosstalk. Black hot glue is the ticket.

HEARING THE CURRENT

Use a cheap amplified speaker to listen to low-voltage logic signals in circuits. You can hear most clicks and saw/sine signals.

PLAN PROJECTS ON PAPER FIRST

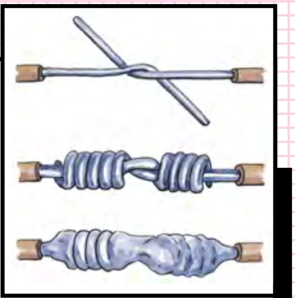
When designing an electronics project, first draw your enclosure on paper, blocking out the components that will go inside. When building a workable enclosure, you need to consider all sorts of variables, such as component placement and needed allowances for screws, stand-offs, and wire runs. Physically placing components onto a piece a paper can help you clearly visualise and think through all of these placement considerations. Once you've worked out all of the placement and measurements, you can transfer your design to a CAD program.



Credit
Richard Sheppard, Tips and Tales from the Workshop

MAKE A LINEMAN'S SPLICE

Credit
Richard Sheppard, Tips and Tales from the Workshop



This time-proven splicing and soldering convention, originally developed for telegraph technology, creates a super-strong wire join that will stay put even under tension.

1. Strip and tin wire leads as normal.

2. Overlap the two wires at their centre-point, and tightly twist each wire at least three turns, with no gaps or overlaps between turns. Trim any excess on the ends to prevent protrusions or overlapping the insulation.

3. Wet all elements of the connection with solder, and make sure the solder flows between all elements of the connection, including the periphery.

DON'T ASSUME ANYTHING IS RIGHT

When troubleshooting hardware and software, it's easy to make assumptions about certain aspects of your system. Don't. Start with the most proverbial 'Is it plugged in?' and work your way up. Think like a detective. Work like a trouble-shooter. Be methodical.

THE RIGHT CODE CAN SAVE YOU POWER

Code can be as important as hardware when it comes to saving power on microcontrollers. With power-optimising code (e.g. turning off power to sensors when not in use), you can keep an MCU running for months on a coin cell battery. To find out what sorts of low-power options are available for your programming environment, just do a web search.

LEARN BY DISCOVERY

Don't be afraid to learn programming by discovery. Change attributes and values and see what that does. Use and modify online sample code. Tinker. Iterate. Ask questions in online forums.

PROGRAMMING

TURN A PROJECT IDEA INTO A PROGRAM

In a highly recommended YouTube video and Instructable, entitled '*Arduino Sample Code Mashups*', Becky Stern simply and beautifully breaks down how to think through the design of a hardware project and the software that you will need to create to run it. While the video is geared to Arduino, almost everything is applicable to any microcontroller programming language. Here are the basic considerations that she outlines:

1 **Define your purpose** – Write down the goals of your project, and what features are required and which ones are optional.

2 **Classify inputs and outputs** – List out all of the inputs and outputs you plan on using.

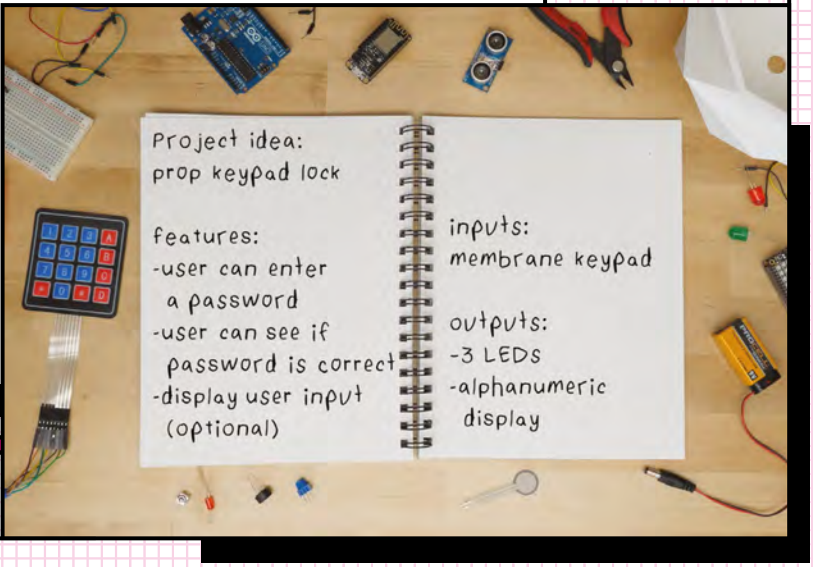
3 **Write pseudocode** – Explain in plain English the steps the program will walk through as it's executed.

4 **Test your hardware** – Test out each subsystem of your design by using sample code known to work with each input and output component.

5 **Create your program** – Paste in all of your now-verified code samples, and start building your program from there.

6 **Make incremental improvements** – Once you get everything working, you can then fix or change any issues you encounter or changes you care to make.

Credit
Becky Stern



Project idea:
prop keypad lock

features:
-user can enter
a password
-user can see if
password is correct
-display user input
(optional)

inputs:
membrane keypad

outputs:
-3 LEDs
-alphanumeric
display

TEN TIPS FOR CLEAN CODE

At WordCamp a few years ago, programmer Michael Toppa gave a lecture on clean code in which he shared the following tips:

1. You are responsible for the quality of your code
2. Use meaningful names
3. Write code that expresses intent
4. Code should speak for itself whenever possible (keep comments to a minimum)
5. Leave the code better than you found it
6. Single-responsibility principle (each section of your code should have a clear conceptual responsibility, and not be trying to do too much)
7. Write tests (run tests to see how users are going to experience your code)
8. Work in short cycles: incremental and iterative
9. Independent architecture (good software allows decisions about frameworks, databases, servers, etc. to be deferred and delayed)
10. Practise, practise, practise

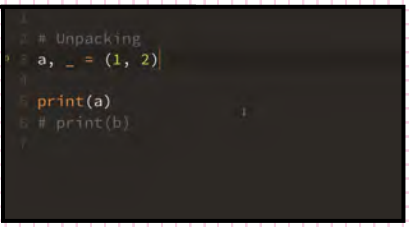
FREE UP RAM BY REMOVING APPS

If you're working with Raspberry Pi and need to free up some disk space, consider removing unwanted apps. For instance, here are the commands for removing Wolfram Engine (all 650MB worth):

```
sudo apt-get purge wolfram-engine
sudo apt-get clean
sudo apt-get autoremove
```

Other chunky apps you may want to consider removing: libreoffice, minecraft-pi, and sonic-pi.

IGNORING VALUES IN PYTHON WITH UNDERSCORE



```
# Unpacking
a, _ = (1, 2)

print(a)
# print(b)
```

Credit
Corey Schafer

Did you know that any time you want to ignore a variable in Python, you can simply use an underscore as a variable name? Using this convention will tell Python (or anyone else reading your code) that this variable is not going to be used anywhere else in the code. This tip was shared by Corey Schafer in an excellent YouTube video, '*10 Python Tips and Tricks for Writing Better Code*'.

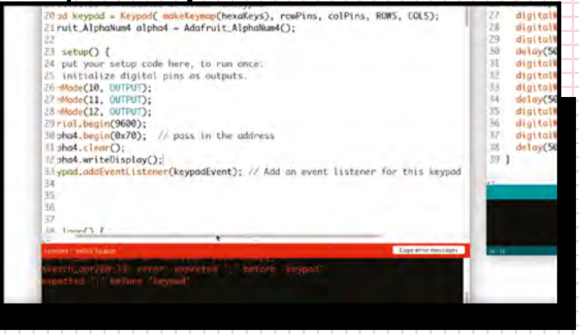
PROGRAM MICROCONTROLLERS WITHOUT CODING

XOD (pronounced 'zode') is a visual, node-based programming environment that enables you to program microcontrollers without using a single line of code. XOD is a bit like Scratch (the visual programming language for kids), but XOD is especially made for microcontrollers and electronic components. There are XOD desktop IDEs for Mac, Windows, and Linux. To see XOD in action, check out the MakeUseOf YouTube video entitled '*Arduino Robotics with No Coding Using XOD*'.

COMPILE YOUR CODE OFTEN

By way of maker, artist, and Arduino guru Becky Stern comes this simple but essential tip. When working on an Arduino Sketch, pasting in sample code and tweaking variables as needed, compile your code often. That way, if you discover an error, you can isolate it to the time since your last save.

Credit
Becky Stern



```
ad keypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
ruit_AlphaNum4 alpha4 = Adafruit_AlphaNum4();

 setup() {
 put your setup code here, to run once:
 initialize digital pins as outputs.
nMode(10, OUTPUT);
nMode(11, OUTPUT);
nMode(12, OUTPUT);
rial.begin(9600);
pha4.begin(0x70);  // pass in the address
pha4.clear();
pha4.writeDisplay();
ypad.addEventListener(keypadEvent); // Add an event listener for this keypad




 loop() {
```

```
sketch_apr20b:11: error: expected ';' before 'keypad'
expected ';' before 'keypad'
```

```
    digitalW
    digitalW
    digitalW
    delay(50
    digitalW
    digitalW
    digitalW
    delay(50
    digitalW
    digitalW
    digitalW
    delay(50
}
```

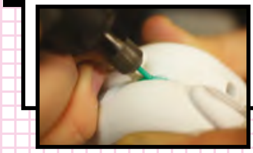
BREAK IT DOWN INTO SUBSYSTEMS

In teaching robotics, one thing this author has always stressed is the idea of breaking the robot (or any other complex technology) into manageable subsystems. Get those to work and then build upon them. The same holds true for microcontroller programming and physical computing.

DESKTOP FABRICATION

A ROTARY TOOL AS PLASTIC WELDER

Did you know that you can join or repair 3D parts using friction welding? All you need to do is to chuck a short piece of plastic rod or filament into a rotary tool. By applying the spinning plastic tip to the area of your join, you will generate friction, and that friction will melt the plastic pieces, forming a serviceable bond.



Credit
Matt Griffin

TIME-LAPSE FILMING OF 3D PRINTS

Scott, of the YouTube channel GreatScott!, came up with a fantastic way of taking high-quality time-lapse photos of 3D prints and turning them into an impressive-looking video that looks as though the print magically grows from the print bed (the photos are taken each time the print head returns to its starting position). Here are the four steps to this process:

1 Hook a microswitch up to the right side of your printer's Z-axis so that the head engages the switch every time it returns to its start position. (Scott's print head didn't quite reach his switch so, cleverly, he used hot glue to create an extension to the switch button.)

2 Attach a suitable cable to your microswitch and plug it into the Trigger Port on a DSLR camera with Manual Focus selected.

3 Download and install the web-based 3D printer controller software OctoPrint, and Octolapse (the time-lapse camera plug-in), on the computer controlling your camera.

4 Set up your camera on a tripod with a good view of the print bed and begin printing. When your print is done, the Octolapse software will spit out a beautiful time-lapse video file of your print.

You can see the entire process, including an earlier setup using a Raspberry Pi Camera Module instead of a DSLR camera, in the video '*How to Create the Best 3D Printing Time-Lapses*' on the GreatScott! channel.

CLEAN UP AFTER MILLING

After you've milled a fibreglass PCB on a CNC milling machine, don't be tempted to blow away the dust. The fibreglass particles are bad news for your eyes and your lungs. Use a vacuum to get up most of the waste, and then use a small brush to carefully clean up any remnants. A scrubby pad (or fine sandpaper) can also be lightly used to do any further deburring. Finish up with a wipe-down of isopropyl alcohol.

CNC JOINERY COLLECTION

Credit
Sean Michael Ragan

Maker Sean Michael Ragan has put together a highly useful and inspiring collection of joints that can be used in CNC panel joinery. Some of the joints, like lap joints and box joints, are easily recognisable from old-school joinery. But others are truly the product of a modern CNC era, such as snap-lock cross joints, captive-nut T-joints, and 'sninges' (aka living hinges). You can access his collection here: **hsmag.cc/LFUfjW**.

TOOL PATH NAMING CONVENTIONS

When using programs like VCarve, you must isolate individual line or vector objects and assign tool paths to those objects. Selecting your objects once may be easy, but making changes will necessitate selecting those objects multiple times. Having consistent naming conventions can help you stay organised and avoid confusion. Create a naming system that helps you cluster the different tool paths into different runs. E.g., if you drill hold-down holes, followed by a series of pocket cuts, then a tool bit change, the naming structure would be something like this:

First Run: File Name: "Run 01"

Tool Path 01, Run 01: "01.Hold_Down_Holes.3mm" [notice the 3mm dictating the bit size]

Pause Action: Screw Down Plate

2nd Run: File Name: "Run 02"

Tool Path 01, Run 02: "02.A.Pocket_Cut.3mm

Tool Path 02, Run 02: "02.B.Inside_Cut.3mm

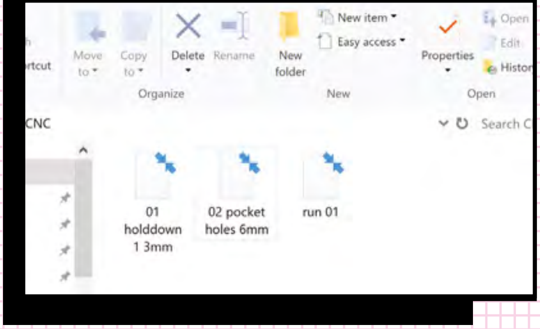
Pause Action: Change Bit to 6mm

Final Run: File Name: "Run 03"

Tool Path 01, Run 03: "03.A.Outside_Rough_Cut.6mm"

Tool Path 02, Run 03 "03.B.Outside_Rough_Cut.6mm"

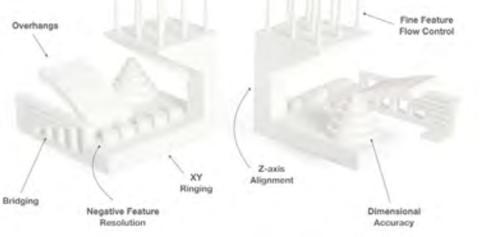
Establishing a consistent naming convention will help you the next time you want to cut a file, or if someone else wishes to do it.



FDM 3D PRINTER ASSESSMENT

There is a fairly sizeable collection of 3D test prints that are downloadable online. These are designed to put your printer to the test and reveal a printer's particular strengths and weaknesses. A recent partnership between Autodesk and Kickstarter has produced what may be the most rigorous and robust test object to date. It's called the FDM 3 Printer Assessment. The object was developed so that Kickstarter creators and backers can test the limits of both the printer hardware and the software, and to see how well a machine is calibrated. You can download the STL file on GitHub (**hsmag.cc/gqXSrq**).

Credit
Kickstarter/Autodesk



MEETING MAKERS

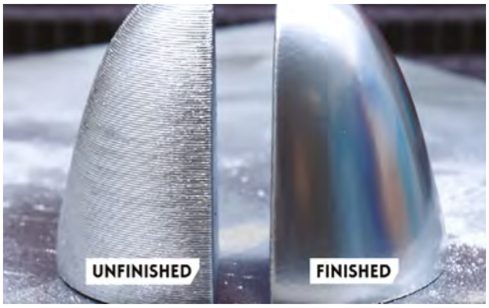
Maker John Ülaszek offers this great suggestion: "Join Facebook groups of people doing stuff that's not normally your thing. For instance, there's a group on 3D-printing fishing lures. I don't care about fishing lures, but solving all of the problems of making 3D-printed lures reveals a treasure trove of tips that I wouldn't have otherwise discovered."

CNC MACHINIST REFERENCE CHART

Bantam Tools has put together a free PDF wall chart (**wallchart.bantamtools.com**) featuring useful info for CNC milling. The chart includes graphs showing metal machinability vs. hardness, plastic hardness vs. melting points, common calculations, a G-code reference, nut sizing, and more.

FINISHING 3D PRINTS

If you want to get a smooth result from your FDM prints, you need to smooth out all of the layers that can be seen in your final print. To achieve this, Bob (of I Like to Make Stuff) recommends first trowelling on a product like Bondo Glazing and Spot Putty. This is applied in a series of layers that are then sanded until you get a reasonably smooth surface. To finish off the part, you then apply several coats of automotive spray filler primer (with sanding in-between each application). The results will be a very smooth surface that does not look like a 3D print.



Credit
Bob Clagett/I Like to Make Stuff

PRINT VERTICALLY FOR HIGHER RESOLUTION

When printing a detailed image, such as a face for a rubber stamp, orient your print so that the image is vertical, not horizontal. This way, you can take advantage of the higher resolution on the X and Y axis.

Credit
John Edgar Park

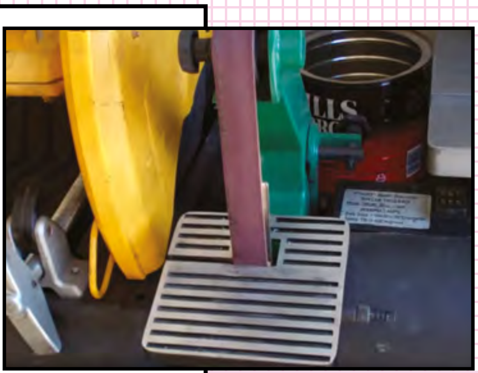
MACHINISTS TO WATCH

If you're new to metalworking, or if you're otherwise unaware, there are two must-see YouTube channels we'd recommend: mrpete222 and This Old Tony. These seasoned machine shop vets consistently deliver amazingly clear, clever, and useful content on all aspects of working metal in a small shop.

Credit mrpete222

CHEAP BELT SANDERS FOR TOOL SHARPENING

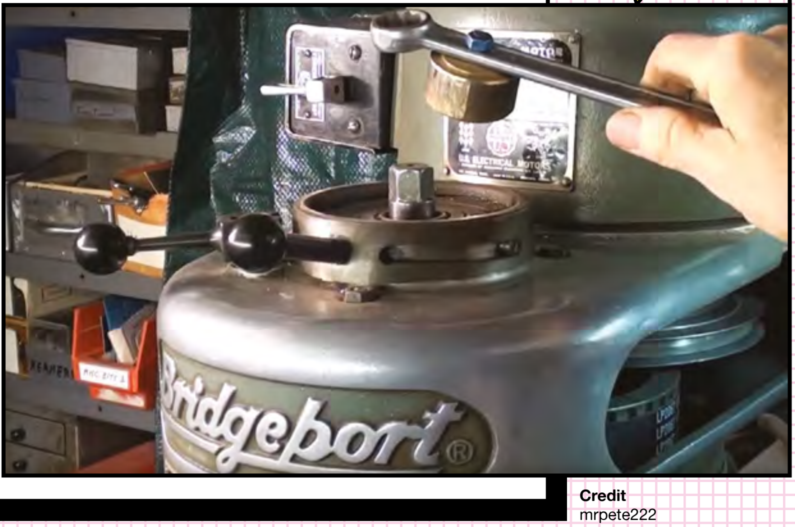
If you have a metal lathe or other shop cutting tools, consider buying a cheap belt sander to use for quick and dirty sharpening before use. You can get really inexpensive benchtop belt sanders at discount tool retailers like Harbor Freight Tools for as little as $30. Such prices make it affordable to dedicate a tool like this to a specific function. Sanding can't compete with a stone, but it will work just fine in a pinch.



METALWORK

MAKE A 'NUT BUDDY' HAMMER WRENCH

Bridgeport mills are known for needing a little percussive encouragement to release the tool from the collet. This is done by hammering on the drawbar nut that holds the collet in place. Annoyingly, this process requires two tools: a ¾ wrench to loosen the nut, and a hammer to bang the milling tool free . To help save some tool-reaching time, YouTube machine shopper mrpete222 created what he calls the 'nut buddy'. It is simply a ¾ wrench with a puck of metal attached to the wrench's arm to act as a hammer head. In two videos on his channel, he shows how he used pieces of round brass and lead stock to create two of these nut buddy wrenches. The two pieces he uses are around 1⅝" diameter and about ¾" thick. All he did was drill a hole near the neck of the wrench and he counter-bored holes in the discs on a lathe, adding a 16% chamfer to the business end. The result is an easy-to-make tool that eliminates having to grab for two tools when one gets the job done. This tool would be useful in any situation where both adjustments and percussive encouragement are commonplace. Take a look at the videos here: hsmag.cc/tqckXV



Credit mrpete222

ADD ‘BILL HOLES’ WHENEVER POSSIBLE

When machining parts, add extra holes for future mounting, attachment points, and fixturing. Add these wherever you think you might want to attach future hardware. Even if you never use them, the extra holes will help reduce the weight of your part. The name derives from a machinist named Bill, who worked in a Silicon Valley foundry and always insisted on designing-in such holes.

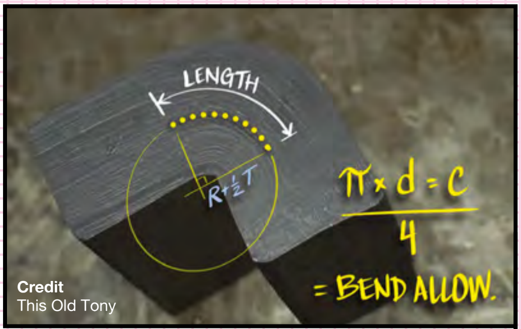
MIX YOUR OWN THREADING FLUID

On a Ray’s Garage YouTube video of lathe tips and tricks, Ray shares a recipe for creating the perfect lathe threading fluid. It involves combining a drilling and tapping fluid with an anti-seize compound (used to lubricate threads against galling or sticking). Cleverly, Ray keeps this mixture in a metal can with rare-earth magnets on the bottom. This does the double-duty of holding the can securely on the lathe, while the magnets attract all of the filings that end up in the solution during application.

USE A MARKER INSTEAD OF LAYOUT BLUE

Need to scribe a cut-mark on metal, but don’t have any traditional ‘layout blue’ handy? Use a solvent-based marker, such as Sharpie, instead.

CONSIDER YOUR BEND ALLOWANCE



Credit
This Old Tony

When you cut material that needs to be a precise geometry after it’s been bent into the desired shape, you need to take into account the ‘bend allowance’. Bend allowance is impacted by the composition of the material and its thickness. You can find web-based apps online that will allow you to calculate these allowances for different materials. Just search for ‘Bend Allowance Calculator’. For more information, take a look at This Old Tony’s YouTube video hsmag.cc/WmTrPi

MIND YOUR TOLERANCES

If you need to send a job out to a machine shop, you can possibly save yourself some money by specifying your tolerances. By being clear on how precise your tolerances need to be, you may be able to save on jobs requiring less precision, because those jobs can be done on lower-precision machinery.

CONSIDER THE WORKING TIME OF YOUR ALLOY

When figuring out the cost of a job (or the cost of hiring a machine shop), newbies often make the mistake of not figuring in the work time of various metals. Steel may cost less per pound than aluminium, but it takes more time to work, and it’s harder on tools.

COPYING THE GEOMETRY OF A SHARP TOOL TO A DULL ONE

On This Old Tony, on YouTube, Tony offers some homely advice for sharpening a bit, blade, or other edged tool that you’ve never sharpened before: “Grind your dull thing until it matches your sharp thing. Just grind your dull one until it matches.” Closely inspect the geometry of the sharp tool, and transfer what you see to the dull one.

WOODWORKING

MAKE A QUICK DADO JIG

Credit
Jimmy DiResta

If you need to make notches, i.e. dado joints, in a 2×4, Jimmy DiResta demonstrates an easy way to make quick dados using a simple jig in his '*Circular Saw Tips*' video on YouTube.

He first created the jig using some thin-ripped plywood. You're creating a wooden 'saddle' that can be fitted over the workpiece and clamped down. You want to make sure that the two top pieces of the jig are square. The width of the inside dimensions of the jig needs to consider both extremes of the saw's base plate travel as it cuts the outer edges of the dado.

Once the jig is clamped securely in place, all you have to do is start the saw at one side of the jig, and cut your way across using a very close sequence of cuts. This jig will allow you to quickly cut a number of notches in a 2×4 to the same width and depth. And, knocking out the thin cuts from the joint is at least as satisfying as popping bubble wrap.

PREVENTING TEAR-OUT

To get a clean cut when cutting with a circular or table saw, apply masking tape to the cutting edge (on the underside where the blade breaks through). This will prevent chips and splinters (known as 'tear-out').

Credit
Bill Livolsi

THE RIP N' FLIP

When repetitively ripping a lot of thin stock on a table saw (e.g. making furring strips), and you want to keep your hands from getting too close to the blade, rip the stock halfway down, flip it over, and then rip the other half. Just make sure to keep the same side of the board against your fence for both cuts.

Credit
Jimmy Diresta

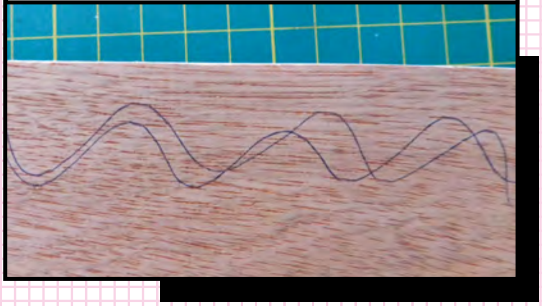
Credit
Bob Clagett,
I Like To Make Stuff

ANGLE YOUR BINS

When making storage bins for tools, parts, materials, and supplies, angle the bins slightly downward towards the back so that items can't roll out.

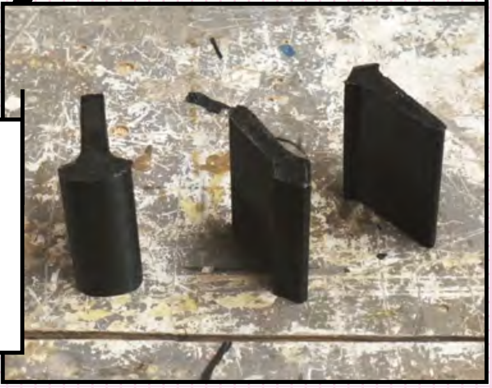
MARK A SQUIGGLY LINE OF UNTRUE

If you have wood stock and you know an edge is not a square and true 'factory edge', mark that edge with a squiggly line on both sides of the board. And only use that squiggle to indicate an 'untrue edge'. That way, you'll know at a glance which edges are true and which aren't.



MAKE CUSTOM SANDING GRIPS

Using 3D design software, you can quickly design custom grips that conform to a specific, unique surface you need to sand. You can then 3D-print these custom grips. If you've designed and 3D-printed the object you're needing to sand, you can even use the negative space information from your 3D design to create grips for sanding inside of these difficult places. Since print resolution doesn't mean much here, you can print at the lowest resolution to accelerate the print time.



Credit
Bob Clagett,
I Like To Make Stuff

SCOOPING UP SQUEEZE-OUT WITH A STRAW

'Squeeze-out' is the excess glue that squishes out when bonding two workpieces. Trying to remove this excess glue from inside corners can be challenging. One way to attack this is to use a plastic drinking straw to scoop up the squeeze-out from these difficult-to-reach areas.

LET WOOD GLUE THOROUGHLY DRY

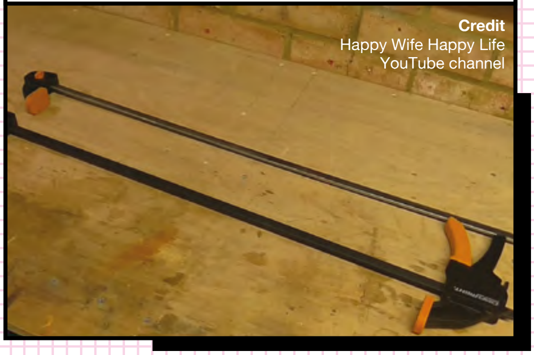
When applying a water-based glue, it will swell the grain of the wood around the join. Make sure the glue is thoroughly dry before doing any sanding.

USING WAX PAPER WHEN GLUING AROUND METAL

When gluing up a project using bar clamps, wood glue can chemically react to the metal of the bar and stain your wood black. To prevent this, and to make clean-up easier, place wax paper over the bar clamps beneath the project before clamping.

MAKE CUSTOM BAR CLAMPS

It is easy to make bar clamps any length that you wish. First, remove whatever rivets are holding the clamping mechanisms on your existing bar. Next, you'll need some bar stock that matches the height and thickness of your existing bar. Now, remove the clamps from your existing bar, and create a new stop for the clamps on the new bar by drilling a hole in the bar and installing the hardware taken from your existing clamp.



Credit
Happy Wife Happy Life
YouTube channel

HOT GLUE AS A TEMPORARY 'CLAMP'

Credit
Jimmy DiResta

If you don't need a high degree of clamping pressure (e.g., when using a router or CNC machine), consider hot-gluing your workpiece right onto your bench. By gluing instead of clamping, you have the entire surface of the workpiece free for working on. When done, it's easy to pry the piece free. You can also use hot glue as a temporary clamp in concert with a longer-curing glue (e.g. carpenter's glue) to allow a quick join before the long-term glue has fully cured.

ZEN AND THE ART OF CUTTING AND ROUTING WOOD

Be mindful of the wood you are cutting or routing. Pay attention to the qualities of the wood as you cut it. Listen to your tool. Pay attention to how the cutting looks, smells, sounds, and how the tool and wood are behaving. When cutting or routing, everything should be stable and steady, with no unusual howling, jumping, or burning smells. □

SAVE UP TO 35%
ON THE COVER PRICE

OTHER WAYS TO SUBSCRIBE

Get your first three issues for £5:

- → The cheapest way to try out HackSpace magazine
- → Automatically renews at our most affordable rate

Rolling subscription for just £5 a month:

- → No long-term commitment
- → No large up-front cost

Six issues for £30:

- → 792 pages of hacking and making
- → Easy to manage

DIGITAL SUBSCRIPTIONS ALSO AVAILABLE

Available on the App Store

SEE PAGE 72

GET IT ON Google Play

Visit: hsmag.cc/subscribe **or call** 01293 312189

How I Made A CNC MILL

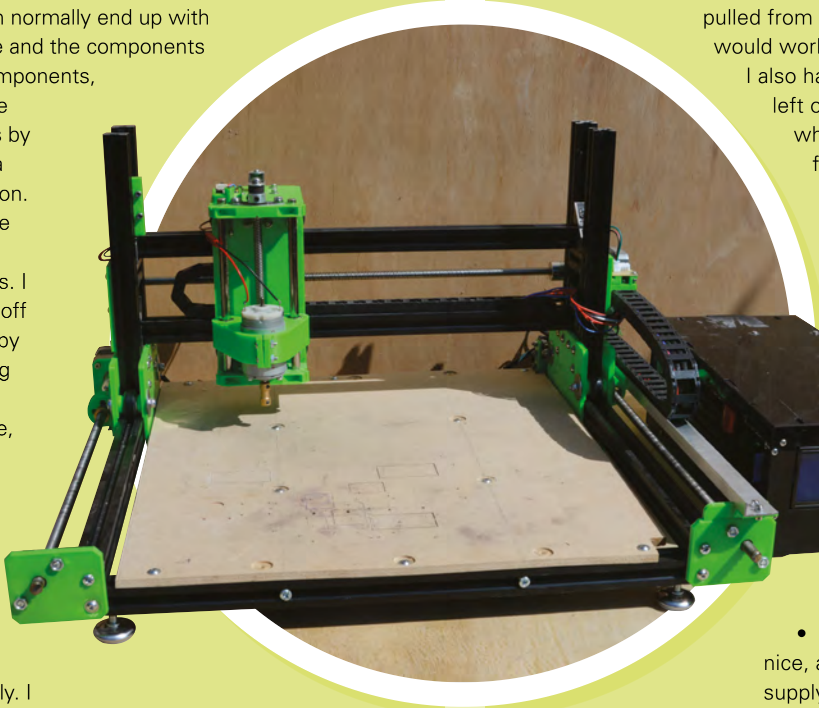
Custom-made PCBs at the press of a button

By **Will White**

Making PCBs by hand can become a bit of a mission! Etching a board takes ages, with numerous processes. Then there is the drilling of lots of holes, which normally end up with them being out of line and the components not fitting. Broken components, large holes, and fragile traces – making PCBs by hand can come with a huge dose of frustration.

It was getting to the point where I was reluctant to etch PCBs. I was sending designs off for them to be made by a board house, waiting two weeks for my custom PCBs to arrive, only to find I'd made a mistake with the design. I wanted to make a board quickly, test it, modify if necessary, before committing it to be made professionally. I decided I wanted a mill for PCBs.

Looking about, there are many small CNC mills on the internet for around £125, but you're not getting much for your money, compared to a 3D printer kit of a similar price. So, I decided to adapt a 3D printer into a CNC mill: this would give me a much bigger work area. Also, the spare parts would come in handy elsewhere.



Above
My CNC mill, ready to create some PCBs

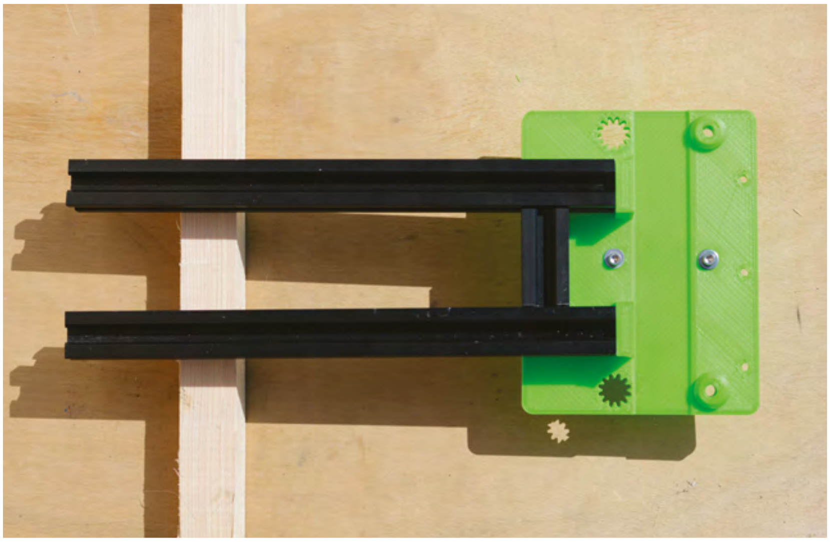
FINDING PARTS

Having a look in my salvaged parts bin, I could use a DC motor and some linear rails pulled from some old inkjet printers. That would work for the spindle and the Z rails. I also have some T-nuts and other bits left over from a 3D printer project, which should keep the cost down for extra parts.

Now for the donor printer – it has to have:

- A frame made from 20 × 20 aluminium profile, not one cut from sheet material.
- Linear rails or (V-groove) wheels – NOT unsupported rods, as these can flex.
- A heated build plate would be nice, as it would have a bigger power supply. I could upgrade the motors and spindle at a later date.

I identified some new 3D printer kits for around £110. It's also worth looking at second-hand machines, especially ones listed as 'for parts' or 'not working'. I finally bought a HC Maker 7 – a CR-10 clone with



Right
The Y-carriage with toothed holes for the eccentric spacers

a 30 × 30 cm heated bed. It was listed as not working.

DIGITAL DESIGN

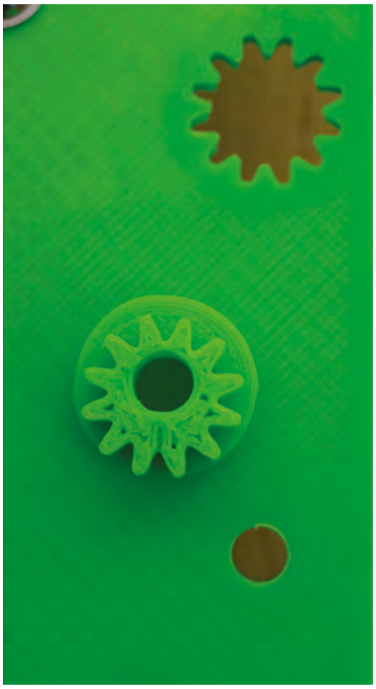
I opted to use Fusion 360 to design the machine, because I could create the whole machine in software first, checking components fit together before printing them. This approach should save me a lot of time and filament by reducing the amount of 3D printing needed. In short, planning out the project first may feel like it's using up time, but in the long term, it saves much more time (and money) than it takes up.

Every part of any project has to be a compromise. Time vs. money vs. functionality vs. aesthetics vs. quality etc. Of course, everyone wants a beautiful machine that does everything, costs nothing, and is working within seconds. Now back to reality! I've got a 3D printer, therefore many of the components will be printed.

I want a machine that:

- Is as stiff as possible. This gives a better-quality finish to workpieces and can open up the range of materials to work with.
- Has minimal play in all axes.
- Can be upgraded at a later date, to use better-quality components and stronger motors.
- Can mill PCBs as a minimum.

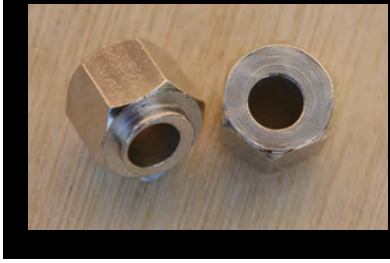
I designed the main components with the idea of being able to replace them with 5 mm aluminium plate in the future. For now, each component was designed with consideration to its orientation during printing, as this will affect its strength. I printed all parts on my Replicator 2X. Structural components were printed at 30% infill, four perimeters, and three top and bottom layers. Non-structural components were printed at 10% infill, two perimeters, and two top and bottom layers. All in ABS.



Left
The offset spacer: turn the cog to a new position and rebuild the carriage to alter the tension on the wheels

Eccentric spacers

Eccentric spacers are spacers designed with an off-centre hole. The bottom part of the spacer sits in a hole in the frame and acts like a cam. The wheel is mounted to the top. The tension of the wheel is adjusted by turning the spacer.



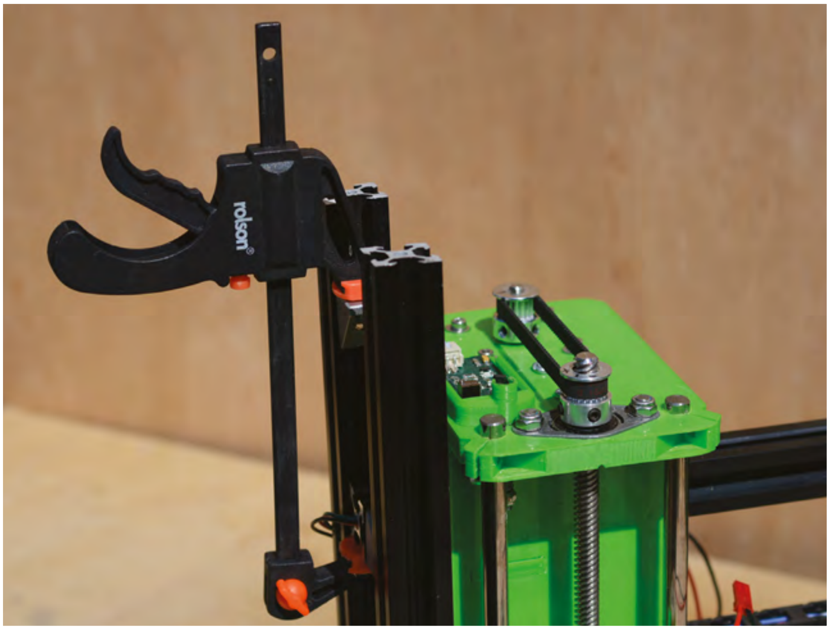
Other decisions to make at this stage were:

- Moving gantry or a moving bed?
- Lead screws or timing belts?

