MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsm mac | June 2022 | Issue #55

Vinyl cutting

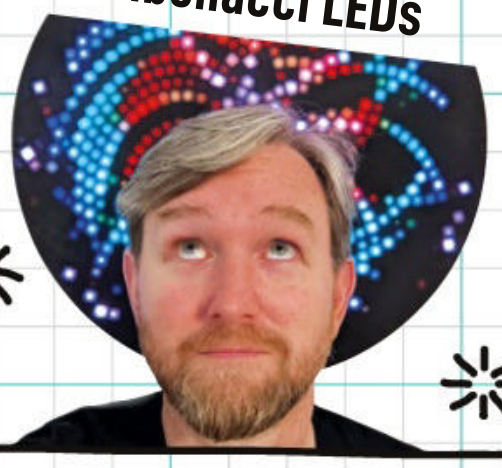
Get started with a Cricut EasyPress

BUILD A PLOTTER

make a robotic drawing machine with Raspberry Pi Pico

Jason Coon

The Evil Genius of Fibonacci LEDs



Lube

Everything you wanted to know but were afraid to ask

Hydraulics

Move mountains with water power

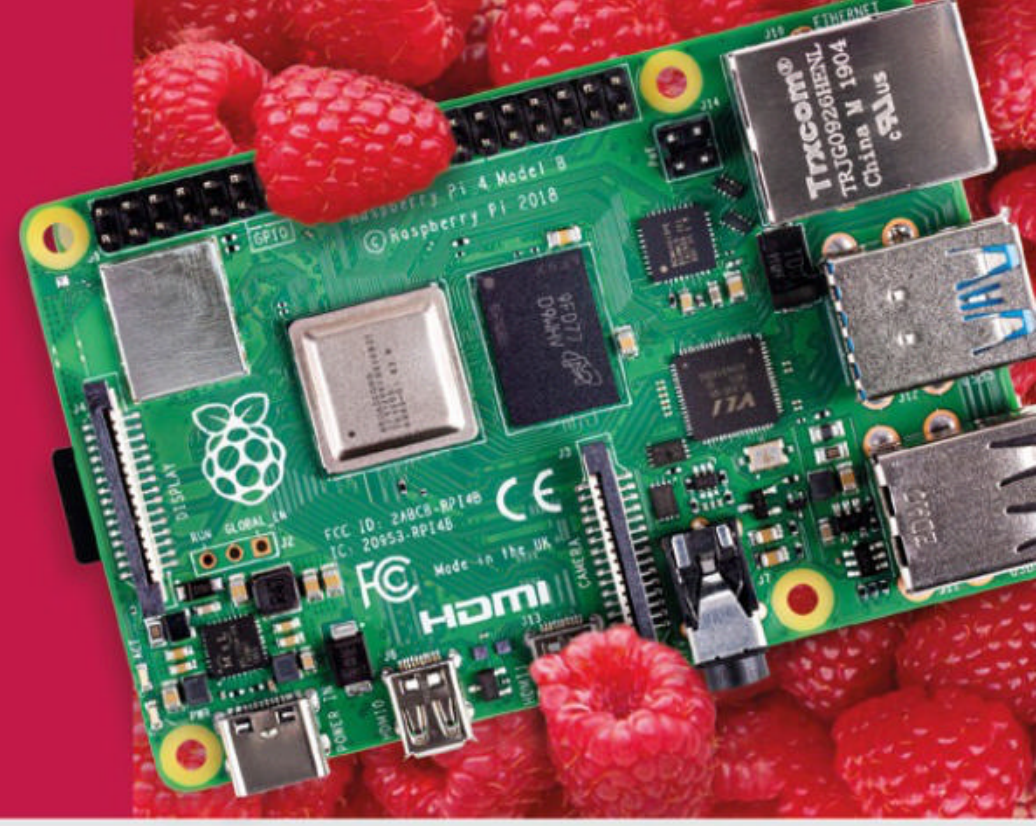
Jun. 2022
Issue #55 £6

9 772515 514006 55

Raspberry Pi PRESS

GUITARS / LASERS / 3D PRINTING / ARDUINO

American **Raspberry Pi** Shop



- Displays
- HATs
- Sensors
- Cases
- Arcade
- Swag
- Project Kits
- Cameras
- Power Options
- Add-on Boards
- Cables and Connectors
- GPIO and Prototyping

Partner and official reseller for top Pi brands:

adafruit · sparkfun ELECTRONICS · PIMORONI · HiFiBerry · and many others!

Price, service, design, and logistics support for **VOLUME PROJECTS**

USA **PiShop.us**

Canada **PiShop.ca**

Raspberry Pi APPROVED RESELLER

Welcome to HackSpace magazine

Plotters are very definitely machines, but they do something that is very human: draw. Watching a plotter in action – particularly a vertical plotter with its less obvious motors – makes you feel like the machine is thinking, maybe even feeling. It's not, of course. It's following a set of programmatically generated instructions, but that's not the point. It's not about what it is, it's about how we perceive it. We developed this plotter in Bristol Hackspace, and every time we had it running a picture, people passing by would stop to watch. It's just that sort of machine.

Watching a plotter in action – particularly a vertical plotter with its less obvious motors – makes you feel like the machine **is thinking, maybe even feeling**

Nobody needs a plotter – it won't solve a problem you have, but it is just fun. It's fun to tinker with, it's fun to watch, and it's fun to look at the end result. That's what drew [ahem] us to this machine, and that's why we tried to create one that you could build at home. Turn to page 32 to find out more.

BEN EVERARD

Editor ben.everard@raspberrypi.com

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

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.com

hackspacemag

hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
Nicola King, Phil King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Olivia Mitchell, Ty Logan

Photography
Brian O'Halloran

CONTRIBUTORS
Jo Hinchliffe, Marc de Vinck, Andrew Lewis, Mike Bedford, Nicola King, Rosie Hattersley, Phil King

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

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

recycle

CC BY NC SA

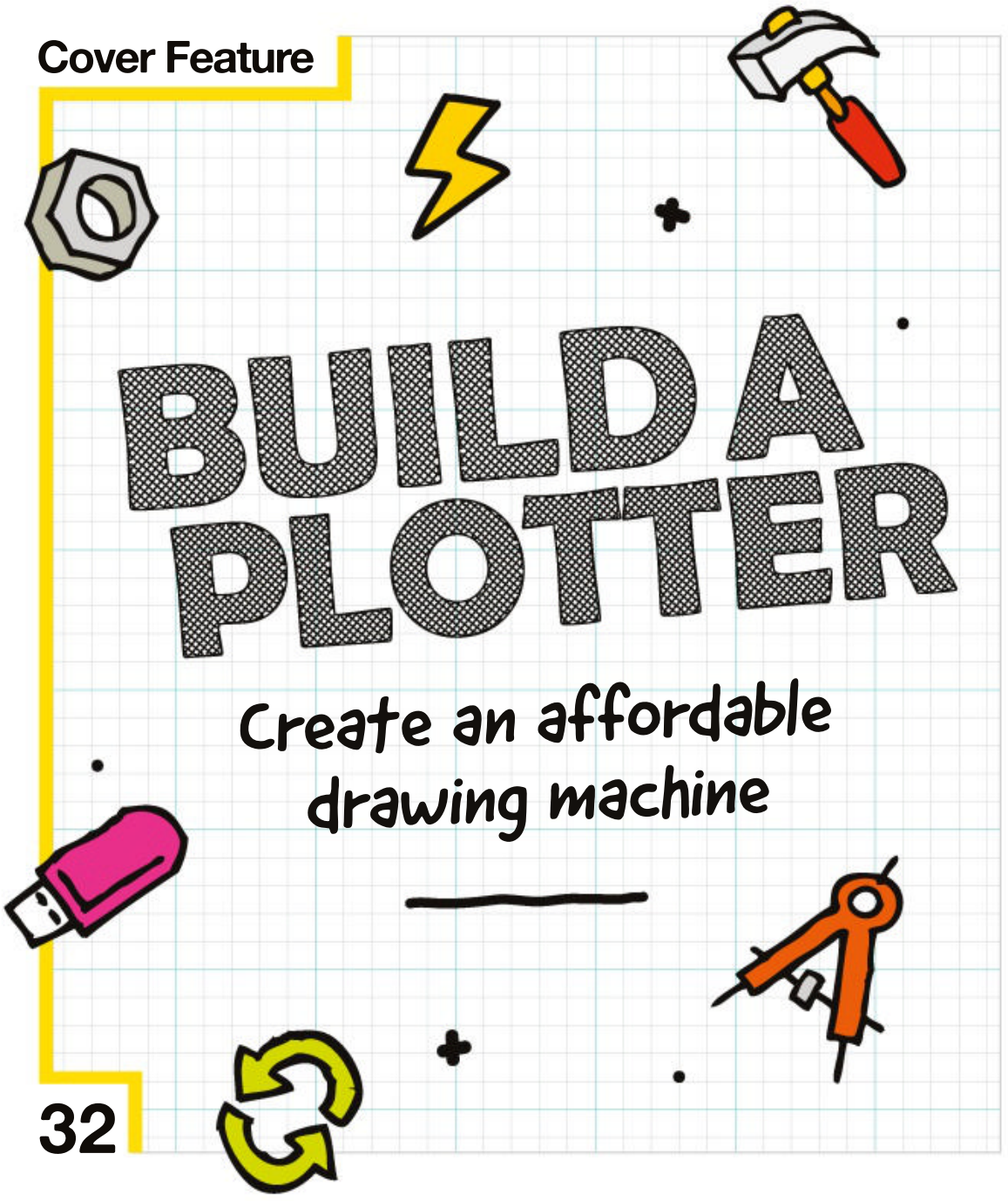
This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi 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. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents

92

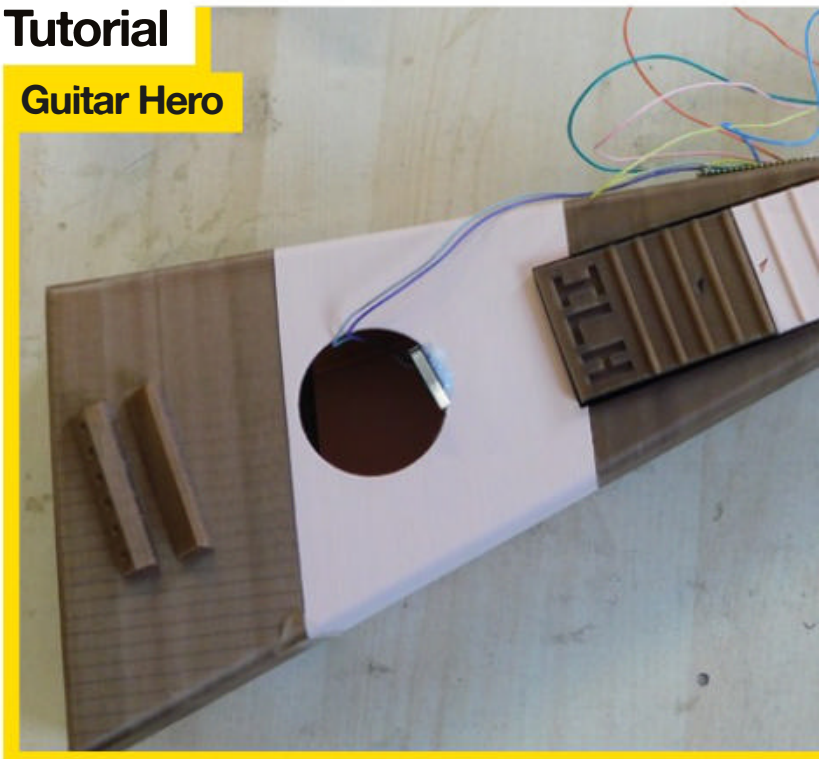
Cover Feature



32

Tutorial

Guitar Hero



72 Turn a failed 3D print into a one-off games controller

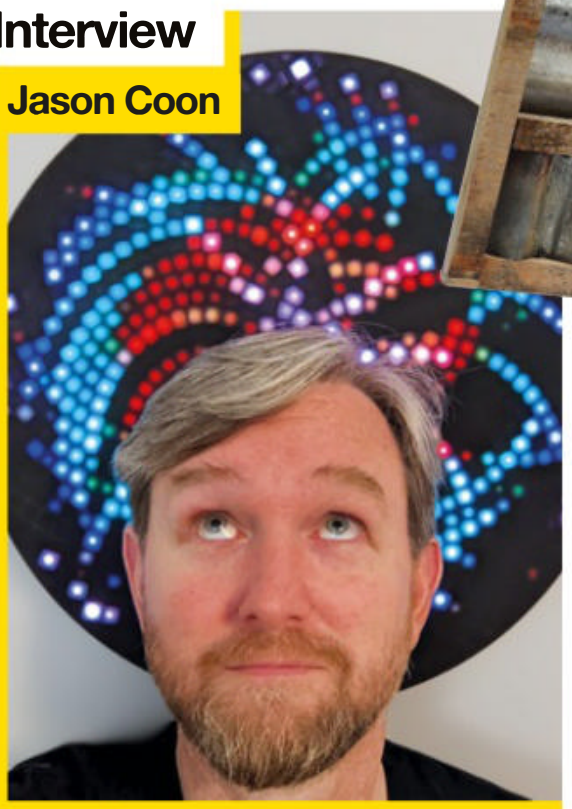
104

98

60

Interview

Jason Coon



52 At the creative confluence of mathematics, art, and nature

Tutorial

Lubrication

82 Stuck? Here are the best ways to get things moving again

66

06

71 FORGE

103 FIELD TEST

HackSpace magazine is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send US and Canadian address changes to HackSpace magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

REGULAR

Walnut laptop

By **mw33121**

hsmag.cc/WalnutLaptop

Reddit user mw33121 made this beautiful laptop, based on a Raspberry Pi 4B. The case is 5mm and 10mm walnut (the darker wood you can see is Indian rosewood; the lighter wood is maple), and it was built with hand tools and a cordless drill.

In the maker's own words: "Inside there's the LCD panel and driver board, a tiny amp, and tiny speakers, a camera and microphone, a 30k mAh laptop power bank, the foldable keyboard/trackpad, with a Raspberry Pi 4B running it all. The USB port charges the power bank and there are controls for the overall power, the monitor control board, and the amp volume on the sides. There's a small slot to change the SD card at one end and a tiny acrylic window underneath to read the power bank LCD charge percentage."

Apparently, you can get six hours of battery life out of this $250 build – enough to watch many, many glorious woodworking videos.

Right
This build started when the maker found the LCD screen on sale at a steep discount — probably because it's such a weird shape


2033
command
alt
option
shift
enter
return

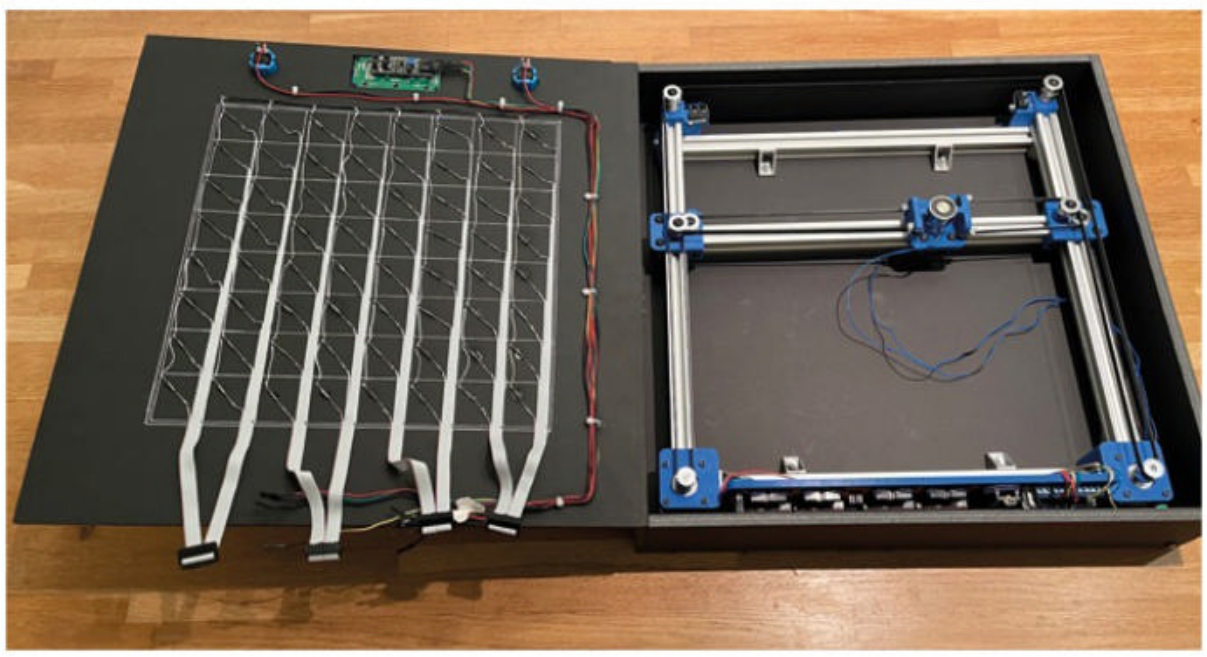
Automated chessboard

By **Greg**

hsmag.cc/AutomatedChessBoard

This automated chessboard by Greg is wonderfully simple, and so cleanly built that we had to ask him for a shot of the underside of the build too, just so you can see that there's more to it than a standard chessboard.

The chessboard houses an electromagnet moving on a set of extruded aluminium rails, very much like a laser cutter. Two stepper motors and transmission belts move the electromagnet up and down and left and right so it can move the chess pieces above into any position on the board. If you remember the excitement around Deep Blue, the IBM supercomputer that beat Garry Kasparov in a match in 1996-7, you'll be amazed to learn that the brains for this machine are provided by a simple Arduino Nano.



Right
The chess pieces have been modified with a small magnet in the base of each one

REGULAR

Cyberdeck

By cyzoonic

hsmag.cc/Cyberdeck2

We love chunky toggle switches, and this build by cyzoonic has four of them. That's what drew us in; what intrigued us is that the keyboard and all the electronics are remounted on a custom-cut aluminium chassis – that's what elevated this build to the next level of polish. That, and the WiFi, 11-inch screen, Cherry keyboard, and the waterproof Pelican case, which ensures that the bearer of this machine will still be able to hack the mainframe (or whatever remains) in the event of an apocalypse.

Right
Cyzoonic has added an ESP32 with an OLED display, two rotary knobs, five buttons, and keypad


SPARK
VIDEO
CONTROL
NETWORK
LAN1
LAN2
USB
VIDEO
IN
POWER
PC
NET
VIDEO
DANGER
CYBERDECK
ONLY TO BE OPERATED BY TRAINED PROFESSIONALS

Portable Savonius wind turbine

By **Adrian Cubas**

hsmag.cc/SavoniusTurbine

Savonius wind turbine is the simplest form of vertical wind turbine you're likely to see. At their most basic, they're made from two halves of a cylinder cut lengthwise and stuck back together around a rotating axis. They're easy to build, but not as efficient as some other designs – which is why Adrian Cubas has created his own design here, adding wind deflectors to improve torque and reduce drag on the side of the cylinder that's moving toward the wind.

Adrian used a stepper motor in place of a generator – it's basically just wires and magnets, and so the only consideration, if you're going to make one of these yourself, is that you really need to wire the thing up correctly.

Right
The central part of this turbine is made, as so many are, from a length of PVC pipe; the rest (the clever bits) Adrian cut out using a CNC, a jig-saw, and an angle grinder


nzxo

REGULAR

Brass mouse

By **Uri Tuchman** hsmag.cc/BrassMouse

People love their computers; they're tools that we're connected to for hours, days, or weeks at a time. We fidget over elbow and wrist positions, we spend crazy amounts of money on the best screens, the most comfortable keyboards… and then we go and use the same lump of plastic to move a cursor around. Uri Tuchman wanted to improve on the bog-standard computer mouse, and so he made this.

The electronics may have been lifted out of an old two-button mouse, but everything else is Uri's work. There's a solid brass base plate and brass stand-offs (milled by Uri – nothing is shop-bought). Even the scroll wheel is lovely, cool, tactile brass. And check out the engraving!

Right
As Uri says, ergonomically this is terrible. But it looks fantastic!

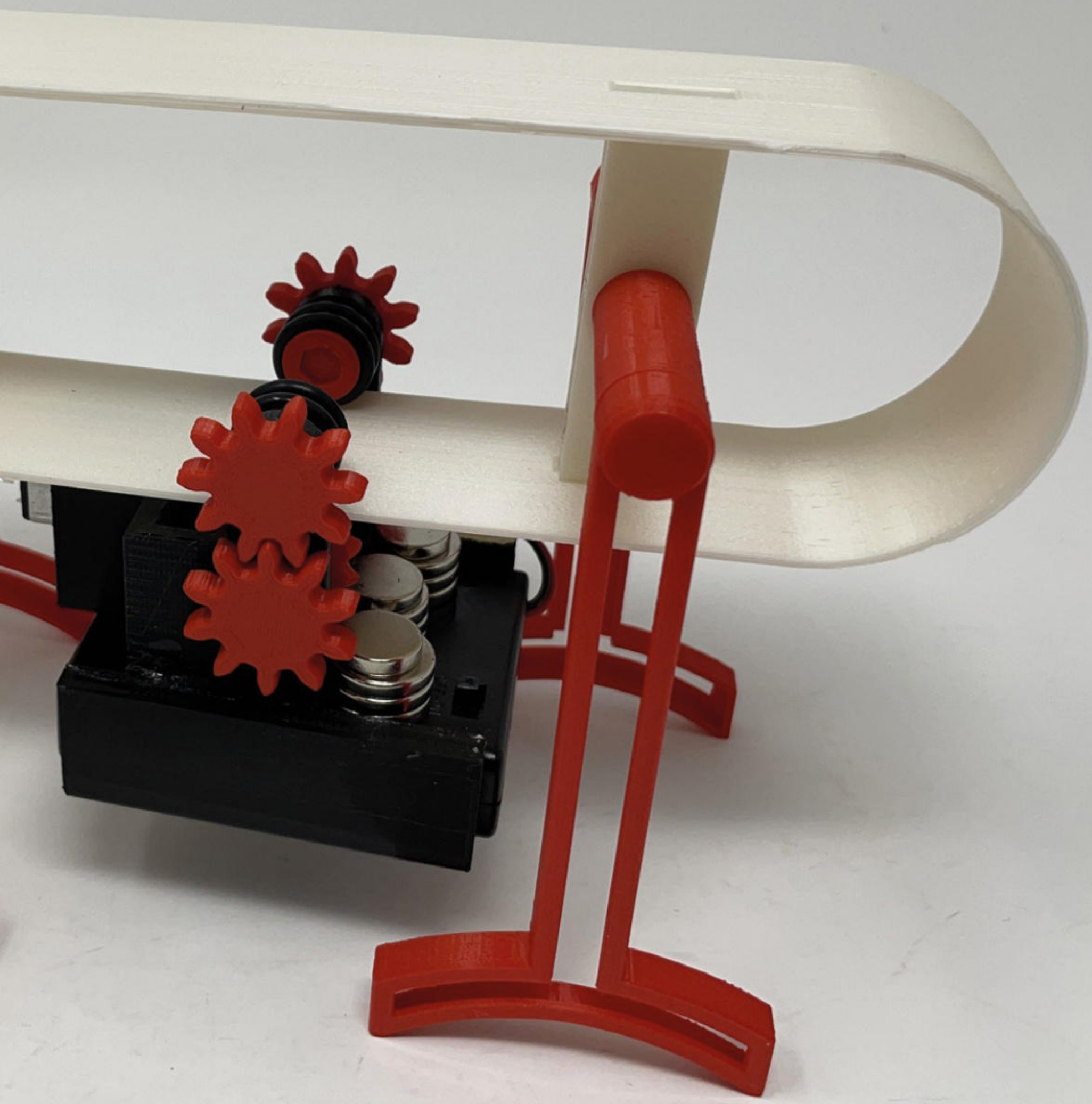
Wacky Walker

By Greg Zumwalt

hsmag.cc/WackyWalker

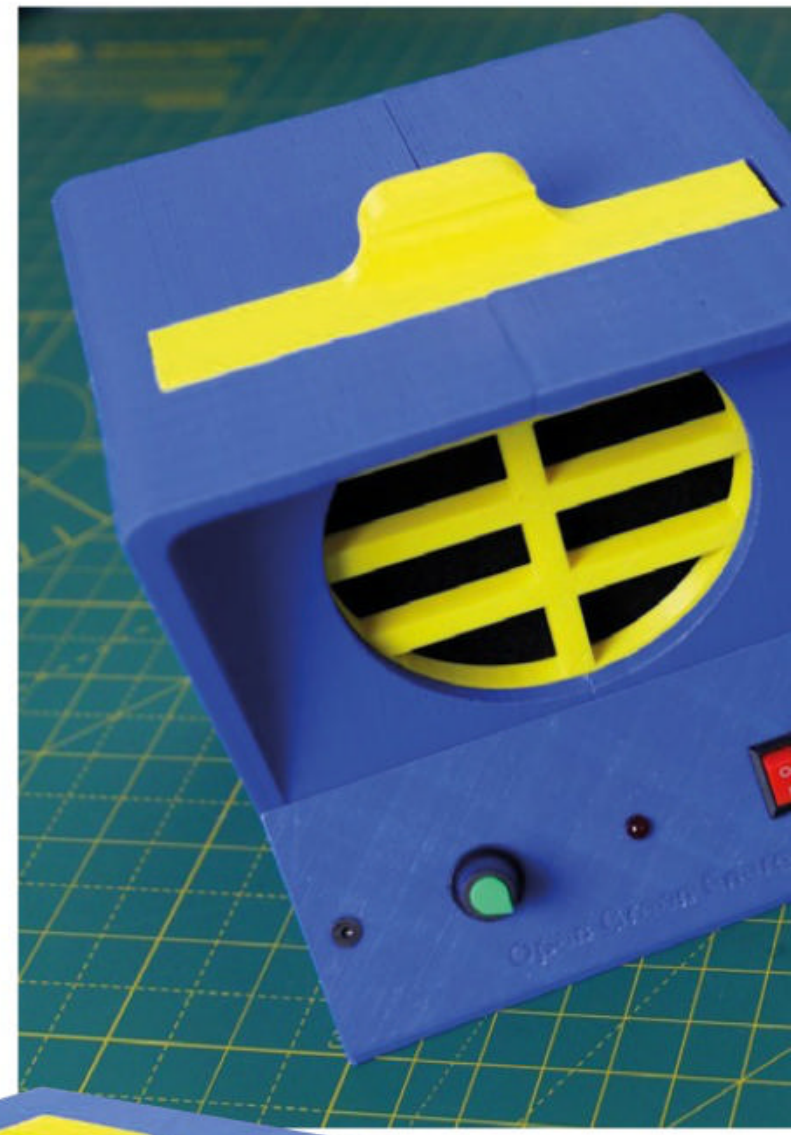
This brilliant creation by Greg Zumwalt stretches the definition of walking a little for us, but it does have such charm and style that we can't help but love it. From standing on four feet, it rises on to two, before leaning forward on to four feet, and then it does it again. That's less walking action than a controlled forward roll in our book, and that's exactly how it works: a counterweight moves along the inside the ellipse that forms the body, propelling the model up and down and forwards. Silly, yes, but also brilliant.