THE BUILD

First, I reorganised the base of the printer. This entailed swapping the central bars with the outside bars along the Y axis, so that the Y-carriage will run on the edge of the frame, and the spoil board will be supported underneath.

My first problem was how to stop the machine from flexing backwards and forwards. The original printer had the Z axis attached to the base by a bolt and a small →



Above Holding the mill together while it's assembled

bracket. There was a lot of flex within the frame, I managed to snap one of the acrylic brackets by just moving the printer around.

My solution was to use two vertical sections of frame braced with two 45 mm sections, on each side.

I originally thought that I would trim the uprights to size at the end, but I decided to leave them to allow for thicker stock materials.

The next issue was how to stop the wheels in the Y-carriage from binding. On my first attempt at making the Y-carriage, the wheels were too close together, only by a fraction of a millimetre, but the carriage would bind up and stop moving. As each carriage took over six hours to print, it was not practical to move the holes for the axles on a trial and error basis. My donor printer uses eccentric spacers to solve this problem (a spacer with an off-centre hole). You turn the spacer in order to adjust the tension of the wheel. These spacers wouldn't work with my design, but they did inspire the solution.

I took the spacer from my design and mounted it onto an offset cog. I then added the corresponding cog-shaped hole into the side of the Y-carriage. A few penny washers ensure everything stays in place. Now, by turning the cog into a new position and rebuilding the carriage, I can change the tension on the wheels.

> To keep the machines cutting in the right place, I reused some endstops from the printer

To set the tension in the X axis, I first fixed the lower rail in place with the X-carriage in situ. Then, using a spare corner bracket, I moved the X-carriage to the end of its rail and clamped the upper rail in position. You need to repeat the process on the opposite side, and bolt everything in position before removing the clamps and the spare corner brackets.

ELECTRONIC CONTROL

To keep the machines cutting in the right place, I reused some endstops from the printer. For the Z axis, I built a Hall effect endstop from open-source designs. The spindle motor is controlled from the MOSFET for the heated bed.

I had originally wanted to use GRBL (firmware) to control my machine as it's designed for CNC routers. However, the motherboard from the original printer had been designed in a way which was not compatible with GRBL, so I had to use a 3D printer firmware. I chose Repetier-Firmware because it had a CNC mill mode, and the online setup tool was easy to use (see boxout).

The Repetier website has extensive documentation on how to set up your machine.

The key points were:

- Select the correct motherboard from the list, else create a custom profile.

GRBL vs. Repetier

GRBL's primary design is for CNC mills, routers, and laser cutters. Repetier-Firmware is designed for 3D printing.

Both firmwares can be used interchangeably, but using firmware designed for a slightly different purpose normally results in some form of compromise.

Repetier adds support for SD cards and LCD screens, whilst GRBL wants a host machine to offer that support.

Repetier does not support all the functions of GRBL. Repetier currently only offers on/off control for spindle motors and no PWM control. Repetier needs to be told every time to work with negative coordinates, otherwise, it thinks it's going to drive the hotend into the build platform.

For endstops, GRBL uses the Pin Change Interrupt (PCINT) function on the AVR chip's dedicated pins, whilst Repetier polls the endstop pins on a regular basis. So Repetier can use any pin for the endstops, whilst GRBL has to use specific pins.

I wanted endstops on my machine, but the 3D printer's motherboard didn't have any free PCINT pins available. This forced my choice to use a 3D printer firmware rather then GRBL.

- Add the machine dimensions.

- Check your endstop settings and motor settings.

- Tick the 'support CNC mill' checkbox.

- Change the 'default printer mode' to CNC mill.

For the first time, download the entire firmware. For adjustments to your profile, you can download the **Configuration.h** file and replace the existing version in the **firmware** folder. Now open **Repetier.ino**, using the Arduino IDE, and upload the firmware to the motherboard.

FIRMWARE FUN

I'm using Fritzing to design PCBs and exporting them as Gerber RS-274X files into FlatCAM.

FlatCAM does a good job of generating G-code from Gerber files. However, because I'm using a firmware designed for additive manufacturing and not one for subtractive manufacturing, Repetier does not like to work in the negative space. So you need to tell Repetier to permit the machine to work in the negative space by adding **G1 S1** to the beginning of the G-code file. You can do this by either editing the G-code in a text editor and adding **G1 S1** to a new line below the first '%', or you can get FlatCAM to do it for you automatically. Under the Options tab, find the text box to add custom G-code to the start of the file. Enter **G1 S1** in that box in the line beneath the '%'.

> I'm using a firmware designed for additive manufacturing and not one for subtractive manufacturing

While you are there, I found it worthwhile adding the lines **G00 Z0.0000** and **M18 Z** to the end G-code box in the lines above the '%'; these commands move the Z axis to zero and disable the Z stepper motor (while keeping the X and Y steppers engaged) once it has finished the job. I use this to change the tool and set it to the zero position before starting the next job.

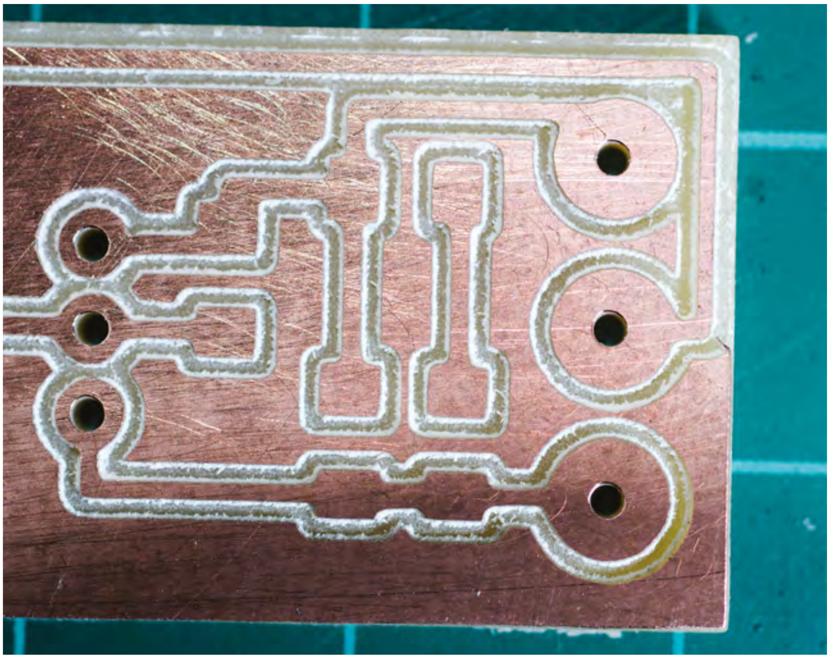
When FlatCAM writes speeds to its G-code file, it does not add G00 to the beginning of the line. Repetier does not like this, and runs the machine at any speed it fancies, so you have to add the correct syntax manually. Open the G-code file in a text editor and find the line near the top starting 'F' – it might say 'F100.00'. Add **G00** to the beginning of this line, so it reads 'G00 F100.00'; this now tells Repetier to run the machine at a speed of 100 mm per minute.

MY PCB MILL

I feel that I have achieved everything I set out to do with this mill. It's got a work area of 290 mm × 258 mm × 99 mm and can cut PCBs and other soft materials. In total, I spent £156 on this machine, of which £125 was for the donor printer. Compare that with a machine from China with a smaller work area of 240 mm × 180 mm × 45 mm for £170, which can cut similar materials.

What I didn't anticipate was how much I would learn along the way. I would encourage people to give it a go: create a pen plotter, a mill, or a 3D printer for yourself!

This machine is not finished – already I'm thinking how can I improve it. Dust extraction and an enclosure are first, followed by a probe for tool length, and aviation-style connectors. Eventually, I'll give it bigger motors and a 32-bit board. □



Left
A milled PCB ready to be populated

Liverpool MakeFest

Saturday June 29th 2019

A free day of fun for all the family!

lpoolmakefest.org

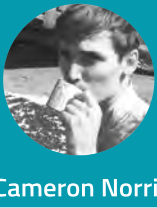
VILROS
BUILD, PLAY, WIN!

USB
5 Pack

50 PROJECTS FOR MAKERS & HACKERS
BOOK OF MAKING
VOLUME 1
BUILD A TREBUCHET
MAKE MUSIC ON A DIY SYNTH
COOK FOOD IN A HOMEMADE TANDOOR
FLY YOUR OWN CUSTOM DRONE
BUILD AN A.I. WATER CANNON
CONTROL YOUR HOME: SMART TECH YOUR WAY
& LOTS MORE
FROM THE MAKERS OF HackSpace MAGAZINE

Open-source hardware certification

Certify your projects as 'genuine' open-source hardware



Cameron Norris

@CameronSNorris

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

As the open-source hardware (OSH) community continues to gather momentum, this article explores what it means to be 'OSHWA certified', as well as how a little-known group of amateur radio enthusiasts helped to launch this exciting movement.

Since the Open Source Hardware Association (OSHWA) launched their highly anticipated certification program in 2016, it has attracted more than 250 successful applicants from 30 different countries.

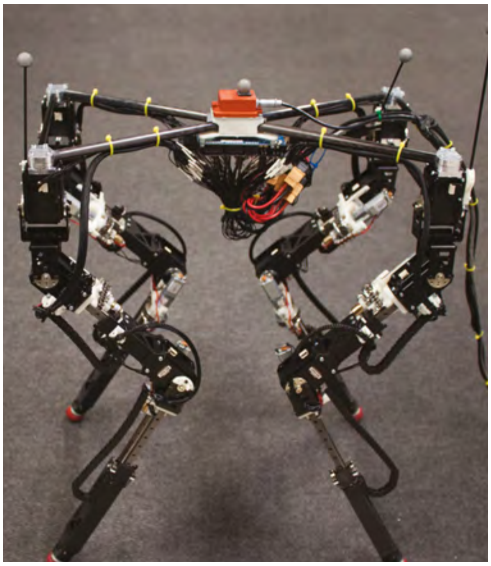
"If you think about food being certified organic or fair trade, it provides an easy to distinguish and understand label for something meeting certain quality criteria," explains Sebastian Pichelhofer of the OSHWA-certified AXIOM project in Austria. Pichelhofer has spent the past eleven years developing a cinema-quality OSH camera, and he firmly believes that a clear quality label to guarantee 'genuine' OSH is necessary to prevent abuse of the term as the popularity and awareness of open-source continues to grow. "That's why we were one of the first projects to apply for the OSHWA open hardware certificate and still carry it with pride," he says.

OSHWA CERTIFICATION IS FREE

OSHWA themselves agree with Pichelhofer's sentiment. "In many cases, creators would label their hardware as being open-source, and use the open gear logo without complying with the community definition," explains OSHWA Board President, Michael Weinberg. "The Certification programme makes it easy to identify hardware that fully complies with the community definition of OSH."

In a nutshell, hardware creators who join the programme are required to enter a legally binding contract that requires certified projects to comply with the community definition of OSH. Only then can the project be assigned a unique identifier and be included in OSHWA's searchable list of OSH.

"I think the OSHWA and its certification is a great initiative," says Tønnes Nygaard, a PhD candidate developing 'evolutionary robotics' at the University of Oslo. "I always have a lot of people asking about it when presenting at scientific conferences or maker faires, and displaying the logo with our robot clearly shows that anyone can make, use, or modify our project for their own needs, as well as signifying that we fully support the open-source movement."



Right ◈
The DyRET robot is a certified OSH project, which means all software and hardware design can be found on Tønnes's GitHub page: hsmag.cc/dyret-documentation

LOGO

Macklin Chaffee designed the Open Hardware logo most people are familiar with in 2011, as a submission for the OSHWA community logo design contest. After being selected as the winning entry, it was released under a Creative Commons Attribution-ShareAlike licence for the rest of the OSH community to use.

Above
The DyRET robot during a 'real world' test outside the University of Oslo

Below
Nygaard's research interests include distributed robotics, mobile computing, and programmable matter

Nygaard's project is a four-legged research robot designed to change the length of its legs while still in operation. "Building a robot like we have comes with a lot of challenges, and hopefully, some of our solutions can be used by others to make their work a bit easier."

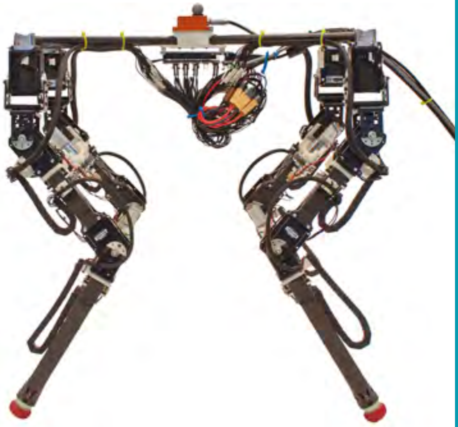
As a PhD candidate, Nygaard is used to sharing his experiments and findings through scientific publications, but these publications do not traditionally include much of the background engineering involved in the research. Going open-source enables Nygaard to close this gap by sharing physical designs, electronics, software, and documentation in a meaningful way. "Science is all about discovering and spreading knowledge, so I see this as a natural extension of that," he says. "Open-source is already a huge part of the maker world, and we are trying our best to expand its use in the scientific community as well."

OPEN EXCHANGE OF DESIGNS

At the last Open Hardware Summit in 2018, OSHWA announced version 2.0 of their OSH Certification programme, "with a new website, a new directory, and lots of new resources for learning about OSH." The new website even includes case studies from the community to help illustrate the best practices for compliance with the community definition of OSH, making it, "easy to be sure that you have provided all of the information needed for someone else to make use of your hardware and removed any legal barriers to use it," says Weinberg.

For those interested in joining the certification program, Nygaard had a few words of advice. "When it comes to open-sourcing a project, a very important part that people often miss is the documentation. It doesn't help if all the design and code is available if it is too hard to start using or modifying it. Writing documentation can take a while, but it's a great investment for the community," he says. Pichelhofer's advice was a lot briefer but no less relevant. "Do it!" he told me with a smile emoji.

To find out more about the OSHWA Certification programme visit: **certification.oshwa.org** □



A short history of open-source hardware

On Tuesday 26 August 1997, Bruce Perens announced the world's first Open Hardware Certification programme. Perens's programme required computer hardware manufacturers to publicly release the documentation for 'programming the device-driver interface' of any products they wished to certify.

The first company to certify a product as Open Hardware was the Cyclades Corporation based in Fremont, California. The 'Cyclom-Y' was a multi-port serial card used to connect things like modems, printers, and scanners to computers. As a certified device, the packaging of the Cyclom-Y was allowed to include the trademarked Open Hardware logo.

Over the next few years, several more attempts were made to apply the principles of open-source software to the design and development of hardware, but most of these early efforts quickly faded into obscurity. It wasn't until 2007 that the Tucson Amateur Packet Radio Corporation (TAPR) created the first hardware-specific open-source licence we recognise today.

Below Over 300 people joined the first Open Hardware Summit in 2010

THE FIRST OPEN-SOURCE HARDWARE LICENCE

TAPR was set up in 1981 by a group of packet radio enthusiasts to 'help advance the amateur art of communications' by providing kits, reproducible designs, and other useful resources to fellow hobbyists. Packet radio is a method of sending and receiving small bursts (or 'packets') of digital data from one computer to another using amateur radio transmissions. It was (and arguably still is) commonly used to send and receive short personal messages, or to interact with local 'Bulletin Board Systems' set up to help packet radio operators connect with others in the community.

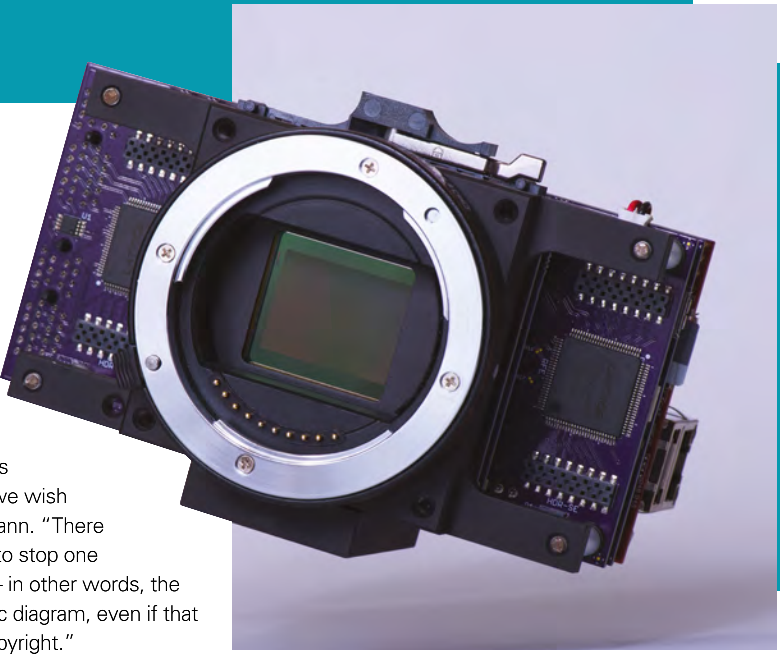
In 2005, TAPR began working with the HPSDR group, a community of 'experimental-minded radio and computer enthusiasts' aiming to develop high-performance, software-defined radio products through 'open and co-operative efforts'. The group were inspired by the effectiveness of the GNU General Public License (GPL) at preventing open-source software from being made proprietary, and hoped that TAPR could help them to apply a similar concept to the protection of community hardware projects.

In turn, TAPR approached fellow amateur radio enthusiast and software licensing attorney John Ackermann to take responsibility for drafting an

OPEN TERMINOLOGY

The term 'open source' was first suggested in 1998 by Christine Peterson to replace the term 'free software', which had been popularised by the Free Software Foundation established by Richard Stallman in 1985. The problem with the term 'free software' was that it often led newcomers to assume it only referred to the price rather than 'free as in freedom', which included many more benefits.

appropriate licence. This was by no means a simple task. The GNU GPL was based on US copyright law, which automatically protected written code as a 'literary work', but did nothing to protect the 'idea' behind it. "This is the fundamental difficulty in creating a licence for OSH: it is the idea, or at least its physical implementation, that we wish to protect?" explained Ackermann. "There is nothing in the copyright law to stop one from implementing the circuit – in other words, the idea – described by a schematic diagram, even if that diagram is clearly subject to copyright."



After almost two years of circulating initial drafts among fellow attorneys, members of the HPSDR group, and other interested amateur radio hobbyists, Ackermann released the first public draft of the TAPR Open Hardware License (OHL) through Perens's blog on **Technocrat.net** (now viewable through the Internet Archive) in 2007. Following a 60-day public review, TAPR formally adopted the licence.

FREE AS IN FREEDOM, NOT AS IN BEER

Three years later, in 2010, another big leap occurred when Ayah Bdeir and Alicia Gibb joined forces to remove the ambiguity around how to do business as an open hardware company. Bdeir was trying to turn her, now wildly successful, littleBits project into a viable business and wanted to understand the legal tools available to her more clearly. She decided to hold a workshop at Eyebeam Art and Technology Center in New York, as an opportunity for open hardware developers facing similar questions to exchange ideas.

Lucky for Bdeir, the date of her 'Opening Hardware' workshop was the same week as a major Arduino NYC community meetup, which attracted several influential figures to the area, many of whom were interested to hear what Bdeir had to say. Partly due to this coincidence, workshop attendees included four of the five co-founders of the Arduino project (David Mellis, Gianluca Martino, Massimo Banzi, and Tom Igoe), the founders of SparkFun and Adafruit (Nathan Seidle and Limor Fried respectively), and Bug Labs R&D director Alicia Gibb, who was in the process of arranging a summit to discuss the complications of manufacturing OSH.

Ultimately, Bdeir's event was a big success, and follow-up discussions between participants eventually led to a clear definition that enabled the community to specify whether or not a hardware licence qualified as open-source – this new 'OSH Definition' was largely derived from the Open Source Definition for software, written by Perens in 1998.

Impressed by the workshop, Gibb invited Bdeir to co-chair the Open Hardware Summit she had been planning for later that year, and Bdeir agreed. Over 300 attendees participated in the debate. The summit became an annual event, and further changes were made to the OSHW Definition to include input from more people who wished to contribute.

The OSHWA that we know and love today was established as a non-profit organisation by Gibb in 2012, and since then has gone from strength to strength. "To continue to spread and improve, open-source hardware needs all industries to participate, especially engineering," says Gibb. "It is my hope that next time you're working on a hardware project, you'll ask yourself if there is an open-source version out there." □

GOING NUCLEAR

The CERN Open Hardware Licence is an OSHWA-compliant alternative to the TAPR OHL. It was published in 2011 by electronic engineers working in experimental physics at CERN. Led by Physicist Javier Serrano, the group viewed 'open source' as a way to improve the efficiency of scientific hardware development while facilitating greater collaboration between members of the scientific community.

Above
The open-source AXIOM Beta developer kit currently offered by apertus°

Left
Alicia Gibb (left) and Ayah Bdeir (right), at the first Open Hardware Summit

HackSpace magazine meets…

Jen Schachter

Mixed-media maven, maker, and mastermind of interactive public art projects

One of the things that always takes us by surprise is the breadth of experience that people bring to their making. We've spoken to self-taught people from all walks of life: electrical engineers, carpenters, sculptors, loads of teachers, and one former funeral director, but we've never had a fine artist in the magazine, until now.

Jen Schachter has gone from art, to bashing together shelves, to working for the Obama White House, to looking after Adam Savage's vast cave of infinite stuff, as seen on **Tested.com**. She's also part of the team behind the incredible *Rosie the Riveter* sculpture, so we thought we'd find out from her how such a thing came into being, why she's carrying ET with her, and what it's like being escorted across Washington by the Secret Service. →

Above
ET's eyes fell out after the heat of the Mojave Desert melted the adhesive holding them in. He's looking much better now, after some duct tape and a little love

Above
We take our hat off to anyone who can coordinate over 700 maker contributers

HackSpace We saw the giant 3D-printed sculpture of *Rosie the Riveter* a while back, and it was glorious. How did it come about?

Jen Schachter That was a crowdsourced 3D-printed sculpture produced with a group called 'We The Builders', which was founded at a hackathon in 2014.

They started a platform for crowdsourcing the pieces. They take a sculpture of an object, usually a person, digitally scan it, and then slice the 3D model into sections, usually about three-inch cubes. All those part files, the STL files, go online and people can sign up and claim a part, print them at home, and ship it to one of their locations to be assembled.

Rosie the Riveter was the fifth one they've done, and the biggest they've made. We had about 750 contributors around the world, 2600+ parts, and the final sculpture was six feet tall. We assembled it in one day, with Adam Savage and the folks from Nation of Makers – it was a big community build. Some of the people who had contributed parts were there on the day; we laid everything out on the floor, we had a blueprint for each layer, and we glued them all together and stacked them up.

HS 2600 pieces must have taken a lot of organisation.

JS There's this super-awesome web developer we have called Marty McGuire who designed the platform. It's pretty simple: you log in with your Google ID, you get assigned a piece at random, but in the back-end, we can log in as admins and check the status of that part. So, as you claim the piece, we have a list of pieces that have been claimed. Once the piece has been printed, you let us know, and we can also view that; so in the back-end, we can pull a status update into a spreadsheet. We'll send you an email if you've been holding a piece for a week and haven't yet printed it, or haven't shipped it to us. We can pull all those parts and generate an email that will say, "Hey, thanks for taking this piece; if you don't print it, we'll re-release it into the system". And then, we also collect tracking information, so we pretty much know where all the parts are at any given time.

HS Are you working on anything similar at the moment?

JS I'm working on a project for the Smithsonian National Air and Space Museum in Washington DC. We're doing a replica of the Apollo hatch, to mark the 50th anniversary of the moon landing. So you've probably seen it, it's a trapezoidal shape with rounded corners. It's a really iconic part of space history, it's important because the original one… there was a fire, and the astronauts couldn't get out. So they made this door to be able to egress from the command module in three seconds flat. It's a really complicated piece of machinery, and there are loads of latches and pressure gauges to stabilise the pressure inside the module.

We're taking a digital scan of the door, which is in the Smithsonian, and we're recreating it in a variety of media. It's a similar idea to the *Rosie the Riveter* sculpture except that, instead of slicing it up into a grid, a really amazing CAD modeller Andrew Birch has taken the entire digital model and direct-modelled each mechanical component. Every single nut, bolt, piston… it is an incredible piece of modelling – it's mind-blowing. He spent about two months working on this.

> "It could be machined, it could be CNC milled, **it could be sculpted in clay, it could be paper mache**"

Every individual component of the entire door is now a full CAD model in Fusion 360. He's been exporting those as an engineering diagram, and then different artists will replicate a piece in their medium. It could be machined, it could be CNC milled, it could be sculpted in clay, it could be paper mache, but they'll all be different representations of a piece of this machinery, and then we'll assemble the entire thing and it will be on display for the 50th anniversary.

HS What are you going to be doing next?

JS I'm going to San Francisco to be Adam Savage's shop manager. He's got a cave in The Mission in San Francisco. He's got a shop where he builds all of his passion projects. It's full of props, and I'll be in there running it. I'm looking forward to getting familiar with the gear, but it's going to be quite a task getting to know where everything is.

HS How much running does a shop like that take?

JS That's a good question. I don't know. I know that there are a lot of things in storage. For example, I was working with a television show called *Savage Builds* that's coming out in June. When Adam needed something – a particular prop or a piece of armour, for example – somebody has to know where all of that stuff is and pull all those items out of the cave, make sure they're polished, clean, and ready. Simple things like transporting machinery, taking inventories, making sure everything comes back. Lots of logistical things, and also helping with project builds. I seem to have a brain for logistics, I end up managing a lot of big projects.

HS You told us earlier that your background is in fine art. Does that lead into what you make? →

Above ↑
Jen's advice for anyone, in any situation: make yourself indispensable

JS In a funny way, I'm not doing anything I went to school for. I spent a lot of time working with oil paint and charcoal and turpentine, and I still have all of that stuff, even though I haven't used it in years. I think the design sense has informed the work that I make, certainly. But, I wish I had gone to school for industrial design, because I really like making big immersive pieces and environments, but I am very much self-taught in woodworking and fabrication – I wish those were skills that I had studied formally.

HS There's a great quote from someone at the BBC Radiophonic Workshop on being self-taught: "Because we weren't experts, we didn't know what we shouldn't be capable of doing."

JS There's definitely a lot of that. I think in working with Adam, you learn things, a lot of self-taught ways. He has his own Adam hacks, ways of doing things. You start to pick up a lot of things that aren't maybe the accepted way to do things, but they're a really efficient, quick, and dirty way to do stuff. It's useful to learn from somebody who's a really proficient, successful hacker.

HS How did you get that job?

JS I was working at a youth tech centre called the Digital Harbor Foundation, based in Baltimore, and they do after-school programmes for youth and various technologies, laser cutting, 3D printing, coding. I was the on-site builder/fabricator. I built tables and storage cabinets, things like that.

I was really interested in making stuff for my house. I needed a table and a shelf for my house, so I bought a saw and somebody else got me a drill, and I started creating things out of necessity.

Let me think… I went from waiting tables and working in a kitchen to working at a corporate – I was freelancing doing window displays. I was dressing mannequins and doing backdrops, lighting, and stuff. That seemed like a pretty natural progression. From there, I went to working a corporate job at Under Armour, which was doing official merchandising. I started writing manuals on how to dress mannequins, how to build this grid wall, or whatever.

The corporate life was not for me at all. I enjoyed communicating how to build something to other people; I enjoyed communicating to other people and using diagrams, and figuring out the right language, but there wasn't a lot of creativity, so I totally ditched corporate and went into non-profits. I had all this experience of throwing things together with my borrowed mitre saw, and I convinced them that I could go and build a makerspace, and I did.

> "Do you want to do a project for the White House with Adam Savage?" **Let me think about it… yes!**

I did a couple of creative projects for a hackathon that they ran, and my boss realised that I could do more creative, more artistic building projects. He then got a job at the science advisory board at the White House. At the end of the Obama administration, they had this festival, called South by South Lawn (as in the South Lawn of the White House), which is a spin on South by Southwest, and I got a call from Andrew, and he said, "What are you working on? Are you busy? Do you want to do a project for the White House with Adam Savage?" Let me think about it… yes!

The catch was that it was due in three weeks. So, I quickly learned how to use a dado set on a table saw, and all these complex angles. And so, as quickly as possible, I fabricated – basically cabinets, but in the shape of letters. I fabricated all the pieces, and Adam and I were sending sketches back and forth this whole time; he would draw something, email it to me, and I would show him what I was working on, and we'd send things back and forth. The project culminated when the day before the festival, Adam came to Baltimore and we met up with 50 kids from the Digital Harbor Foundation, and in a frenzied 13-hour day, we assembled the whole thing together, wired it up with LEDs, painted the whole thing, and then drove it to the White House the next morning.

We arrived, and it took a really long time to get through the Secret Service, because they have to check everything, bomb-sniff everything. Once they've checked everything, they lock the truck, and you have an armoured escort to drive you to the location. So, we were driving the U-Haul with front and back armoured cars through Washington DC to the White House, because the checks take place about a mile away.

We get there just before the festival opens, and we had eight minutes to install this centrepiece. I designed it so you could bolt the back pieces on to it, so there's no drilling or anything, just a through-bolt construction. I think we had a minute to spare. It was the centrepiece of the event, so people could tweet during the event, and certain hashtags would cause the lights to change colours.

The colours would change throughout the event, and once the sun went down, people had picnic blankets out. If you look up South by South Lawn, that's the image that comes up most often, these big letters.

That was my first collaboration with Adam. I was kind of in the eye of the storm, managing all these things. Adam came to me afterwards and said: "This is really awesome; we should do more stuff". That's how it started.

HS What did Barack think of it? →

Above ◈
Check out schac_attack on Instagram for loads of flawless clever, artistic builds

JS It's funny, he never stood in front of the sign. For security reasons, he's not allowed to be in areas where there are above a certain number of people if he's not behind a barricade. He was there, on stage behind a thing. I think he liked it.

HS Finally, we always ask everyone about failure, because it's important.

JS Failure is an interesting one for me. I don't look back at any of my projects and think that they've failed, even if they turned into something different than I thought they would. The biggest thing that I run into is impostor syndrome. All the time. I don't feel qualified to be doing the things that I'm doing.

I got a call right before the holidays, inviting me to come and work on a television show. Working on that show was a time when I was around a lot of people who had been in the industry for at least a decade. They're all extremely talented at what they do. I was so humbled to be among them. And I keep finding myself in situations where I don't feel like I belong; I don't feel like I deserve to be there. I have to keep telling myself that I'm there for a reason and I'm working really hard. Adam pulled me aside after a particularly rough day – I think he must have seen it on my face – he said: "Listen, I know I pulled you into this hurricane, and I can tell that you're worried about this, but I brought you here because you know how to think, and I can teach you everything else you need to know. Because, as long as you know how to think and you know how to learn, you can make it in this field." I really, really needed that that day. I was shrivelling inside because I felt that much impostor syndrome. Working alongside people like Simone Giertz and Laura Kampf, whose work I have admired from a distance for a long time, and then suddenly we're working alongside each other in a shop…

But people are just people. I think I said on-stage, that a good thing to keep reminding yourself is that if you are indispensable in every job that you apply yourself to, whether it's a job job, or you're volunteering at something, if you are indispensable at everything you do, if you work hard, you're easy to work with and you're willing to learn, then people will keep asking you back. □

Left We asked Barack Obama what he thought of the SXSL sign. He said "I like it"

Below Jen turned a 2D image of Rosie the Riveter into a 3D model

> I don't look back at any of my projects and think that they've failed, **even if they turned into something different**

SODA CANS

IMPROVISER'S TOOLBOX

The hand-held food preserver is a gift that keeps on giving

Mayank Sharma

@geekybodhi

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

Over 200 years ago, it was the can, or rather its very crude progenitor, that helped Napoleon save his troops from dying of hunger and scurvy. The lightweight recyclable containers have come a long way since the days of the crude tinplate canister, but they still pretty much do the same thing as before: shield their contents from light and oxygen. On top of this, the can doesn't rust or corrode, which is why it offers one of the longest shelf lives as far as packaging consumables is concerned.

Aluminium cans were initially used for storing frozen juice concentrate. Coors is often credited as being the first to package their beverages inside an all-aluminium can, back in the early 1960s. They also kickstarted the recycling movement by offering a cent for every can that was returned to the factory. Can production increased throughout the 1960s, and went through the roof when both Coca Cola and Pepsi started packaging their drinks inside aluminium cans.

Another characteristic that fuelled the popularity of the aluminium can is its weight. The cans have shed weight over the years, and now weigh about 15 grams. Yet they are sturdy enough to support the pressure required to package carbonated drinks. In fact, studies show them to withstand 90 pounds per square inch of pressure.

The can's thin aluminium body is a conducive surface for printing, and has a lining inside it that helps protect the flavour of the beverage from tasting metallic. While the exact combination of metals used to create the can varies from one country to another, a majority of the cans are made up of recycled metals.

Thanks to these properties, besides food and beverages, cans are also used for housing several other items, including spray paint and aerosol products.

Cans are simple objects, as far as their construction is concerned. The early cans didn't have a pull-tab, and to get to its contents, you had to use a device known as the churchkey. Ermal Fraze, who owned a tool company, devised the pull-tab after spilling most of the contents of his cans trying to open them without his churchkey. The first generation of pull-tabs came completely off the can and were a health hazard as they would frequently drop inside the drink and were sometimes accidentally swallowed. Daniel Cudzik is credited for creating the stay-on-tab in 1975 that we continue to use today.

You will have to make sure you take the pull-tabs completely off to replicate the hacks featured over the next few pages. Also, wear gloves while working with the cans since they can be pretty sharp, and remember to smooth any rough edges. □

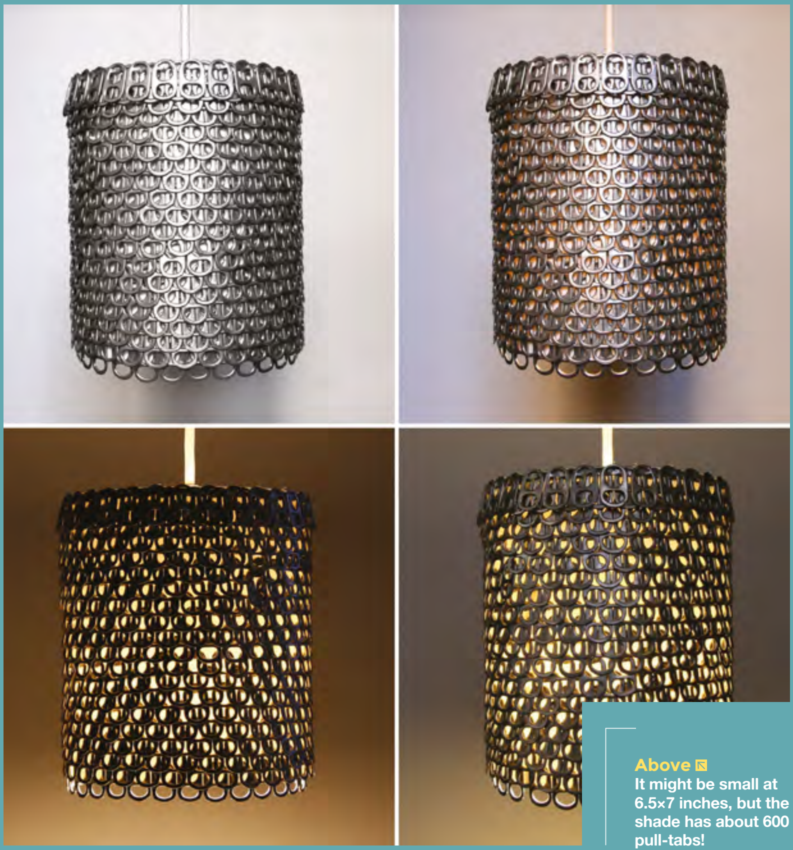
CAN TAB LAMP-SHADE

Project Maker
Allison Patrick

Project Link
hsmag.cc/CanTabShade

Although not a fan of canned drinks, a discussion with a friend on the materials she could use for future lamp-shades led Allison to soda can tabs. It took her four attempts to get the build right. She first tried to sew the tabs with a wire mesh, which turned out to be too rigid. Next she tried using a piece of leftover mesh laundry bag, which turned out to be better than the wire, but it didn't have any structure to it. Then she turned to using a window screening, which worked really well. "But about seven tabs in I realised, this is going to take forever! And I just did not have the patience," writes Allison. So instead she broke out her trusty glue gun and simply pasted them onto a plastic Löbbo lamp-shade. Her blog makes an interesting read for anyone looking to replicate the build as she's shared some valuable tips. Before starting, Allison suggests, take the time to draw guidelines on the shade. She was a third of the way when she realised that the sides weren't aligned. You should also insulate the surface on which you are assembling the shade as the pull-tabs can leave little scratch marks all over it. □

> " ABOUT SEVEN TABS IN I REALISED, THIS IS GOING TO TAKE FOREVER! "



Above
It might be small at 6.5×7 inches, but the shade has about 600 pull-tabs!