Right
Greg's a master of 3D-printed mechanisms



Objet 3d’art

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



We always recommend soldering in a well-ventilated space – but why settle for an off-the-shelf extractor fan when you could print one yourself to match the colours of your soldering station? That’s what Debasish, aka Open Green Energy, has done in this smart build.

The solder fume extractor fan is a simple build – it’s just a DC jack, a switch, a status LED + resistor, a speed controller, and a 12 V fan. The body is more advanced, as it’s made from five separate parts, designed in Autodesk Fusion 360, and connected with threaded M3 inserts and hot glue.

A fan on its own would merely spread the fumes about, not contain them, so this device also has a changeable carbon filter to collect the many and varied pollutants that come from soldering. □

hsmag.cc/SolderFumeExtractor


HAKKO
UP
ENTER


HAKKO
350
UP
ENTER

REGULAR

Meet The Maker:
Sam Battle
aka Look Mum No Computer

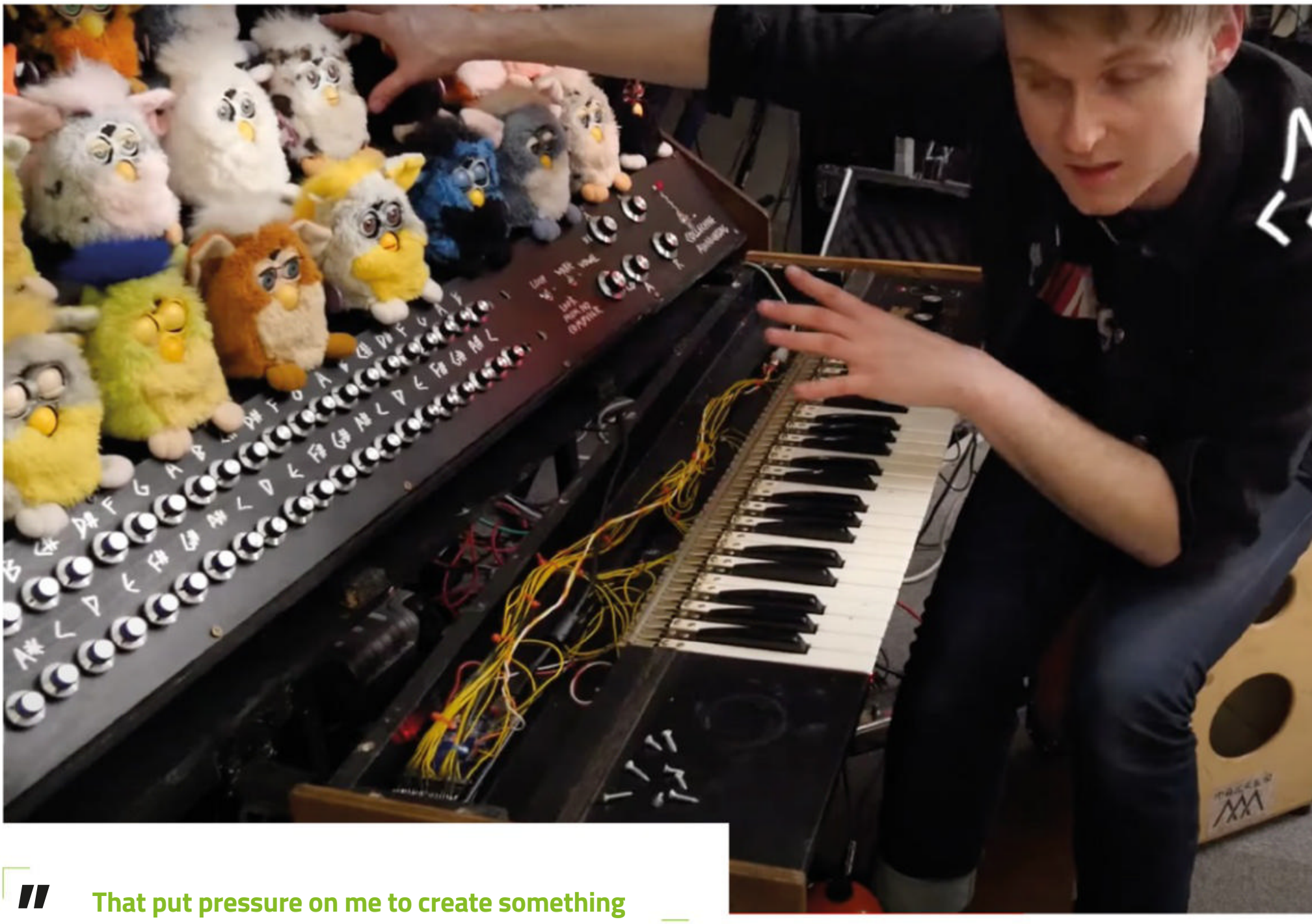
A harmonious nightmare machine

Sam Battle, aka Look Mum No Computer, wears many hats. He's a musician, who records and performs. He's an inventor, bringing electronic musical instruments into life from out of the depths of his brain. He's also a music producer, a TED talker, and a self-taught electronics engineer.

That's more than enough, but he's also a brilliant teacher – if you want to know about the fascinating, bizarre rabbit hole that is electronic music, you couldn't find a better place to start than his site **lookmumnocomputer.com**. We spoke to Sam when he created the Furby Organ: a polyphonic synthesizer that cries out with the voices of 44 Furbies. It's an amusing, weird, fascinating instrument that relies on Arduinos, abandoned 1990s animatronic toys, and a lot of soldering. →

Below
One man, 44 Furbies

REGULAR



> **That put pressure on me to create something tangible,** rather than just some random doodle

Above
The keyboard is a separate module, wired up to an Arduino Mega, which interfaces with anything else that Sam cares to plug it into

HS This Furby organ is the absolute stuff of nightmares.

SB I know, it was a bit of an odd one… it was just something that had to be done, you know what I mean? I've had that idea for a long while – seven years – and it was finally time to make it happen.

HS It looks like a hell of a lot of work. I assume you weren't working on it solidly all day, every day for seven years?

SB No, not at all. Like, seven years ago, I had a few Furbies laying around and I thought would be cool if I made something. Like, it would be cool if they were all singing in a choir. That idea kind of lay dormant for about six years. And then in 2018, I was like, I'm gonna do it. I'm gonna finally do it. I'd been telling loads of people all the time, 'I'm going to build this big Furby organ.'

That put pressure on me to create something tangible, rather than just some random doodle. So I bought a load of Furbies last year and it was my New Year's resolution of 2018 to get it done.

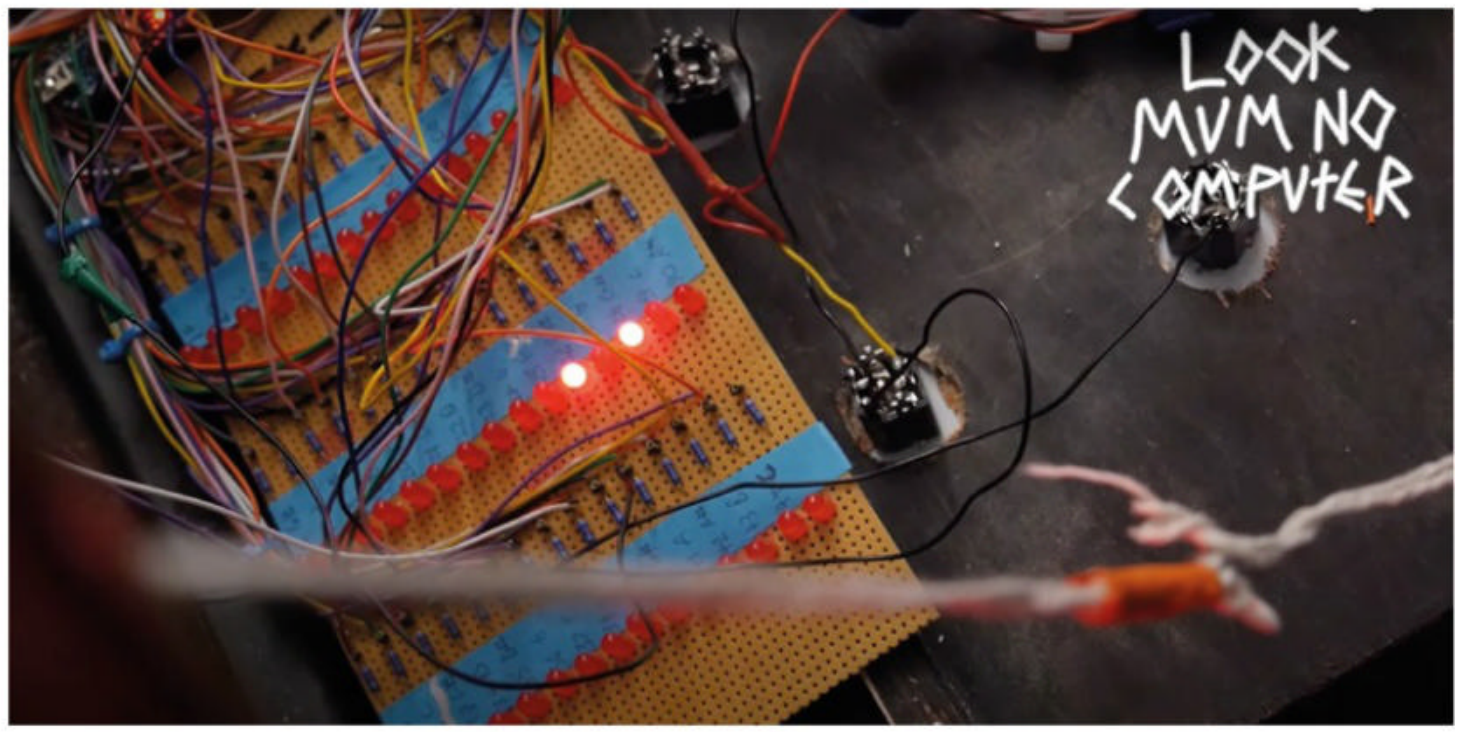
I think I sat down for maybe a week and a half, just solidly soldering and wondering what the point was, because it nearly broke me – the amount of soldering. It was very tedious.

HS Wondering what the point was… that's a very relatable feeling.

SB Yeah, they were testing times.

HS How does it all work technically? I can see that there's a lot of wiring at the back. Is there a microcontroller for each Furby, or do you treat

Below
This perfboard circuit translates messages from the keyboard into trigger commands, telling the Furby to play a note

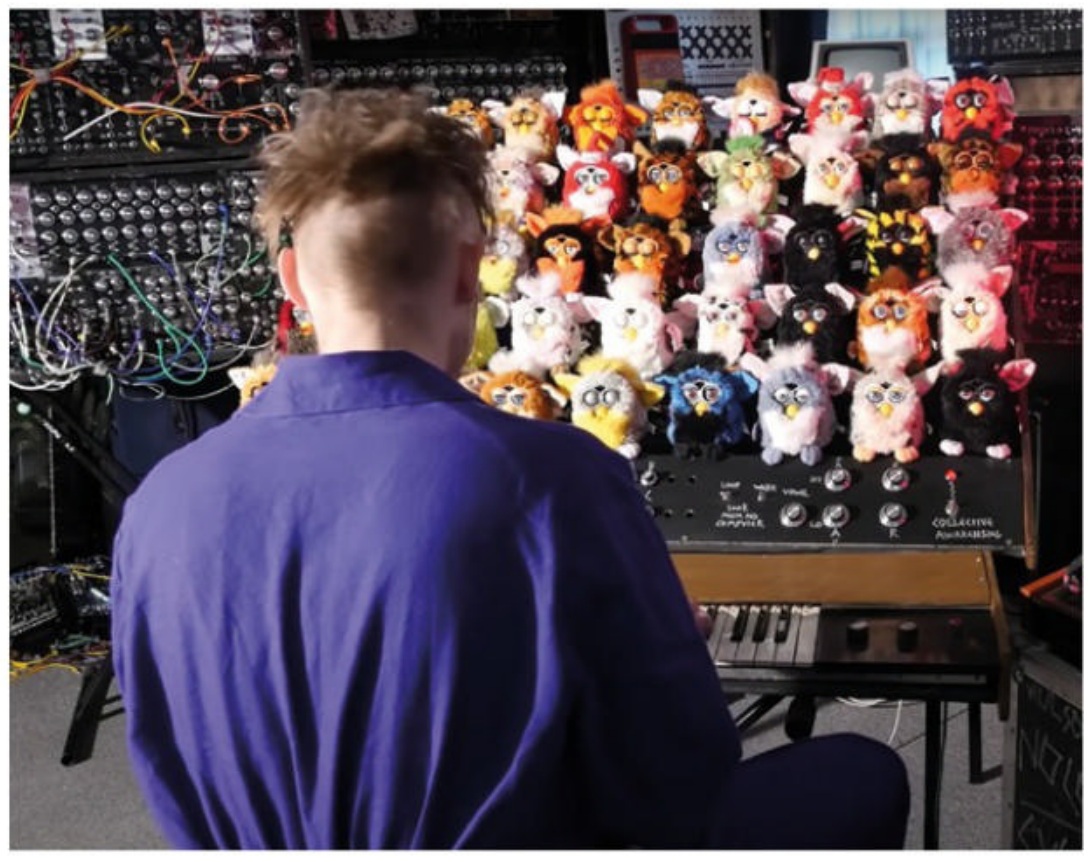


them as individual speakers and control them from a central point?

SB I think the roadblock I had about seven years ago was the fact that I couldn't get any constant tones out of the Furby. The thing is, it depends on the point at which you kind of freeze its brain and it makes a nice note, but you can't get a constant pitch that's reliable for that note.

So what I did was, I've got two microcontrollers per Furby – it just seemed easier that way, because I'm not very good at coding and I thought it would look cool if there was a hell of a lot of stuff. So each Furby has one microcontroller that controls all of its stuff, from its eyes to its mouth to its senses, the lights in the back of its eyes and stuff. And the same microcontroller controls its brain, so it's still saying stuff and acting like a Furby.

And then the other microcontroller kind of makes a formant note for it, which is hard to explain but it's basically the Furby vowel. And then that goes out →



REGULAR

Left
MIDI: it's like USB, but for the Flintstones, according to Sam

into this speaker as well. So it's kind of like, speaking and singing at the same time. Then they're all going out of each speaker, and I didn't know how else you would do it. I just built it not knowing whether it would work.

HS You must be pretty happy with how it's turned out?

SB I'm very happy. Finally, I've got a Furby organ! Finally it's sitting here, and it's not just a mangle of sounds that are truly off-the-rails oddness, but there's a button on the back that kind of makes it go into glitch mode and just sounds like all the Furbies are vomiting on each other.

> **I could get all philosophical about it,** but at the end of the day they're just Furbies

Right
Sam built a portable version of the Furby organ to take abroad; because airlines have weight limits, he removed the poor Furbies' skins to shave a few grams off the build

There's a synthesizer called a Polymoog, which is a keyboard that has a separate synthesizer voice for each key. So it was a truly polyphonic synth with an 88-note keyboard that had 88 little synth voices in it. I took the idea from that to just give each Furby a voice.

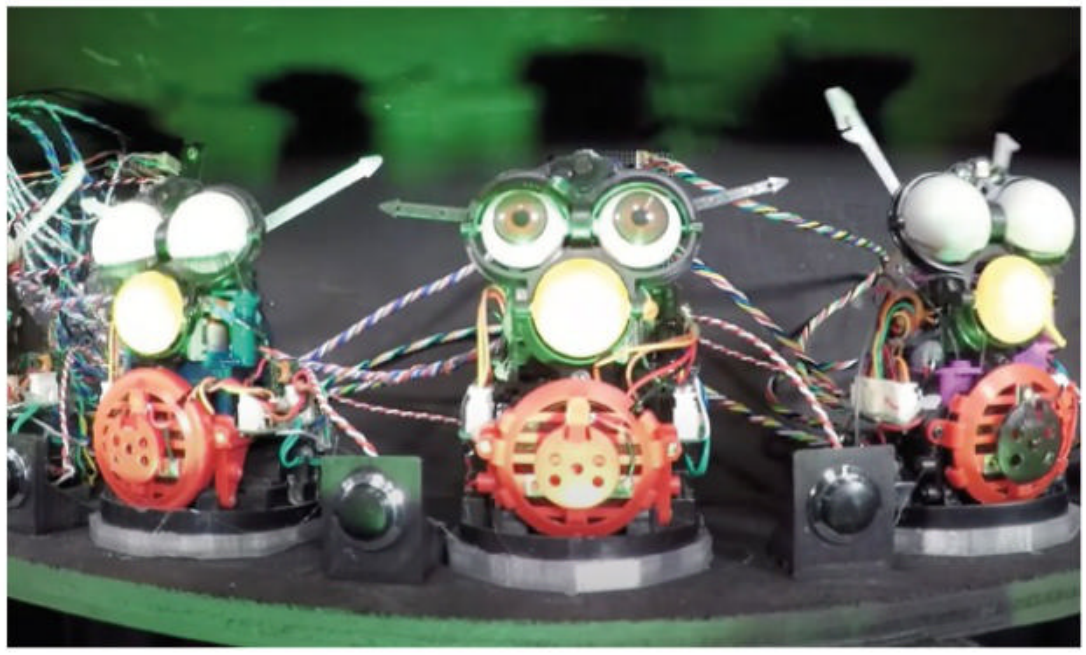
HS I know someone who's going to love that. He doesn't even collect synths, he collects the SID chips from old Commodore computers. He'll buy the computers just to get the sound chips out of them.

Synthesizers and electronic music in general feels like a really esoteric field that you can really geek out on.

SB I've done the same! I've got about ten SID chips. I see Commodore 64s at car boot sales and I'm just like 'I'm gonna buy that. I'm gonna buy that. I'm gonna buy that', but then you get home and you realise that you've got too many Commodore 64s.

HS What other special features does the Furby organ possess?

SB Well, there's the Wake Up switch: that one is wired straight to all of their accelerometers – the sensor that tells them whether they've been shaken around or turned upside down or something. So each Furby's accelerometer has been wired to a switch so it makes them feel like they've gone upside down, which wakes them all up.



The Collective Awakening switch is just the switch to turn it on; when it turns on, all of them reset and just wake up for the first time. It's resetting their brains, *Men in Black*-style so they don't know where they are or who they are, or why they're here. Every time you turn it off, they forget who they are. I could get all philosophical about it, but at the end of the day they're just Furbies.

I like to think that, in return for giving them a kind of dream-like world where, you know, they have no regrets and they have no worries and qualms because they forget what they are at the end of the working day, I get their souls in return for that. It seems like a good bargain.

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**

UPCYCLING

One thing stood out from the last issue... nothing to do with making really, but the appliance you chose to modify for In The Workshop. What on earth is a bean slicer? How does it differ from a knife, which I can use to slice all sorts of things including, but not limited to, beans? What's the point of these things existing? Can they be all that common in British households, when I've lived here for 40-odd years and never heard of them? We need to be told!

Jane
Birmingham

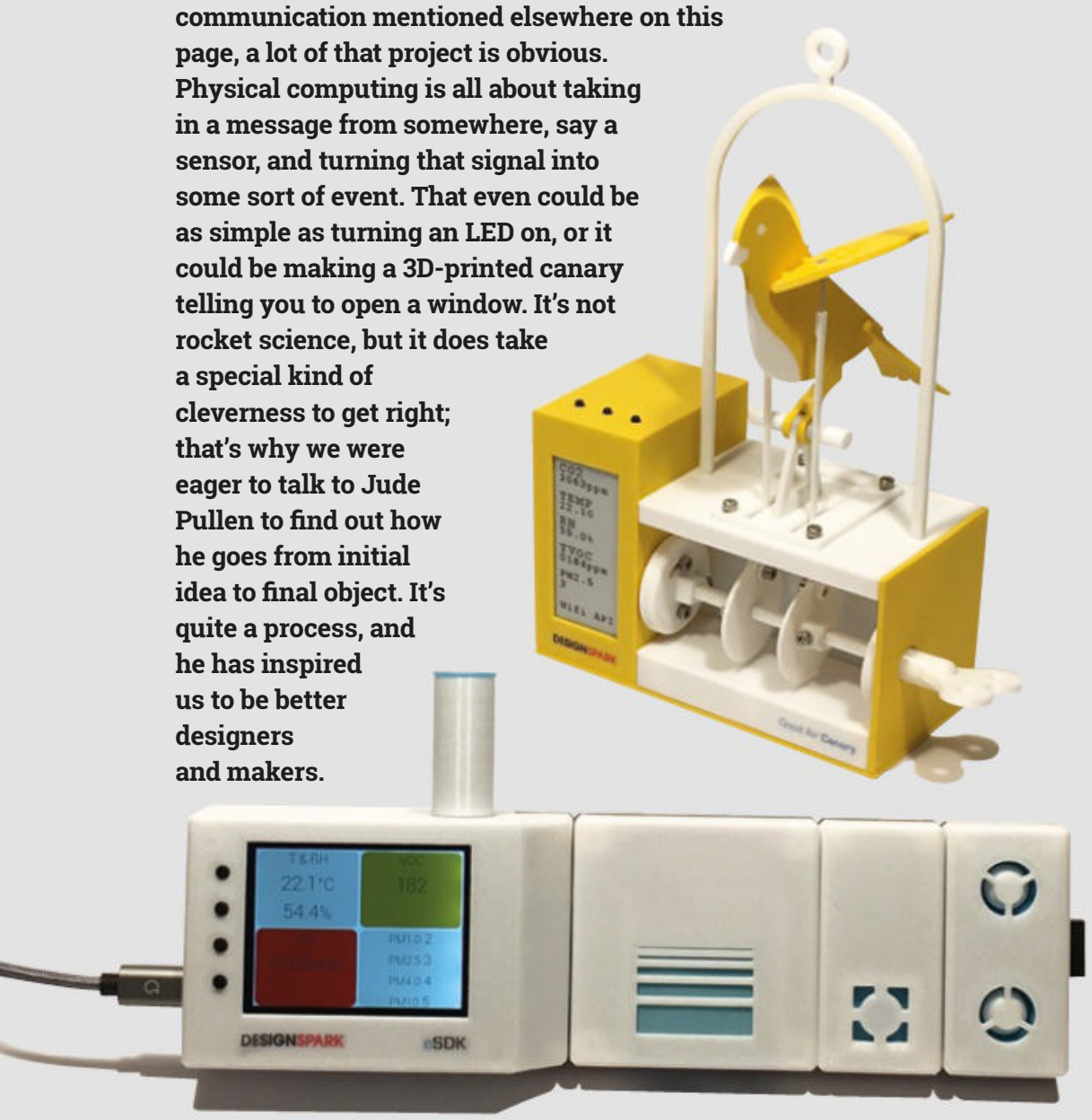
Ben says: Apparently bean slicers were a common sight in country kitchens: if you've got the space for it, you can grow beans easily, but they produce gluts of food and can get very tough if they're not picked at the right time. Hence the need for a bean slicer! If you're not convinced, maybe you're starting to realise why I repurposed mine to turn it into something actually useful.

GOOD AIR

Following on from your interview with product designer Jude Pullen, I fell down the rabbit hole of the Good Air Canary. What a brilliant project: gathering air quality data and displaying it via an IoT device, rather than just showing it on a screen, is a great idea. And the way that the other makers involved in the project have taken up the idea is brilliant: each one is different in its own way. We're a clever bunch, us humans.

Paul
Newcastle

Ben says: We are indeed. And, just like with the earth communication mentioned elsewhere on this page, a lot of that project is obvious. Physical computing is all about taking in a message from somewhere, say a sensor, and turning that signal into some sort of event. That even could be as simple as turning an LED on, or it could be making a 3D-printed canary telling you to open a window. It's not rocket science, but it does take a special kind of cleverness to get right; that's why we were eager to talk to Jude Pullen to find out how he goes from initial idea to final object. It's quite a process, and he has inspired us to be better designers and makers.



EARTH COMMUNICATION

Thanks for (again) opening my eyes to cool old technology – this time communication by ground conduction. It makes perfect sense that you can use the conductivity of the earth to send messages similar to radio waves, but I'd never thought of it until I read issue 54.

Andrew
York

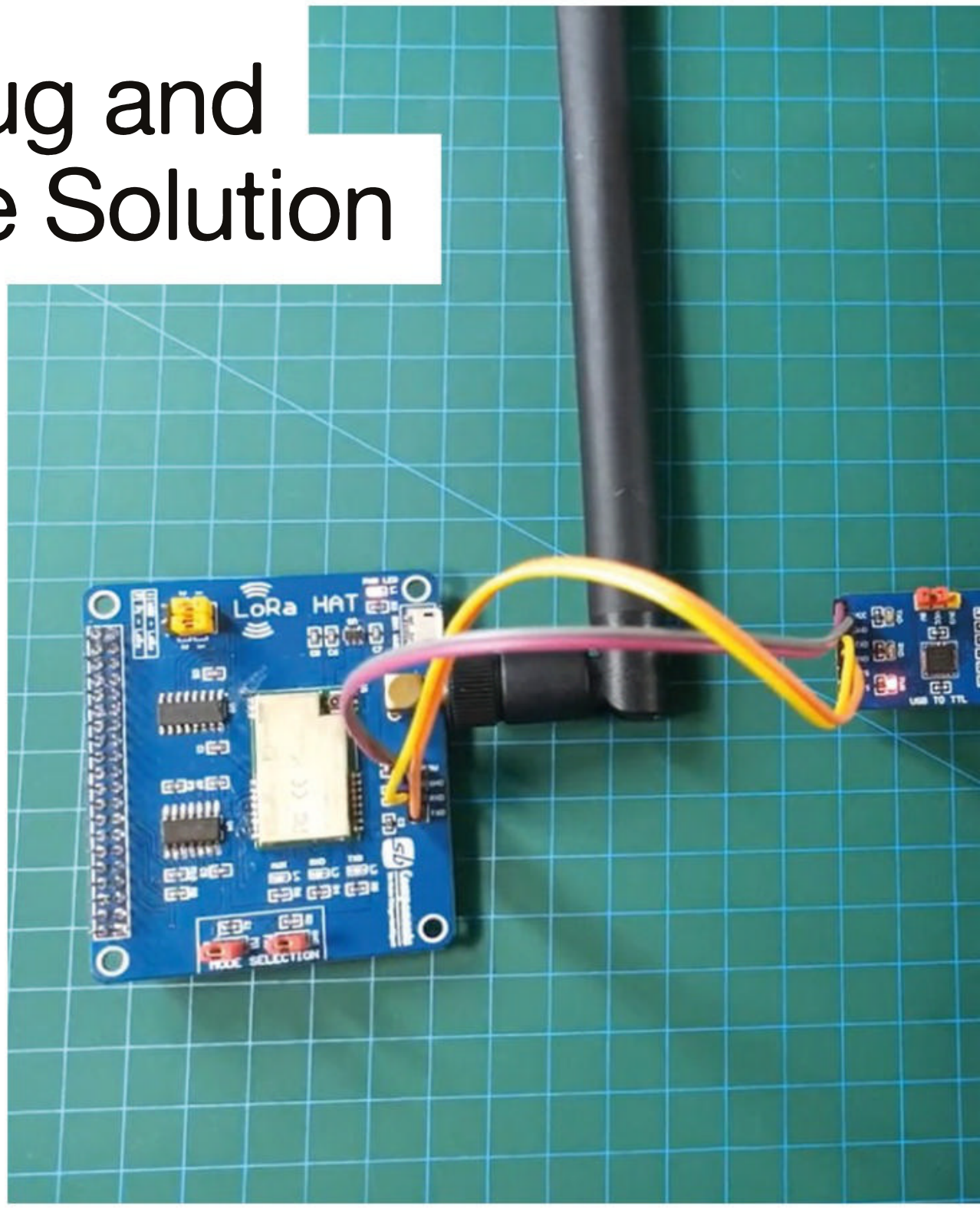
Ben says: It is obvious when you think about it, which is a sign of it being a good idea. As Mike Bedford wrote in his piece, earth communication is still used by cave rescue teams in areas where there's no WiFi or phone signal, or where stupid vanity submarines can't reach. Turn to page 86 to find another way to send data – humble light beam communication.