SODA CAN STOVE

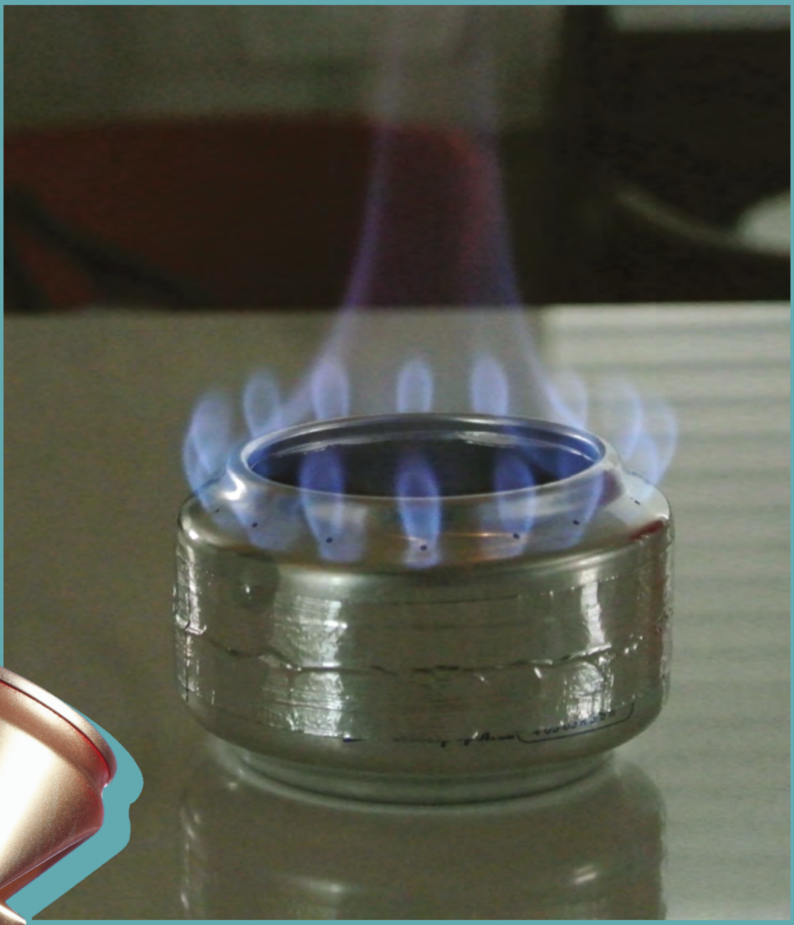
Project Maker
Andrew

Project Link
hsmag.cc/CanStove

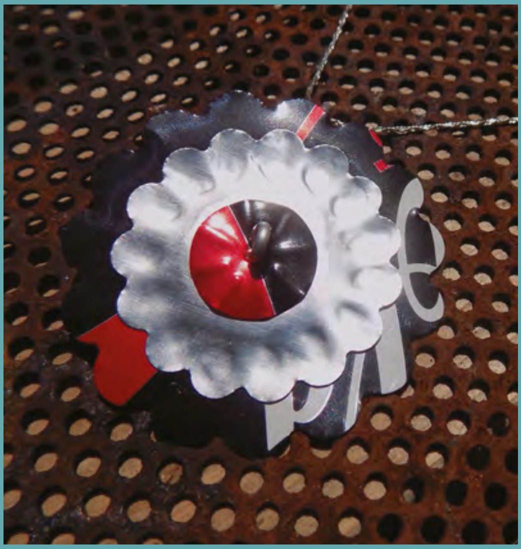
If you love camping, you'll love Andrew's nifty and portable soda can stove. This is actually Andrew's second build, which is a lot more efficient than the original design, as he demonstrates in the video. The build requires a couple of cans to create the three parts of the stove; a top ring, an inside wall, and a bottom. Andrew cuts the parts using simple cutters that he's made specially for this build. The cutters are just blocks of wood cut in a particular height with a utility blade screwed to the top. Andrew says you can cut the cans with a knife as well, but the cutters will help you save a considerable amount of time, especially if you want to make several stoves. Follow the video to see Andrew cut out the various parts from the two cans and then use a file to add creases to the inside wall of the stove. He then drilled 1/32" holes in the top ring of the stove. The holes define the size of the flame, so you can use larger drill bits for a bigger flame. He finally tapes the parts together using some metal tape to create a leak-proof seal. □

> " Cutters will help you save a considerable amount of time "

Below Andrew suggests you use 20ml of denatured alcohol or methyl hydrate to fuel the stove



SODA CAN PENDANT



Project Maker
Brooke Brei

Project Link
hsmag.cc/CanPendant

Brooke is an art teacher at Bridgeport High School who's turned some cans into fashionable accessories. "It only took me a half hour to create this, and you can easily put your own spin on it by using different cans or different shapes," writes Brooke. You'll surely need gloves for this one as you use scissors to cut the can and straighten it into a sheet. She then used flower-shaped punches to cut out the sheet. Her design calls for three punch-outs, but you can create your own design. She then used a leather modelling tool to shape the punched-out pieces, before punching a hole through all the pieces. Finally, she just passed a wire through the pieces and curved it into a loop using pliers. Now, just cut off the excess wire and make another small loop at the top of the pendant, through which you can pass any pendant chain, and show-off your newly minted accessory. □

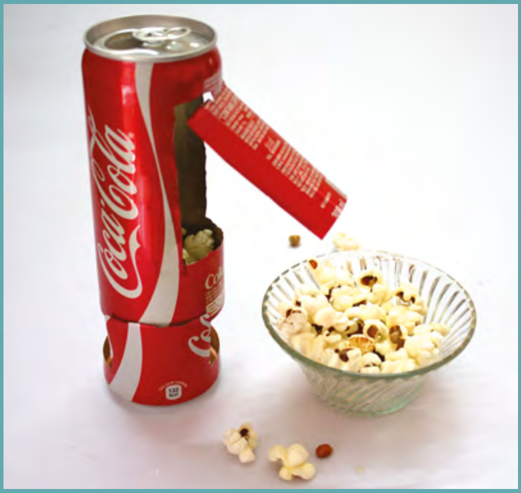
Left Brooke has an impressive portfolio with over 100 Instructables that cover everything from jewellery to electronics to food

CAN POPCORN MACHINE

Project Maker
Debasish Dutta

Project Link
hsmag.cc/PopcornMachine

Here's another project for campers. Debasish shows you how to take a couple of soda cans and refashion them into an emergency snack maker in little time with nothing but a Leatherman multitool or a Swiss Army knife. He first traced and cut one of the cans that'll direct the popcorn into a bowl. He then made a burner from the second one by first slicing it in half, and then marking it and cutting it in a simple but particular shape. The final step of the build calls for using a hammer to make sure the bottom of the stove is completely flat. That's all there is to it. To enjoy the snack, Debasish just lit and placed a tea light inside the flattened burner before covering it up with the other can. He then poured the corn kernels into the can, and before long they popped into the bowl he had placed beside the contraption. □



Right Speed up the popping process by adding more wicks to the tealight candle to make it produce more heat

MAKE BUILD HACK CREATE
132 PAGES OF MAKING
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
July 2019
Issue #20
BUILD A WORD CLOCK
A homemade chic timepiece
DIY VEGAN LEATHER
Animal-friendly
TOP MAKER TIPS
50 WAYS TO BUILD BETTER PROJECTS
Homemade CNC milling
Hydroponic gardening
The best OLEDs for makers
Laser-cut loom
TIN CANS TAPPING RADIO CONTROL
BUILD A ROBOT
TESTED MAKER GPS MODULES
MAKE THIS ROBOT FROM SCRATCH
LASER CUTTING RUST NUTS & BOLTS MAKEFEST

FORGE

HACK | MAKE | BUILD | CREATE

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

Electronics 101.12: DC motor drivers

Learn how to control the speed and direction of your DC motors

Dave Astels

daveastels.com

Dave's career started in the 8-bit days, with the Z80 and 6502, and he's been working with computers ever since. Check him out at: **daveastels.com** and **learn.adafruit.com**

In this series, we've discussed a variety of basic components, as well as more complex integrated circuits. We've explored some example circuits along the way. With this issue, we're going to switch to focusing on circuits. We'll look at something that's very common: DC motor control. We'll start simply with a switch, and progress to a robust, full-featured controller that can be connected to a microcontroller board. We'll use an Arduino Uno, since it is probably still the most common. However, any board with at least two digital outputs and one PWM output will work.

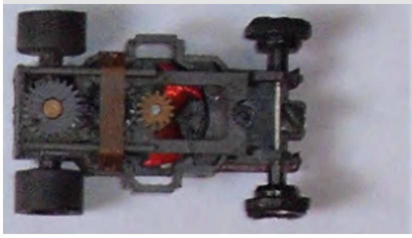
A DC motor is made up of a set of electromagnets mounted on a shaft that spins freely (the armature), surrounded by permanent magnets (the stator). **Figure 1** shows a basic motor. The coils of the electromagnets are connected to contact pads (orange in the diagram), known as the commutator. The + and - connections of the motor are in contact with these pads (yellow and cyan). This energises the electromagnets, and so their magnetic fields cause the motor shaft to spin, to better align their fields with those of the permanent magnets. At that point, the connections to the commutator change, causing the electromagnets' field polarity to change, causing attraction/repulsion with the opposite permanent magnets. That causes the shaft to spin another 180 degrees, at which point the connections change again. This happens continuously for as long as power is supplied. The result is a spinning shaft to which gears, pulleys, etc. can be attached to make use of the rotation. A motor has two connections. They're not polarised: their relative voltage determines the direction in which the motor spins.

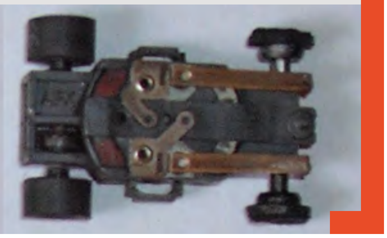
> "The contacts with the commutator can take different forms"

The contacts with the commutator can take different forms, but the classic technique is to use moulded pieces of carbon or graphite (the brushes) that rub against the commutator. Over time, the brushes wear to conform to the shape of the commutator. That wear continues over time, and the brushes will eventually need replacing. That's one of the disadvantages of brushes and having the electromagnets on the rotating shaft. Brushless motors swap the positions of the permanent and electromagnets, and use electronics to alternate the magnetic fields. These motors are more complex and expensive, but in some cases, it's worth it. In many applications, however, simple brushed motors are a better, or even preferred, solution.

SLOT CARS

The cars for slot car racing use simple DC motors. One brand that this author had as a child was AFX. Their cars were so small (the same HO scale that is very popular in model railroading) that the motor and gear-box were incorporated directly into the car's chassis. Their original design used a 'pancake' motor, where the shaft was vertical and connected to the rear axle by a series of round gears with a final crown gear on the axle. The commutator was a flat disc on the bottom of the motor, facing down, that was sliced into areas, like a pizza. Brushes were below that, with springs holding them against the commutator. The brushes connected to wipers that rubbed against the power rails in the track.





Simple switch control

Since the voltage difference between the motor's connections determines its rotation, the simplest

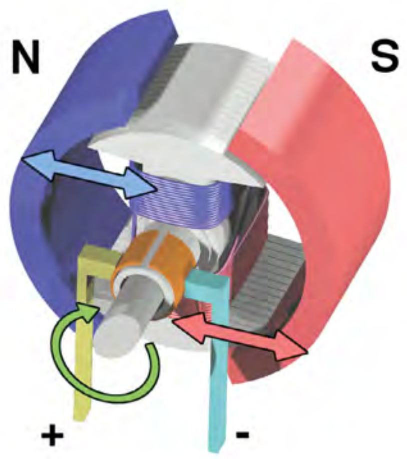


Figure 1
The construction of a basic DC motor

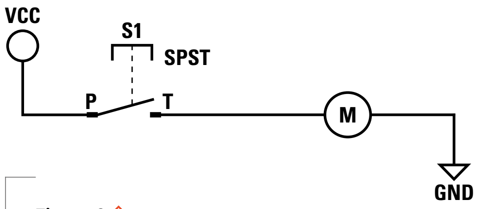


Figure 2
Simple switch control of a motor

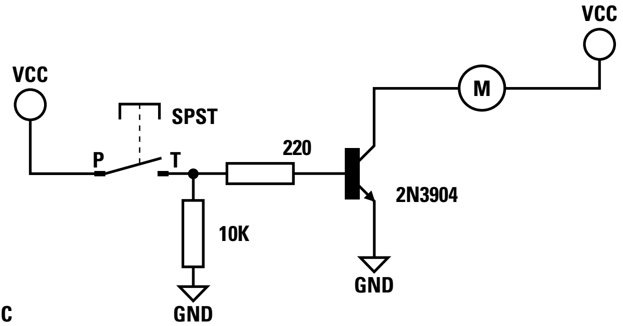


Figure 3
Transistor control of a motor

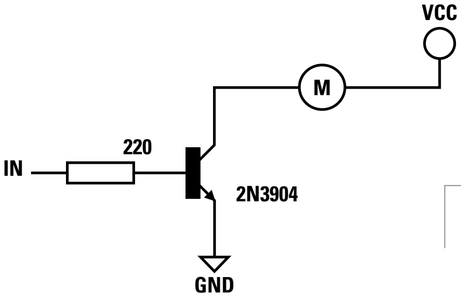


Figure 4
MCU control of the transistor

way to control a motor is to use a simple switch, as shown in **Figure 2**.

This has at least three shortcomings:

1. The motor is either on, or off: there's no way to control the speed.
2. There's a button that needs to be pressed to run the motor. It's completely manual. And that's no fun.
3. The motor only turns in one direction and that's of limited use.

Let's deal with these issues.

GOING DIGITAL

One step away from a manual switch is to use the switch to control a transistor, shown in **Figure 3**. The bias on the base of the transistor is to let the transistor switch between off and fully saturated. Take notice of the diode between the emitter and collector. As discussed in issue 12, a diode can be used with an inductive load (which a motor is) to protect the transistor from the voltage spike when power to the motor is removed.

This still isn't much different. But the next step is to get rid of the manual switch, replacing it with a microcontroller digital output pin. **Figure 4** shows the connection, and the code below repeatedly turns the motor on for a second, then off for a second.

```
void setup() {
  pinMode(8, OUTPUT);
}

void loop() {
  digitalWrite(8, HIGH);
  delay(1000);
  digitalWrite(8, LOW);
```

→

YOU'LL NEED

- Solderless breadboard and hook-up wires
- 5 V power supply for the breadboard
- 2 × 2N3904 PNP transistors
- 2 × 2N3906 PNP transistors
- 2 × TIP102 NPN Darlington transistors
- 2 × TIP107 PNP Darlington transistors
- 4 × 1N4936 or comparable power diodes
- LTV847 or comparable optocoupler
- 4 × 220 Ω resistors
- 4 × 1 kΩ resistors
- 2 × 5.6 kΩ resistors
- 4 × 10 kΩ resistors
- SPST push-button switch
- DPDT toggle/slide switch
- 5 V DC motor
- Arduino Uno, or other microcontroller board

```
  delay(1000);
}
```

This gives us computer control of the motor. But it's still only off and on, with no speed control. That's the next step.

CONTROLLING SPEED

Now that we are controlling the motor by using an MCU pin, we can look at varying its speed. One way to do that is to vary the voltage across the motor. That will vary the strength of the magnetic fields in the motor, which will vary the speed.

This works fine, but that has some disadvantages. Varying the saturation of the transistor that way means that it's always conducting some, so the power not used by the motor is turned into heat, which can cause the transistor to burn out. Also, while some microcontrollers have actual analogue outputs that we could use for this, not all do. Typically, when we use **`analogWrite`** in an Arduino sketch, an analogue value doesn't get output on the pin. Instead, a PWM (see box below) signal is generated.

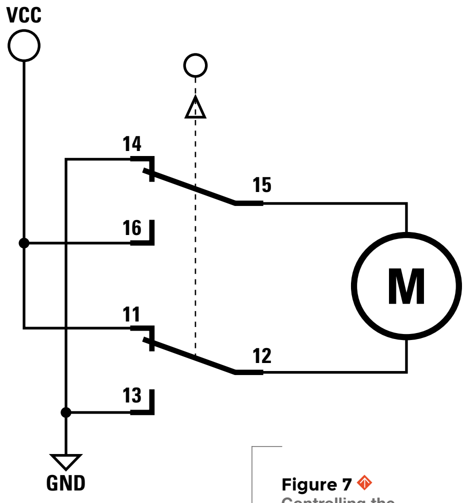


Figure 7 Controlling the direction of a motor

PULSE WIDTH MODULATION

A PWM (pulse width modulation) signal is a digital signal that switches between high and low very rapidly. The amount of time that it's high, compared to the entire cycle, is called the duty-cycle and can vary from 0% to 100%. The bigger the duty-cycle, the higher the average value. When you control an LED with a PWM signal, its brightness will reflect the duty-cycle, because it takes some time for the LED to adjust to changes in current and it also takes our eyes time to adjust to changes in the LED brightness. Similarly, when a motor is controlled with a PWM signal, the rotational inertia of the motor smooths out the changes from the signal. In both cases, when the frequency of the PWM signal is high enough, you don't notice it. If the frequency is too low, you will see jitter in the LED's output. If it's really low, you'll see the LED blinking.

The upshot of this is that we can drive the transistor with a PWM signal to control the speed of the motor. The circuit is the same, but we just need to use a pin that is PWM capable. Instead of pin 8, as we used before, we can use pin 5. The following code alternately runs the motor at half speed and full speed.

```
void setup() {
 pinMode(5, OUTPUT);
}

void loop() {
 analogWrite(5, 128);
  delay(1000);
  analogWrite(5, 255);
  delay(1000);
}
```

CONTROLLING DIRECTION

The direction that a motor spins in is determined by the polarity of the voltage applied to it (i.e. which connection is more positive). With our sample motor, we connect one pin to +5V, and the other to ground. Going back to our switch circuit, we can replace the push-button with a more elaborate switch. **Figure 7** shows using a DPDT switch. DPDT means double pole, double throw: it has two separate switches (double pole) that each connect a centre connection to one of two others (double throw). The motor connections connect to the centres of the two switches. The opposite outside connections are connected to +5V and ground. So, by flipping the switch one way, we connect one side of the motor to +5V and the other to ground, and so the motor spins. Flip it the other way and the motor voltage is reversed and the motor spins the other way. If we want to be able to stop the motor, we need to use either another switch to control the voltage going to our direction switch, or use a 'centre-off' DPDT switch, which can be put into a centre position where nothing is connected to the motor connections.

The problem with using a DPDT switch is the same problem as controlling a motor using a switch: it's manual. A common motor control circuit that provides direction control electrically is the H bridge, so named because it looks sort of like an 'H'. The control transistors form the legs, and the motor is

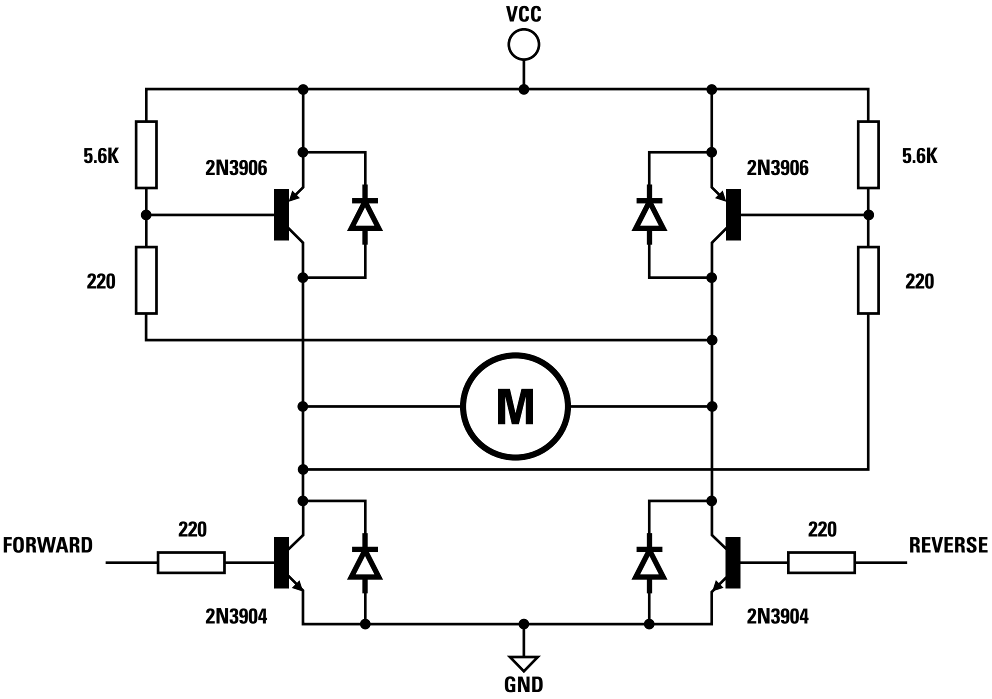


Figure 8
A simple H bridge circuit

the cross bar. **Figure 8** shows an example circuit. Notice that the transistors at the top (which connect the motor to +5V) are PNP, while the bottom ones (which connect the motor to ground) are NPN.

The idea behind the H bridge is that we switch on the top and bottom of opposite sides of the 'H'. This allows current to flow between ground and +5V, through the motor. The direction of current through the motor (i.e. which side is +5V and which is ground) depends on which pair of transistors are on. Since each pair is controlled by a separate signal, there is the possibility that both signals can be high simultaneously. That would mean that all four transistors would be on, and the +5V line would be connected to ground through them. That would not turn out well at all. Two or more transistors would burn out, possibly rather violently.

> **Two or more transistors would burn out,** possibly rather violently

If we use a PWM signal for the control input, we can control speed as well as direction. One input always has to be set to zero to avoid the above mentioned shorting.

There are some problems with the circuits presented thus far.

OPTOCOUPLERS

An optocoupler is a fancy term for an LED paired with a phototransistor in a light-tight enclosure. We are already familiar with LEDs from issue 9. A phototransistor is a special kind of transistor that can use photons (instead of electrons) to activate the base-emitter junction, turning the transistor on.

The first is power line noise. DC motors can be very noisy. A big part of that is the action of the brushes. When they make and break connections on the commutator, they can spark. That can introduce noise into the power line that can find its way to other parts of the circuit. Microcontrollers can be especially sensitive to this. We can use a separate supply for the motor, but as long as there is an electrical connection (even a ground) between the circuits, noise can get through.

We've used small transistors in the circuits. That limits the motors we can use to fairly small ones.

Finally, by using a common power supply, we're limiting the motor selection to those that use the same voltage as the controller. In the case of an Arduino Uno, that's 5V. 12V motors are quite common, and it would be nice to be able to use them in some situations.

This final circuit, which appears in a tutorial by Chuck McManis (**mcmanis.com**), is a classic solution and takes care of all three problems. →

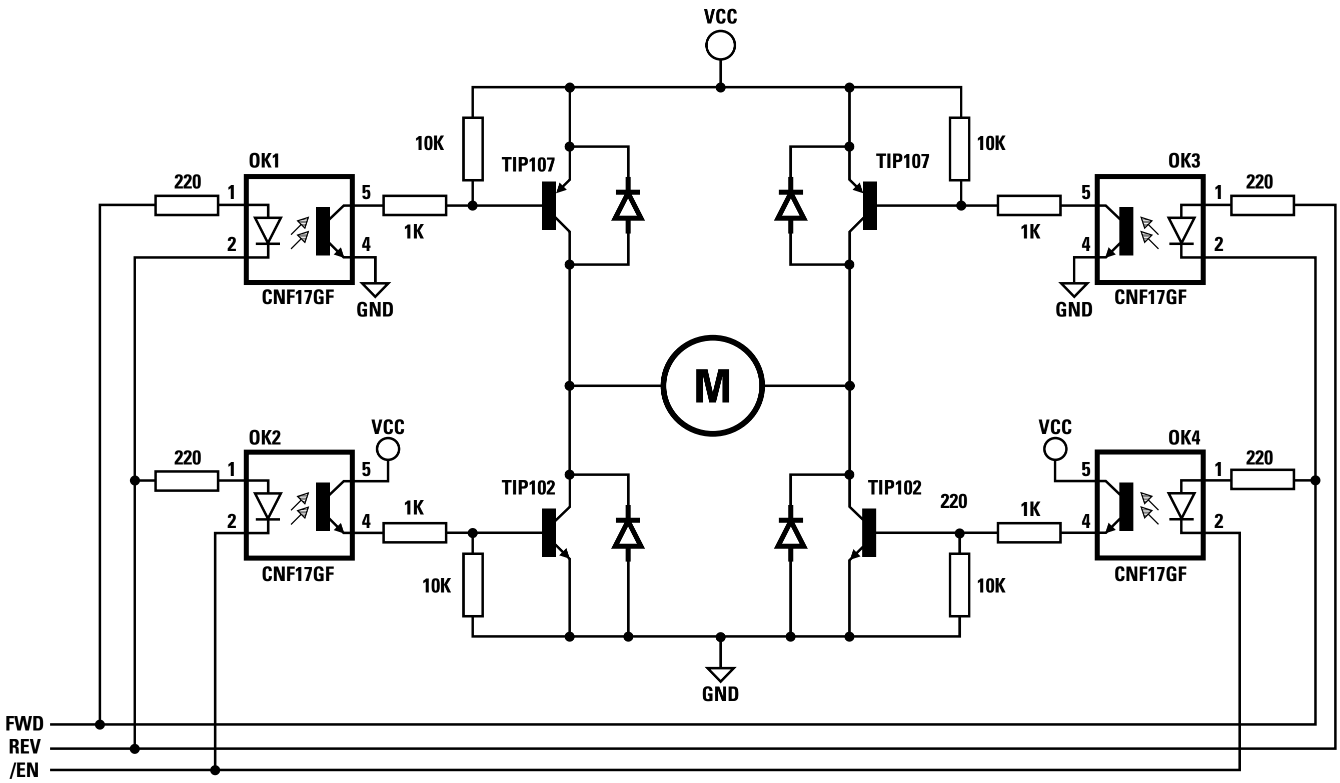


Figure 9 A more full-featured H bridge circuit

It uses transistors that can handle more power: TIP102 and TIP107 Darlingtons (see issue 13). It also uses optocouplers to connect the controller (the Arduino Uno in our case) to the transistors. This means that there is no electrical connection between the controller circuitry and the motor and its drivers; not even a ground. No noise can get back into the controller, and the voltage used for the motor circuit can be different than that of the control circuit. **Figure 9** shows the circuit.

There are three digital signals needed from the Arduino:

forward – high makes the motor spin in one direction
reverse – high makes the motor spin in the other direction
enable – high disables the motor, low lets it spin (i.e. it's an active-low enable)

TIP107 drivers

The optocouplers that drive the TIP107s (at the top) are wired between the forward and reverse lines. This avoids the case where both forward and reverse being high causes a short (which was a problem in the simple H bridge). When both are high, neither of the TIP107 optocoupler LEDs are active, since the anodes and cathodes of each LED are both high, so there isn't the voltage across them needed for them to light. When forward and reverse have different values (one high, the other low), one of the TIP107 driver LEDs is active. Note that only one will be active, since they are wired between forward and reverse in opposite directions.

TIP102 drivers

The TIP102 driver optocouplers have their anodes wired to either forward or reverse, and their cathodes wired to the active-low enable line. If both forward and reverse are high, both TIP102 optocouplers could be active, turning on both TIP102s. This isn't a problem, since the two TIP107 optocouplers will be disabled, as discussed above. Neither of the TIP102 optocouplers will be active unless the enable line is low.

> **In practical use,** you would likely use an integrated circuit version

Enable

If enable is high, nothing happens since both TIP102s are shut off because their optocouplers are inactive. If enable is low, one (assuming only one of forward or reverse are high) of the TIP102 optocouplers will be active, causing the associated TIP102 to switch on. This means that enable (as expected from the name) can allow voltage to be applied to the motor based on the states of the forward and reverse lines, or disconnect any voltage from it. We can use this by supplying a PWM signal to the enable line to control the speed of the motor.

Coasting

When both forward and reverse are low, all of the optocouplers are inactive, so all the transistors are off, and the motor is completely disconnected. This allows the motor to spin freely in either direction, i.e. it coasts.

Braking

When both forward and reverse are high, and enable is low, both of the TIP102 (lower) optocouplers are active, and both TIP102s are on. This connects both sides of the motor together and to ground. This, combined with the electromagnetic effects of spinning the motor (induced current), has the effect of resisting any spinning of the motor by external forces, effectively braking and actively trying to stop the motor. By using the PWM signal of the enable line, the strength of this braking effect can be controlled.

The code below exercises the H bridge. It sets the motor to spin in one direction (forward), gradually spinning it faster, to a maximum, then back to a stop. Then, it does the same thing in the opposite direction. Note that writing 0 to the speed-control pin is full speed, while 255 is full stop. This is because the enable signal is active low. So, the lower the duty-cycle, the higher the percentage of time that enable is low, and the more the motor is powered.

```
const int speed_pin = 5;
const int forward_pin = 7;
const int reverse_pin = 8;

void setup() {
  pinMode(speed_pin, OUTPUT);
  pinMode(forward_pin, OUTPUT);
  pinMode(reverse_pin, OUTPUT);

  analogWrite(speed_pin, 0);
  digitalWrite(forward_pin, LOW);
  digitalWrite(reverse_pin, LOW);
}

void loop() {
  digitalWrite(forward_pin, HIGH);
  for (int i = 250; i >= 0; i -= 10) {
    analogWrite(speed_pin, i);
    delay(100);
  }
  for (int i = 0; i <= 250; i += 10) {
    analogWrite(speed_pin, i);
    delay(100);
  }
  digitalWrite(forward_pin, LOW);

  digitalWrite(reverse_pin, HIGH);
  for (int i = 250; i >= 0; i -= 10) {
    analogWrite(speed_pin, i);
    delay(100);
  }
  for (int i = 0; i <= 250; i += 10) {
    analogWrite(speed_pin, i);
    delay(100);
  }
  digitalWrite(reverse_pin, LOW);
}
```

This article has gone over the basics of controlling DC motors. With the circuits presented, you could build motor drivers for a small robot or other motorised devices. In practical use, you would likely use an integrated circuit version of an H bridge, of which there are several, which work more or less the same as we've discussed. There are motor driver add-on boards for many common microcontroller boards. An integrated, all-in-one solution like this can get you up and running very quickly, but it's still useful to know how these driver circuits work.

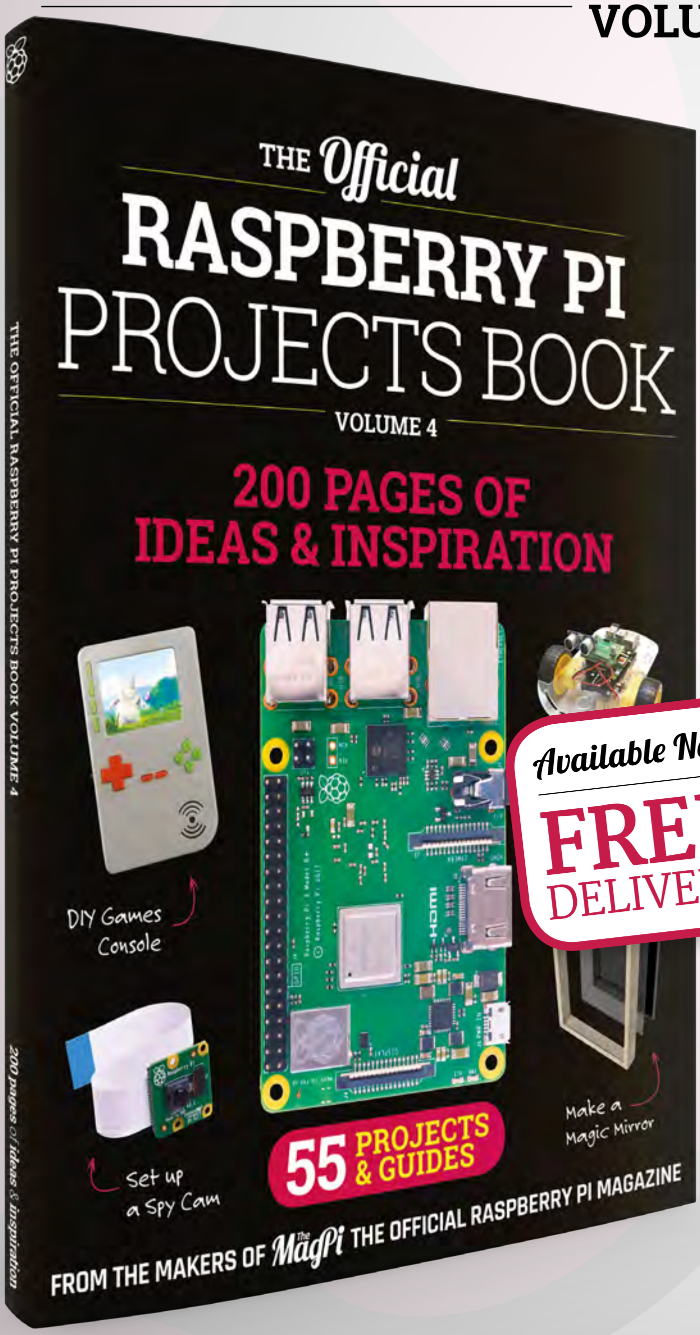
> **By using the PWM signal of the enable line,** the strength of this braking effect can be controlled

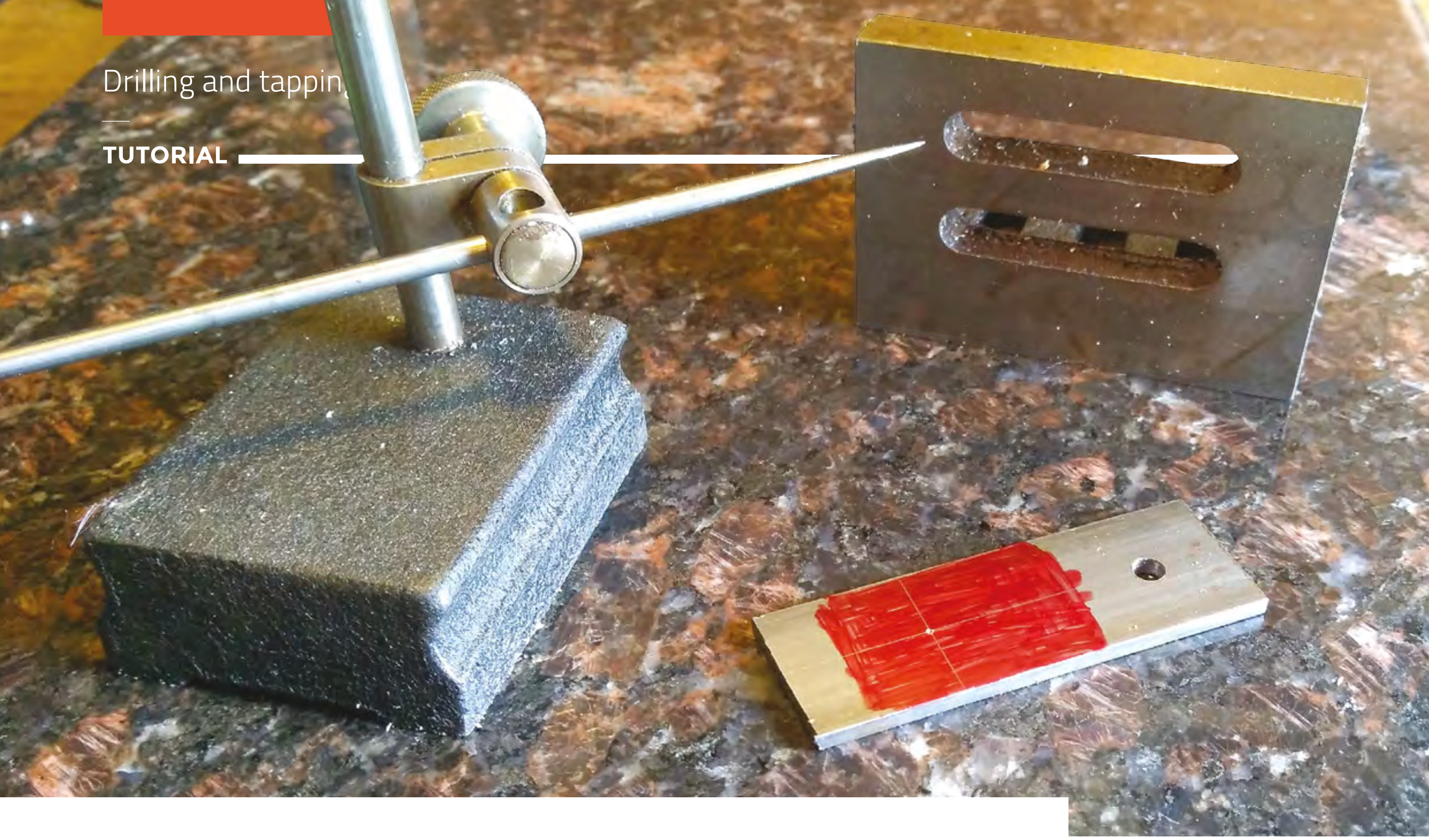
Having a fundamental understanding of basic electronic components and circuits will aid you in using more convenient, integrated solutions. They are, after all, built from the same basic components and circuit designs. Finally, knowing the basics not only helps with designing and building circuits, but also with trouble-shooting them. □

QUICK TIP

You can read more about motor control in Chuck McManis's tutorial at mcmanis.com

£3
ALL FORMATS
Wireframe
LIFTING THE LID ON VIDEO GAMES
BROKEN SWORD
Revolution's fight for survival
BLAST OFF
Create a jetpack in Unreal Engine 4
Issue 15 £3
wfmag.cc
TEARAWAYS
JOYFUL NOSTALGIA AND COMIC ADVENTURE IN KNIGHTS AND BIKES
Recompile
A game-changing 3D Metroidvania
Crime an
BILL OVERDUE
$3235
Going solo
Life as a lone game dev
Play detective
Inside the evolving deduction genre
mic puzzler,


THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 4
THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 4
200 PAGES OF
IDEAS & INSPIRATION
DIY Games Console
Set up a Spy Cam
Make a Magic Mirror
55 PROJECTS & GUIDES
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
200 pages of ideas & inspiration



Drilling and tapping

Let's learn how to accurately drill and create threads in metalwork

Figure 1 A workpiece made from EN1A mild steel, marked out and centre-punched ready for drilling

Jo Hinchliffe

@concreted0g

Jo Hinchliffe 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 may seem simple to drill a hole into material, but it's easy to go wrong! Having learnt, in issue 19, the skills to accurately mark out materials, it's a shame to lose accuracy with a drill skittering across the workpiece or drilling off-centre. Let's look at good practice with drilling – and also tapping, where we will create a threaded hole to receive a bolt.