CROWDFUNDING NOW

USB to TTL: Complete Plug and Play interface Solution

An easier way to plug into your projects

From £3 | **hsmag.cc/usbttl** | **Delivery:** July 2022



We often think of crowdfunding campaigns as having to be some revolutionary, life-changing thing. Perhaps this is sometimes due to the hyperbole of some campaigns and some of the platforms that host them. However, it doesn't have to be the case. Crowdfunding is a way of bringing projects to life, and that can be true for small and simple campaigns as well as huge, complicated ones.

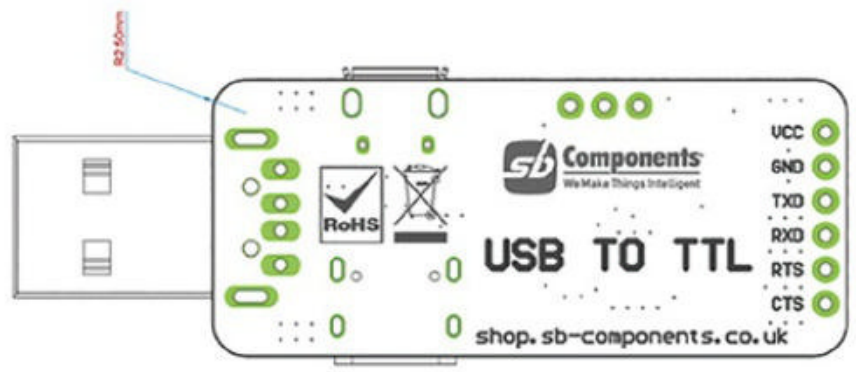
The USB to UART bridge is a common maker component. It lets you control a wide variety of different hardware, including getting data off microcontrollers (and even programming some). The trouble is, there are a few differences. Some run at 3V, some at 5V. Usually, they have a USB-A plug mounted directly on the PCB, and this can be awkward to use with some computers.

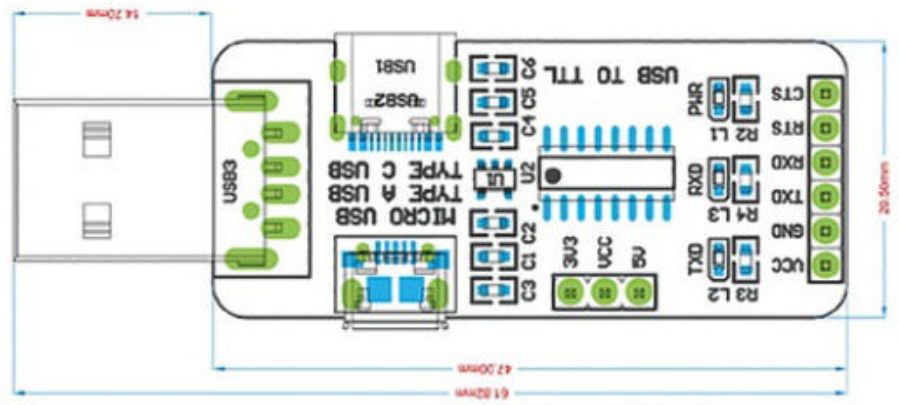
This project is one of the most flexible USB to UART connectors we've seen. It has a range of different USB connectors (A, C, and micro), as well as a jumper to select the voltage. It's not going to revolutionise your life, but if it works as expected, it'll be one of those little things that just makes life a little bit easier.

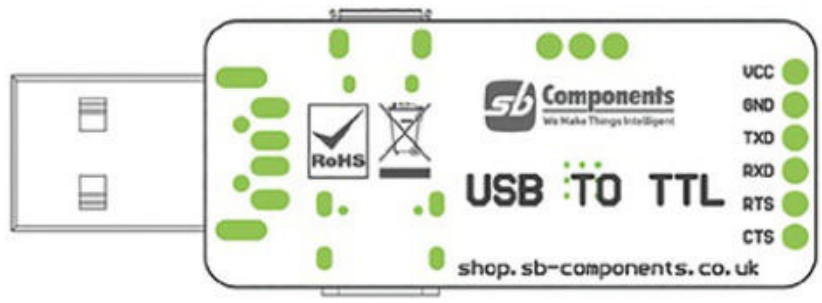
BUYER BEWARE

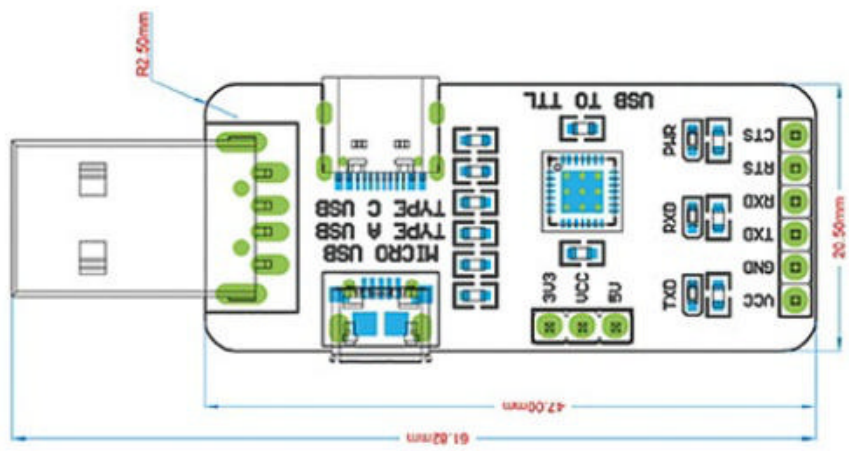
When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the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

Below
Three different USB connectors mean this will plug into most laptops









sinclair

Commodo

the

COMPUTERS THAT MADE BRITAIN

OUT NOW

"The Computers That Made Britain is one of the best things I've read this year. It's an incredible story of eccentrics and oddballs, geniuses and madmen, and one that will have you pining for a future that could have been. It's utterly astonishing!"

- **Stuart Turton**, bestselling author and journalist

Available on amazon

Buy online: **wfmag.cc/ctmb**

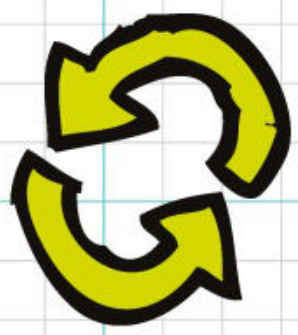
HackSpace
TECHNOLOGY IN YOUR HANDS

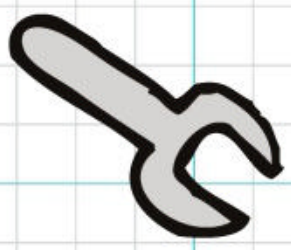
LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

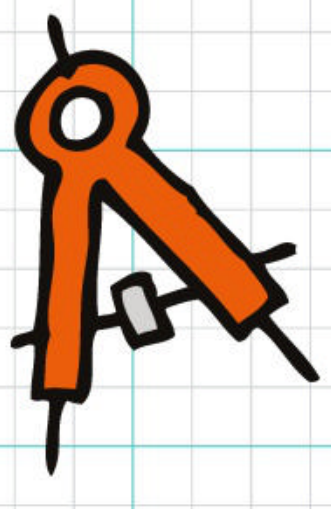
FEATURE



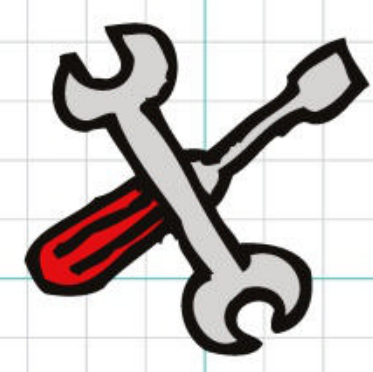


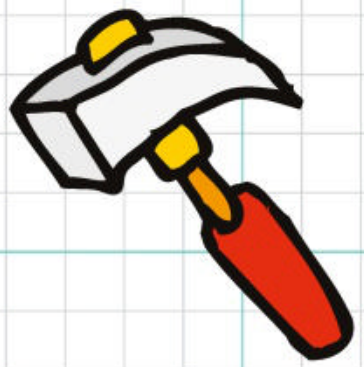
BUILD A PLOTTER

Plotters are popular machines for makers. They're expensive to buy, but simple enough to be easy to build out of commonly available parts



A TER

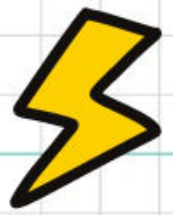




Unlike many computer-controlled maker machines, plotters can be forgiving of slight inaccuracies. For example, in a line representation of a photo, a line a millimetre or two out of place might be unnoticeable. Similarly, a bit of wobble in an already wobbly line won't really be visible to the eye. A 3D printer – which is mechanically a similar machine – might be useless if it had this level of error, but a plotter can still be a useful and fun machine. This makes it a great first machine to build. You can learn about the different trade-offs and challenges in building a computer-controlled hardware without having to get everything perfect the first time.

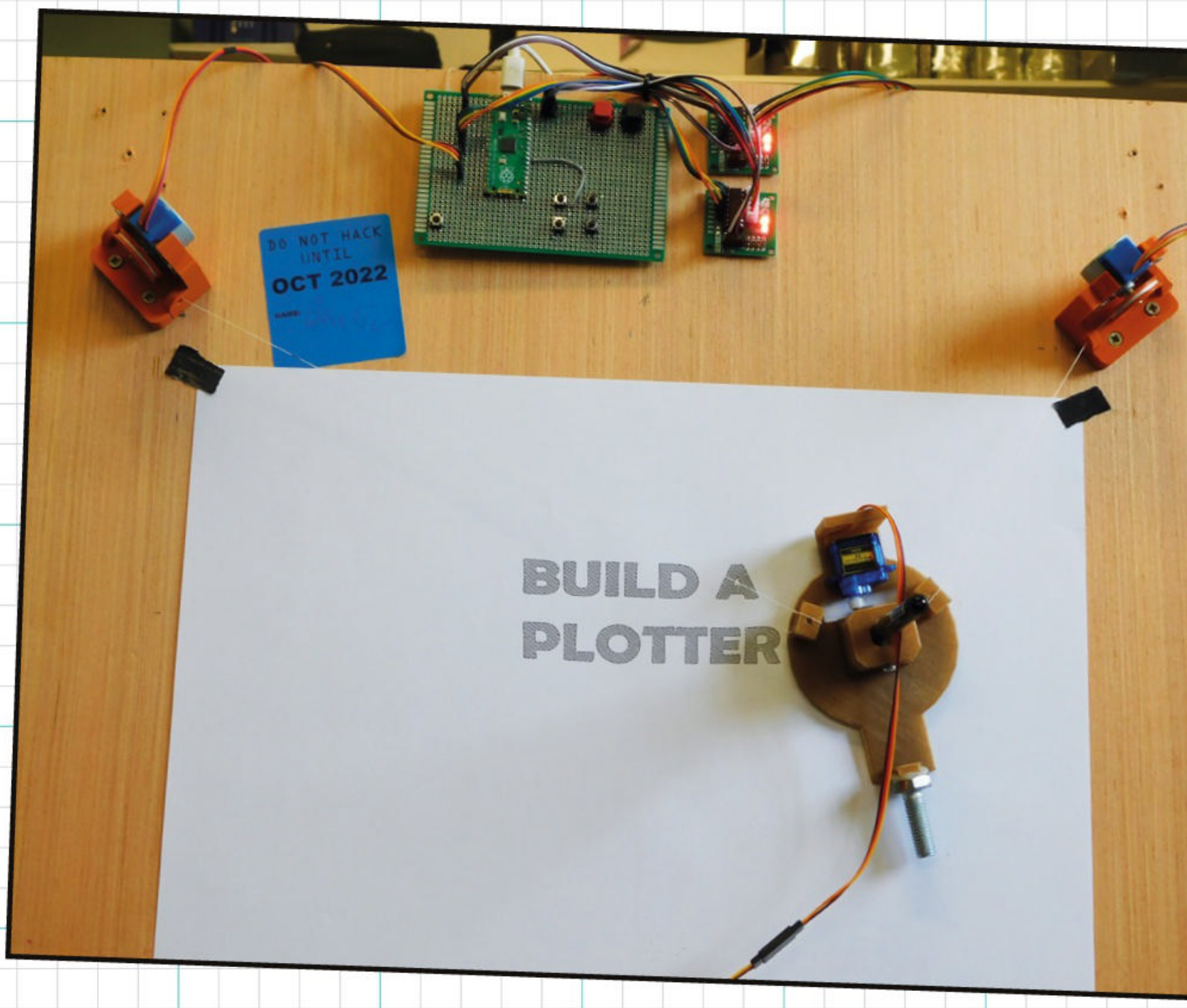
As in so many cases, accuracy really comes down to cost. Expensive stepper motors can be more accurate than cheap ones (at least to a point), and accuracy requires a sturdy frame which will need to be made of more expensive materials. More complex mechanisms are needed to keep things moving in the right way, and again, more complex usually equals more cost. What we're building is a cheap, not particularly accurate, plotter that is easy to understand and is a great starting point for exploring the world of plotters. Once you've got a working plotter, you can tweak it and see how that affects the quality of the images.