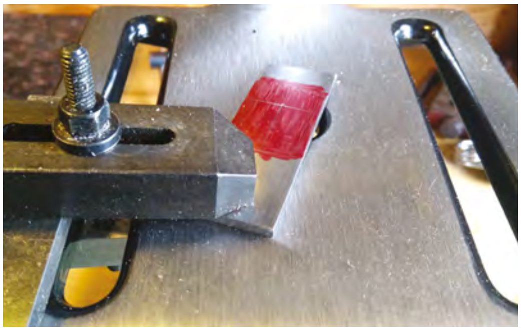
Drills and drill bits come in many forms, from hand-cranked to cordless, to drill press stands, and pillar drills. In the task of making a hole, we are usually aiming for the same criteria: we usually want the hole to be vertical and centred on the desired point. Keeping a drill vertical is challenging with a hand drill or a cordless drill and so, for repeatable accurate work, often people will invest in a pillar drill or a drill press stand, which receives a handheld electric drill and converts it into a vertical pillar drill.

Having marked out and centre-punched a piece of metal (**Figure 1**), let's transfer this to the pillar drill. It's important that the workpiece is held securely on the drilling platform so that when drilled, it cannot be thrown off, potentially damaging the work but, more importantly, damaging the machinist! Pillar drill tables have slots in them to allow clamping equipment to be attached, commonly using a tee nut so that the nut under the slot cannot rotate. In **Figure 2**, we can see that clamping accessories have been used to clamp down the work. If you are building up equipment, however, and don't have clamping accessories, a common G-clamp can be used to clamp the workpiece securely, as in **Figure 3**.

GET A GRIP!

Before we fully tighten the clamp, we need to centre the drill chuck over the coordinate we have marked and punched for drilling. We are going to use a

Figure 2 Using milling clamping accessories to clamp a workpiece



VIRTUES OF VICES

A drill vice is a great accessory for a pillar drill or drill press stand. They can be bolted to the pillar drill table, and will hold work firmly. They are particularly useful if you want to repeat a task on multiple objects of the same dimensions that can be repeatedly fitted to the vice without moving the vice relative to the chuck.

special drill bit called a centre drill (**Figure 3**) to help us line up, and to ensure we drill on the centre of the mark. Move the workpiece until the tip of the centre drill engages with the centre-punched mark. If you can, lock the pillar drill spindle with the centre drill in the punched hole, to hold the workpiece while you clamp it down. If you can't clamp the spindle, then take care not to move the workpiece while clamping, and double-check once clamped firmly.

STAY CENTRED!

Regular drill bits have an annoying property in that they are flexible and (especially with thinner diameter bits) it's easy for the drill bit to flex off-centre as it contacts the workpiece. It can then continue to drill off-centre. The centre drill allows us to start the hole accurately. Centre drills are made with only a very small cutting area at the end of the bit, with the rest of the hardened steel bit being solid and much more rigid than a regular drill bit. Centre drill geometry is such that they can be used to create a starter hole with a 60-degree angle which matches the cutting angle of a regular drill bit. This means that the regular drill bit will sit in the centre-drilled starter hole perfectly, and the angled walls of the hole will mean it is unlikely to flex off-centre. It's important, with a centre drill, that we use one that creates a 60-degree hole that is less than the diameter of our desired hole, or we use a wider one with care and only drill to a depth creating a narrower starter hole.

> "It's important that we create the **right-sized hole for the type of thread we desire**"

Having selected the correct speed, turn the drill on and lower the centre drill into the workpiece. Drill down until the 60-degree shoulders of the centre drill cut into the work.

Turn off the drill and clear the chips away. Without moving the workpiece, reset the drill and fit a 5 mm drill bit. Repeat the process and drill through the workpiece. Optionally, and to reduce wear on your drill bits, using some tapping/cutting fluid on the tip of the drill bit acts as a coolant for the process. When drilling through thick materials, it's a good idea to do it in a few passes, lifting the drill head and using a small brush to clear the debris from the drill bit tip.

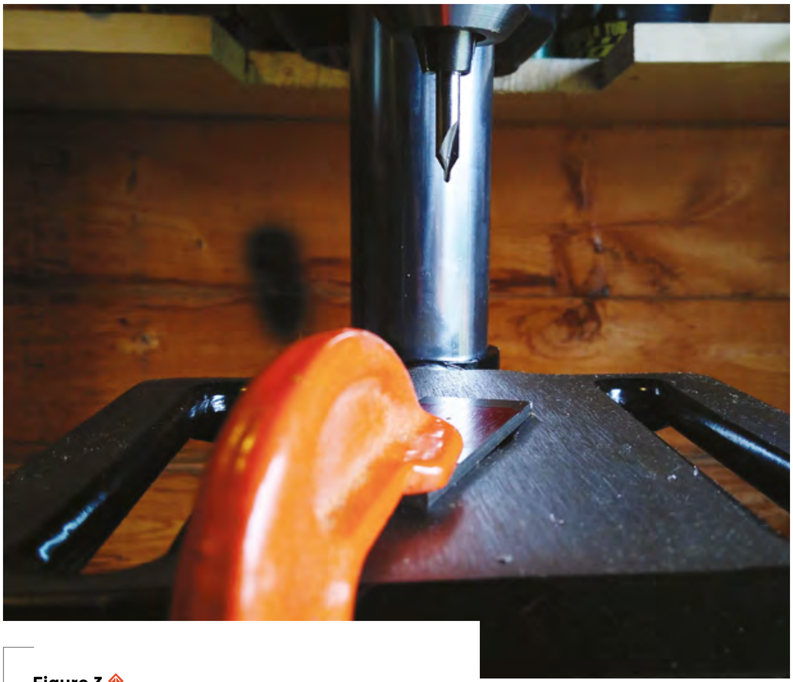


Figure 3
Using a G-clamp is a good budget way to clamp work securely. The centre drill is already mounted in the chuck to begin to drill accurately

So, now we have a nice clean 5 mm hole through our workpiece. Let's continue by tapping a thread through the hole. To form threads in holes, we use a tap (or a series of taps) which cuts a known size of thread. It's important that we create the right-sized hole for the type of thread we desire. Diameter charts are available online, or come in a booklet with a tap and die set. Conveniently the 5 mm hole we have drilled is the perfect size for an M6 thread, which has a pitch (distance between each thread) of 1 mm, a common metric thread size.

Whilst powered threading certainly is possible, it is beyond the scope of this article, and so we are going to use hand tapping techniques. It can be useful, however, to fit the tap into the pillar drill and hand-turn the tap into the hole whilst the chuck is still centred over the workpiece. It is very important that the next operation is with the drill UNPOWERED. To be extra safe, unplug the drill. Hand-turning the tap in a chuck may be a tricky procedure to get right, but is a good way to ensure that the tap starts the thread cutting perfectly perpendicular to the workpiece. Let's look, however, at the more standard way of tapping a thread using a tap wrench. →

YOU'LL NEED

- **A pillar drill or drill stand**
- **A G-clamp**
- **A 5 mm drill bit**
- **A centre drill**
- **An M6 tap**
- **A tap wrench**
- **A bench vice**
- **Some cutting/ tapping fluid** (or, at a push, 3-in-1-type oil will do)

QUICK TIP

Drilling produces hot shards of metal! PUT ON YOUR SAFETY GLASSES. Stow any loose sleeves, hair, jewellery that risks becoming entangled in moving chucks and, if your drill is fitted with guards, make sure they are in place and secured.

REFERENCE CHARTS

It's important we set the correct speed of rotation that will give us the best chances for a successful drill operation. Drilling speeds are calculated based on the diameter of the drill bit and the hardness of the material. We don't have to run the calculations, however: there are many drill speed information sites online, or books with drilling speed tables. The classic machinist's workshop guide is the pocket *Zeus* book, which contains drill speed information, as well as tapping hole size information, and a myriad of other useful data. For the drilling operations in this article, we used a speed of around 1300 rpm.

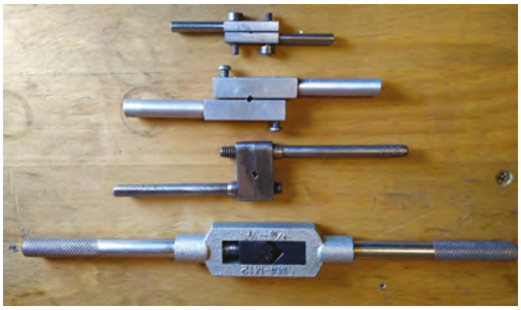


Figure 4
A collection of tap wrenches of varying sizes

QUICK TIP

Never leave a chuck key in a chuck! It's incredibly dangerous if a chuck key is flung out of a chuck accidentally when the machine is turned on.

GET WRENCHING!

Tap wrenches (**Figure 4**) are essentially a set of handles that clamp a tap and hold it firmly. (They are quite a nice little project to make in themselves.) They allow a lot of force to be exerted into the tap. Clamp the M6 tap into the tap handles, and the workpiece in a bench vice, and add a drop of cutting/tapping fluid (or oil) to the end of the tap (**Figure 5**). Introduce the tap to the hole, and begin to turn the tap clockwise whilst applying a downforce. Ensure you apply even pressure on both sides of the wrench, and keep the tap vertical with regard to the hole. Getting a hand tap started is a difficult task that takes some practice, so don't despair if it takes you some time and attempts. Once the tap is engaged, when cutting a thread it is very important that we always make a small ¼ turn anti-clockwise for every ¾ turn we make clockwise. This backward turn pushes the chip we are cutting inside the hole back into the channels you can see are ground into the tap. If we don't make those backward turns, eventually the

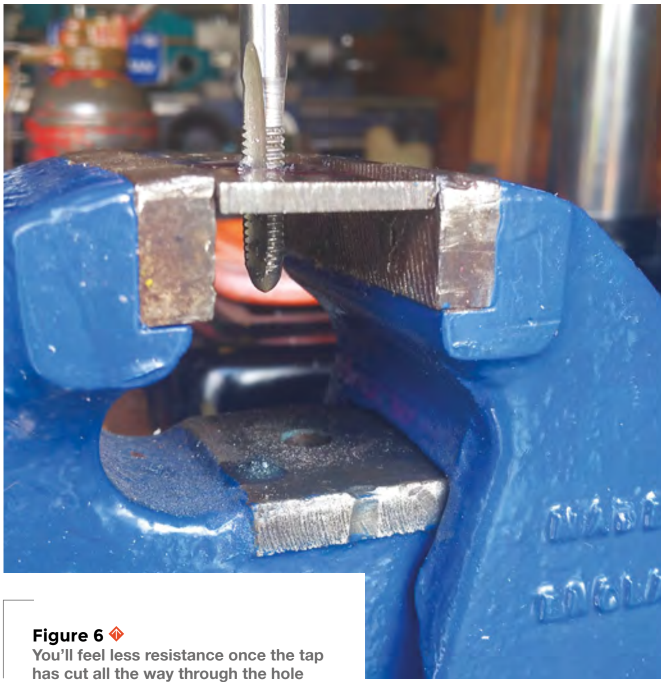


Figure 6
You'll feel less resistance once the tap has cut all the way through the hole

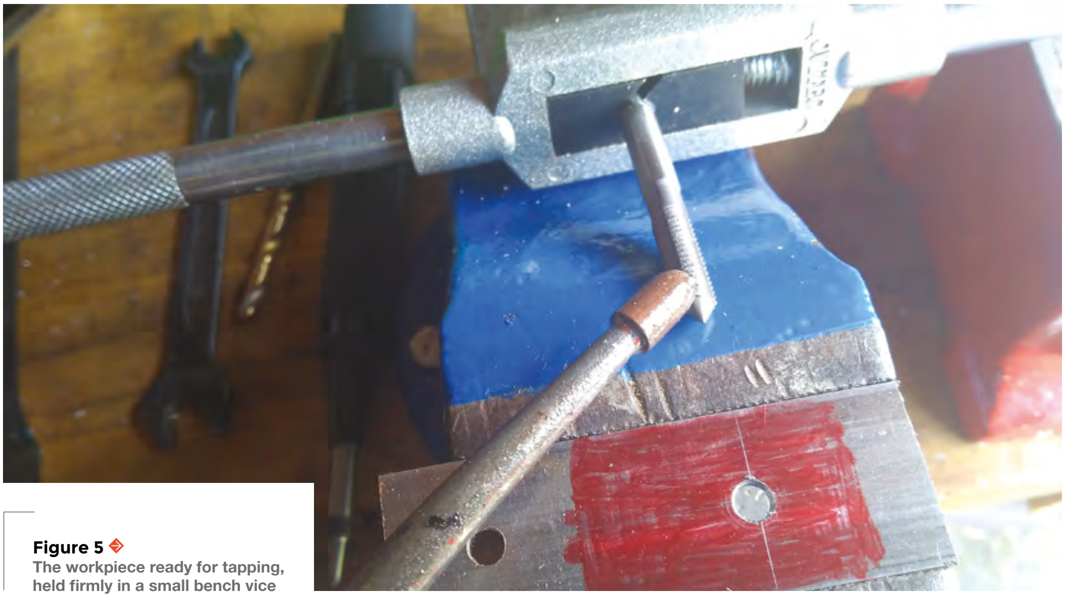


Figure 5
The workpiece ready for tapping, held firmly in a small bench vice

QUICK TIP

Get into the habit of safely clearing the metal chips away with a small brush, rather than by blowing it away or by using your hand. This saves on cut fingers and other injuries in the long run!

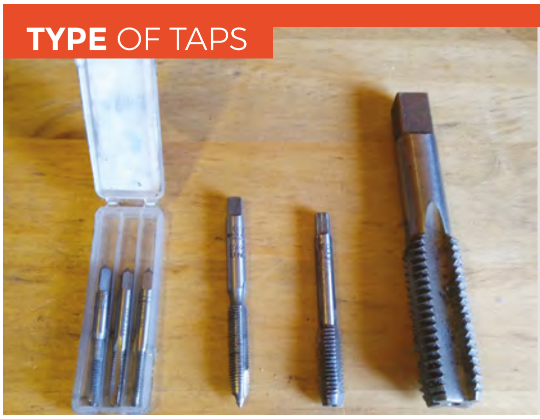
tap gets tight and, if we continue to turn, it is very easy to snap a tap and then subsequently it can be very difficult to remove a broken tap from a piece of work.

Having continued to cut the thread all the way through the hole, the tap will suddenly feel looser as it no longer is cutting through and is now just passing through the already cut threads. Continue to spin the tap down and, before removing, clean the underside of the tap (**Figure 6**) to make it clear of chips so it won't wind them back through the thread. Once removed, clean any chips from the threaded hole and try the thread with the correct-size bolt (**Figure 7**).

Next month, we will use the techniques we learnt in this article, and the marking out article in issue 19, to make a hand tool for our workshop. □

> "Introduce the tap to the hole, and begin to turn the tap clockwise **whilst applying a downforce**"

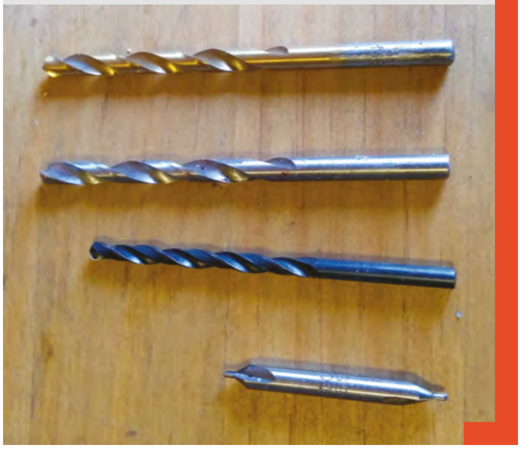
TYPE OF TAPS



There are a few different types of taps. For high-quality hand tapping, most people use serial taps (left-hand side of the picture) which often come in a pack of three labelled first, second, and third. Often delineated with small engraved rings around the shank, no ring is first tap, one ring second tap, and two rings third tap. Using the taps in order, series taps create a very accurately sized thread. However, taps are often sold (particularly in tap and die sets) that just have a single tap per size that cuts a complete thread in one go. Taps can be bought with different geometries for different types of hole. Taper taps (centre-left of picture) have a tapered end which introduces the thread-cutting surfaces gradually and makes it easier to start the process. They are made for through-hole operations where the tap will come out of the other side of the work. Sometimes we will need to tap a 'blind hole': a hole that doesn't go all the way through a material. For this, 'bottoming' taps (two taps on right-hand side of picture) are available that have less taper, and will thread further into a hole with a flat-bottom surface. It's also possible to buy taps with longer shanks for harder-to-reach holes. If possible, always buy a quality tap made from good, high-carbon steel.

BIT LENGTHS

'Jobber' drill bits are the most common type for metalwork, and are suitable for soft metals and softer mild steels and some other materials, such as some plastics. Jobber drill bits have a longer length of flutes (the spiral cutting edges) compared to 'mechanics length' drill bits, which have shorter flutes; this means they are much stiffer and stronger, so more appropriate for harder steels. All metalwork drills tend to be made from HSS (high-speed steel) but some other types are available. Some HSS drill bits may be coated with titanium (the golden-coloured one in the picture) which makes them more suitable for cutting harder materials such as cast iron. Others are made from a cobalt and steel alloy (silver-coloured one in the picture) which, again, makes them tough enough for harder materials.



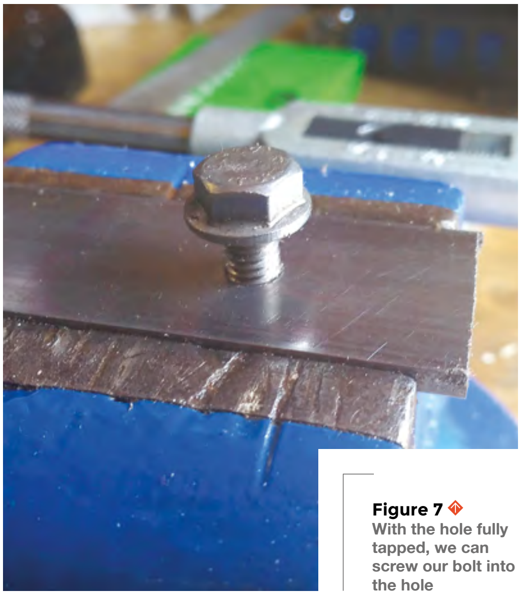


Figure 7
With the hole fully tapped, we can screw our bolt into the hole

QUICK TIP

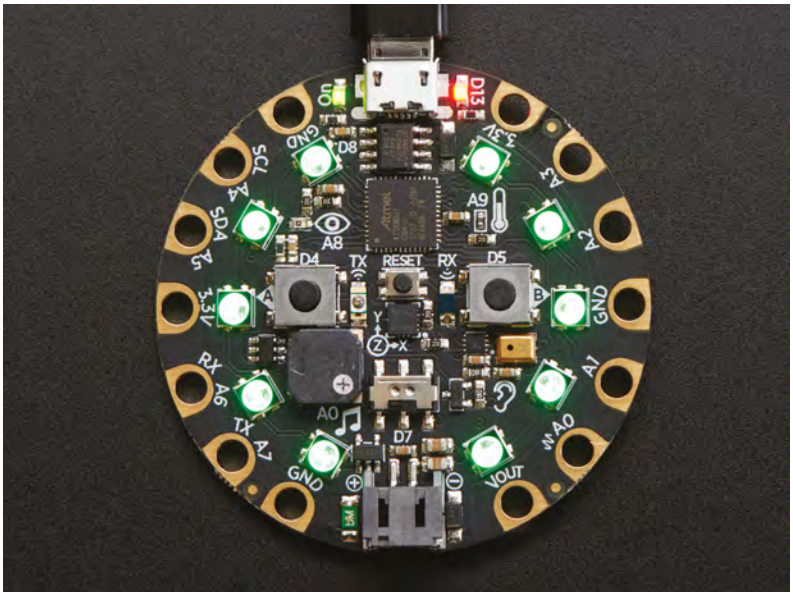
If you are wanting to drill a larger diameter hole, it's often better to work through a range of diameter drill bits, increasing the hole size to the desired amount. This reduces strain and wear on both the drill and the drill bits.

QUICK TIP

Budget tap and die sets are OK, and a good way to get started, but lots of people prefer to build up a tap collection by buying higher-quality taps in the sizes they need, as they need them.

Minimalist games console

Writing time-wasters on a Circuit Playground Express



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.

Modern games development seems obsessed with resolution, polygons, and the number of players on a server. The essence of games, though, has nothing to do with this. It's about creating a challenge for the player, and venturing into unusual form factors can make a simple game a bit more engrossing.

We're going to look at a really simple game inspired by Moon Buggy – an old terminal-based Unix game where you're driving a buggy across the moon, and have to jump over boulders that litter your path.

In our game, we'll use the ten pixels on a Circuit Playground Express to represent our moon, with red pixels indicating rocks, and white pixels representing our buggy as it drives around this tiny planet. Touching one of the pins will be the only input to the game – it will make the buggy jump for one step. If you hit this, you can clear a rock and keep going, otherwise, you crash out and have to start again.

Let's take a walk through the code. First, there are a bunch of modules we're going to need, and we need to set up the NeoPixels and the touch input.

```
import board
import random
import neopixel
from time import sleep
import touchio
pixels = neopixel.NeoPixel(board.NEOPIXEL,10)
touch = touchio.TouchIn(board.A1)
```

Then, we need to declare a few variables, the colours we'll use, and the initial position of the buggy and the rock. The speed is the amount of time (in seconds) that the game will pause each loop. We'll also hold a score and a variable that remembers whether or not we're currently jumping.

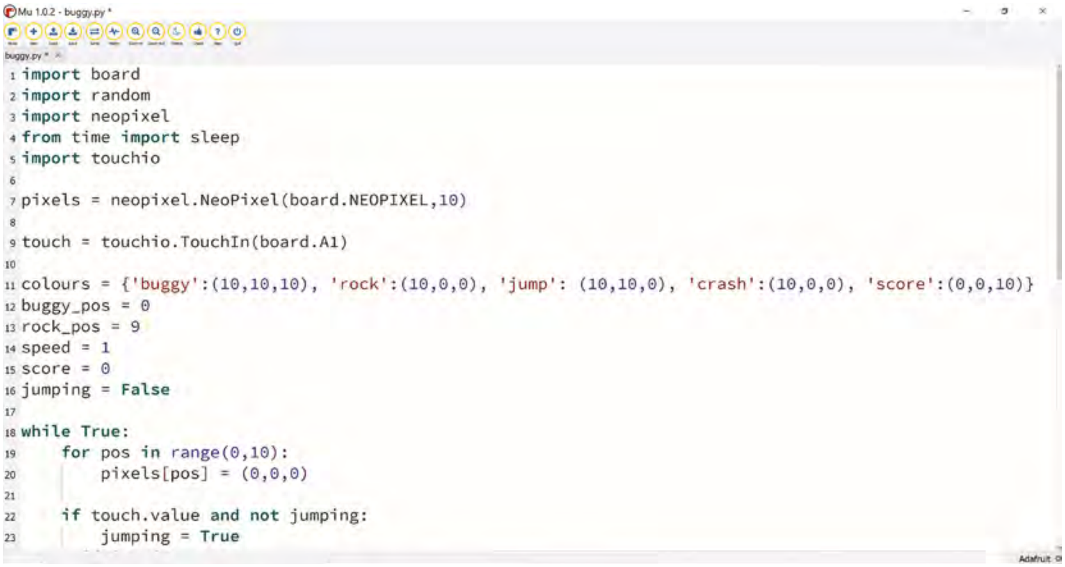
```
colours = {‘buggy’:(10,10,10), ‘rock’:(10,0,0),
‘jump’: (10,10,0), ‘crash’:(10,0,0),
‘score’:(0,0,10)}
buggy_pos = 0
rock_pos = 9
speed = 1
score = 0
jumping = False
```

We now reach the main game loop. The first part blanks all the pixels. Then, it works out whether or not we're currently jumping – we are if we're touching A1 and we're not already jumping (as jumps last for one step only).

```
while True:
    for pos in range(0,10):
        pixels[pos] = (0,0,0)
    if touch.value and not jumping:
        jumping = True
    elif jumping:
        jumping = False
```

Now we know whether we're jumping, we need to draw our buggy in the right colour:

```
    if jumping:
        pixels[buggy_pos] = colours[‘jump’]
```



```
import board
import random
import neopixel
from time import sleep
import touchio

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

touch = touchio.TouchIn(board.A1)

colours = {'buggy':(10,10,10), 'rock':(10,0,0), 'jump': (10,10,0), 'crash':(10,0,0), 'score':(0,0,10)}
buggy_pos = 0
rock_pos = 9
speed = 1
score = 0
jumping = False

while True:
    for pos in range(0,10):
        pixels[pos] = (0,0,0)

    if touch.value and not jumping:
        jumping = True
```

Left
It takes less than 50 lines of Python to get this game running

> **We turn all the pixels red for five seconds**, then light up pixels denoting the score

```
    else:
        pixels[buggy_pos] = colours[‘buggy’]
```

The main game mechanics are all in one **if** statement that triggers if the buggy and the rock are in the same position. If they are, and we're not jumping, we turn all the pixels red for five seconds, then light up pixels denoting the score, reset the variables, and start again. If we are jumping, we increase the score and create a new rock in a random position.

```
    if buggy_pos == rock_pos:
        if not jumping:
            pixels.fill(colours[‘crash’])
            sleep(5)
            speed = 1
            for num in range(0,10):
                if score >= num:
                    pixels[num] =
colours[‘score’]
                else:
                    pixels[num] = (0,0,0)
            sleep(5)
            score = 0
        else:
            rock_pos = (rock_pos + 4 + random.
randint(0,5)) % 10
            score = score + 1
```

The final bit of code lights up the pixel where the rock is, speeds up the buggy (reducing the speed variable actually speeds up the buggy, since this is the amount of time the script stops for), then does the actual sleep.

```
    pixels[rock_pos] = colours[‘rock’]
    speed = speed*0.95
    sleep(speed)
    buggy_pos = (buggy_pos + 1) % 10
```

The last line moves the buggy along one. The **% 10** section means divide by ten and take the remainder – this ensures that we keep cycling back to 0 (and so restarting our loop), rather than launching off into the distance (or at least off the end of the NeoPixel chain).

That's it. Upload this code to your Circuit Playground Express, and you'll be able to play the game. □

GOING FURTHER

You could adapt this game in many ways. The in-built speaker on the Circuit Playground Express (CPX) could be used to add sound effects, or you could add even more graphical effects.

Rather than adding to the audio-visual splendour, you could also change the way you interact with the game. There are a few sensors on the CPX that could be used, but perhaps the most exciting alternative is the accelerometer. We looked at this last month, but as well as detecting orientation, it can detect whether the CPX is being held, or whether it's in the air (or jumping). You could adapt this code so you have to 'throw' the CPX over the rocks.

Desktop hydroponic gardening

Grow your own food with an Arduino and some rain guttering

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.

In this project, you'll be making a scalable hydroponic growing system that uses easily sourced components to control the flow of water, light, and heat to your plants.

Hydroponic systems use regular or constant flows of nutrient-enriched water to grow plants without soil, and are a great way to grow vegetables if you have restricted space or access to natural light.

This project represents a couple of years' worth of experimentation with homemade growing systems, and is a variation of a hydroponic technique called ebb-and-flow – where the water is tidal, and floods through the system several times a day. Plants are rooted into an absorbent substrate that holds water next to the roots of the plant when the flow of water stops.

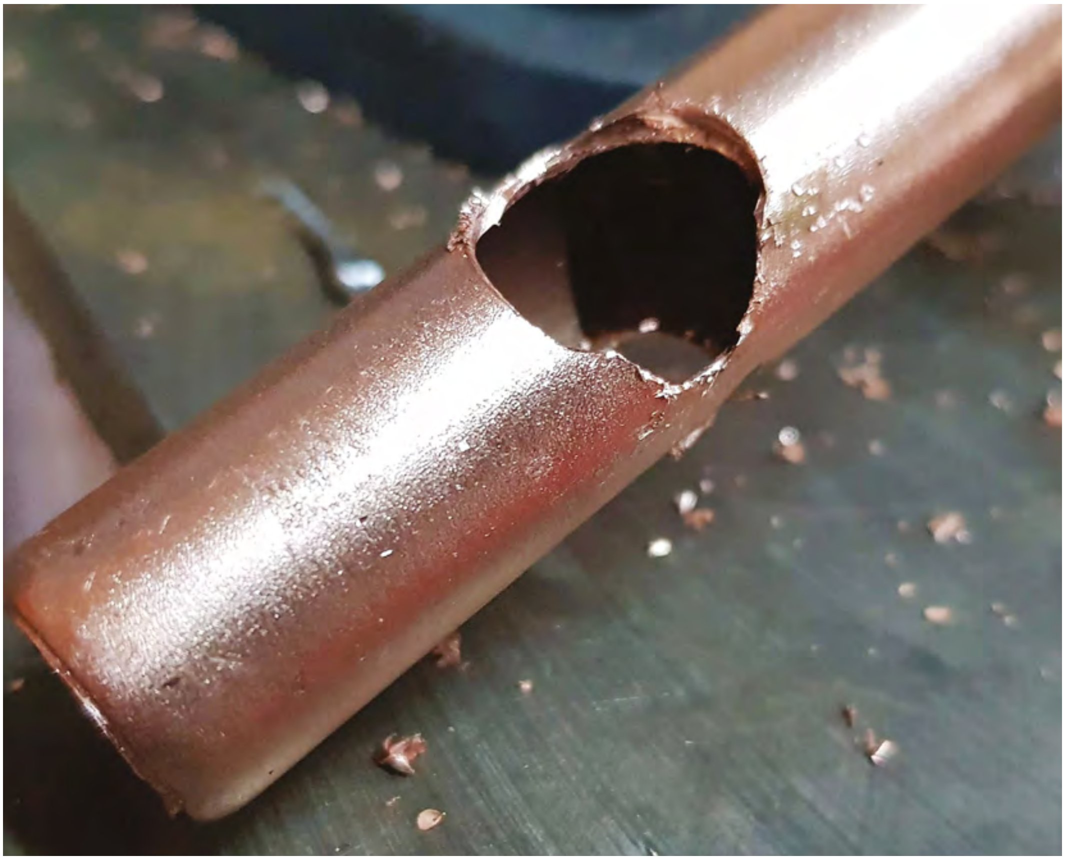
If the pump in a constant flow system fails, the plants will die very quickly, while an ebb-and-flow system can survive for several hours in the event of a power failure.

GROW VERTICALLY TO SAVE SPACE

The hydroponic system described here has three main parts: the support system, the water system, and the control system. The support system is essentially a wooden box and a shelf unit, and is the easiest part to make.

Lay the smallest piece of plywood flat, and use some coins or hex nuts to raise it slightly from the surface it is resting on. Next, take the four planks, and arrange them to make a rectangular box around the smaller piece of plywood. Screw the planks together using corner brackets at the top and bottom of each corner, and then secure the plywood base to the planks using screws and glue. You should now have a simple box that you can use for the base of your hydroponic system.

You will use copper pipe to make shelf brackets. Measure 450 mm from the end of the 28 mm pipes, and drill a 15 mm hole right through. Drill a second



hole through the pipes, 750 mm from the end. The 15 mm pipe should slide through the hole in the 28 mm pipe and make a rudimentary shelf bracket. To figure out the length of the shelf brackets, measure the width of your guttering and add about 40 mm. Cut four pieces of 15 mm pipe to this length, and add a copper elbow to the end of each pipe. If you have a blow-lamp and solder, you can use this to join the pipes and elbows; otherwise, you can just use hot glue.

> "Hydroponics are a great way to grow vegetables **if you have restricted space**"

Fix the 28 mm pipes 150 mm from the sides at the back of the box, using the 28 mm pipe clamps to hold each pipe in place vertically. You can now slide the 15 mm brackets into place, and fix them in position using solder or glue.

Next, you are going to extend the brackets to support LED lights above the guttering. Cut four pieces of copper pipe 250 mm long, and fit these vertically into the elbows on the shelf brackets. Cut another four pieces of copper pipe slightly less than half of the width of your guttering, and connect them to the 250 mm verticals using elbows, so that they hang over the guttering. Cut two final pieces of copper pipe to join the left and right brackets together over the guttering.

The final piece of pipework will carry water from the pump to the top watering channel. The pipe is fitted vertically in the middle of the back of the box using 15 mm pipe clamps, and is approximately 800 mm long with a U-shaped section at the top to direct water into the guttering. It is much easier to use the copper pipe as a conduit for a length of narrow-bore, silicone pipe push-fitted to the pump than it is to connect to the copper pipe directly. Feed the silicone pipe through the 15 mm copper pipe and around the U-shape, leaving about 30 cm hanging out in the bottom of the box.

With the last of pipework done, you can complete the woodwork. Drill, or cut, a 10 mm hole through the back of the box at the right-hand side to accommodate your power cable, then drill a larger hole on both sides of the back to allow ventilation. To finish the box, cut slots in the larger piece of plywood so that it will fit onto the box as a lid without hitting the pipes. Mark the centre of the →

YOU'LL NEED

Electronics

- **12 V 20 A power supply**
- **Arduino Uno**
- **60 W waterproof greenhouse tube heater**
- **20×4 I²C LCD screen**
- **4 × Momentary push buttons** (normally open)
- **2 × TIP120** (or similar) transistors
- **1 × 10 A solid state relay**
- **2 × 2.2 kΩ resistor**
- **10 kΩ resistor**
- **10 kΩ thermistor**
- **5 m LED grow light strip** (only 2 m needed)
- **12 V water pump**
- **3m length of 240 3-core flex**
- **13 A plug**
- **DuPont signal cables and 3 A rated cable**

Other hardware

- **2 m length of flexible silicone tubing, approx. 6 mm bore**
- **2 × 1 m lengths of 28 mm copper pipe**
- **2 × 3 m lengths of 15 mm copper pipe**
- **20 × 15 mm equal copper elbows**
- **4 × 28 mm pipe clamps**
- **2 × 15 mm pipe clamps**

YOU'LL NEED CONT...

- 2 × 900 mm lengths of square guttering
- 4 × square guttering outside end caps
- 2 × Stewart 52 cm premium gravel trays
- 60 cm square piece of polythene
- 2 × planks, approx. 900 mm x 150 mm x 25 mm
- 2 × planks, approx. 500 mm × 150 mm × 25 mm
- 8 × metal corner braces
- 12 mm plywood sheet 950 mm × 550 mm
- 12 mm plywood sheet 850 mm × 500 mm
- 1 × approximately A4-sized sheet of 3 mm plastic
- 2 × pieces of aluminium to act as heat sink for transistors
- M3 brass PCB supports

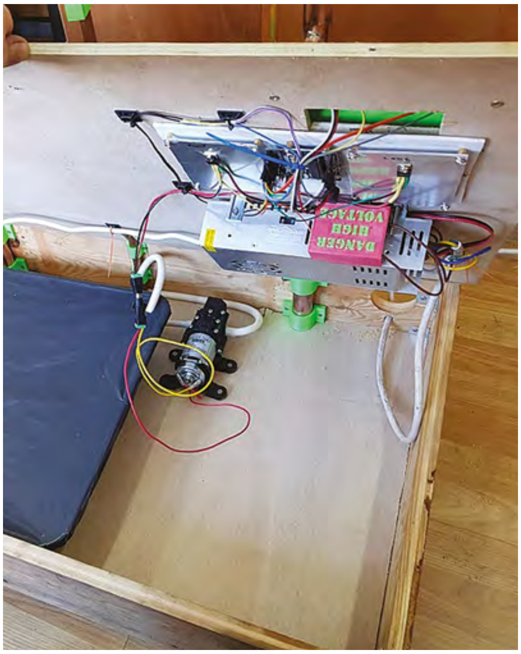
longest side of the plywood, and line this mark up with the centre of the box (where the water pipe goes up). Now use a square to mark the position of the pipes on the larger piece of plywood, and notch out the pieces using a jigsaw or fretsaw.

TIME TO GET YOUR FEET WET!

Water is pumped to the top of the system from the water tank, then drains down through a series of water channels under the force of gravity until it goes back down into the water tank. The tank is very simple to make from a 52 cm Stewart gravel tray. Apply double-sided tape or glue around the top of the gravel tray, and simply stick the polythene sheet to the top, so that the tray is completely covered. The tray is now a closed water tank.

Place the second Stewart gravel tray on the lid of the box, between the two 28 mm pipes. This tray will support larger potted plants, and will always have a few centimetres of water in it. Drill a 25 mm diameter hole through the slightly raised section at the bottom left-hand side of the tray, and continue the hole through the plywood lid of the box. This hole is where a water fitting will connect to the tank at the bottom, so there needs to be a hole in the water tank here.

> "This tray will support larger potted plants, and **will always have a few centimetres of water in it**"



Right The inside of the base, showing the electronics mounted to the underside of the lid, and the water tank on the bottom-left of the box with the pump connected to it

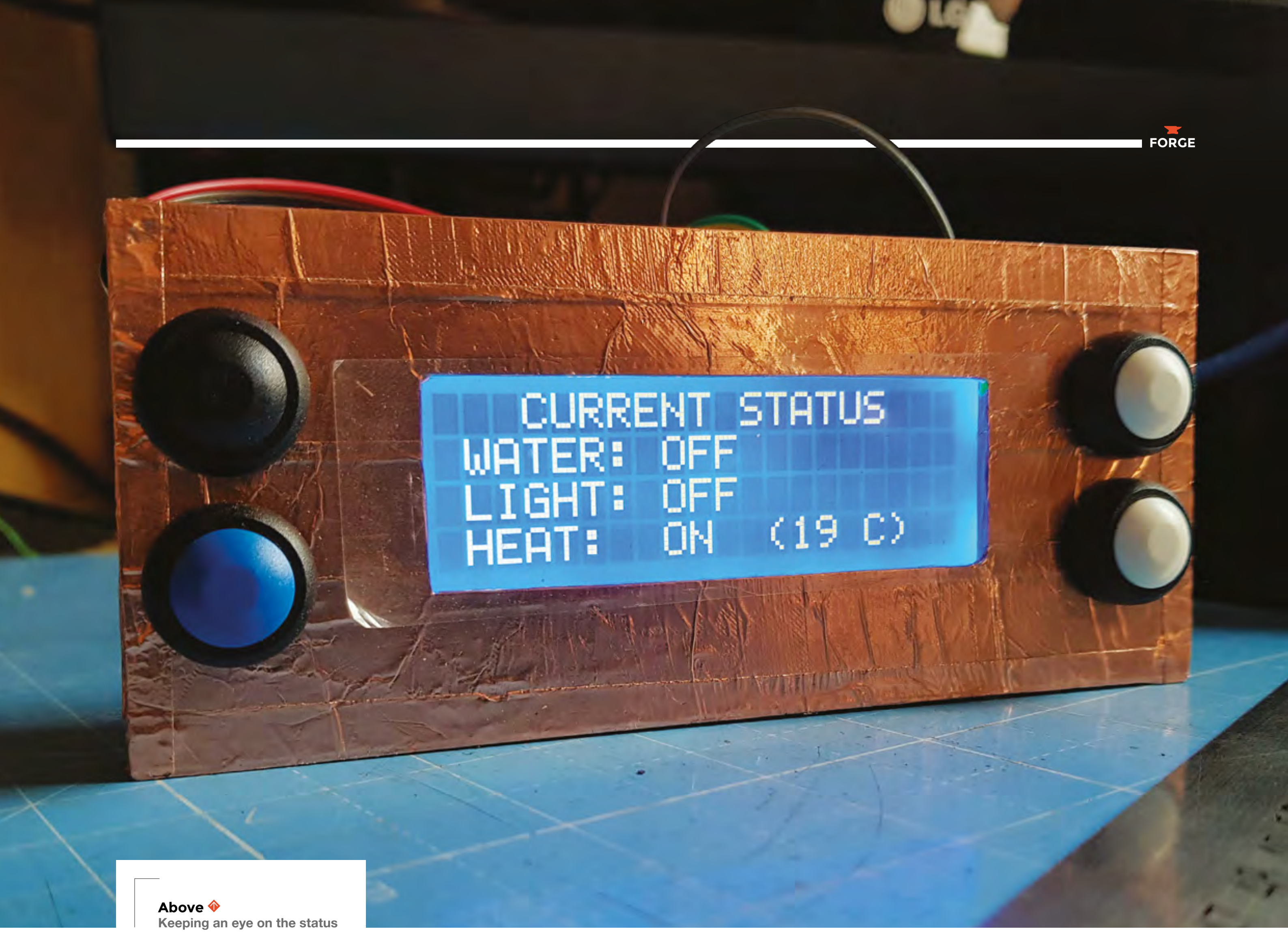
Remove the tray and plywood lid from the box. Position the water tank at the left-hand side, and slit the polythene near the back of the tank. This slit is where the pump will draw water from. Replace the lid, and cut into the plastic through the hole you just made in the plywood. Glue the water tank into place with hot glue, and reinforce the polythene around the hole with a 15 mm washer or gaffer tape.

Screw the water pump into place near to the tank and connect the outlet to the silicone pipe, then use another length of silicone pipe to connect the pump's inlet to the water tank through the slit in the polythene. To make sure that the inlet pipe rests on the bottom of the tank, weigh it down or use rigid copper pipe to hold it in place. Use tape to seal the slit in the polythene.

Two pieces of guttering (with the end-caps fitted) make your high-level growing channels. Controlling the flow of water from one channel to another is critical for the proper operation of the hydroponics rig, and this project uses a special 3D-printed variation of a greedy cup (or Pythagoras cup) to do this. To make the greedy cups, you will need three small jam jars, three of the 3D-printed cup pieces, and three lengths of 15 mm copper pipe long enough to reach from slightly below the top of one channel to slightly above the top of the next channel.

Drill a 25 mm hole in each of the water channels, staggering the holes between the left and right side. Fit the 3D-printed cup pieces in the holes, and screw them on tightly with a little bit of waterproof sealant around the thread. Fit the third greedy cup into the hole in the Stewart tray, and use a short piece

Above Fitting the greedy cup into the hole in the water channel, with a little silicon rubber to ensure a waterproof seal



Above Keeping an eye on the status

Above The system under test, showing water flow from the outlet, and the greedy cup mechanism on the left of the photo

of 15 mm pipe to connect the tray and the water tank below.

All that remains is to develop a control system for the Arduino, and wire up the electronics. We've already written some commented code and a wiring diagram to make this step less complicated. The control system uses a 20×4 LCD and four buttons to navigate between pages, tab between items, and alter values up and down. Print out the panel for the LCD and buttons and the housing for mounting the panel onto the lid of the box, and then wire the buttons as shown in the diagram **(Figure 1**, overleaf).

Position the LCD housing on the lid of the box near to the front. Drill a hole through the plywood for the LCD wires, and screw or glue the mounting into place. Mount the Arduino and heat sinks to a plastic sheet, and screw the sheet to the underside of the lid near to the hole for the LCD wires. Mount the TIP120 transistors to the heat sinks. Add the 12 V power supply to the underside of the lid near to the cable hole you drilled at the back. Make sure that all the electronics, but particularly the high voltage side is mounted so that it's protected against water. You also need to ensure that the high voltage cable is not at risk of coming loose, and make sure that anything

> "All that remains is to develop a control system for the Arduino, **and wire up the electronics**"

QUICK TIP

Don't put both TIP120 transistors on the same heat sink. The mounting tab is connected to the transistor base.

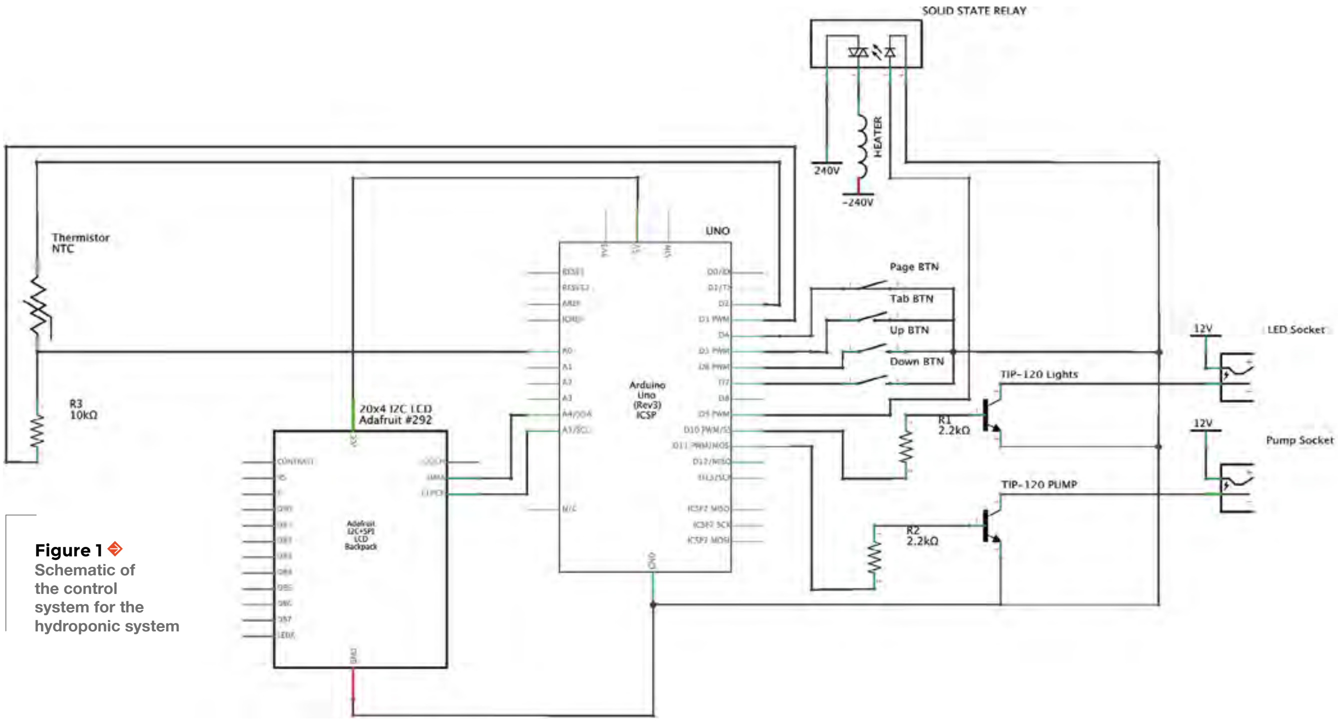
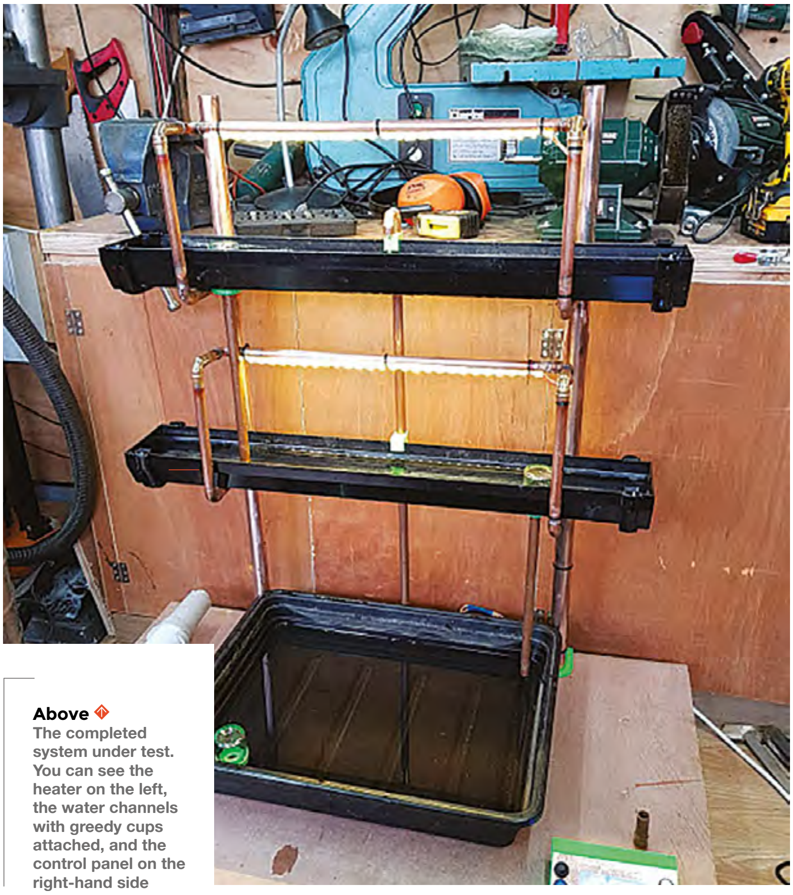


Figure 1 Schematic of the control system for the hydroponic system



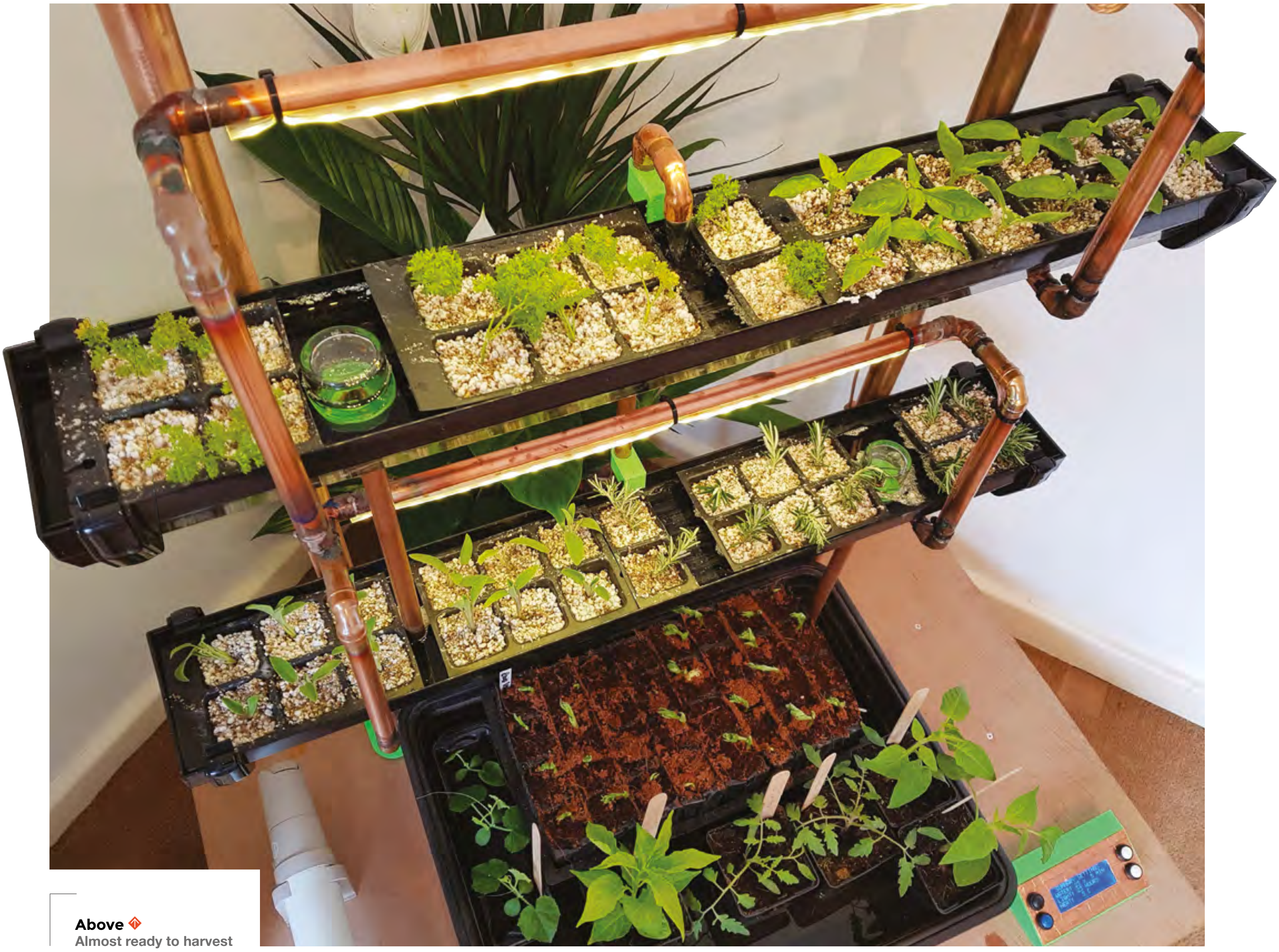
Above The completed system under test. You can see the heater on the left, the water channels with greedy cups attached, and the control panel on the right-hand side

that needs grounding is connected to ground. Working with mains voltage safely takes experience and if you're not experienced enough to work safely with mains voltage, seek advice from someone who is before proceeding with this build. Take your safety seriously as you don't always get a second chance! You can mount the PSU using four metal brackets.

Add the LED lights to the tubes above the water channels, and run the wires down to the box using cable ties to hold them in place. Position the thermistor about halfway up the right-hand pipe using cable ties, and connect the wires to the Arduino as shown in the diagram (**Figure 1**).

Add the heater to the left-hand side of the box, and drill a hole to pass the cable through to the underside of the lid. Run the cable along the underside of the lid to the PSU. Mount the solid-state relay to the side of the PSU by drilling into the case and using machine screws or bolts. Wire the electronics as shown in the circuit diagram, and ensure that any live contacts are well protected with insulating material. Cables should be routed using cable ties and clips. Flash the Arduino with the hydroponics sketch (from **hsmag.cc/issue20**), and test out the interface using just the USB power to make sure everything is working.

If the Arduino seems to be working, test the water channels by pouring water into the top channel and tracing its path back to the water tank. Watch for any leaks or blockages. If the water channels seem



Above Almost ready to harvest

Above Germinating seeds ready for growing

SAFETY

This project combines electricity and water, which can be a tricky, and dangerous, combination. You need be knowledgeable enough to work with this combination before embarking on this build. You can reduce the safety issues by using an external power supply. We'd also recommend using a residual current device (RCD) for further protection.

OK, pour in a whole bucket of water to the Stewart tray. The water will drain into the tank, and you can reconnect the power once you're sure there are no leaks to worry about. Now you can test the pump by setting the water flows, and adjust your light settings. It's recommended to start the pump at a low setting (maybe 25% power). If you're planning on using the heater, you'll need to put the hydroponic unit inside a polythene tent to contain the heat. The tent can be made with a few garden canes or pieces of PVC pipe, and held together with bulldog clips. □

QUICK TIP

The thermistor only gets power just before a reading is taken, because supplying it with constant power can make it heat up over time.

Getting the most from a component datasheet

Turn forgotten tools into workshop favourites

Gareth Branwyn

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.

An electronic component datasheet is sort of a deeply nerdy version of a typical user's manual. There is no standardised format for such spec sheets, and some have more detailed information than others. The arrangement of datasheets can vary greatly, with different titles given for similar sections of the sheet. Some are only a few pages long while others, for more complex parts, can run dozens (even hundreds) of pages long and include a table of contents. Regardless of the eccentricities of the page layout, most datasheets for discrete components and logic devices have similar content and a similar goal: to convey the basic functions, operations, limitations, requirements, and design considerations of the part. Since the majority of the users of datasheets are engineers, these documents are not written with laypeople in mind. But that doesn't mean they can't be extremely helpful to the hobbyist, even if you don't fully understand some of deeper engineering esoterica presented.

Let's have a look at some of the more common datasheet sections, and what they contain.

FEATURES

This section is common to all datasheets and usually contains the headlines about the component, its bullet points. It contains things like the operating voltage ranges, power consumption, inputs and outputs available, and other key features of the part. One thing every actual engineer knows is not to rely too heavily on the features list. It can contain ratings that are misleading, if not downright incorrect. You always want to look at the more detailed specs for each feature that are unpacked further down on the sheet.

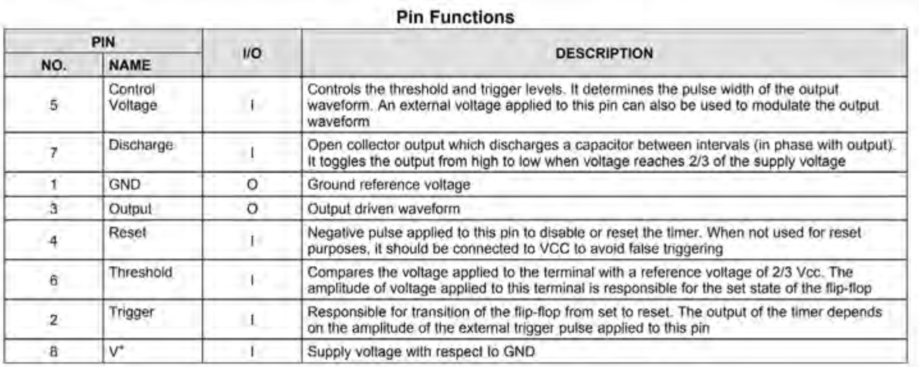
APPLICATIONS

This is a list of some of the more common applications of the part. For commercial engineers, this is somewhat useless information, but for hobbyists, it can give you other ideas for future projects where this part can be used.

DESCRIPTION

This section describes the basics of the part itself, what it's designed to do in general terms, the various packages (parts enclosures) that it is available in (e.g. DIP, SOIC, SOP, PDIP, TO, etc.), and the

Blow
The pinout list is essential for using any chip



Pin Functions

PIN		I/O	DESCRIPTION
NO.	NAME		
5	Control Voltage	I	Controls the threshold and trigger levels. It determines the pulse width of the output waveform. An external voltage applied to this pin can also be used to modulate the output waveform
7	Discharge	I	Open collector output which discharges a capacitor between intervals (in phase with output). It toggles the output from high to low when voltage reaches 2/3 of the supply voltage
1	GND	O	Ground reference voltage
3	Output	O	Output driven waveform
4	Reset	I	Negative pulse applied to this pin to disable or reset the timer. When not used for reset purposes, it should be connected to VCC to avoid false triggering
6	Threshold	I	Compares the voltage applied to the terminal with a reference voltage of 2/3 Vcc. The amplitude of voltage applied to this terminal is responsible for the set state of the flip-flop
2	Trigger	I	Responsible for transition of the flip-flop from set to reset. The output of the timer depends on the amplitude of the external trigger pulse applied to this pin
8	V^+	I	Supply voltage with respect lo GND

NOT ALL SHEETS ARE THE SAME

Sometimes it's worth comparing datasheets from different manufacturers for the same part. Some manufacturers will include more detailed parts information, performance graphs, or additional footnotes that can be very useful in more deeply understanding your part. Just make sure it is the exact same component that you are looking at.

PAY ATTENTION TO THE FINE PRINT

Don't ignore the footnotes often found on datasheets. They can offer valuable cautions, exceptions, design notes, and recommendations for optimal use of the part.

dimensions for each package type. Any variations of the part (e.g. TPS61090 TPS61091, TPS61092) and the different packages for each are usually listed here, too.

Package dimensions are often given here (or appear elsewhere on the sheet), which are useful in designing your circuit in a schematics program. But if you know the package type you're going to use, most schematic design programs will already recognise standard package dimensions.

FUNCTIONAL BLOCK DIAGRAM/ PINOUTS/PIN FUNCTIONS

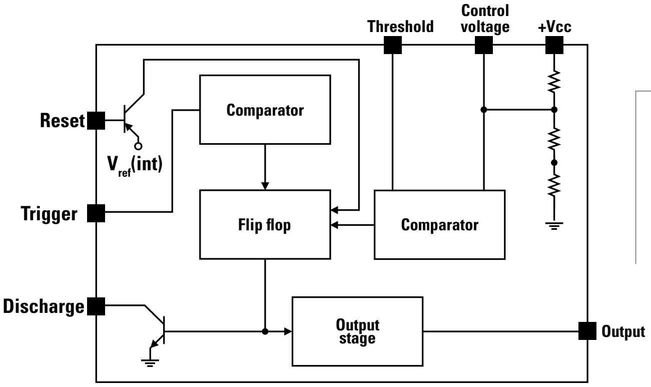
This section (or sections) details the internal functions of the part. It will usually contain a diagram of the component and a table showing the pins on the part and what each of them do. The functional and pin-outs diagrams often use acronyms that can be intimidating to newbies. Many of these are explained later on in the datasheet. You can also do an online search to find out what they mean. For ICs, the diagrams usually show you a top view, and include an indicator of where pin 1 is on the chip (often indicated on the actual part with a mark or divot on the top of the package).

SPECIFICATIONS

Under specifications, you will usually find the chart for Absolute Maximum Ratings (e.g. supply voltage, input/output current, continuous output current). Absolute Max Ratings are the values a part can withstand before you risk generating the dreaded magic blue smoke (i.e. severe electrical stress and a dead part).

RECOMMENDED OPERATING CONDITIONS

The safe operating ranges for your part are found under Recommended Operating Conditions. These are the specs you actually want to pay close attention to when designing your project. This table will list the minimum, maximum, and typical operating parameters for the part.



Left The block diagram from the datasheet of an LM555 gives you a high-level overview of how the chip works

ELECTRICAL CHARACTERISTICS

Here is where you'll find more specifics on supply inputs, voltage regulation, temperature specs, and the min, max, typical operating conditions for voltage and current.

PERFORMANCE CHARACTERISTICS/ PERFORMANCE CURVES

Some datasheets will contain graphs detailing various performance characteristics of the part (e.g. supply voltage and current, and temperature ranges). Pay attention here to the safe zones of operation that are indicated.

TRUTH TABLES

These logic tables are used to determine what inputs will produce what outputs. An H in a table indicates that an input requires logic 'high' to switch on (usually applying positive voltage, Vcc) and an L indicates logic 'low', usually ground (GND). An X in a truth table means that the component doesn't care if the input is high or low.

ADDITIONAL SECTIONS

Datasheets for complicated parts can include a lot of other application information such as more details on pin functions, memory tables (if applicable), sample schematic diagrams, and layout considerations (e.g. dealing with heat issues, potential mounting issues, and noise-reduction considerations). You can get very valuable information about the part, its operation, and its limitations by spending the time reading all of these sections. □

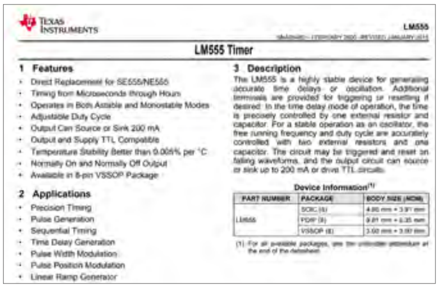
CHECK THE DATE

Always pay attention to the revision date on the datasheet, and make sure you have the most recent one. The newer the component, the more likely that it has multiple recent revisions.

YOU'LL NEED

- **PDF datasheet for the part you want to investigate**

Below At the top of the datasheet, you'll usually find a summary of how to use the device



TEXAS INSTRUMENTS

LM555

SNAS548D – FEBRUARY 2000 – REVISED JANUARY 2015

LM555 Timer

1 Features

- Direct Replacement for SE555/NE555
- Timing from Microseconds through Hours
- Operates in Both Astable and Monostable Modes
- Adjustable Duty Cycle
- Output Can Source or Sink 200 mA
- Output and Supply TTL Compatible
- Temperature Stability Better than 0.005% per °C
- Normally On and Normally Off Output
- Available in 8-pin VSSOP Package

2 Applications

- Precision Timing
- Pulse Generation
- Sequential Timing
- Time Delay Generation
- Pulse Width Modulation
- Pulse Position Modulation
- Linear Ramp Generator

3 Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200 mA or drive TTL circuits.

Device Information[1]

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM555	SOIC (8)	4.90 mm × 3.91 mm
	PDIP (8)	9.81 mm × 6.35 mm
	VSSOP (8)	3.00 mm × 3.00 mm

[1] For all available packages, see the orderable addendum at the end of the datasheet.

Modular NeoPixel patches

These light-up patches look cool on a denim jacket and can be swapped around

Sophy Wong

@sophywong

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

What could possibly be more amazing than hook-and-loop tape? Conductive hook-and-loop tape! That's right, your favourite fastener is available in a silver-coated version that is perfect for making a modular circuit with custom swappable components. In this tutorial, we'll use it to create modular NeoPixel patches for a jacket. The microcontroller and battery for this circuit will be located in the jacket, and be connected to the loop side of the hook-and-loop tape. NeoPixel LEDs will be sewn into the removable patches. Whichever patch you slap onto your jacket will light up!

A Gemma M0 microcontroller from Adafruit will control the NeoPixels, and we'll program the Gemma with CircuitPython. Anytime you want to change the code, just plug the Gemma into your computer and edit the code directly on the device. We'll power the whole circuit with a small battery pack that can hide in a pocket. A sewn circuit is perfect for this project, as it will keep the patches thin and soft.

To achieve good swappability, the hook-and-loop tape will have to be in exactly the same position on the back of each patch, and all the patches should adequately cover the hook and loop tape underneath. The NeoPixels can be in different locations from patch to patch, but it's definitely easiest if they are vertically centred and spread out, as shown on our cactus patch.

Embroidered iron-on patches are perfect for this project. Try to choose patches that are at least 8 or 9 cm long and 6 or 7 cm wide. You'll need some space

Right We'll add glowing hands to this cactus using NeoPixels

Left A selection of patches and components

Below Draw a border around the patch after ironing it to the felt

to spread out the hook-and-loop, and it will be easier to avoid shorts if your conductive traces have lots of room between them. If you want to use smaller patches, iron a few to a single piece of felt to give yourself more surface area to work with. If your patches are small enough, the conductive hook-and-loop may be enough to hold them on securely, but in most cases you'll want to add some regular (non-conductive) hook-and-loop tape for extra security.

Look for patches with graphics that will be easy to incorporate the NeoPixels into. Our circuit will have two NeoPixels, which is perfect for eyes on a face, or glowing Iron Man hands on a cactus (why not?). Get creative with your designs, and keep your circuit in mind while designing. Sketch out your sewing paths before you glue anything down, to make sure none of the conductive traces will cross or touch each other. Pick out your patches and let's get started!

DESIGN THE CIRCUIT

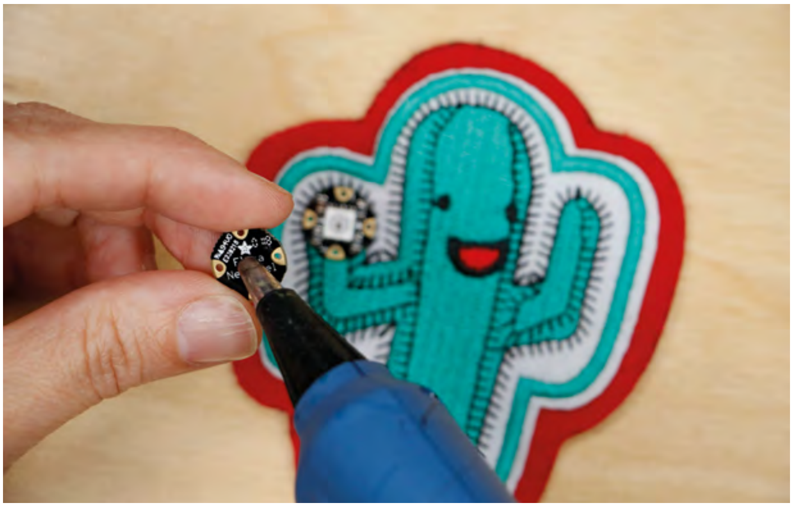
As always, start by designing your patch and sketching out the paths for sewing your conductive traces. Look at all the patches you plan to use and decide on a general layout that will work for all of them. Make sure both LEDs are in the same orientation: the negative pin should be at the top, the positive pin at the bottom, and the data pins' arrows should point to the right. When looking at the front of your patch, the NeoPixel furthest to the left is the first NeoPixel. Design your circuit so that data flows from D0 on the microcontroller to the data-in pin on the first NeoPixel. Then sketch a path from the data-out pin on the first NeoPixel to the data-in pin on the second NeoPixel.

The back of each patch will have three pieces of hook tape (the scratchy side). These should be spaced exactly the same way on each patch. For our patches, the top piece connects to the ground (-) pins, the middle piece connects to power (+), and the bottom piece connects to data-in on the first NeoPixel. Because our NeoPixels are mostly in the centre of our patches, this arrangement made it easy to sew thread traces without crossing over each other. Take the time to work your paths out before you start so you can avoid frustrating shorts and troubleshooting later.

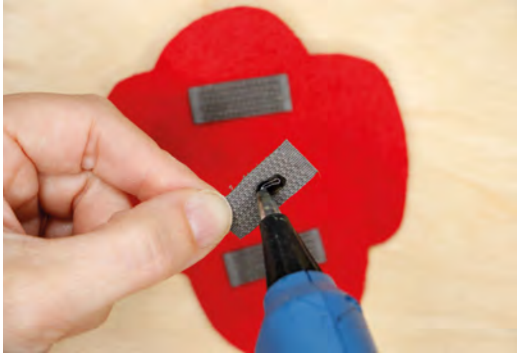
Notice that in our design, the top two pieces of hook-and-loop tape are horizontal, and the bottom piece is vertical. This configuration acts as a visual aid to help keep track of things while working on the back of your patch. With this layout, you'll always know which side is the top of your patch, and what you're connecting to. It also helps to ensure that you'll always put the patch on your jacket the right way up, so your connections will be correct. →

YOU'LL NEED

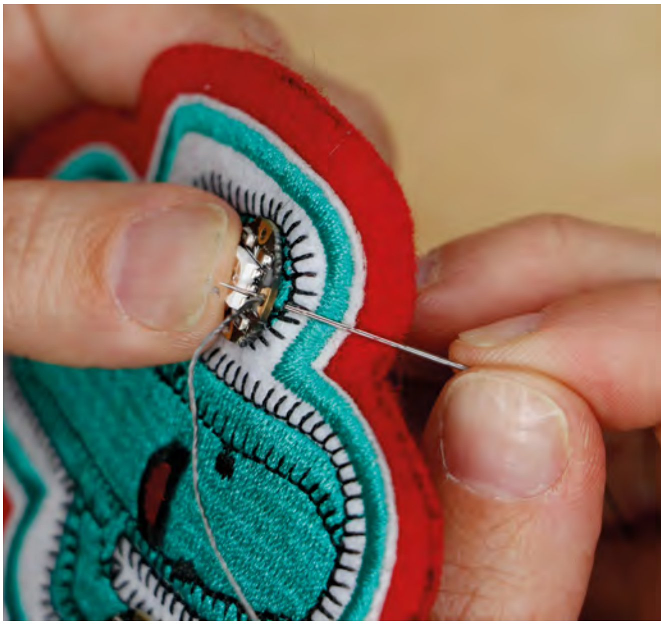
- Purchased patches
- Felt in coordinating colours
- Sewable NeoPixels (two per patch)
- Conductive hook-and-loop tape (3 cm per patch)
- Gemma M0
- 2 × 2032 coin cell battery holder with JST connector
- Conductive thread
- Hot glue gun
- Fray Check or cyanoacrylate glue
- Needle and sewing thread
- Regular (non-conductive) hook-and-loop tape
- Alligator clip leads for testing



Above Hot-glue each NeoPixel into place on the patch



Right Glue three pieces of conductive hook tape to the back



Above Connect the NeoPixel with conductive thread

To make the iron-on patches removable, we'll be affixing them to a piece of felt before adding the hook and loop tape. The felt will be visible, so choose colours that coordinate with your patches.

When you've got a clear sketch for your patch design, you're ready to start building the circuit!

> **When you're finished sewing, trim all thread tails short** to keep them from touching other lines of your circuit

MAKE THE FIRST PATCH

Place the patch on a piece of felt that is at least a few centimetres wider than the patch on all sides. Following the manufacturer's instructions, iron the patch to the felt. When the patch is securely attached, cut the patch out of the felt, leaving a border of about ½ cm on all sides.

Using a small dab of hot glue, glue each NeoPixel in place on the patch, making sure to orient each one correctly according to your design. Remember, the data arrows should be pointing to the right.

Next, cut three pieces of conductive hook-and-loop tape, making each about 1 cm long. Pull the hook and loop sides apart, and set aside the loop pieces (you'll use them for the jacket). Flip the patch over and glue the three pieces of hook tape to the back. Leave at least 1.5 cm of space between each piece to ensure they will not touch. Remember to make the top two pieces horizontal, and the bottom piece vertical.

Now, grab your sewing needle and conductive thread, and start sewing the traces of your circuit. Refer to your sketch, or draw out your paths right onto the back of your patch. First, sew the ground lines, which will go from the ground pins (-) of each NeoPixel to the top piece of hook tape. Start from the back of the patch, and take a few tight stitches around the ground pin. Then, use a running stitch to sew along the back of the patch to the top piece of hook tape. Try to keep the stitches from showing on the front by just sewing through the felt and not the patch. When you get to the hook tape, take several tight stitches at the edge for a good connection, and tie the thread off. Snip the thread tail short, and dab a little Fray Check or cyanoacrylate glue on the knot to keep it from unravelling. Then, sew the positive lines and data lines of your circuit in the same way, following the paths in your sketch.

When you're finished sewing, trim all thread tails short to keep them from touching other lines of your circuit. Dust off the back of your patch well, and make sure there aren't little conductive thread clippings stuck in your felt. Now let's program the Gemma and make sure the circuit works.

PROGRAM THE GEMMA AND TEST THE CIRCUIT

Grab three alligator clip leads, and clip one to the Gemma: attach one to the GND pin, one to the Vout pin, and one to D0. Attach the other end of each lead to a piece of the loop tape you've set aside for the jacket portion of the circuit. Then, connect the hook-and-loop tape to the patch to mock up the circuit – ground at the top, Vout in the middle, and D0 at the bottom.

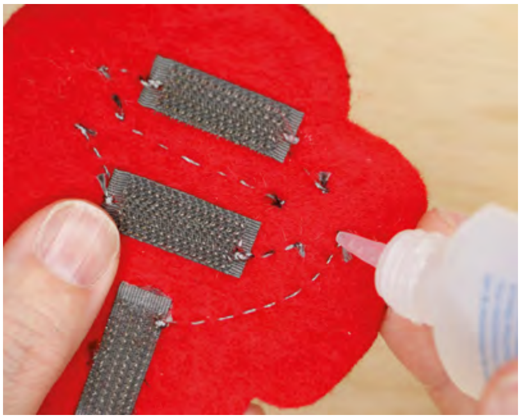
If this is your first time working with CircuitPython, visit **hsmag.cc/ZKvDqa** to get started. Follow the instructions there to update your Gemma M0 with the latest version of CircuitPython.

When connected to your computer with a micro USB cable, the Gemma will appear on your desktop as a drive. Open the drive to find the **main.py** file – this file contains the CircuitPython code that the Gemma is running, and it can be edited in any text editor. Saving this file will make it run automatically on the Gemma M0.

If your Gemma M0 is brand new, there will be sample code in your **main.py** file. Select it all and delete it. We'll write our own code for a simple, colour-cycling animation that will look good on all of our patches.

As per usual, we will start by importing the necessary libraries, setting up variables for our data pin and the number of NeoPixels we're using, and we'll create the NeoPixel strip as an object with two NeoPixels.

```
import time
import board
import neopixel

pixpin = board.D0
```

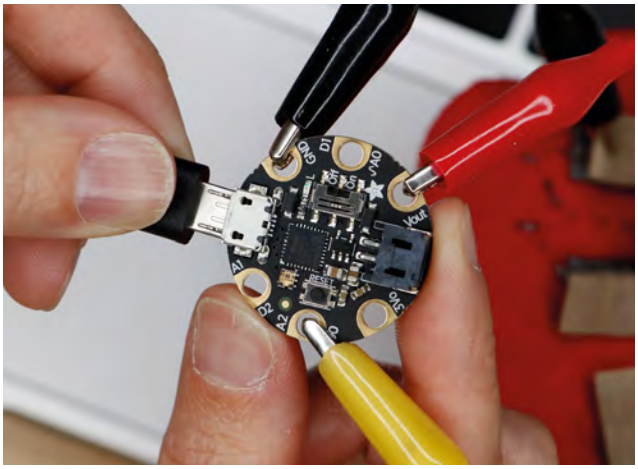


Above & Left
The back of the patch won't be visible, so it doesn't have to be too tidy

```
numpix = 2

strip = neopixel.NeoPixel(pixpin, numpix,
brightness=.5, auto_write=False)
```

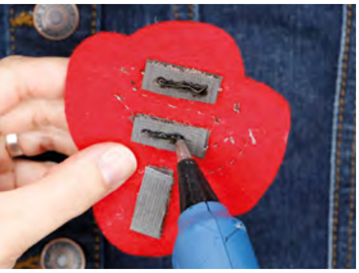
We'll also set up variables with starting values for the red, green, and blue values of our strip. These values can be anything from 0 (off) to 255 (full intensity). For dynamic colour change, we'll want to change each colour value at a different rate, so we'll also need to create a change value for each colour. You can tweak these values later to tune the colour changes to your liking:

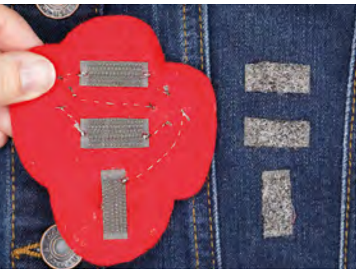
```
r = 0
g = 85
b = 170

rChange = 1
gChange = 5
bChange = 12
```



Below Left
One cactus, almost ready to light up the night

Below
Crocodile clips make it easy to test the circuit





Top
Apply hot glue to the back of each piece of loop tape

Above
Hold it in place on the jacket, then pull the badge away, for perfect tape alignment

In the main loop, start by filling the strip with the starting RGB values:

```
while True:
    COLOR = (r, g, b)
    strip.fill(COLOR)
    strip.show()
    time.sleep(0.03)
```

To make the colours change over time, let's add the increments we declared in our setup. This will increase each value by these increments every time the loop is repeated.

```
    r = r + rChange
    g = g + gChange
    b = b + bChange
```

Finally, because RGB values only range from 0 (off) to 255 (full), we need to set upper and lower limits for each value. For this, we'll use conditional statements. Starting with the `r` value, when 255 or 0 is reached, we'll flip the direction of the change by making `rChange` negative. Repeat this for both `g` and `b` values. This will make the colour values change smoothly from 0 to 255 and back again.