The price is the main feature because cheap things are less pressure to build and tinker with. No matter what you do, the most expensive part you can break on this plotter is £4, so how bad can tinkering be? Try changing a part, try changing the code, see what happens and what the end results look like. This is a machine for makers and tinkerers, not for people wanting a perfect finished machine to place upon a pedestal. →

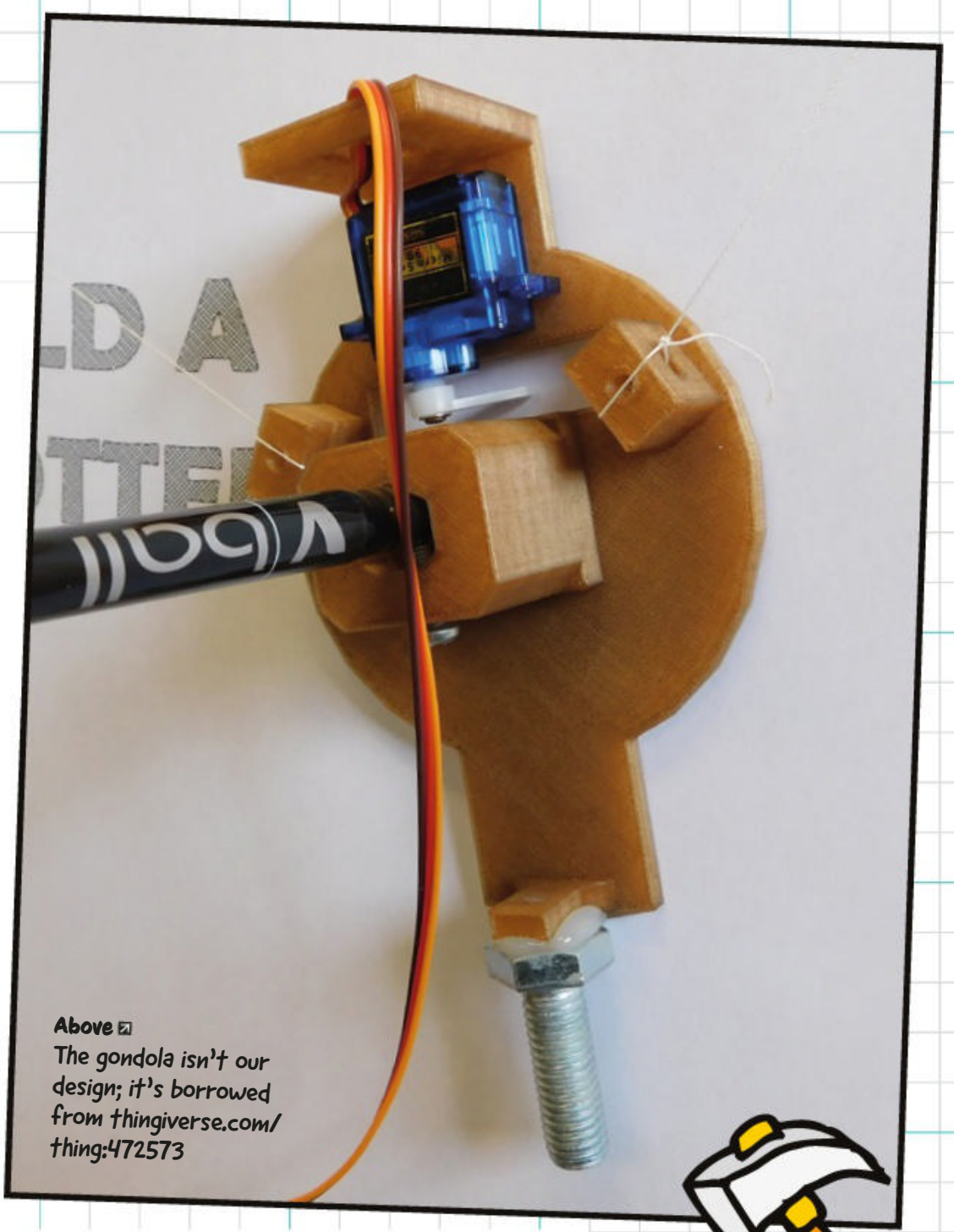


FEATURE

Vertical PLOTTER



Right
The final plotter up and running



Above
The gondola isn't our design; it's borrowed from thingiverse.com/thing:472573

The idea behind a vertical plotter is that there are two motors each attached to a spool of cable. At the other end of these cables is a gondola holding a pen. By winding or unwinding the string different amounts from each motor, the plotter can position the pen wherever it wants. At least, that's the theory.

No plotter is perfect and hopefully it comes as no surprise that, if you build a plotter for £15, it's going to have some compromises. Let's take a look at what they are, starting with the potential problems with a vertical plotter.

Firstly, it's hard to eliminate wobble. The mechanism is just two lengths of string, so it's inherently flexible.

Secondly is that while the theoretical drawing area is quite large, you'll only get an accurate drawing in a small part of this. Errors compound the further from the centre of your drawing area you travel, so things tend to get a little distorted if you try to fill the area.

Thirdly, there's no easy way that we could find to auto-home the plotter. There's nowhere to put limit switches, so each time you start the plotter, it's up to you to make sure the gondola's in the right position. There is a different design of vertical plotter that, instead of wrapping the cables up in a spool, has the loose cable dangling down, typically with a weight on the bottom. This is a slightly more complicated design, but it does allow auto-homing because you can put bits to trigger a limit switch on the loose end of the cable. We opted not to go with this style of design because it increases the complexity of the build.

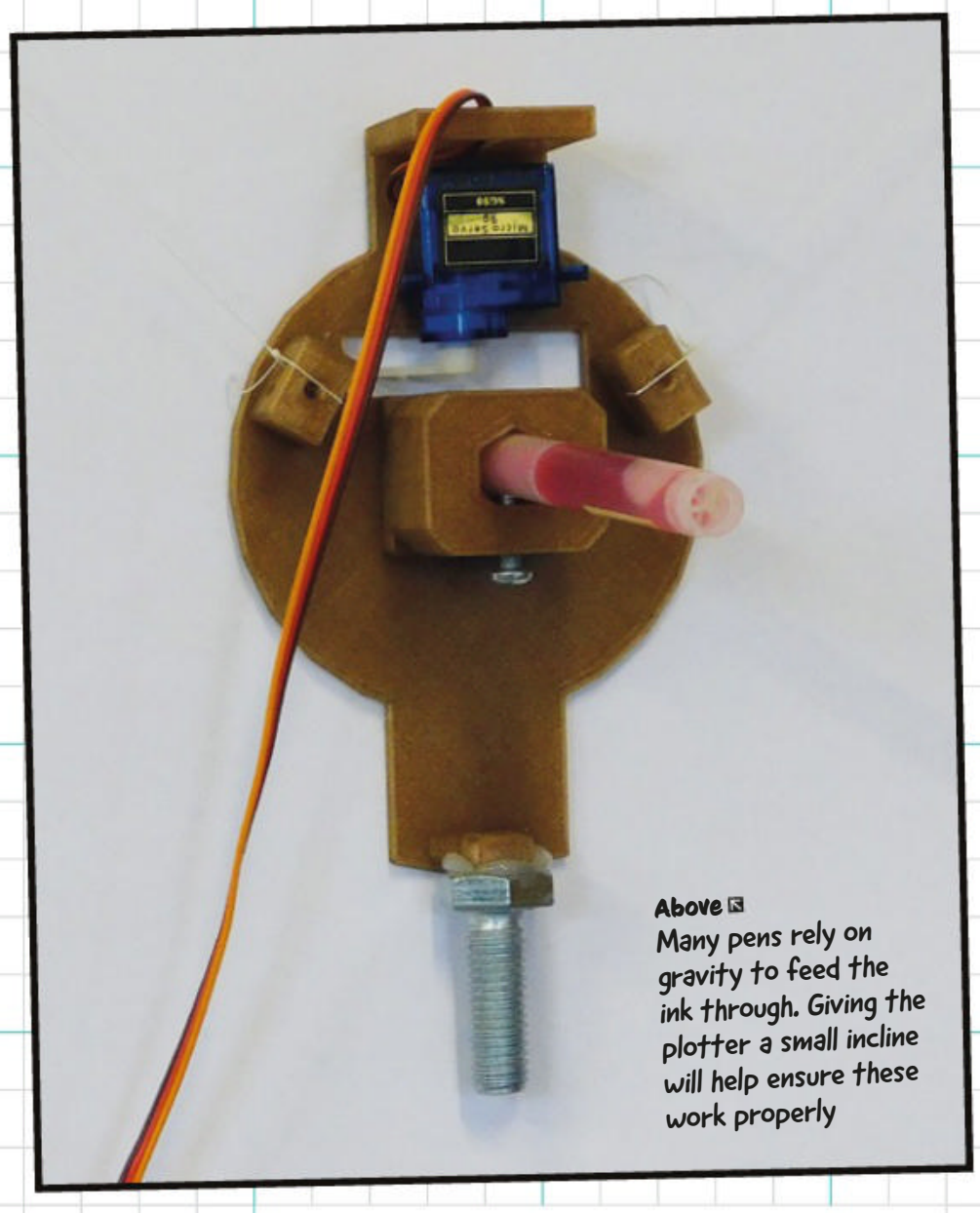
Those are the downsides of this design; what are the upsides?

Firstly, it's cheap, and this is an important feature. Not just because it costs less money, but price has further implications. Less money means less risk to hacking around and changing things. If a plotter cost £150, then you'd expect it to work better, but you might be more reticent about making changes to the code or design should you break it. For £15, it's automatically more hackable because you can't do that much damage (in financial terms). This also means that more people will build the project. The hypothetical £150 plotter might work better, but you're likely need a good reason to buy one. The cheaper it is, the more people will build it just to have a play.

Secondly, it's small. By that, we mean that it doesn't take up much floor space as it can be mounted on the wall. For the overwhelming majority of makers, floor space is at a premium, and can be one of the biggest limiting factors on what tools →

It's cheap, and this is an important feature

FEATURE



Above
Many pens rely on gravity to feed the ink through. Giving the plotter a small incline will help ensure these work properly

we have. On the other hand, wall space is often spare – not just spare, but attaching an interesting-looking tool to the wall can be an asset. Got an ugly bit of peeling paint on the wall? This is cheaper than some art to cover it up (and it can produce its own art!).

Thirdly, it's big. This isn't a contradiction to the previous point because this time we mean that it can make big drawings. How big? Well, how big do you want to make it? With vertical plotters, the maximum size you can print is guided largely by how accurate you want the drawing to be. Ours can draw A4 well, and it's OK if you go bigger than this. You can move the motors further apart to give yourself a bigger area, but you will run into problems with accuracy. How far can they go? We don't know, you figure it out!

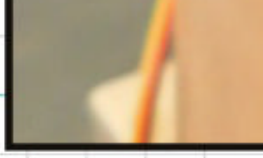
Only you can decide if these particular trade-offs are right for you. There are a myriad of other options if you want a plotter that has different pros and cons.

The physical hardware is quite straightforward. You'll need two motor mounts (we designed these to be symmetrical so you can print two of the same mount), two spools, and a gondola. The 3D files are all downloadable from **printables.com/model/175691**. These can all be 3D-printed without supports. We tested it out with both PLA and recycled PETG, and both printed without problems – we'd expect more-or-less any non-flexible filament to work, so just use what you have to hand.

There aren't particularly large stresses on any of these, so an infill of 15% or so should be fine. Make sure they're orientated correctly. The motor mount should print with the 'horns' pointing up; the pulley should print flat on the bed, as should the gondola.

As well as the 3D-printed parts, there are a few other bits of hardware:

- A board. We've used 6mm ply, but this can be absolutely anything that's large and flat.
- Screws or bolts to attach the motor mounts to the board. There are two countersunk 4mm holes which are designed for wood screws in each motor mount. You could also glue these down if screws won't do the job. Double-sided sticky tape would also make a good temporary fix if you wanted to reserve the option of removing the mounts.
- A 20mm M3 bolt to hold the pen in place.
- Some fine thread to hang the gondola. Cotton thread should work fine. It doesn't have to take any great force, so don't worry too much about how strong it is.
- A pen. Not all pens work well with vertical plotters as some require gravity to feed the ink down. Felt-tips work well but don't always have the graphical effect you want. We found that as long as the plotter had a slight angle off vertical, it wasn't too much of a problem, but if you get some pens refusing to draw, this might be why.



- A weight. We glued an M10 nut to our gondola, but anything that weighs about 50g and doesn't rub on the workpiece should be fine.