> **For a good connection and no shorts,** the loop tape on the jacket must line up with the hook tape on the patches

```
    if r > 255:
        r = 255
        rChange = -rChange
    elif r < 0:
        r = 0
        rChange = -rChange

    if g > 255:
        g = 255
        gChange = -gChange
    elif g < 0:
        g = 0
        gChange = -gChange

    if b > 255:
        b = 255
        bChange = -bChange
    elif b < 0:
        b = 0
        bChange = -bChange
```

Above
Tight stitches help make a good connection on the pad

When you're finished writing your code, save the file and the Gemma will auto-load the new code. You should see both NeoPixels on your patch light up and slowly change colours. If everything is working as expected, do a little dance and move on! If not, check your connections and make sure your thread traces do not touch each other. Use a multimeter, if necessary, to check for shorts in your circuit. When your circuit is working properly, you're ready to sew the receiving end into your jacket!

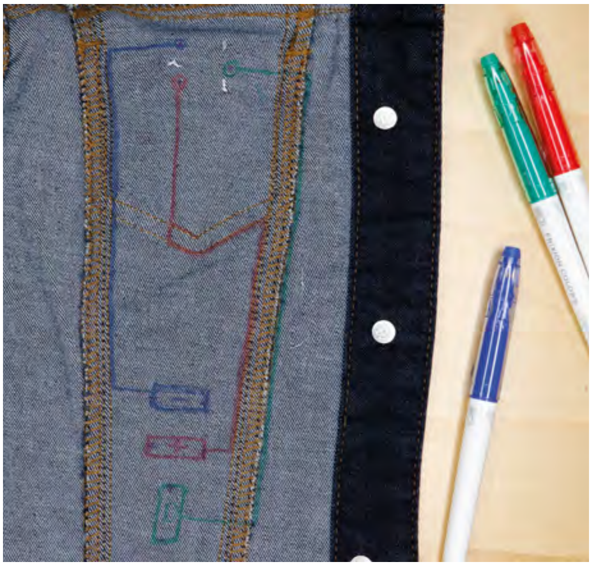
ADD LOOP TAPE TO THE JACKET

Remove the alligator clip leads from your patch. For a good connection and no shorts, the loop tape on the jacket must line up with the hook tape on the patches. To ensure that it matches perfectly, press the loop tape pieces onto the hook tape pieces on the back of the patch. Apply hot glue to the back of each piece of loop tape, and press the badge into place on the jacket. Hold it there for a moment while the glue solidifies, and then pull the hook-and-loop tape apart to release the badge. Voilà! Your loop tape is guaranteed to be in the right place for your badge. Sew around the edges of the loop tape with regular thread for a secure hold.

You may want to add some additional pieces of regular (non-conductive) hook-and-loop tape to hold the patches securely to the jacket. Place these around the edges of your patch, and repeat the step above to line them up on your jacket.

INSTALL THE GEMMA M0

Next, decide where your Gemma M0 will live. A pocket is the perfect place to hide it, and if you're



Above Left To hide the circuit, sew the conductive thread along the jacket's stitch lines

Above Connect the thread to the loop tape

Above Install the Gemma M0 in a pocket if your jacket has one

lucky, there will be room for the battery pack in there too. If your pocket has a metal button that might touch the Gemma, like ours did, cover the back of the button with electrical insulating tape or tuck a spare piece of felt into the pocket as a barrier.

If your jacket doesn't have a pocket, choose a spot that will protect the Gemma M0 from the elements, but keep it easily accessible for turning off and on, and reprogramming. Orient the Gemma so that the micro USB port is at the top, and the JST battery connector is at the bottom. Sew the Gemma in place with regular thread through the pin holes we won't be using for this project: D1, D2, and 3Vo.

Now have a look at the inside of your jacket, and plan out the paths you will use for sewing conductive traces from the Gemma to the three loop tape pieces. Remember that GND will connect to the top piece, Vout will connect to the middle piece, and D0 will connect to the bottom piece, and none of the traces can cross or touch each other. To hide the lines of your circuit, try to run the traces along the stitching lines of your jacket. Lightly drawing out your paths with different coloured pens is extremely helpful here.

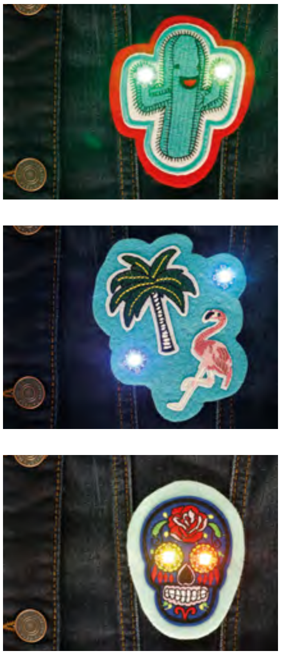
When you've designed your circuit, start sewing the traces with conductive thread, as before. Start with a few stitches around the pin on the Gemma, then follow your drawn line and finish with a few tight stitches to the loop tape. Make sure your thread is long enough to sew the whole line in one continuous piece. Cut thread tails short and secure all knots with a dot of Fray Check or CA glue. When your traces are sewn, you're ready for batteries!

The Gemma M0 can take up to 6VDC, and can be powered with a 2× 2032 coin cell battery pack, a 3× AAA battery pack, or a small 3.7 V lithium-ion battery. If you choose to go with a lithium-ion battery, protect it by wrapping it in gaffer tape and make sure it will not be compressed or punctured while wearing.

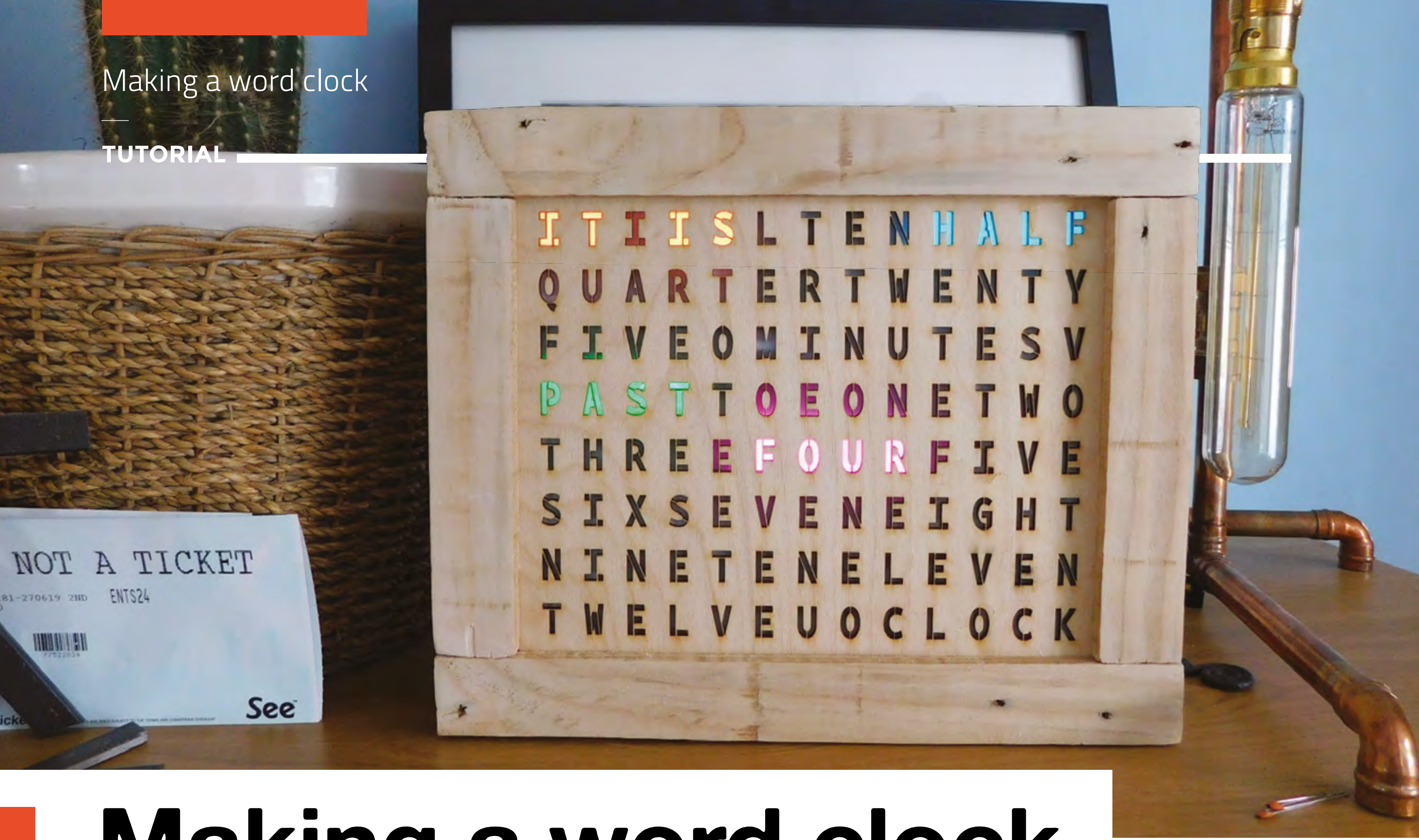
With your patch in place, plug your battery into the JST port on the Gemma M0 and switch the Gemma on. Don't be discouraged if your NeoPixels don't light up as they did before: troubleshooting is part of the process, especially with sewn circuits. Check the connections you just sewed into your jacket and eliminate any shorts until your circuit is working correctly. Don't forget to remove the batteries before cutting or sewing traces.

MAKE MORE PATCHES!

Now that you've perfected one patch, repeat the process to build your collection of NeoPixel patches! Experiment with other shapes and themes, or make a custom patch for a special event. Conductive hook and loop tape can be used to make other modular soft circuits, as well as circuits that open and close around the body, like belts and bracelets. Show us your modular makes at **hackspace@raspberrypi.org**! or on Twitter **@HackSpaceMag**. □



Above Over to you. Now get creative!



Making a word clock

Build your own attractive timepiece

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.

OK, settle in. This project turned out to be a bit more complex than expected. Actually, complex isn't quite the right word. There's nothing in here that's fundamentally hard, but it did test our skills in quite a few different areas of making, and each area posed its own little challenges that needed to be overcome. We'll guide you through it as best as we can.

In this project, we used quite a wide range of equipment and parts. These represent the tools and parts we had available, rather than a canonical set of things you actually need. There's no 'right' way of doing this, and you can find alternatives to almost everything we've used if you need to.

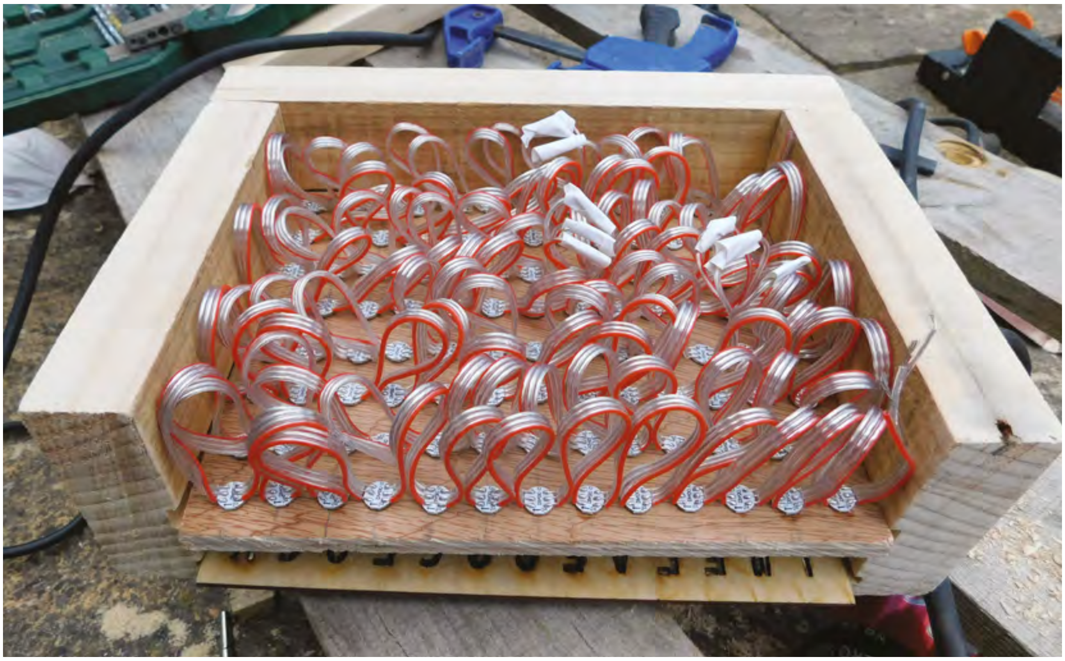
Above The finished clock on display. Eagle-eyed readers may also spot projects from previous issues

Right Drilling holes for the LEDs. Precision isn't essential in this step

The basic way a word clock works is that it shines a light through letters spelling out the words to say the time. The heart of our clock, then, is these letters and the LEDs to make the light. We used laser-cut 3 mm plywood for our clock face, but other people have had success using printed acetate sheets (the sort used in overhead projectors that older – but not too old – readers will remember from their school days). Thinner laser-cut sheets would also work, but we'd recommend going no thicker than 3 mm as this will reduce your viewing angle.

You can grab our design from **hsmag.cc/issue20**, but it's fairly easy to create your own (or modify ours if you'd prefer). The crucial point for the lettering is that we need to use a stencil font – this ensures that there's a connection to any isolated parts of a letter (such as the middle of the letter O), so they don't fall out when laser-cut. It makes layout easier if the font is monospaced – we used BP Mono Stencil (**hsmag.cc/BPMonoStencil**).

The LEDs must be held in the appropriate place behind the letters. There are two approaches that you can take here – you can design your letters so that they line up with off-the-shelf LED strips, or you can use strings of LEDs to line up with whatever spacing you use for the letters. We opted for the latter, but the former would make a more



Left The LED string inserted. We had to join three strings together to get 104 LEDs for the clock. In hindsight, only 100 are needed, as some letters are never lit

YOU'LL NEED

- WiFi-enabled microcontroller (such as the MKR1000)
- String of 104 NeoPixels
- 1 A diode
- 9 mm plywood
- 3 mm laserply
- Wood for frame
- Laser cutter
- Modelling foam

straightforward build if you're less fussy about the size of the clock.

We then need a way of holding the LEDs in place behind the letters. There are a few parts to this – first, you need a way of holding the LEDs in place far enough behind the letters so that they illuminate them evenly; then you need a way of minimising the amount of 'bleed', where lighting one letter illuminates the letters either side of it; finally, you need something to diffuse the light.

Our setup used plywood with 7 mm holes drilled into it. This is just large enough for surface-mounted 5050 LEDs to be pushed in place and held with a drop of superglue. These shone through the holes in the plywood and into a square honeycomb made of modelling foam hot-glued together. Finally, it hit a double-layer of diffusion fabric before shining through the laser-cut face. All we needed was a frame to hold it in place. We made this from 4×1 inch reclaimed wood with routed grooves to hold the face and plywood LED panel in place.

Let's take a closer look at this process before diving into the microcontroller brains.

THE BUILD

First, you'll need to laser-cut your clock face – that's the easy bit of woodwork. Now on to the manual part…

As mentioned, we started building our frame with reclaimed 4×1 inch wood that cost just £1 from our local wood recycling project. We sanded this down to give it a smooth finish, but it lacks the hard corners of planed wood. There are also holes from old nails which combine with the rustic joining technique to give the look we wanted for our clock.

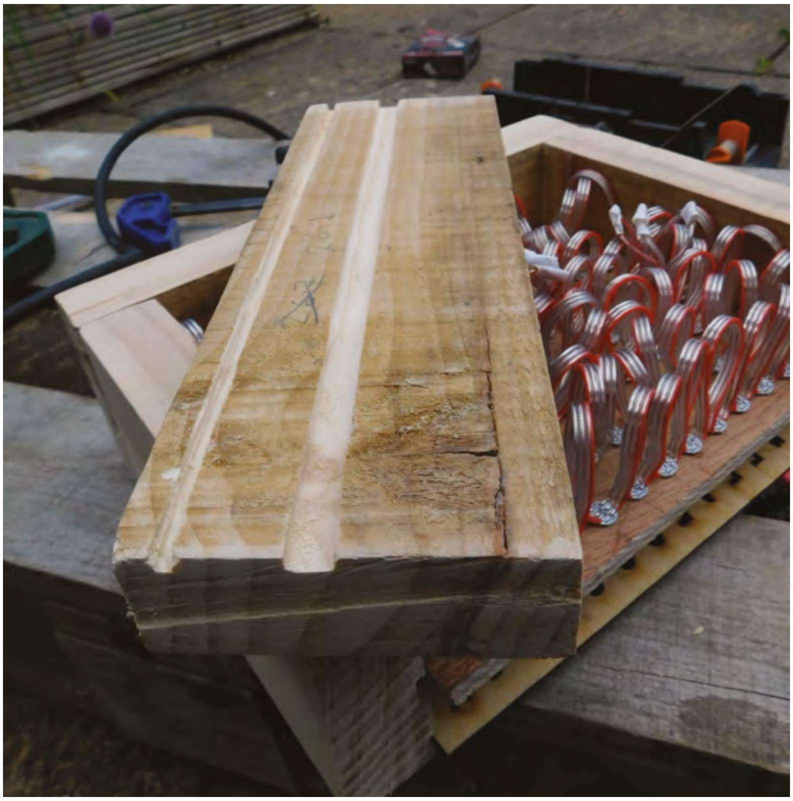
If you're an experienced woodworker, you may choose a more elegant method for making the frame, but as we're not, we'll keep it simple. We've used butt joints in the corners which are held together with two screws each. First, we routed two grooves in one side of the wood – one to hold the 3 mm face, and one to hold the 9 mm plywood LED panel. 9 mm ply is overkill for such a frame, but we happened to have some spare from a previous project – you could easily get by with 3 mm or 6 mm plywood, and MDF would work just as well. We routed these grooves 3 mm deep into the wood. →

> "It hits a double-layer of diffusion fabric before shining through the laser-cut face"

WIRING

The simplest wiring for the clock is to connect the 5 V and GND pins and one data point (we used pin 6) from the microcontroller to the 5 V, GND, and data input pins on the first LED. The LED chain will then propagate power and data along the strip. However, there are a few problems with this.

Firstly, this results in an out-of-spec power situation which you might, or might not get away with (see 'Power problems' box on page 107). Secondly, the jitter on the power line may cause problems – putting a capacitor between the 5 V and GND lines can smooth that. Thirdly, you should include a 470 Ω resistor between the Arduino pin and the data-in line. You might get away without this, but it will prevent any problems with too much current being drawn.



Above The two grooves routed in our wooden frame to hold the clock face and the LED board

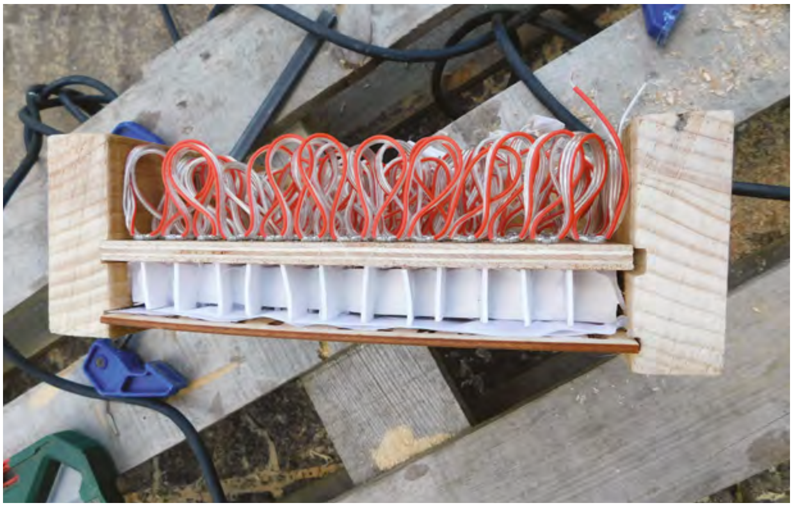
If you've got a plunge router bit, you might choose to do this groove-cutting later, and not rout all the way to the edge of each section of frame as this will give a better finish.

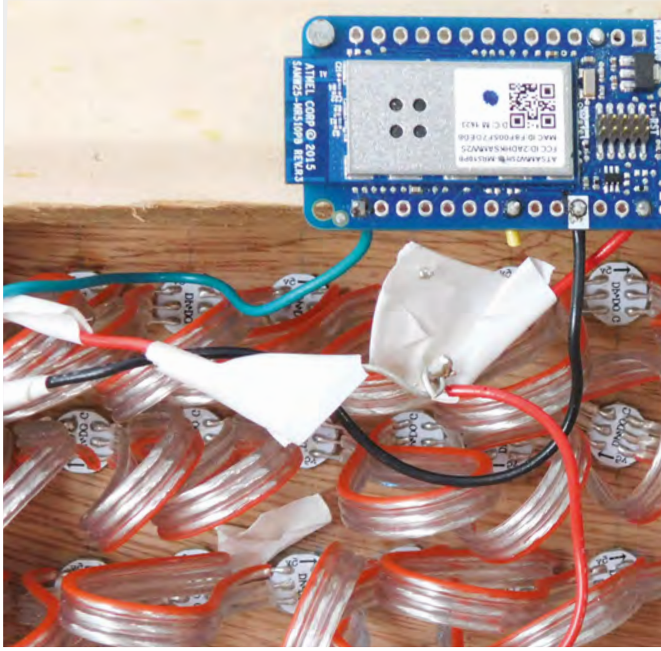
We then needed to cut the wood into four appropriate length sections. You need two for the top and bottom that are:

length = width of face + (2 × width of frame wood) – (2 × depth of groove)

And two for the side that are:

length = height of face – (2 × depth of groove)





You should now be able to hold them all by hand and everything should fit together (don't screw or glue them together yet). If they don't fit, you'll need to make adjustments before moving on. This might entail routing the grooves a little deeper, or trimming down the wooden frame.

INNARDS

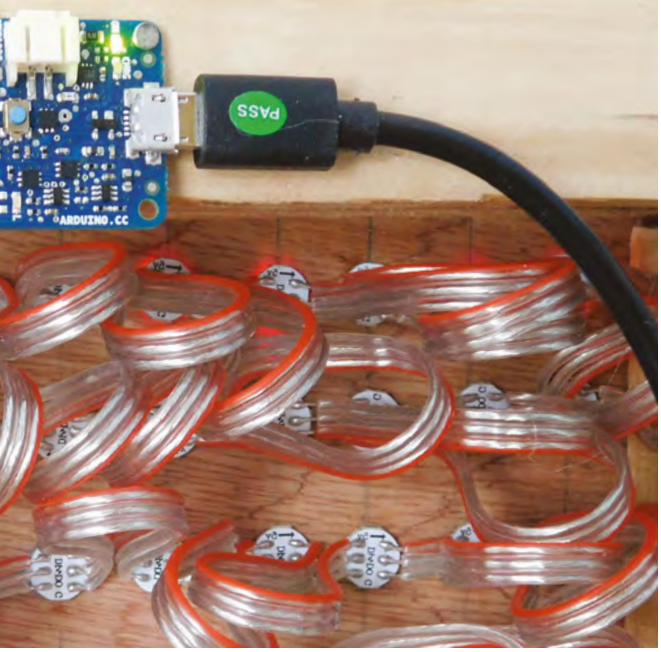
The quickest way to mark up the plywood LED holder is by eye. It needs to be the same size as the clock face, and you can pencil-mark the spots for the LEDs very quickly without the need for a tape measure (though measure and mark properly if you'd prefer).

As previously mentioned, we drilled these out with a 7 mm drill bit. The diagonal on a 5050 surface-mount part is just larger than 7 mm, so it's a tight fit. We used strings of WS2812 LEDs (often known as NeoPixels). Each LED is on a small, circular PCB. We applied a drop of superglue to the edge of each LED, then pushed it into the hole in the circuit board. They take a bit of force to get in, but be careful, as we pushed too hard on one and dislodged a resistor (if you do this, just cut out the LED in question and join the wires with solder).

TESTING

Your build will almost certainly be slightly different to ours, so rather than just following along by rote and hoping that the results are the same, now's a good

Left The foam honeycomb – if we'd made this fit better, we'd have less bleed between the different letters



point to pause and check that everything's working as you'd like.

Connect the microcontroller up to the NeoPixels (we used crocodile clips, but you can solder it up if you don't have these). See the 'Wiring' instructions box on page 103.

We used the test code from the Adafruit NeoPixel Überguide to make sure everything was working properly (**hsmag.cc/ArduinoLibraryUse**). Bear in mind that lighting up all the pixels at once will take quite a bit of current, so you will want to either use an external power supply or dial down the brightness (we tested ours with a colour of (10,10,10) and this worked with the on-board regulator on the MKR1000).

With this and a mess of wires in place, and everything working, let's move on with the assembly. Screw together three sides (one long side and two short sides) of the frame. To ensure that it is in the right place, it's a good idea to use an F-clamp to hold it together with the clock face and the plywood panel in place while drilling and screwing.

Leave one F-clamp in place, holding the two ends of the wood on the exposed side together while finishing the internal assembly.

We used 1 mm-thick white modelling foam for the square honeycomb inside the frame. You may want to consider laser-cutting this out using something like the tray insert pattern from **hsmag.cc/TrayInsert**; however, we didn't. We cut long strips the width of the frame and the height of the gap between the plywood and the back of the face, and small 'separator' strips to split it up vertically. Gluing this together was a bit more challenging than we anticipated, but with the right technique it's not too hard.

First, anchor one end of one long strip to the frame and wait for the glue to harden. Then put a 'U' of glue in where you want one of the separators to go, and then slot the separator into this glue (don't try to hold it in place while you put the glue in). With practice, you can do several of these 'Us' of glue at a time (we found four or five was a good number), then insert all the separators in one go. Before you finish one row, anchor the next long row strip to the frame, as this gives it time for the glue to harden before starting that row.

Left
We didn't have screws small enough for the mounting hole, so we used nails. In hindsight, this was a very risky move, and one we don't recommend you copy

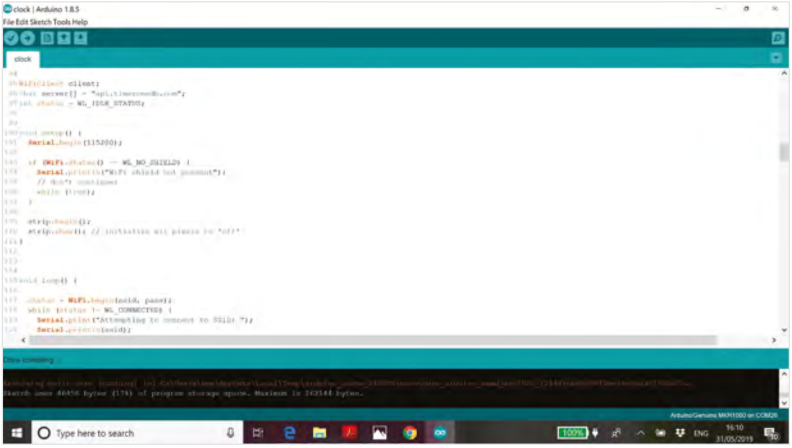
> **"Experiment with what you have** to see what creates the aesthetic that you want"

DIFFUSION

The final thing to add before assembly is diffusion. This can be anything that's translucent and thin enough to fit in the space. We used photographer's diffusion fabric (essentially a thin, white nylon material), and we found that we needed two layers of this to get the look we wanted, but it's not standard fabric, so experiment with what you have to see what creates the aesthetic that you want.

We cut this to size and placed it over the square honeycomb. A few dabs of hot glue on the corners held it all in place (and this won't be visible once it's fully assembled). →

Below
The Arduino code checks the time on the internet every minute and displays it on the clock



```
WiFiClient client;
char server[] = "api.timezonedb.com";
int status = WL_IDLE_STATUS;


void setup() {
  Serial.begin(115200);

  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while (true);
  }

  strip.begin();
  strip.show(); // initialize all pixels to "off"
}



void loop() {

  status = WiFi.begin(ssid, pass);
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
```

Done compiling.

```
[illegible]
Sketch uses 44456 bytes (17%) of program storage space. Maximum is 262144 bytes.
```

Arduino/Genuino MKR1000 on COM26

Above NeoPixel strings are easier to mount in custom spacings than NeoPixel strips, but still have the advantage of not having to solder every LED

Once you're happy with the amount of diffusion, you can attach the final side of the frame, and that's the hardware setup complete. Now let's take a look at the software.

The full code for this is available from **hsmag.cc/ClockCode**, but let's take a look at the most pertinent bits.

Obviously, our clock needs to know what the time is. We could have used a real-time clock, but this would still necessitate setting the clock time manually and adjusting the time for daylight savings. Instead, we decided to grab the time from the internet – specifically, **timezonedb.com**.

You'll need to register for a free API key, but we'll be staying well within the limits of free use. Once you've got that, you can grab the current time in a particular location by pointing your web browser to **api.timezonedb.com/v2/get-time-zone?key=KEYHERE&format=xml&fields=formatted&by=zone&zone=Europe/London**.

You'll need to replace **KEYHERE** with your key, and if you're not in the UK you'll need to update the zone to your location. The result comes back in XML, and should be something like:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<result>
<status>OK</status>
<message/>
<formatted>2019-05-30 14:52:07</formatted>
</result>
```

There are two parts to getting and processing this in Arduino. First, we have to download this XML, and then we have to extract the time from it. The method of connecting to WiFi differs a little depending on what hardware you're using. We used the WiFi101 library, but if you're using different hardware (such as an ESP8266) you might need to do it slightly differently. Take a look at your board's example WiFi sketches for details.

Once connected, we have a client object linked to the **api.timezonedb.com** server (see the full code for more info on this). We can then extract the appropriate line in the response with the following:

```
    client.println("GET /v2/get-time-zone?key=YOUR
KEY&format=xml&fields=formatted&by=zone&zone=Euro
pe/London HTTP/1.1");
    client.println("Host: api.timezonedb.com");
    client.println("Connection: close");
    client.println();
    }
delay(10000);
payload = "";
    Serial.println("stand by for data");
    while (client.available()) {
    char c = client.read();
    Serial.write(c);
    if (c == '\n') {
    payload = "";
    }
    payload += c;
    if(payload.endsWith("</result>")) {
    parse_response();
    }
```

This reads the HTTP response character by character and builds up a string called `payload`. If it reaches a newline character, it empties `payload` as we only want one line. If it reaches the string `</result>`, it knows that it's got the data it needs, so it called the function `parse_response`.

The key parts of this function are as follows:

```
 int colon = payload.indexOf(':');
// Set the first colon in time as reference point
    nowday = payload.substring(colon - 5,
colon - 3);
    d = nowday.toInt();
    nowmonth = payload.substring(colon - 8,
colon - 6);
    mo = nowmonth.toInt();
    nowyear = payload.substring(colon - 13,
colon - 9);
    y = nowyear.toInt();
    nowhour = payload.substring(colon - 2, colon);
    h = nowhour.toInt();
    nowmin = payload.substring(colon + 1, c
olon + 3);
    mi = nowmin.toInt();
    nowsec = payload.substring(colon + 4,
colon + 6);
    s = nowsec.toInt();
```

Since the time and date is in a specific format, we can locate the particular part we want relative to the first colon. This code pulls the string apart and converts the relevant segments into integer values for the hours, minutes, and seconds. It also extracts the date, but we don't use that. We adapted this code from Arduino forum user Aggertroll – thanks Aggertroll!

> **This reads the HTTP response character by character** and builds up a string called 'payload'

Now that we've got the time, we need a way of displaying it on the NeoPixel strip. This is done by first creating a series of arrays that hold the locations for the pixels in different words, such as:

```
int itis[] = {8,9,11,12};
int five[] = {35,36,37,38};
int ten[] = {4,5,6};
```

We also created a function that turns the LEDs in one of these arrays a specific colour:

```
void lightup(int letters[], int letters_len, int
red, int green, int blue) {
    for(int i = 0; i<letters_len; i++) {
    strip.setPixelColor(letters[i], red, green,
blue);
    }
    strip.show();
    }
```

POWER PROBLEMS

Once we wired up our clock, we found that it frequently glitched out and flashed strange colours. After unsoldering all the connections and rewiring it all up, we realised that the problem wasn't a cold joint, or even code problems, but a voltage mismatch.

We powered the LEDs from the 5V pin on the microcontroller (we can keep the LED numbers and brightnesses sufficiently low to allow this to work); however, the data pins on the MKR1000 are 3.3V. The input to the LEDs should be (according to their datasheet) at least 0.7 times the power voltage (3.5V), so we're going out-of-spec by powering it at 3.3V. Usually we can get away with this, but the particular LEDs we used proved to be particularly finickety about this.

There are two basic solutions to this – increase the input voltage or decrease the power voltage. We opted to do the latter by putting a diode with a forward voltage of 0.8V on the power line. This diode has to be able to take the full current of the LEDs (we used a 1A diode, which should give us plenty of leeway). Alternatively, you can use a level shifter (these are available in both module and IC form) to increase the voltage from the data signal to 5V.

The final code for lighting up the correct time is as follows:

```
strip.fill();
lightup(itis, 4, 100,100,0);
int hour = h;
if (mi > 33) { hour+=1;}
 if (hour > 12) { hour -= 12;}
if (hour==1) { lightup(h_one, 3, hour_red, hour_
green, hour_blue); }
if (hour==2) { lightup(h_two, 3, hour_red, hour_
green, hour_blue); }
...
//past or to?
if (mi > 3 && mi < 34) { lightup(past, 4,0, 150,
0); }
if (mi > 33 && mi < 58) {lightup(to,2,0,150,0);}
if (mi > 57 || mi < 4) {lightup(oclock,6,50, 50,
100);}
// minutes
if (mi > 3 && mi < 8) {lightup(five, 4, mins_red,
mins_green, mins_blue); lightup(minutes, 7,mins_
red,mins_green, mins_blue);}
if (mi > 7 && mi < 14) {lightup(ten, 3, mins_red,
mins_green, mins_blue); lightup(minutes, 7,mins_
red,mins_green, mins_blue);}
...
```

The first line in this code blanks the whole strip, then the line **lightup(itis, 4, 100,100,0);** lights up the words 'it is'. We then have to find the first hour, bearing in mind that as soon as the minutes have gone past 34, it will switch to 'twenty-five to' the next hour. The code then ends with a series of **if** statements that find the correct letters. □

Moving mechanisms with Fusion 360

It's all about having the right moves

Andy Clark

workshopshed

For the last ten years, Andy has been making and repairing in a shed at the bottom of the garden. You can see more of his exploits at **workshopshed.com**

YOU'LL NEED

- **A computer capable of running Fusion 360**
- **An internet connection**

QUICK TIP

Use 'Joint' to align and link the components. Use 'As Built Joint' when the parts are already in the correct places.

When you want to add motion to a project, simple motors, servos and solenoids will get you quite a long way. But, to get more complex motion, you'll need some mechanisms expanding these circular or linear movements. By modelling these using CAD, it is possible to experiment with the motion before you commit to a physical model.

To make a movable model in Fusion 360, the design needs to be created in a specific way. The moving parts need to be separate components, the fixed parts need to be marked as 'ground', and the parts need to be connected together using joints. To make our design so it can be experimented with, we need to take advantage of the parameters and constraints of the model.

PISTON

The piston shows off a couple of these joints: the revolute joint and the slider joint.

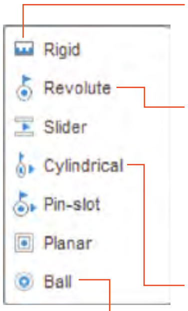
To model the piston, we will use simplified elements, but feel free to go back and embellish these.

Create a cylinder 5 mm in diameter and 6 mm long. Choose the 'New Component' operation. Rename this component 'Pin' and right-click on it and select 'Ground'. This will fix our pin in place.

Repeat the new cylinder operation by drawing a concentric circle on the same axis, this time 50 mm in diameter and 5 mm in height. Again, create as a new component and name this second cylinder 'wheel'.

From the Modify menu, choose the Combine option and cut the pin from the wheel; remember to tick the 'Keep Tools' checkbox.

Select the wheel and the pin component and from the Assemble menu choose 'as built' joint. Specify the centre of the pin as the axis of rotation, and check the animation to ensure it is rotating as expected.



Components are fixed together and move as one

One component spins or pivots in respect to the other, such as a wheel on a shaft

The components can slide and rotate like a pin in a tube

The component can rotate and slide, where the sliding motion is perpendicular to the rotation

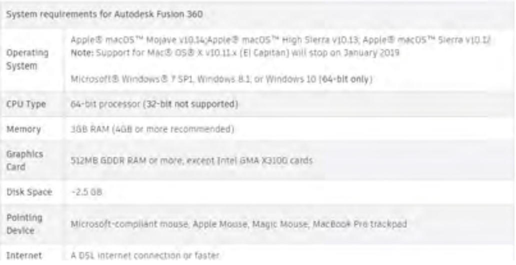
The components move on the same plane, like a box sliding on a tray

The components are linked in a single dimension

The components are free to rotate in all axes, like a hip joint

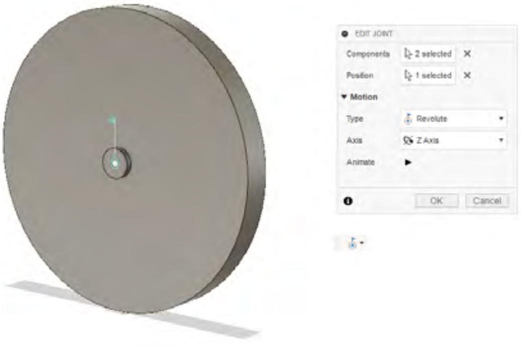
LICENCES

Fusion360 is a powerful piece of software, but that power comes with a price tag attached. For a hobbyist, teacher, or small business that only infrequently uses the software, a licence for the software is quite high at $495.00 per year. However, there is a start-up licence that can help you out giving you free access for a year. There's also a 30-day trial which should give you enough time to work out if you'll derive enough value from the product to justify the price. For more details, visit: **hsmag.cc/cVZM0H**



System requirements for Autodesk Fusion 360	
Operating System	Apple® macOS™ Mojave v10.14;Apple® macOS™ High Sierra v10.13; Apple® macOS™ Sierra v10.12 **Note:** Support for Mac® OS® X v10.11.x (El Capitan) will stop on January 2019 Microsoft® Windows® 7 SP1, Windows 8.1, or Windows 10 (**64-bit only**)
CPU Type	64-bit processor (**32-bit not supported**)
Memory	3GB RAM (4GB or more recommended)
Graphics Card	512MB GDDR RAM or more, except Intel GMA X3100 cards
Disk Space	~2.5 GB
Pointing Device	Microsoft-compliant mouse, Apple Mouse, Magic Mouse, MacBook Pro trackpad
Internet	A DSL internet connection or faster

Above
System requirements



EDIT JOINT

Components: 2 selected

Position: 1 selected

Motion

Type: Revolute

Axis: Z Axis

Animate: ▶

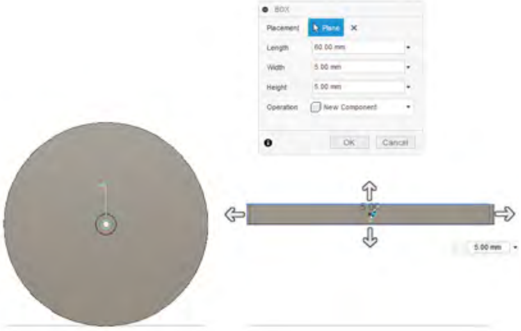
Above Modelling the coupling rod with a simple box

GEAR TERMINOLOGY

Pitch Diameter – The line at which the teeth make contact. Used to determine the spacing between the gears.

Module – Pitch diameter divided by the number of teeth. Gears with the same module value should mesh together.

Pressure Angle – The angle of the meshing faces. The industry standard is 20°. Controls power transfer and efficiency.



BOX

Placement: Plane

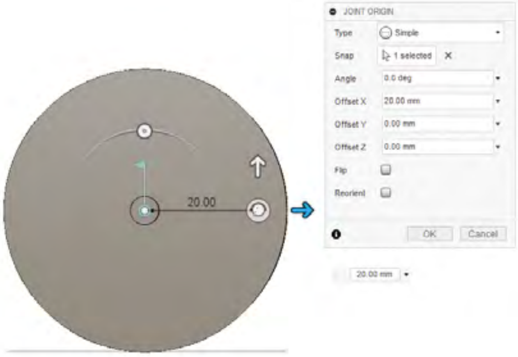
Length: 60.00 mm

Width: 5.00 mm

Height: 5.00 mm

Operation: New Component

Above Simple revolute joint



JOINT ORIGIN

Type: Simple

Snap: 1 selected

Angle: 0.0 deg

Offset X: 20.00 mm

Offset Y: 0.00 mm

Offset Z: 0.00 mm

Flip: □

Reorient: □

Left Use a joint origin to control the position of a joint

To create the pin that drives the connecting rod, we don't need to model the components, just the joint. Activate the wheel component and from the Assemble menu, select 'Joint Origin' and select the wheel. The joint origin should appear in the centre; move it to the right by 20 mm.

Create a box, 60 mm long, by 5 mm, by 5 mm to the right of the wheel. Again, select 'New Component' and name this 'Coupling Rod'.

Select the new joint origin on the wheel and, from the Assemble menu, select Joint. Select the coupling rod as Component1 and pick the top-left corner as the point of rotation. Enter the Offset X as -2.5 mm and Offset Y as 2.5 mm.

You can test this now by choosing Assemble menu, Drive Joints, and selecting the joint in the middle of the wheel. Spinning the wheel should move the connecting rod around too.

To complete the coupling rod, add another joint origin to the right-hand end of the rod, where it can connect to the piston.

Create a 20×30 mm box to the right of the coupling rod; set the height to 10 mm. This new component will be our piston. Add a 2 mm hole, 10 mm from the left end of the block and on the centre line. This will be our pivot point.

Assemble the piston to coupling rod revolute joint, selecting the origin and hole as the two locations to align. Offset the Z position of the piston so that it does not overlap the coupling rod.

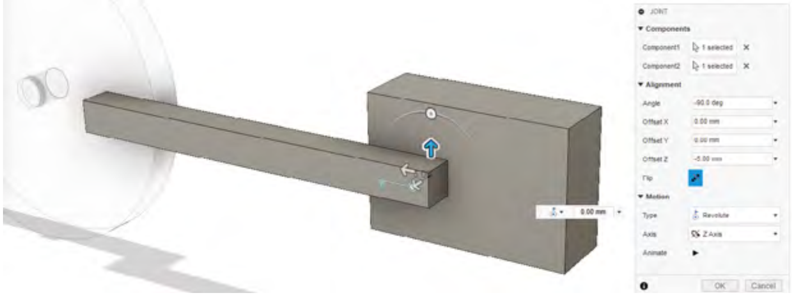
Create a sketch and draw a simple slot for our piston to slide into. Stop the sketch and extrude the profile to create the final component, the cylinder. Choose ground to keep this component fixed.

Choose the piston and cylinder components and create an as-built joint. This time, we will use the slider joint. Note how the piston crashes through our new cylinder. To constrain this behaviour, we can add a limit. Right-click on the joint and choose Edit Limit. Tick minimum and maximum and adjust the limits so that the piston is constrained within the cylinder. →

QUICK TIP

Joint Origin can define the position of the joint when there is no appropriate geometry to select.

Below Position can also be done when creating the joint



JOINT

Components

Component1: 1 selected

Component2: 1 selected

Alignment

Angle: -90.0 deg

Offset X: 0.00 mm

Offset Y: 0.00 mm

Offset Z: -5.00 mm

Flip

Motion

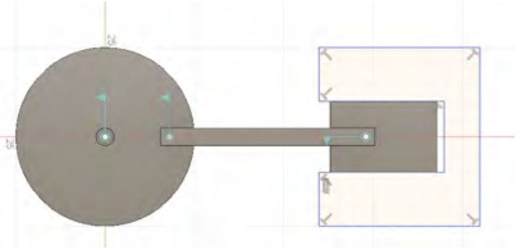
Type: Revolute

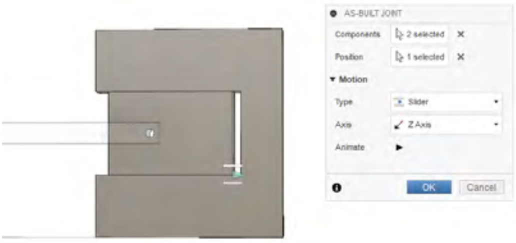
Axis: Z Axis

Animate: ▶

Right
Sketching allows more complex shapes to be produced

Below
The arrow on the slider joint indicates the direction of motion





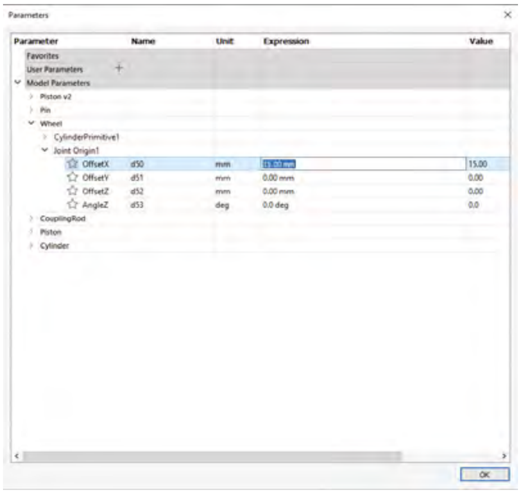
AS-BUILT JOINT

Components: 2 selected ✕
Position: 1 selected ✕

▼ Motion

Type: Slider
Axis: Z Axis
Animate: ▶

OK Cancel



Parameters

Parameter	Name	Unit	Expression	Value
Favorites				
User Parameters +				
Model Parameters				
Piston v2				
Pin				
Wheel				
CylinderPrimitive1				
Joint Origin1				
OffsetX	d50	mm	15.00 mm	15.00
OffsetY	d51	mm	0.00 mm	0.00
OffsetZ	d52	mm	0.00 mm	0.00
AngleZ	d53	deg	0.0 deg	0.0
CouplingRod				
Piston				
Cylinder				

OK

Above
The Parameters menu allows models to be changed retrospectively

QUICK TIP

In the joint dialog, component2 is the parent and component1 moves in respect to that.

QUICK TIP

Components that are linked don't have to be touching.

To demonstrate our piston in motion, we can use a Motion Study. Select the joint in the middle of the wheel, slide the timer to 60s, and add a new point of 360°. You can slide the timeline left and right to see the motion or use the play option to watch it in motion.

For my initial design, the piston was completely leaving the cylinder as the wheel rotated. This can be fixed by selecting the Modify menu and Change Parameters. The parameter to change for this is the X Offset on the joint Origin for the Wheel. Reduce this value and return to the Motion Study to see the effects.

> Reducing the speed via gearing also **increases the torque at the output**

GEARS

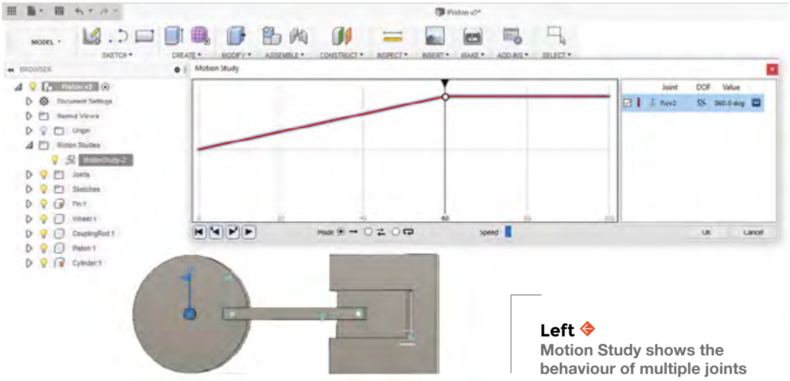
Gears can change motion in two ways. They can reverse the direction of rotation, and they can change the speed of rotation. The ratio of the speed of the two gears can be calculated based on the ratio of the numbers of teeth.

Reducing the speed via gearing also increases the torque at the output. Gears can be chained by having multiple gears on one shaft. It is often possible to buy these double gears ready-made, which simplifies the physical construction of your project.

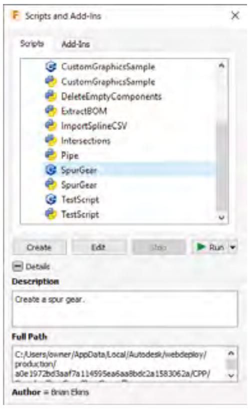
The geometry of gears is quite complex, and creating those by hand would be a challenge. To save time, there are gear generators already available in Fusion 360.

From the Add-Ins menu, select Scripts and Add-Ins… and scroll down the samples to the SpurGear by Brian Ekins. There is a C++ and a Python version of this add-in on the Windows platform, although the performance of both seems very similar – they have the same functionality.

You can experiment with different values and see how they affect the pitch diameter. The default module size seems to be aimed at industrial applications, so for models, you will find a value of 1 or 0.5 produces good gears. Note that you will see that not all values produce a valid gear. It also



Left
Motion Study shows the behaviour of multiple joints



Scripts and Add-Ins

Scripts | Add-Ins

- CustomGraphicsSample
- CustomGraphicsSample
- DeleteEmptyComponents
- ExtractBOM
- ImportSplineCSV
- Intersections
- Pipe
- SpurGear
- SpurGear
- TestScript
- TestScript

Create | Edit | Stop | Run

Details

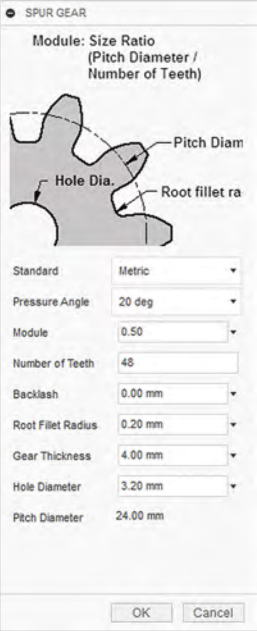
Description

Create a spur gear.

Full Path

C:/Users/owner/AppData/Local/Autodesk/webdeploy/production/a0e1972bd3aaf7a114595ea6aa8bdc2a1583062a/CPP/

Author = Brian Ekins



SPUR GEAR

Module: Size Ratio (Pitch Diameter / Number of Teeth)

Standard	Metric
Pressure Angle	20 deg
Module	0.50
Number of Teeth	48
Backlash	0.00 mm
Root Fillet Radius	0.20 mm
Gear Thickness	4.00 mm
Hole Diameter	3.20 mm
Pitch Diameter	24.00 mm

OK | Cancel

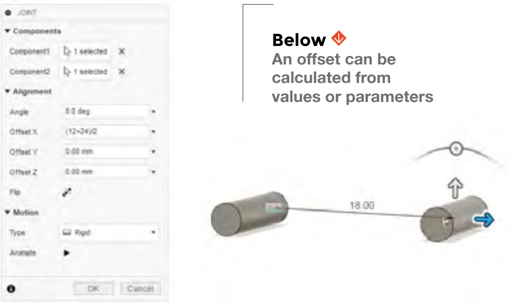
Above/Right
Fusion 360 can be automated using Python to create complex shapes like the SpurGear

isn't possible to change these values once the gear is created, so you will need to delete and recreate them for changes. When determining the hole in the middle, you want this to be a 'clearance' size for the selected shaft. So a 3 mm shaft would need a 3.2 mm hole. For 3D-printed parts, you might want to increase this further. Make a note of the pitch diameter, as you will need it at the next step. Repeat this with a second gear.

Once you have two gears, we can add some axles for them to run on. Create two cylinder components for the axles and use a rigid joint to fix their centres at the sum of the two pitch diameters divided by 2.

Mark both axles as ground to fix them in place. Add revolute joints so that the gears can spin on their respective axles. Adjust the positioning and the angle of the second gear so that the gears mesh correctly.

Fusion 360 can make one gear drive another using contact sets where the teeth push against one. This requires lots of calculations, so is a slow process. There is an alternative way: the motion link.



JOINT

Components

Component1	1 selected
Component2	1 selected

Alignment

Angle	0.0 deg
Offset X	(12+24)/2
Offset Y	0.00 mm
Offset Z	0.00 mm

Motion

Type	Rigid

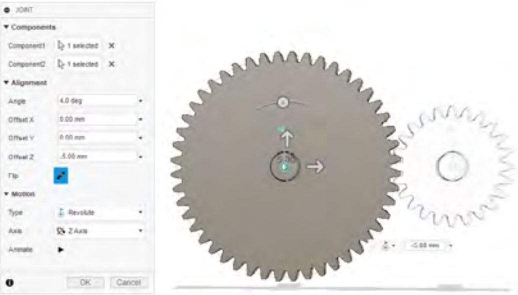
OK | Cancel

Below
An offset can be calculated from values or parameters

WHAT NEXT?

Now you've mastered the basics, you can try your hand at some more advanced projects. Here are a few ideas to get you started:

- Model a steam engine with sliding valves
- Model a car engine with multiple cylinders
- Use a motion link to show the motion of a roller or ball bearing
- Combine multiple gears in a compound gear train
- Cut the shape of a gear from a cylinder to make an internal gear



JOINT

Components

Component1	1 selected
Component2	1 selected

Alignment

Angle	4.0 deg
Offset X	0.00 mm
Offset Y	0.00 mm
Offset Z	-5.00 mm

Motion

Type	Revolute
Axis	Z Axis

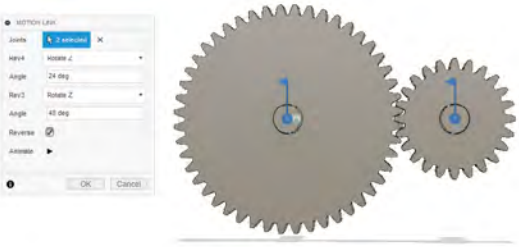
OK | Cancel

Left
The angle parameter controls the initial positions

A motion link describes how one joint relates to another. From the Assemble menu, select motion link. Select the two revolute joints created earlier. To adjust the ratio of the speeds, you can use the numbers of teeth for the number of degrees each rotates.

Fusion 360 is a parametric CAD system and so, by using those parameters, you can create models that can be used to experiment with motion. Joints link up the components and can be driven directly, via a motion study or indirectly through other joints or motion links. □

Below
Controlling the relative motion of two joints using a motion link



MOTION LINK

Joints	2 selected
Rev4	Rotate Z
Angle	24 deg
Rev3	Rotate Z
Angle	48 deg
Reverse	☑

OK | Cancel

QUICK TIP

For higher torque capability, increase the width and module of the gears.

QUICK TIP

Gear trains with no common denominator in the number of teeth will wear more evenly. Therefore, prime numbers are often used for the number of teeth.

FILM STORIES
£5.99. LIFE IS SHORT, THE WORLD IS WIDE...
THE UK'S BRAND NEW FILM MAGAZINE
DIRTY GOD
VICKY KNIGHT AND THE BREAKOUT BRITISH PERFORMANCE OF 2019
STARTER FOR 10 REVISITED
DIEGO MARADONA
HOW THE DOCUMENTARY WAS DONE
YESTERDAY
EXCLUSIVE: DANNY BOYLE, RICHARD CURTIS, HIMESH PATEL & LILY JAMES ON THE BIGGEST BRITISH FILM OF THE YEAR
WHERE'S BEST TO SIT IN THE CINEMA?
THE LOST TIMOTHY DALTON BOND MOVIE
WHO REPLACES FOX NOW THE DISNEY DEAL IS DONE?
PLUS
THE LATEST RELEASES REVIEWED
A NEW FILM MAGAZINE, FROM THE CREATOR OF DEN OF GEEK
JACK & SARAH
REVISITING AN OVERLOOKED 90S BRITISH HIT
CONFESSIONS OF A MOVIE PR REP
UNDERRATED MOVIE APOCALYPSES
PLUS
ANNA SMITH
DEBBIE MOON
HOPE DICKSON LEACH
ROMESH RANGANATHAN
SIMON BREW
ISSUE 2, FEBRUARY 2019
WIN
THREADS BLU-RAYS
MOVIE THEME PARK RIDES
CELEBRATING THE ROMCOM
(AND THE UNDERRATED ROMCOM STAR OF THE 90S)
PHYSICAL MEDIA
CAN THE HUMBLE DISC BE SAVED?
SCARY KIDS FILMS
WERE THEY SUPPOSED TO BE SCARY?
ISSUE 3
HAS CINEMA GOT ITS FUNNYBONE BACK?
ILLNESS: ITS IMPACT ON HOW WE FEEL ABOUT FILMS
FIGHTING WITH MY FAMILY
STEPHEN MERCHANT
FIGHTING TALK
12A CERT
BLAST
UNDERRATED 90S
EAST IS EAST, WILD WEST
& REPRESENTATION
CONFESSIONS OF A CINEMA MANAGER
THE INCREDIBLE, OVERLOOKED BOX OFFICE RUN OF BETTE MIDLER
AUTISM IN THE MOVIES
FEATURING
ROMESH RANGANATHAN
DEBBIE MOON
HOPE DICKSON LEACH
ANNA SMITH

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

REVIEWS

DIRECT FROM SHENZHEN

Remote-controlled car kit

Build your own remote-controlled car – just add motors, and a car

By Ben Everard

@ben_everard

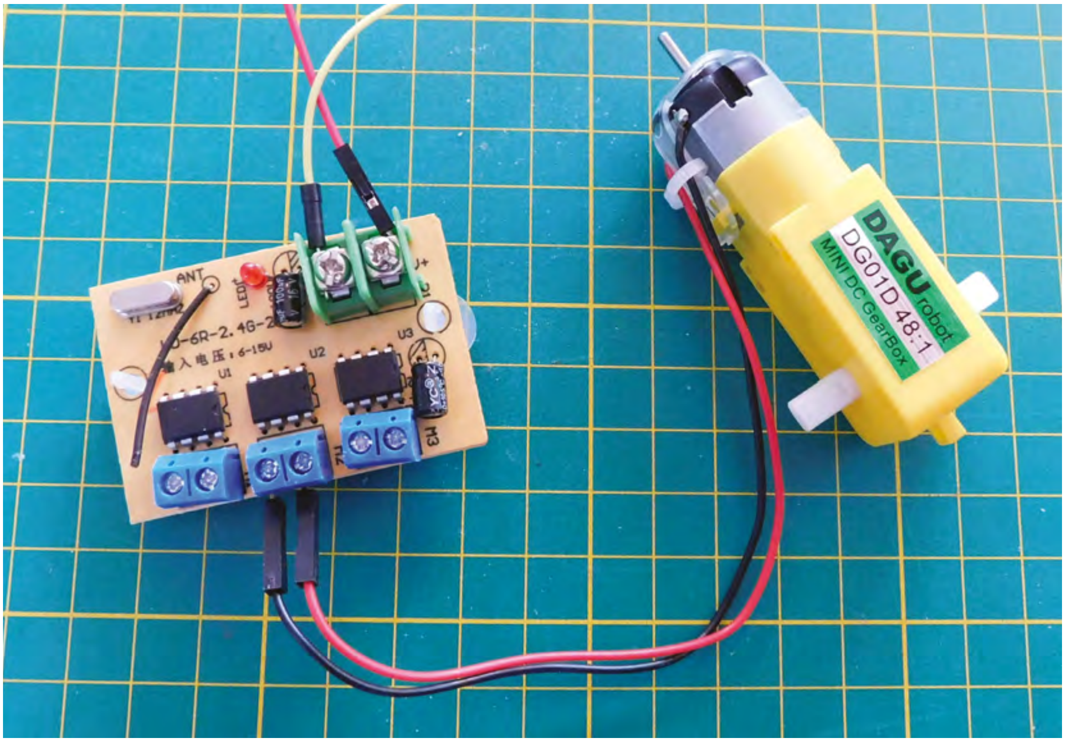
Ever wanted to build a remote-controlled car, or remotely control anything else? There are loads of kits out there, at almost every conceivable level, but we opted for the most bare-bones kit we could find – the snappily titled, '6CH High-power 2.4G 50 Meter Remote Control with Receiver 6-15V for Car Model Ship DIY Tool Remote Control Toys Parts' from Children's Interesting World Store on AliExpress. It's got a hand-held transmitter, receiver board, minimal instruction manual, and even a little screwdriver for attaching wires to terminals. This all came for £9.16, including delivery to the UK.

The kit includes a six-channel transmitter and receiver (that can control three motors in two directions each). These channels are controlled by two sticks (both with forwards/backwards motion) and two buttons.

Each channel is simply on/off, so there's no nuance in controls – you can go full speed forwards, full speed backwards, or stop. That's it. The kit didn't include any motors, but you can use any DC motors that take between 6V and 15V. The spec shows the board can work with up to 5A – though it's unclear if this is referring to the total or per motor. The motor drivers themselves (TS6586s) can each drive up to 9A, but they have a thermal shutdown at 130°C, which may stop them before this, depending on how enclosed they are, and whether or not they have a heat sink (they don't come with one, but this could be added). The PCB trace for the power is quite narrow (circa 1mm), but it is thickened with solder, which should increase its current carrying capacity, but it's hard to know by how much. How much current can they take then? That depends on how much you're willing to risk. If you're going to be pushing the limits on this, keep an eye on the temperature of the board, and don't let it get too hot.

There's not much assembly to do. You'll just need some batteries for the remote control (two AAs), and

Above The receiver board with motor (not included) attached



Left More or less what you'd expect from a remote control transmitter – wiggly sticks and buttons

a power source for the receiver board. We used four AAs, but anything that can provide between 6V and 15V will do, provided that it can supply enough current. There is a bit of a current spike when the board is first powered up, and we were unable to power it from our bench power supply as it drew more than 2A (this didn't last long, but was enough to trip the current protection). Beyond this, you just need some DC motors. Everything attaches via screw terminals.

> " It's easy enough for children to do ... **it's perfectly well suited to adults as well** "

SKIDDING ABOUT

With just on-off control, you're a bit limited by how to manage the steering. You could link up one of the pins to a microcontroller through an optoisolator, and use this to drive a servo mounted on a steering jig. However, if you're going to that level of effort, you might be better served by a more featureful transmitter setup. Our preferred choice for this is skid steering. Simply hook up one motor to the wheels on one side, and one motor to the wheels on the other. This way, you can turn your remote-controlled creation without any complex mechanical creation. If building in this style, all you need is two motors with appropriate gearing and suitable wheels (you can get these as all-in-one packages for a few pounds, though these won't be particularly quick), and something to act as a chassis – this could be as simple as an ice cream tub, or as complex as the Mars rover. Wire in the motors, attach a power source that can supply enough current, and you've got yourself a DIY remote- controlled car. It's easy enough for children to do, but we don't have to let little ones have all the fun – it's perfectly well suited to adults as well.

The build quality of this is about what you'd expect for the price. It's all a bit flimsy, and the terminals flex quite a lot as you tighten the wires in them. Provided it's handled sensibly, though, it should be strong enough for the job at hand.

We had great fun with this kit. There are much better ways of building remote-controlled vehicles. Ways that are more efficient, easier to control, more flexible, and generally better in almost any conceivable way, but what this kit has going for it is that it's cheap and easy to use. Pair it up with a glue gun and some scrap, and you can be driving about in a custom-built RC car in five minutes. The third motor gives you the ability to add a bit more than just steering. You could add a digger contraption for a child's toy, or almost anything else you can think of. There's no need to limit yourself to just the idea of vehicles – this could equally power a pan and tilt mount for a camera, or all manner of other remotely-powered doohickeys. It's basic, but sometimes basic is all you need. □

ONLY THE BEST

Organic light-emitting diodes for DIYers

Are OLEDs the future of displays?

By Marc de Vinck @devinck

In a world where visuals supersede any simple, text-based media consumption, it's no wonder that many DIY enthusiasts are looking to include high-quality displays in their projects. You typically have a few solutions for displaying images and graphics, some easy, and some can be fairly difficult.

LEDs have always been a popular feedback display for almost all electronics. But sometimes LEDs aren't enough, and they have very limited flexibility when it comes to what they can actually display. Next comes the seven-segment displays, such as an old alarm clock. Again, limited functionality, but at least you can display characters. If you want to display something more than just letters and numbers, you'll most likely move on to LCD and OLED displays. Although, if you are working with a Raspberry Pi, you can simply plug an HDMI cable into a monitor or TV, but sometimes that's overkill. And, if you are using a microcontroller like an Arduino, adding HDMI is a near impossibility.

That's where LCDs and OLEDs come into play. They can both be used in a variety of situations and are very cost-effective. They are also very well-documented, since they are used by so many DIY electronic enthusiasts. And, because of their popularity, we've already done one roundup of a few of our favourite displays, both LCD and OLED, in issue 4 of *HackSpace* magazine.

In this Best of Breed, we'll be looking specifically at OLED displays. The prices have come down significantly in the past years, and the quality is hard to beat. There are a lot of different displays out there. From low-resolution monochrome, to high-resolution full-colour versions with integrated microcontrollers. The displays are always very crisp in appearance, due to their high contrast, and often they also have vivid colours to match. With that said, when you need a small, high-quality screen for your next project, make sure to check out all the OLED displays available on the market. Let's jump in and take a look at a few of our top picks.

Above If you want to learn about the various displays available to DIY electronics enthusiasts, check out *HackSpace* magazine issue #04 (page 112) for reviews on some of the more interesting products available, including a flexible OLED from SparkFun

Adafruit Monochrome vs SparkFun Micro OLED Breakout

Our go-to boards for a wide range of projects

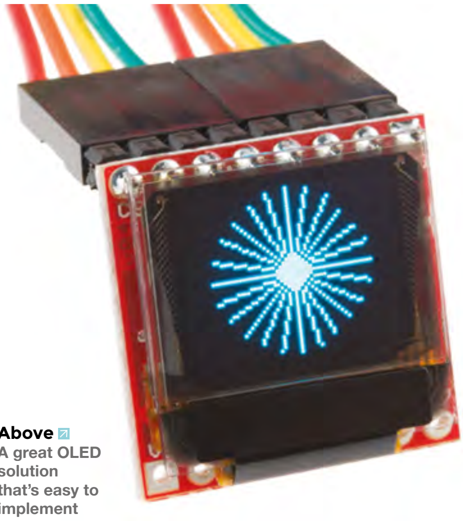
ADAFRUIT ◈ $17.50 | adafruit.com

SPARKFUN ◈ $15.95 | sparkfun.com

The Adafruit Monochrome OLED displays are small – really small. Measuring in at only 1" diagonally, you might think they're hard to read, and you'd be wrong. Thanks to the high contrast of the OLED, and the engineering behind this product, this little display is easy to read and even easier to use.

This display features a grid of 128×32 individual white OLED pixels on a black background. And, since the display makes its own light, you don't have to worry about a backlight that a typical LCD display requires. This reduces the power consumption and overall dimensions of the board. The driver for the OLED communicates via I²C only, which is fairly common and compatible with most microcontrollers and computers. You'll only need three pins to communicate with the IC in the OLED display. They have lots of tutorials, libraries, and projects to get you up and running fast. □

Below
This little display is so small, yet very easy to read



Above
A great OLED solution that's easy to implement

The SparkFun Micro OLED Breakout Board features a monochrome blue-on-black OLED. The display is crisp, as most OLEDs are, and despite its small size of only slightly over half an inch, you can still fit a surprising amount of data on it, including custom graphics and text. The interface can be either of the familiar SPI or I²C protocols, making it compatible with almost any microcontroller or computer.

It's a good solution when space is tight, and you want to display some custom sensor values. And, if you are familiar with their MicroView – see page 120 – then it may look familiar. Yes, it's the same display, but this breakout board doesn't include the built-in microcontroller or case. If you can fit your graphics into a 64×48 pixel grid, then this display might be a good fit for your next project. □

VERDICT

Adafruit Monochrome

Small and oh so sharp!

9/10

SparkFun Micro OLED Breakout

Affordable and easy to use

9/10

Pimoroni 1.12" Mono OLED Breakout

Plug and play on your Pi

PIMORONI ◆ $19.75 | shop.pimoroni.com

Right A simple and elegant solution for adding an OLED to your Pi

The Pimoroni 1.12" Mono OLED Breakout features a bright 1.12" OLED that's perfect for adding a small OLED display to your Raspberry Pi or Arduino project. It can display graphics or text on its monochrome 128×128 pixel display. There is plenty of room to display some useful sensor data, favourite icon, or text-based user feedback. Since it's an OLED, the screen is crisp and bright, making it ideal for conditions that require high contrast.

If you plan on using this with your Raspberry Pi, you can simply plug it right onto the header pins, once you have solder on the included right-angle female header pins. And, if you already use Pimoroni's Breakout Garden platform, you can simply plug this into one of the available slots and get coding. It couldn't be easier. □

VERDICT

Simple and easy to use

9/10

Crystalfontz 128×128 round OLED display

Thinking outside the box

CRYSTALFONTZ ◆ $20.85 | crystalfontz.com

Right Not all displays are rectangular

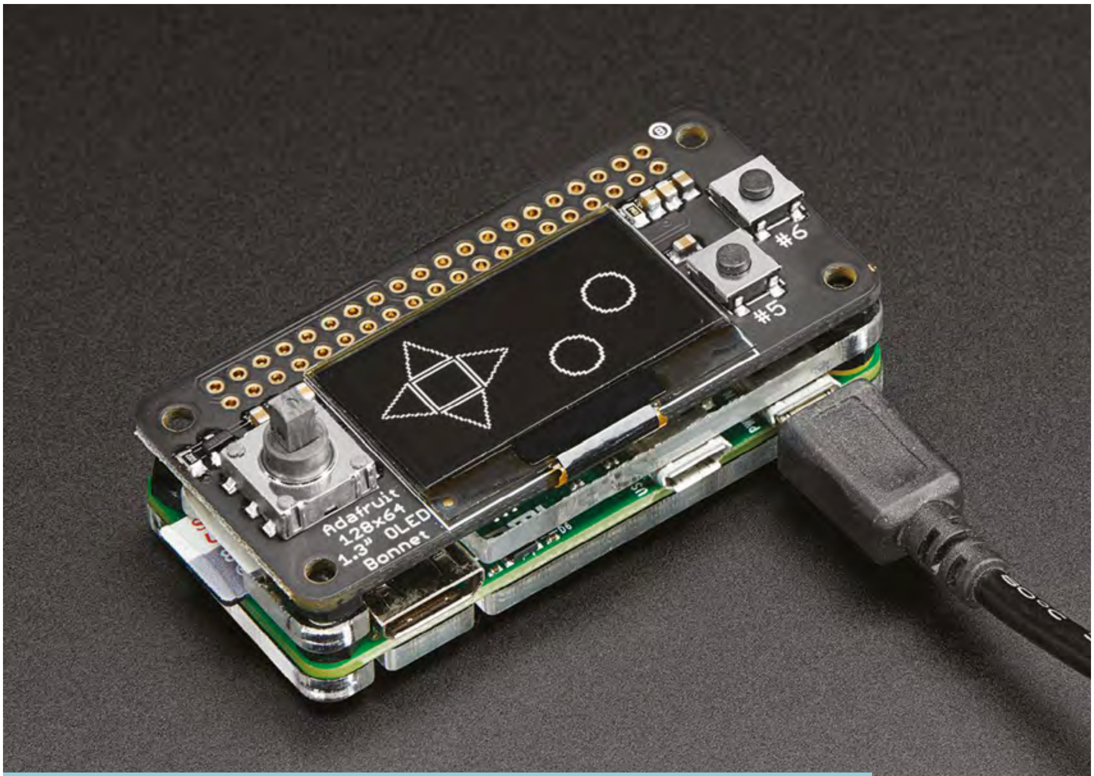
VERDICT

A great option when you need a round OLED

9/10

There is a lively community of DIY digital watch makers out there on the internet. We've seen everything from a simple LED face, to LCD versions, and even a few Nixie tube watches. But, up until now, most of them have had either a square or rectangular digital display. You rarely see a round display. Until recently, acquiring a round display of any kind was a bit challenging. Now, thanks to Crystalfontz, sourcing and using one isn't too difficult.

The round OLED screen is available at a reasonable price, and comes in both white on black, or yellow on black. The company supplies lots of information on how to use the display, including Arduino source code and a comprehensive datasheet. The screen features a 1.1" active display area, and communicates with your own controller via SPI or I²C. If you have ever dreamed of rolling out your own digital watch, then this is a must-have for your prototypes. □



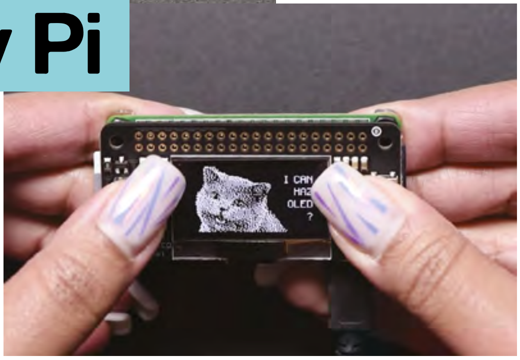
Left
A perfect OLED solution for your Raspberry Pi

Below
Yes, you can haz OLED!

Adafruit 128×64 OLED Bonnet for Raspberry Pi

Input and output

ADAFRUIT $22.50 | adafruit.com



The Adafruit 128×64 OLED Bonnet for Raspberry Pi is a really elegant solution for adding, not only an OLED display to your Pi Zero, or any flavour of Pi for that matter, but it also includes a five-way joystick and two buttons. Yes, we're all thinking about gaming at this point!

> "The OLED is not a replacement **for an HDMI monitor**"

The preassembled, and ready to use, bonnet features a 1.3″ 128×64 OLED display that fits perfectly on top of your Raspberry Pi Zero. Adafruit recommends using Pygame for drawing or writing text, and provides example code on its product page. This would make for a fun game controller where the OLED displays useful secondary information. Remember, the OLED is not a replacement for an HDMI monitor, but with the high contrast and ease of use, it would make for a perfect little feedback display. □

VERDICT

If you have a Feather board, this is your best choice for simple GPS

10/10

SparkFun MicroView

The screen with a programmable brain

SPARKFUN ◈ $42 | sparkfun.com

Left ◈
A polished-looking display for your next project

Calling the MicroView an OLED display is a bit misleading. This little bit of electronics is actually a fully-fledged Arduino-compatible board, that also includes a 64×48 OLED display. Just like all the other OLED displays, it's perfect for showing sensor data or most types of user feedback but, unlike most other products, this one includes an ATMEL's ATmega328P. It makes integrating a display with your Arduino code simple.

Keep in mind that to program the MicroView, you will also need an external programmer that is sold separately. If you are familiar with the Arduino programming ecosystem, then you will be up and running fast, as it's 100% code-compatible with the Arduino Uno (ATmega328P). The MicroView is a good solution for anyone seeking a more polished-looking device for their next project. □

VERDICT

Works well

8/10

Adafruit OLED Breakout Board – 16-bit Color 1.27"

Now in glorious colour

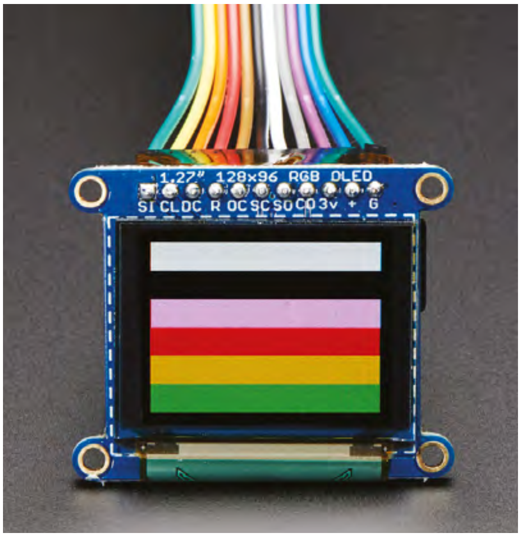
ADAFRUIT ◈ $34.95 | adafruit.com

Right ◈
Great contrast and bright colours make this a great OLED display

LEDs are monochrome, but not this Adafruit OLED Breakout Board. It features a 1.27" diagonal OLED display, with a grid of 128×96 RGB pixels. The pixels have a 16-bit resolution, allowing for a theoretical tonal range of 65,536 unique colours. And, since it's an OLED and does not require a backlight, the blacks are really black, and the contrast is very high. Despite its small size, you get a lot of clarity and functionality.

And, there is a nice surprise when you flip the board over. A built-in microSD card holder. This allows you to access an image from a microSD card, and display it via an SPI interface. Simple!

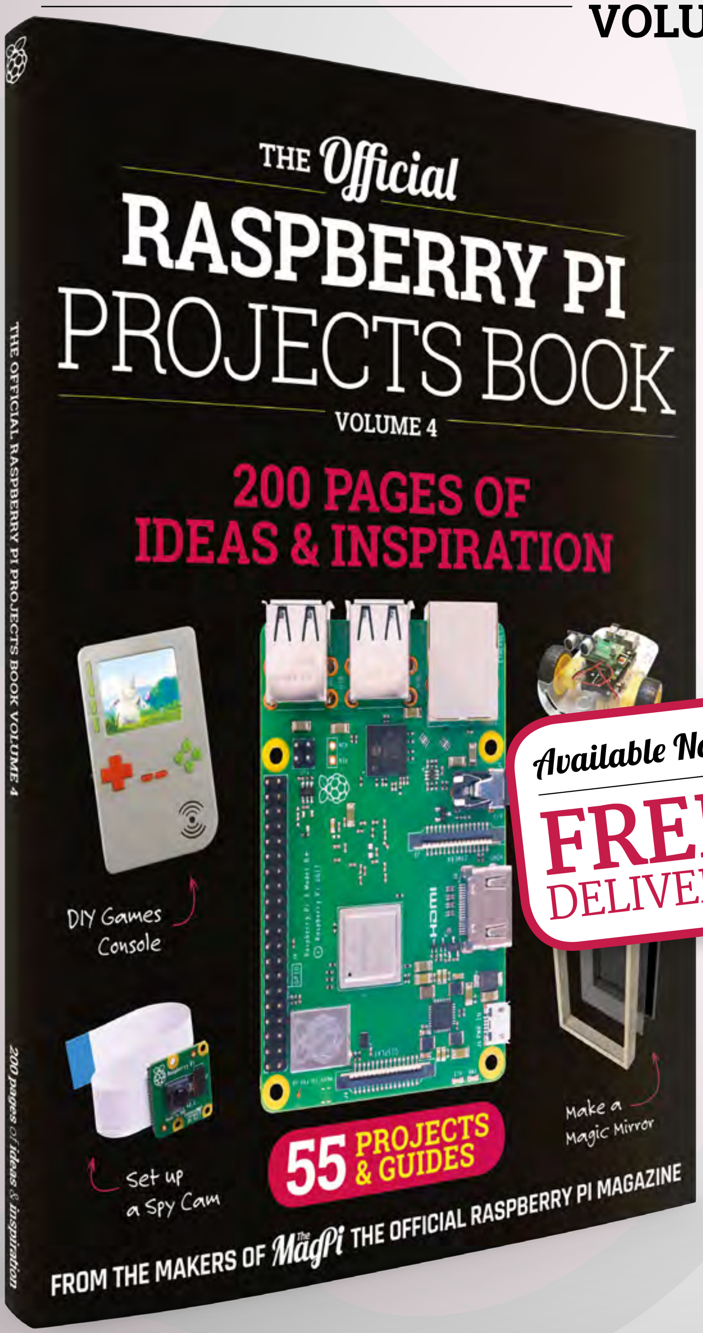
And, in typical Adafruit fashion, they have a comprehensive guide that walks you through all the things this board is capable of doing. Our favourite project that uses this board is the Electronic Eye using a Teensy, by Phillip Burgess. Halloween will be here before you know it! □



VERDICT

Vivid colours and high contrast that's tough to beat

10/10



THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 4
200 PAGES OF
IDEAS & INSPIRATION
DIY Games Console
Set up a Spy Cam
Make a Magic Mirror
55 PROJECTS & GUIDES
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 4
200 pages of ideas & inspiration

Can I Hack It?

A bubble machine?

Can we hack a bubble machine to be controlled over the internet?

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**

Are you tired of blowing your own bubbles? Do you yearn for the Internet of Things (IoT) to provide you with bubbles on demand? Well, your automated bubble blowing days are here! For just £2, we have found an extremely hackable bubble machine that can be used with Raspberry Pi, Arduino, ESP8266, etc. to give you bubbles on demand. So let's take a look inside it and see what we can hack!

Coming in a two-part plastic frame, the bubble machine is made from easily worked plastic. In fact, the plastic is a little thin in places, so take care! Measuring 185 mm tall, 130 mm wide, and 75 mm deep, there is plenty of space for your creation.

The frame is held together with six cross-head screws, all of equal length; when they are removed, the front and back sections are clipped together with small connectors around the perimeter. When unclipped, the two frames are tenuously held together by wires from the battery compartment. In fact, when we pulled ours apart, we broke one of the wires, but it was easy to solder back on. Inside the bubble machine we have another plastic section, a trough that holds the bubble solution, and a top cover. This provides a barely adequate level of water resistance, but some hot glue would be useful here.

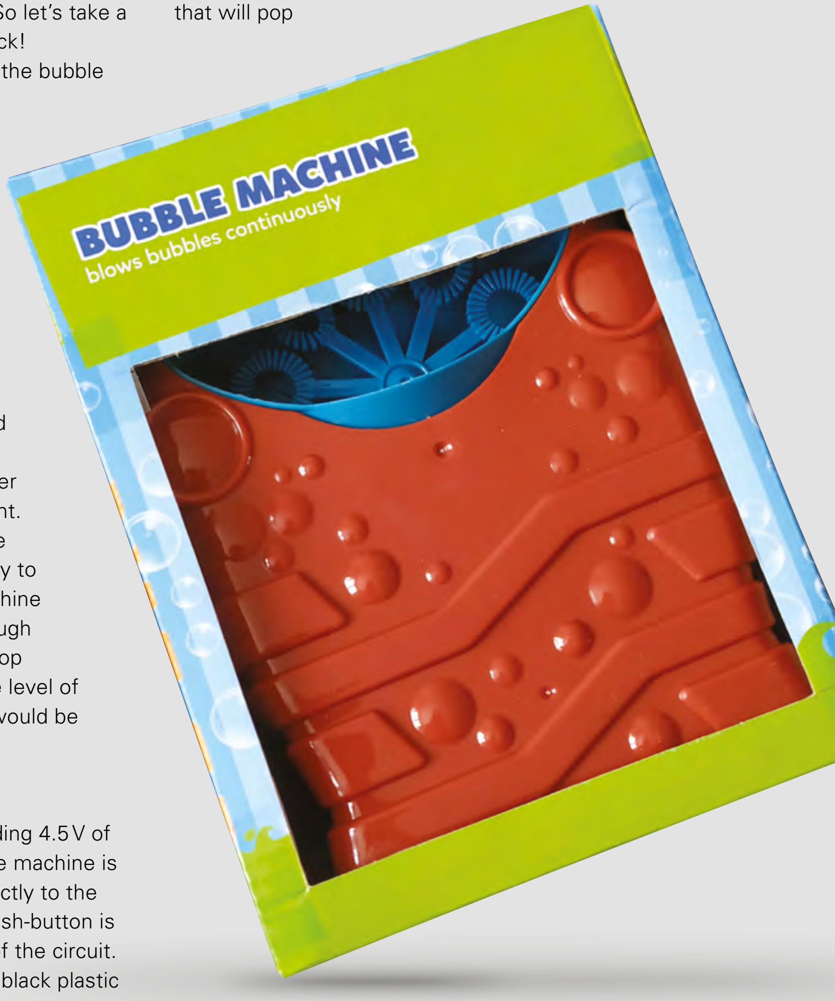
ELECTRONICS

Powered by three AA batteries, providing 4.5 V of power for the on-board DC, the bubble machine is electronically simple. Power goes directly to the motor, but along the way, a locking push-button is used to close and open the Vcc side of the circuit. The DC motor is contained in a small, black plastic housing inside the front frame. There are two cross-head screws to remove before the motor is free of the frame. You will also need to remove the circular 'wand' used to blow bubbles, as this is connected to the motor via a hexagonal shaft. A third screw is located on the reverse – take care when unscrewing, as the DC motor has a number of gears that will pop

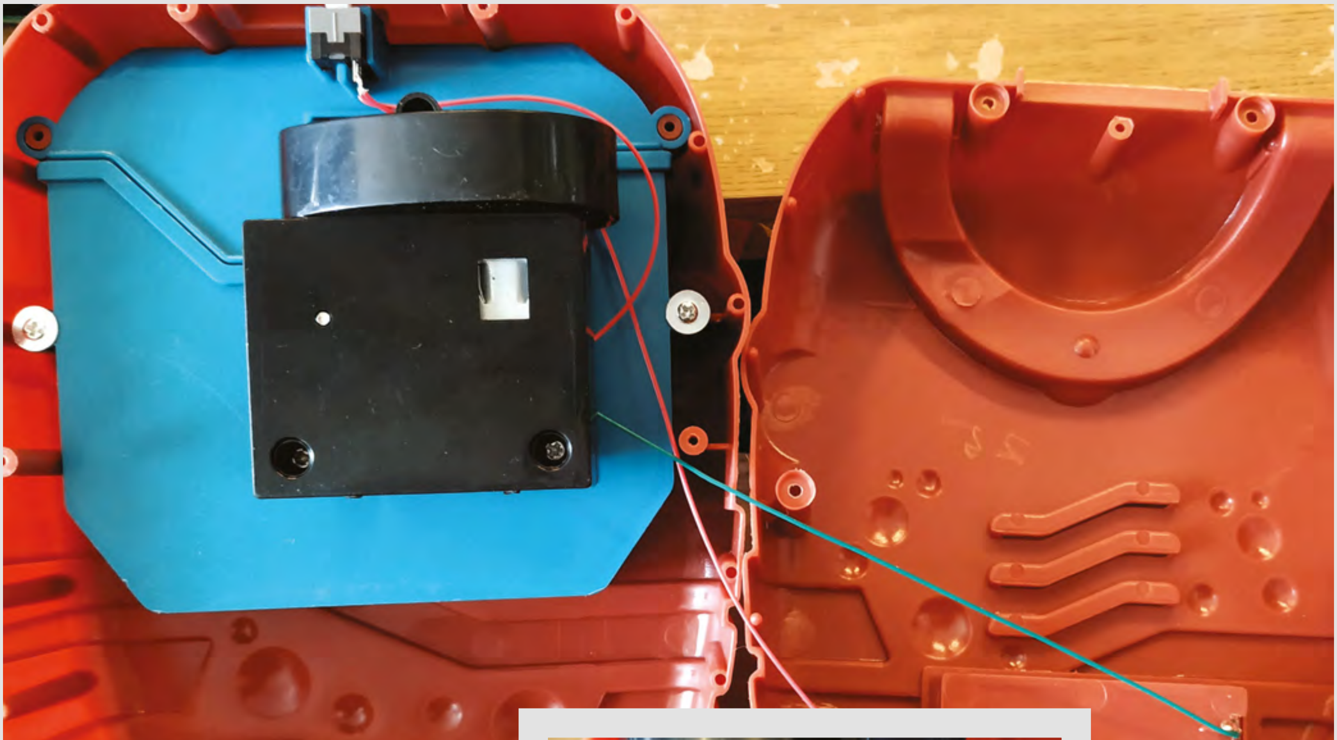
YOU'LL NEED
Bubble machine

COST
£2 from Poundland (UK)

WHERE
Alternatives available via amazon.co.uk



Right Inside this unassuming cardboard box lies your introduction to the Internet of Things

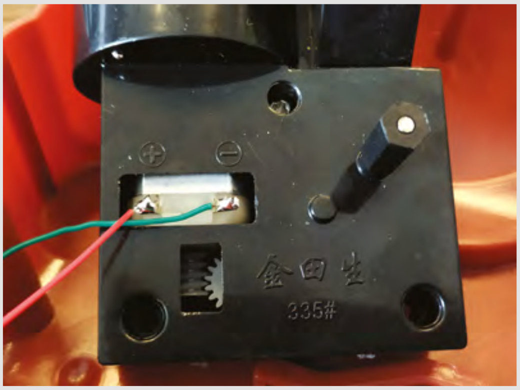


out once opened. The DC motor is a generic cheap DC motor, like those found in really cheap motorised toy cars. In fact, when the motor is running, you can smell the faint 'burning' smell of it, just like those toy cars.

HACKABILITY

For £2, this is an awesome cheap project to hack. Firstly, we have so much space inside the case. There is enough space for a Raspberry Pi Zero W and a motor controller – we tried the L298N controller to directly control the DC motor. All we needed were a few wires from the Pi to the L298N and some power. Sure there is a 3 × AA battery holder on the board, but we chose to make a small hole in the battery box and feed in a 5 V, 2 A power supply for the Pi and motors. It was then just a case of controlling the motors using the GPIO of the Pi via the L298N.

This is an ideal introduction to IoT-based projects. Sure, the plastic frame is a little thin, and the 'waterproofing' needs a little hot glue to ensure your electronics survive. But you have an ideal cheap entry to IoT projects. The cavernous space inside the machine means we can hide everything away to create a simple-looking device. The on-board battery pack provides 4.5 V, which can be used with a voltage booster (like Adafruit's PowerBoost board) to provide 5 V for your chosen board and motor controller. Time to get hacking! □



Above
The two-part case means we can easily gain access to the spacious insides and the electronics within

Left
The motor is used to spin the wands that will hold our bubbles, and to power a small fan that will blow the bubbles

NITTY-GRITTY

The Internet of Things has been with us for some time now, but you would be forgiven if you thought that it was not within your reach. We need data centres, clouds, and top coders, right? Well, no, we don't.

Take our bubble machine for example. We need just a Raspberry Pi Zero W computer, which costs $15. Then we need a motor controller which, for an L298N, is around $3 online. A 5 V, 2 A power source is next, which retails for around $10 for a good supply. Then we need to wire the L298N to the 5 V power supply, and also power the Pi Zero W from the same supply. Connect the two GPIO pins on the Pi to the inputs on the L298N, and then connect the outputs of the L298N to the motor terminals. A quick test in Python to turn the GPIO pins on and off will confirm that the wiring is complete. Then, we just need to trigger the machine to life with IoT – for that, the simplest route is Twitter. We used Node-RED to make a simple app that scans Twitter for our chosen hashtag; when it sees the latter, it will turn the correct GPIO pins on and, in turn, spin the motor for ten seconds, before returning to wait for the next trigger. The trigger can be Twitter, MQTT messages, HTTP POST, etc., but all of it can be done with a Raspberry Pi and under $30 of equipment!

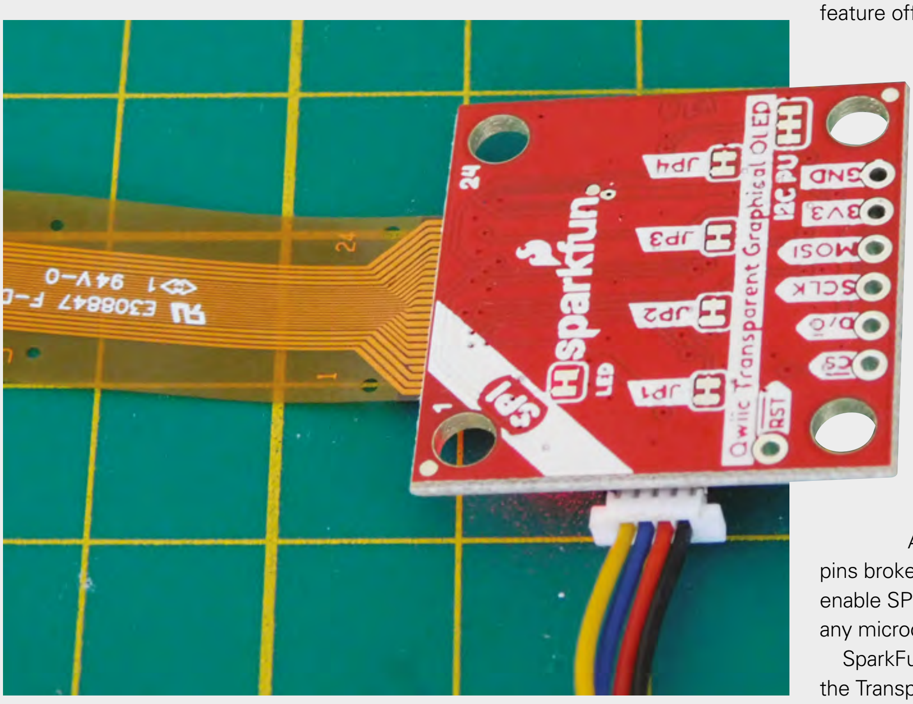
SparkFun Transparent OLED

For all your see-through project's needs

SPARKFUN ◆ $39.95 | sparkfun.com

By **Ben Everard** @ben_everard

Organic light-emitting diodes (OLEDs) are LEDs created by sandwiching a layer of organic compound between two electrodes. Unlike LCD displays, OLEDs themselves emit light so don't need a backlight – they are really a matrix of lots of tiny lights. In this display, all three parts of the OLED are transparent, so, when everything is powered off, the display looks like glass. Slightly grey glass (if you look closely, you can see the grid of LEDs), but it's transparent enough for most uses.



Below ◆ The control board has connectors for Qwiic, I^2C, and SPI

There are 8192 LEDs, arranged in 128 lines of 64 pixels, packed into a 35.5 mm × 18 mm screen. This glass screen is attached to the breakout board via a flexible connector that allows it to be mounted in tight spaces.

Power it up and send the right signals over (more on this later) and you can turn these LEDs on to draw lines, text, and images. They're a bright blue-green (you can't control their brightness, it's just on/off) and stand out easily even in daylight, with or without a light source behind them. With some of the pixels lit, the rest of the display remains transparent – the SparkFun announcement video for this project shows this feature off as an interactive Nerf gun-sight that can add extra info, as well as crosshairs, to your foam dart guns.

The screen part of the unit is just the bare glass – there are no mounting holes or other concessions to make it easy to fix to a project. Presumably, this is done to keep it as transparent as possible, but this means that the user will have to find a way of fitting it in their project, whether that's adhesives or creating a mount to hold it.

CONNECTIONS

The easiest way of hooking it up to your microcontroller is via the Qwiic connector. This lets you hook it up to any compatible microcontroller without any soldering. Alternatively, there are also I^2C and SPI pins broken out (you'll need to cut a copper trace to enable SPI), and from there it can work with almost any microcontroller.

SparkFun has created a library to make it easy to use the Transparent OLED with Arduino (other languages are not supported at this point). This works with the HyperDisplay library, which gives you high-level drawing primitives. For example, the following code draws a line and a circle (in a previously created object called **myTOLED**).