Alongside the hardware, you'll need some electronics. We used a Raspberry Pi Pico as the brains. We're programming it via Arduino IDE – other Arduino-compatible boards might work, but we did have unexplained crashes when we tried this on some AVR boards, so we wouldn't recommend this.

You'll need two 5V 28BYJ-48 stepper motors – these come in both 12V and 5V variants, so make sure you get the 5V version. You'll also need two ULN2003 driver boards – typically these come with the stepper motors.

The final electrical component is a 9g 'micro' servo.

The motors and servo should come with wires, but these wires will be too short for the plotter. Extension cables are available, but you can also cut the wires and solder on some additional wire if needs be.

That's all the hardware. Now let's mount it all together. →

Origins

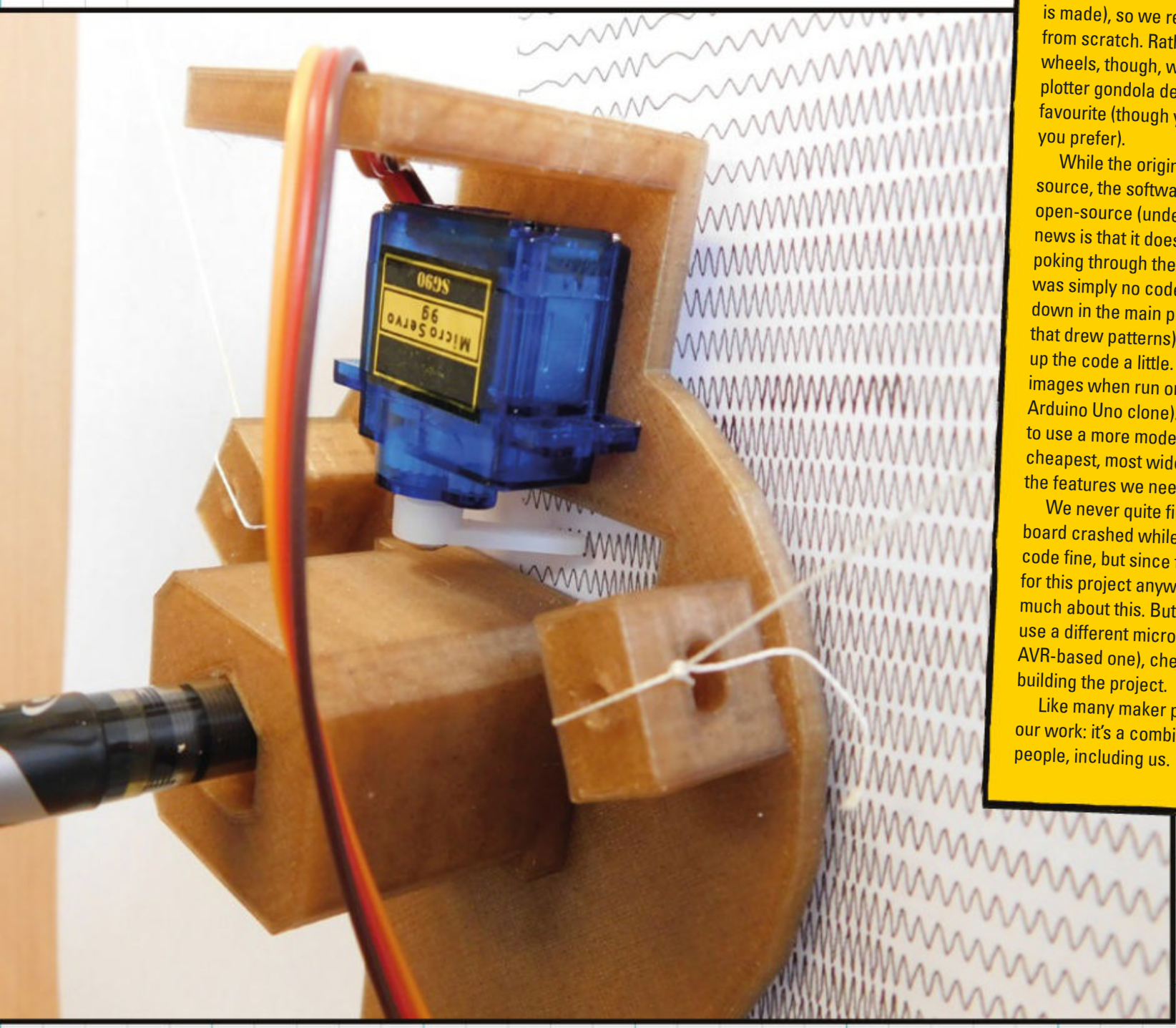
We didn't actually set out to design a plotter. We wanted to test out a cheap vertical plotter that is widely available on direct-from-China websites. However, when we received it, we found that, while it wasn't great – the laser-cut parts didn't fit together very well and the software basically didn't work – it did have a lot of potential. Rather than simply leave the review like this, we decided to get stuck in and improve it.

The hardware designs weren't available, and besides, far more people own a 3D printer than a laser cutter (which is how the original is made), so we redesigned the motor mounts from scratch. Rather than reinvent all the wheels, though, we scoured the internet for plotter gondola designs and selected our favourite (though you can use a different one if you prefer).

While the original hardware was closed-source, the software of the original plotter is open-source (under the MIT License). The bad news is that it doesn't work. At all. After a bit of poking through the code, we found that there was simply no code to move the pen up and down in the main program (only in the test code that drew patterns). We fixed that and tidied up the code a little. Also, it crashed on a lot of images when run on the supplied hardware (an Arduino Uno clone). Since it's 2022, we wanted to use a more modern microcontroller – and the cheapest, most widely available board with all the features we need is the Raspberry Pi Pico.

We never quite figured out why the original board crashed while the Pico ran the same code fine, but since the Pico is a better fit for this project anyway, we didn't worry too much about this. But be warned: if you want to use a different microcontroller (especially an AVR-based one), check it runs properly before building the project.

Like many maker projects, this plotter isn't our work: it's a combination of work from lots of people, including us.



Left ←
If you look closely, you can see that the squiggles aren't perfect, but does it really matter?

FEATURE

Assembling the PLOTTER

Putting it all together

Attach the motor mounts to the back-board. They should be in the top two corners. There aren't any hard and fast rules for how far apart they should be. In our setup, they're 43 cm apart. If you place yours the same distance apart, you can use the plotter without having to edit and recompile the source code.

While in theory, you should be able to place them almost any distance apart, we haven't tested what happens if you take this to extremes, so be prepared for some hiccups.

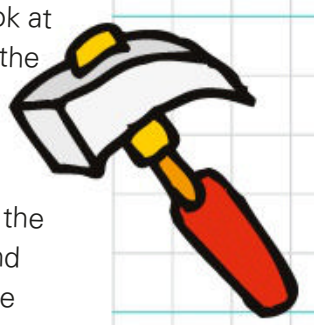
Attach the motors to the mounts using a pair of screws for each motor.

You can then push the pulleys onto the motors. It should be a snug fit, but you can add a few drops of superglue if you want to make sure they stay in place.

Superglue some thread to the pulleys and tie the other end to the gondola. The length of this will depend a bit on how far apart you have your motors and how far down you want to be able to plot. On our plotter, we have the motors 43 cm apart and the drawing area 40 cm high, so we need threads at least 45 cm long, but it's best to err on the side of caution here and go for around 60 cm.

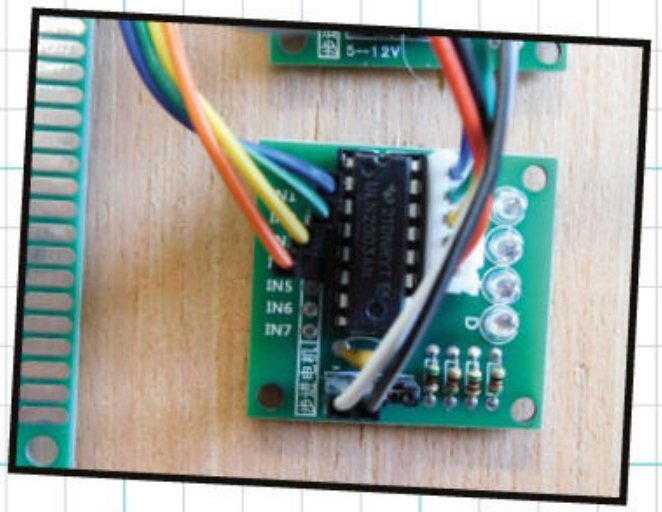
The gondola prints in one piece, but you need to attach a servo. This is pretty straightforward, but the one gotcha is that you have to make sure that the servo horn is in the right position. Since servos have a limited amount of movement, if the horn is in the wrong orientation, you won't be able to use it to lift up the gondola. For now, leave the servo unattached, and we'll look at how to get it on correctly once we have the electronics done.

Below
The angle of the motor mount helps reduce friction, but probably doesn't affect the machine too much



Adding electronics

There's two sections to the electronics – the input which comes in a set of buttons, and the output which is the connections to the motors and servo. Let's take a look at the input electronics first.



Above
28BYJ motors often come with simple driver boards that are suitable for our needs

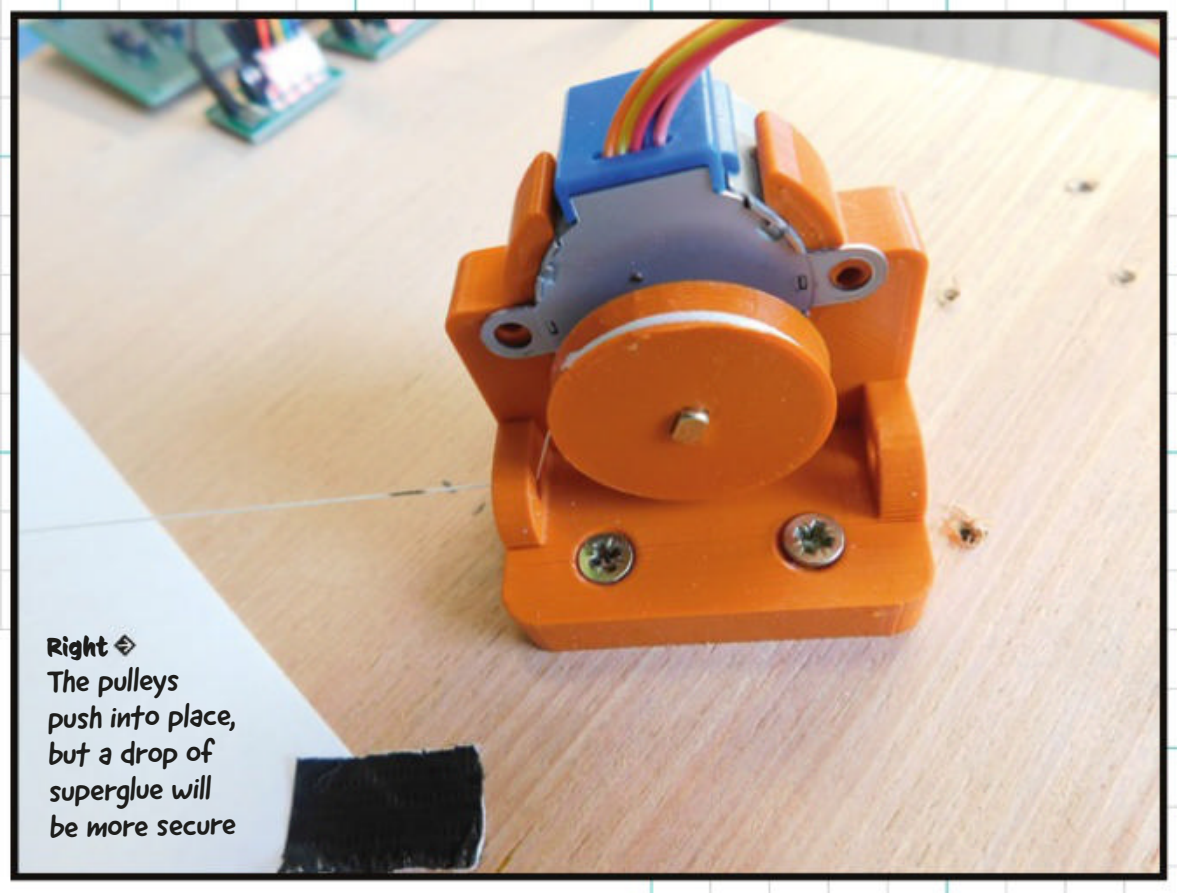
We mounted the Pico and buttons on a piece of protoboard, but you could use stripboard, design a custom PCB, or just use wires (though you'll need a way of mounting the buttons).

There are six buttons. Each one is connected between a GPIO pin and ground. The buttons are connected as follows:

Function	GPIO pin
Pen Up	16
Pen Down	17
Motor 1 Down	20
Motor 1 Up	21
Motor 2 Down	18
Motor 2 Up	19
Start	13

The motors are attached to the motor drivers via a cable that can only be plugged in one way around. This is usually about 20 cm long, so make sure you position the motor drivers close enough to the motors. There are four data connections between the motor drivers and Pico (your driver may have seven inputs, but you can ignore inputs 5-7). Inputs 1-4 are connected as follows:

Motor connection	GPIO pin
Motor 1 In 1	6
Motor 1 In 2	7
Motor 1 In 3	8
Motor 1 In 4	9
Motor 2 In 1	0
Motor 2 In 2	1
Motor 2 In 3	2
Motor 2 In 4	3



Right
The pulleys push into place, but a drop of superglue will be more secure

Each motor controller also needs a power connection. There should be a four-pin connector along the bottom. The rightmost two pins are a jumper to enable power to the motor – this jumper should be on. The leftmost two pins are – and +, and should be connected to ground and 5V (VBUS) respectively.