```
myTOLED.lineSet(x0, y0, x1, y1);          // Draw a
simple line from (x0, y0) to (x1, y1)
myTOLED.circleSet(x0, y1, r);  // Outline circle
centered at (x0, y0) with radius r
```

For more free-form images, SparkFun has created a Python script to convert JPEG images to primitives that can be displayed. This outputs a series of pixels that can be turned on, creating the drawing. Obviously, this compresses colour and brightness into a simple on/off value, so not all JPEGs will end up looking good.

Following through the learning guides, we were able to get up and running quickly. There's nothing particularly complex in either the wiring or programming of these displays, and it's all well documented – including using the jumpers to change the I²C address or to set up SPI. We wouldn't expect anyone familiar with the Arduino IDE to have any challenges getting up and running with this display.

IN USE

The performance of the screen is best described as 'alright'. Adding new primitives is quick, but blanking and then redrawing the screen comes with a noticeable delay. It works well for displaying text and static images. Animations can work well if you keep things simple, but if you need to display a lot of primitives (such as images made up of dots), performance can struggle. Switching to use SPI rather than the default I²C does help with this.

> " **These transparent screens feel wonderfully futuristic**, and they're the sort of technology that enable whole classes of projects "

These transparent screens feel wonderfully futuristic, and they're the sort of technology that enable whole classes of projects that simply aren't possible without them. Perhaps the most impressive use we've seen of this so far is ten of the boards stacked on top of each other to create a volumetric 3D display (**sparkfun.com/news/2885**).

This screen needs a little creative thought to come up with good uses for it, but we're sure there are more great projects out there waiting to be made, whether that's cosplay effects, info displays, or transparent cases that show off the whole of your project's electronics, or something else entirely. □

Left The screen remains transparent even when images are displayed

VERDICT

This screen opens up some really striking visual effects.

9/10

Open-source loom

Weave your own fabric

TONI BUCKBY ◆ Free | theinterlaceproject.com

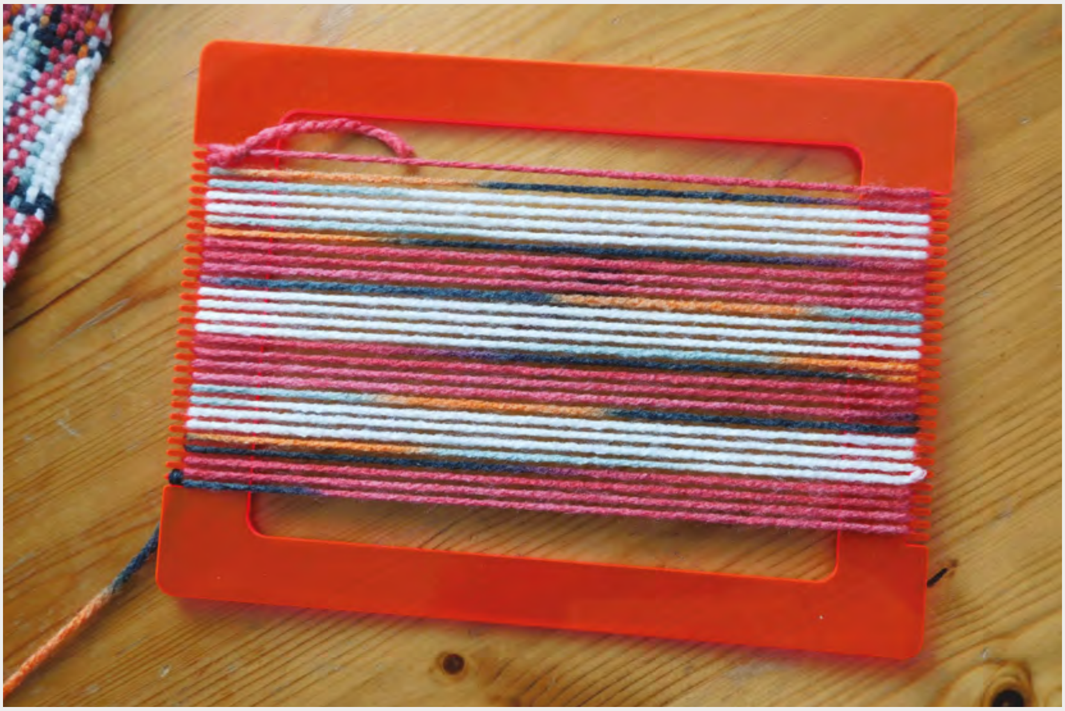
By Ben Everard

@ben_everard

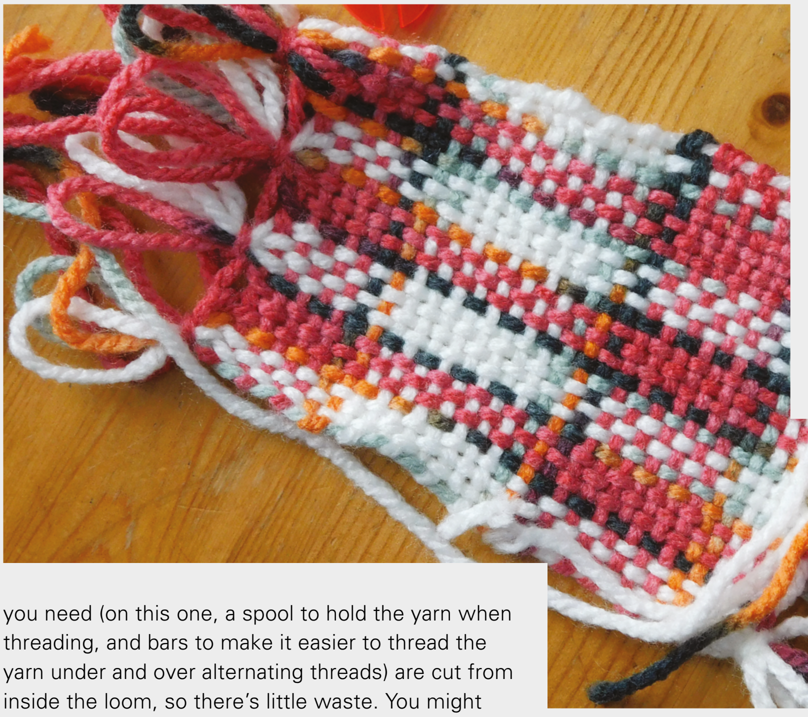
As part of Derby Museum's 'Derby Silk Mill Museum of Making' project, artist Toni Buckby designed and shared two weaving looms: a simple frame loom and a more complex heddle loom. The frame loom is entirely laser-cut from any 3 mm material (we opted for acrylic, while the designer used MDF – it might be worth avoiding plywood as any splinters may snag on the yarn). The heddle loom is a bit more complex and needs a few bits that can't be laser-cut, such as bolts and dowels.

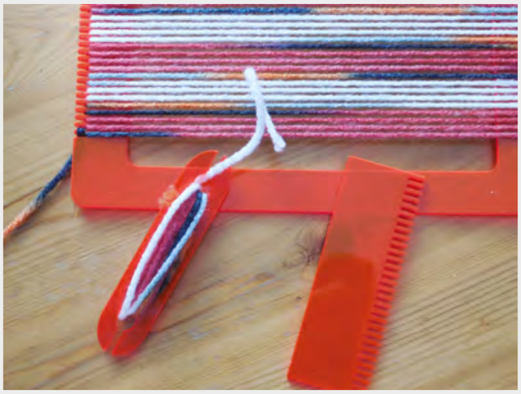
As we're weaving novices, we focused on the simpler and easier frame loom. The designs come as SVG files. Depending on your laser-cutting software, you may need to change the colours or size of the lines, but they're easy to work with, and we had no trouble cutting them out.

The frame loom consists of a hollow square with teeth along the top and bottom, as well as associated bits for weaving. In this case, the bits



Right
Threaded up and ready to weave





Above The design includes a shuttle for threading the warp, and the comb for keeping the weave tight

Left A finished square of fabric – the pattern comes from the colour changes in a single thread of yarn

you need (on this one, a spool to hold the yarn when threading, and bars to make it easier to thread the yarn under and over alternating threads) are cut from inside the loom, so there's little waste. You might need to set your cutter to cut these out first as the material may move slightly if the outer frame is cut first.

Weaving with the loom is all fairly straightforward. Some yarn is threaded over the comb sections on the top and bottom of the square to create a series of parallel threads (this is the warp in weaving terminology). Some more yarn is wound around a shuttle and threaded under and over the warp (this bit of yarn is known as the weft) to form a line of weaving. The weft is threaded back and forward to gradually build up the fabric. Once you've filled the frame with fabric, you can cut it out and tie up the loose ends of the warp to stop it coming undone. These small squares of fabric can be sewn together to make larger items, or simply used independently (such as for drinks coasters).

> **Of course, your weaving doesn't have to** just be about aesthetics

You can change the yarn you use in the weft to make patterns in the fabric – adding different colours and textures of yarn can lead to startling effects in the weave. For more advanced patterns, change the way you thread the weft through the warp. By going under and over in orders other than 'under one over one', you can build up designs in the weave of the fabric itself. There's not much limit to the complexity of pattern you can weave – if you're willing to take the time. There's an active weaving community on YouTube which should help you get started and build up to more advanced techniques.

Of course, your weaving doesn't have to just be about aesthetics, and the Interlace Project have experimented with weaving in conductive thread alongside the yarn, to make touch-sensitive areas of fabric. There could well be more opportunities to incorporate technology into the actual substance of the material itself, rather than creating wearable electronics by adding components to pre-existing materials.

SPEED BUMPS

While this loom is simple, it's quite time-consuming to weave because you have to thread the warp over and under each thread of weft every time you take it across the fabric. The more complex heddle loom has a block that raises or lowers alternate threads of weft to make it much quicker to thread the weft. This loom also allows you to make longer pieces of fabric as it is rolled up as it's woven. The downside of the heddle loom is that it's a more complex loom to build and set up.

While the frame loom is quite small, and you'd struggle to build up a large amount of material with it, it's easy to make and gives you the chance to learn about weaving and experiment with patterns. You shouldn't have much trouble expanding the design to a larger frame to make larger weaves, but you might be better served by the heddle loom. □

VERDICT

A fun little laser-cuttable project to help you learn about fabric making.

9/10

Parkside battery-powered soldering iron

Electronics on the go

PARK TOOLS ◆ **£9.99** | **lidl.co.uk**

By Ben Everard @ben_everard

This portable soldering iron appears periodically in the Lidl supermarket chain. At £9.99 for a rechargeable iron, it seems like a great offer, so we decided to find out if it actually performs well enough to be of any use.

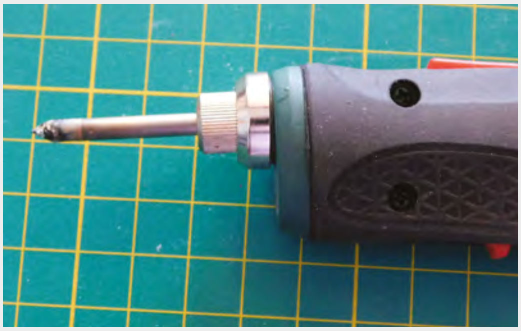
The iron itself is a minimal affair with a conical tip, an 'on' switch, and a trigger to power it. When the trigger is depressed and it's switched on, the tip heats up (this means that it automatically powers off when you put it down). There's no temperature control, other than what you can manage flicking the trigger (but you're unlikely to want it much cooler than its maximum anyway). It comes with integrated rechargeable battery, and even a small amount of solder, so you've got everything you need to start work in the box.

Included in the box is a charging dock, which holds the tip safely and slots a charger into the micro USB port. We do have a few concerns about the strength of the micro USB adaptor on the charging port but, if this becomes damaged, you could charge the iron from any USB charger.

For the most basic soldering, this iron works – it takes a little time to get up to temperature, but once it does, it works perfectly well for small jobs. However, if you come across anything remotely tricky, you're likely to get into problems. The low power of the machine (no wattage is stated) means that it struggles to heat up anything more than thin wire – we found we couldn't solder the leads of four 1 A diodes together – and the blunt conical tip isn't our favoured choice. The tip connects with what looks like an old-fashioned, screw bulb connector and, while replacement tips are available, they're only available in this blunt-cone profile.

The battery lasted for several jobs between recharges (and you could pair this with a USB power bank, if you need longer between recharges).

If you're after a cheap iron to do simple repairs while out and about, this can do the job well – provided the repairs are connecting thin wires, or simple through-hole work. We've found it a genuinely useful addition to our toolbox (which previously only included wired irons). However, you will come across joints that this iron simply can't handle, so it's only suitable as a back-up iron to be used when convenient, then put away when more difficult jobs need to be done. For £9.99, we feel that this is an acceptable compromise. □



Above ◆ The iron and charging dock are compact and easy to transport

Below ◆ The blunt conical tip isn't easy to get into tight spaces

VERDICT

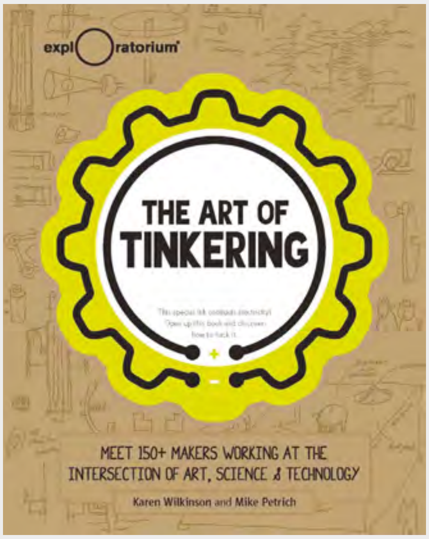
Cheap, cheerful, and useful for on-the-go repairs, but too limited to be your only iron.

7/10

The Art of Tinkering

Karen Wilkinson & Mike Petrich ◆ $32.50 | exploratorium.edu

By **Ben Everard** @ben_everard


explOratorium
THE ART OF
TINKERING
MEET 150+ MAKERS WORKING AT THE INTERSECTION OF ART, SCIENCE & TECHNOLOGY
Karen Wilkinson and Mike Petrich

STEAM – Science, Technology, Engineering, Art, and Maths – is a sometimes controversial acronym. Does art have a place among the other technical subjects? Here at *HackSpace* magazine, we feel that it should. A simple LED circuit includes science, technology, engineering, and maths, but add in some art and you could have sewable patches to adorn your clothes, a word clock, or all manner of other interesting and beautiful objects.

The Art Of Tinkering is an exploration of the work of over 150 makers who are bringing their artistic touches to the world of technology. While there are some short guides to help you get started with some of the areas looked at, such as light painting, paper circuits, and building with wire, the majority of the book is looking at the work of the featured makers. These makers cover a vast area of the world of makers, including puppet-makers, circuit benders, photographers, and much more. If we were being picky, we'd say that we'd like to see a bit more coverage of electronics in there. There's plenty of simple bits in it, with LEDs, motors, servos, etc., but nothing covering the arty side of electronics – the badgelife scene, or the designs of Boldport, for example. However, that's just our personal preferences coming through.

The featured creations are beautifully photographed, and this book is so packed full of inspiration, it's hard to put down. It's one that will stay on your coffee table to be flicked through in idle moments. □

VERDICT

Maker inspiration packed up inside paper and put on a shelf.

9/10

next month

issue #21

ON SALE 18 JULY

GET THE MOST OUT OF YOUR

RASPBERRY PI

ALSO

→ TOOL-MAKING

→ LORAWAN

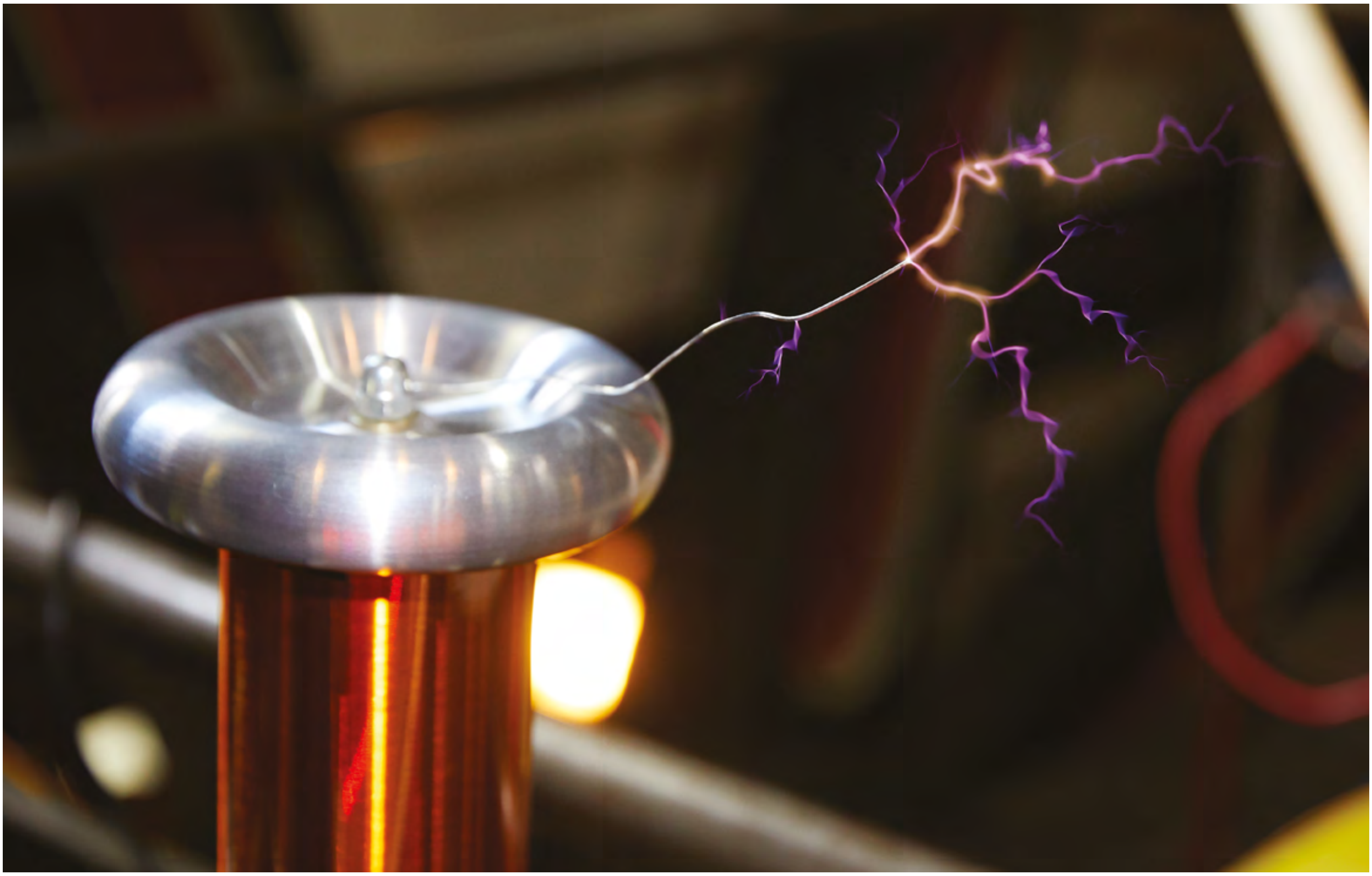
→ ARDUINO

→ MONITOR AIR POLLUTION

→ AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe



Tesla Coil

In 1891, Nikola Tesla designed this iconic circuit that bears his name. It can create huge voltages (sometimes over a million volts) that ionise the surrounding air and allow electrical discharge. While early Tesla coils were important tools for physics research and radio transmission, these days, their use is mainly for entertainment.

— hsmag.cc —

goBILDA®

DREAM • MAKE • REPEAT

WHEELS FOR ANY PROJECT

TERRAIN

OMNI

RHINO

DISC

MECANUM

AVAILABLE AT goBILDA.COM
sales@goBILDA.com · 620.705.5959

BROUGHT TO YOU BY THE
CREATORS OF **ACTOBOTICS®**