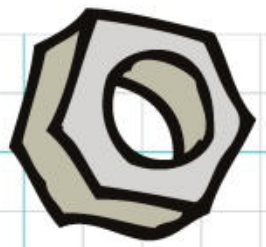
Your motor driver probably has pin headers attached. You can solder wires directly to these to connect to Pico. However, we attached pin headers to our protoboard and made the connection with socket headers.

The final electrical connection is to the servo. This has three connections which are typically brown, red, and orange.

- Brown connects to ground
- Red connects to 5V (VBUS)
- Orange connects to GPIO 14

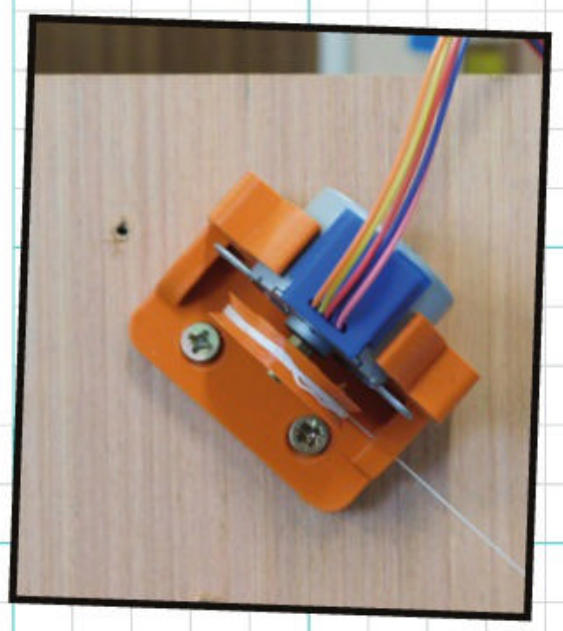
If yours are different colours, take a look at the manufacturer's website for details on the pin out.

You'll probably find that the servo's wire isn't long enough. You can get servo wire extension cables, or you can cut it and solder in place additional wire. →



Right
There are screw holes to hold the motor in the mount

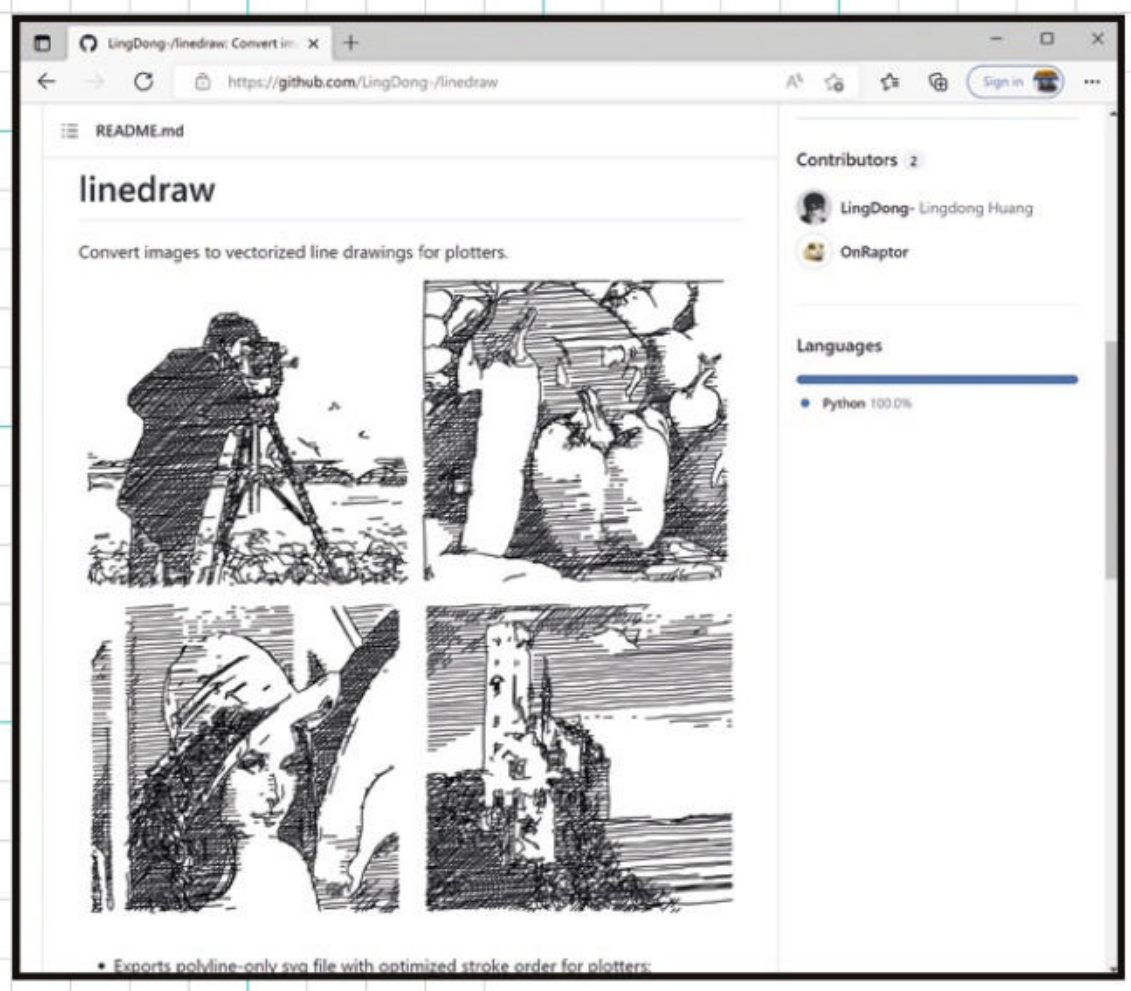
The final electrical connection is to the servo



FEATURE

Software

Let's get everything running



Above
LineDraw is a Python script that makes sketch-like images out of photos. You have to play with the parameters to get good results

The first thing to decide is whether or not you need to compile the firmware. If you've used the same GPIO connections as us, and made your plotter the same size, then there's no need to. You can load our UF2 file (download **WallDrawRP2040.ino.uf2** from **hsmag.cc/walldrawino**) directly on your Pico. Unplug Pico from your computer, hold down the BOOTSEL button, and plug it back in. It should now mount as a USB drive and you can drag-and-drop the UF2 file onto it. The firmware should now be loaded and you're ready to go. However, if you want to make changes, see the box on compiling the firmware (overleaf) for details.

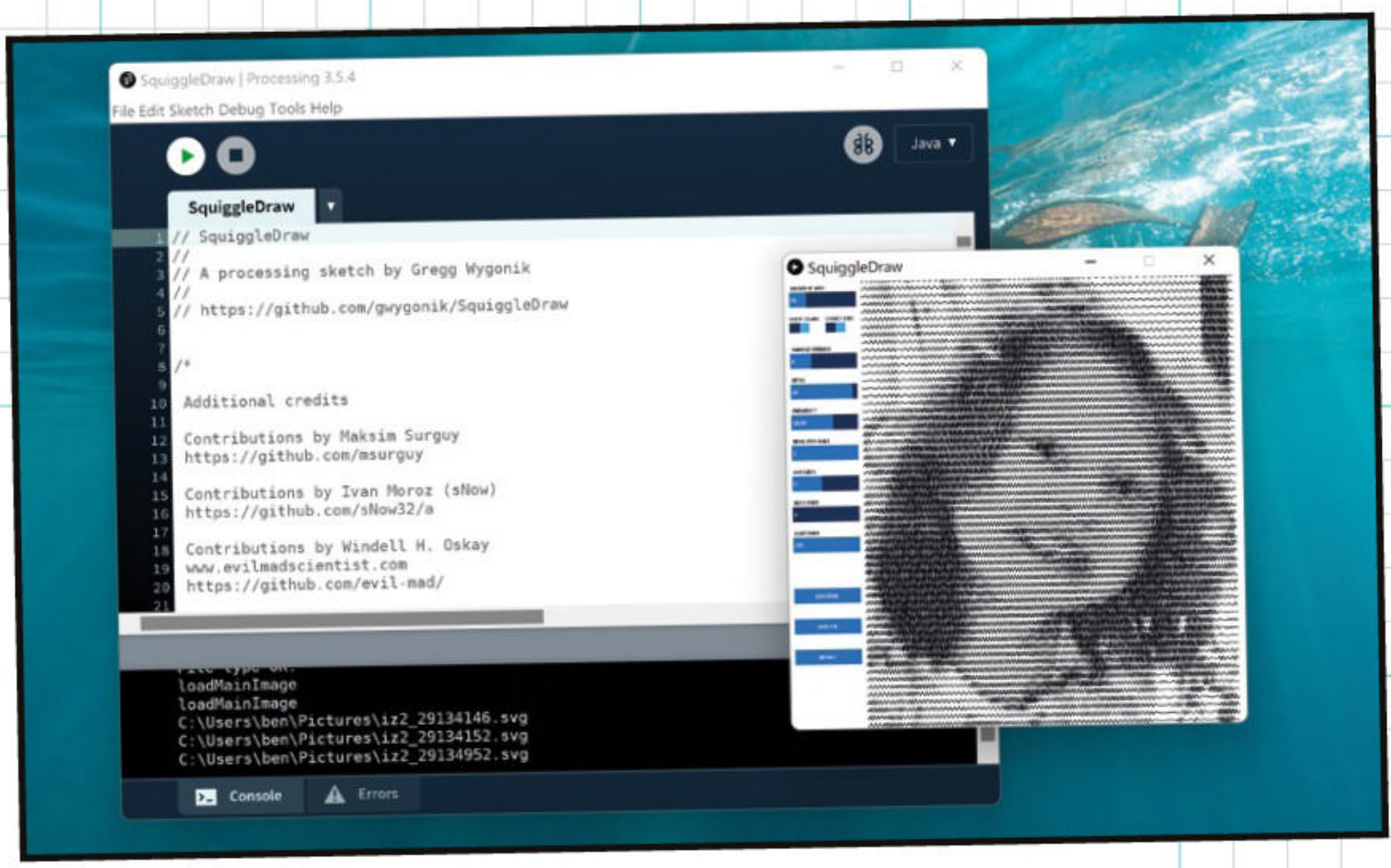
Once you have the firmware, you can attach your servo. First, plug Pico into your computer and make sure that the servo is connected to Pico. Then press the pen-up button on your control board. This should move the servo into the up position. If you don't hear it move, then press the pen-down button. If it doesn't move this time, then something's gone wrong – check the wiring and re-upload the sketch.

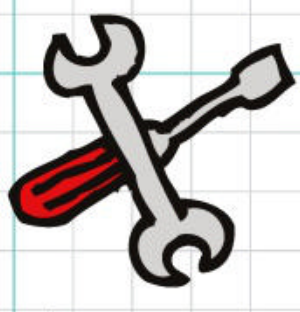
Once the servo has moved into the up position, attach the horn in-line with the servo. Now press the down button. This will rotate the servo – glue the servo onto the gondola using superglue (CA glue) so that the horn pokes through the servo slot when pen-down is pressed.

Now we just need something to draw.

When plotting, you need to make sure your artwork is in a vector format. This means that the graphics are stored as lines rather than as pixels. There are plenty of vector image editors around, but we like Inkscape. This is freely downloadable from **inkscape.org**. Creating images for plotters is a massive subject that we can't get into entirely, but remember that Inkscape can create images with both pixel (or raster) elements and vectors. It's important to make sure anything on the image is converted to vectors before plotting. In Inkscape, this means selecting it and clicking on Path > Object to path. There is a little more to it than this, and we'll look again at it later.

A slightly unusual aspect of our plotter is that the origin – point 0,0, is in the middle of the drawing area. In Inkscape, it's in the bottom left corner. If you want an image to be drawn centrally on your plotter, it needs to be





Left The SquiggleDraw program converts images into squiggly lines – ideal for plotters

half off the left of the paper and half off the bottom in Inkscape (see **Figure 1**, overleaf).

You don't need to worry about this too much for your first drawing though, because we've included some test SVG files that you can run.

G-code

This brings us to the next part of the software chain. The plotter takes G-code – the same file type that your 3D printer uses. This contains movements to tell the plotter where to move the pen. We need some software to convert the vectors from the SVG file to G-code that the plotter can understand. There are a few options for this, including some built into Inkscape, but the one we found most reliable was a Python script. You can download this from **github.com/benevpi/gcodeplot**.

You need to run this from the command line. You'll need Python installed on your machine. If you're using Windows, we recommend using Anaconda (**anaconda.com/products/distribution**) which helps manage the versions of the different packages that you have.

You need to run this from the terminal, so open a console (an Anaconda console if using Windows, otherwise an ordinary console) and navigate to the **gcodeplot** folder that you've just downloaded. The GitHub repository includes some test images, including the text from the front of this magazine. We've already generated the G-code for this for you, but you can create G-code from your own images with:

```
python gcodeplot.py --area=0,0,150,150
--lift-delta-z=4 --work-z=1 images\
buildaplotter.svg > buildplot.gcode
```

You can replace `images\buildaplotter.svg` with any file you want to plot, and this generates the file **buildplot.gcode**.

In the **gcodeplot** folder, you'll also find a the file **calibrate10×10.svg** which is a grid of 1 cm squares to help make sure that your plotter is running correctly.

The final piece of the software puzzle is the G-code sender. This is the bit of software that actually streams G-code to your plotter. There are quite a few different bits of software that do this, but the one we prefer is Universal Gcode Sender, or UGS. You can download it from **hsmag.cc/UGS**. This isn't designed specifically for plotters, and it can send G-code to any machine that understands this language, including 3D printers and CNC machines.

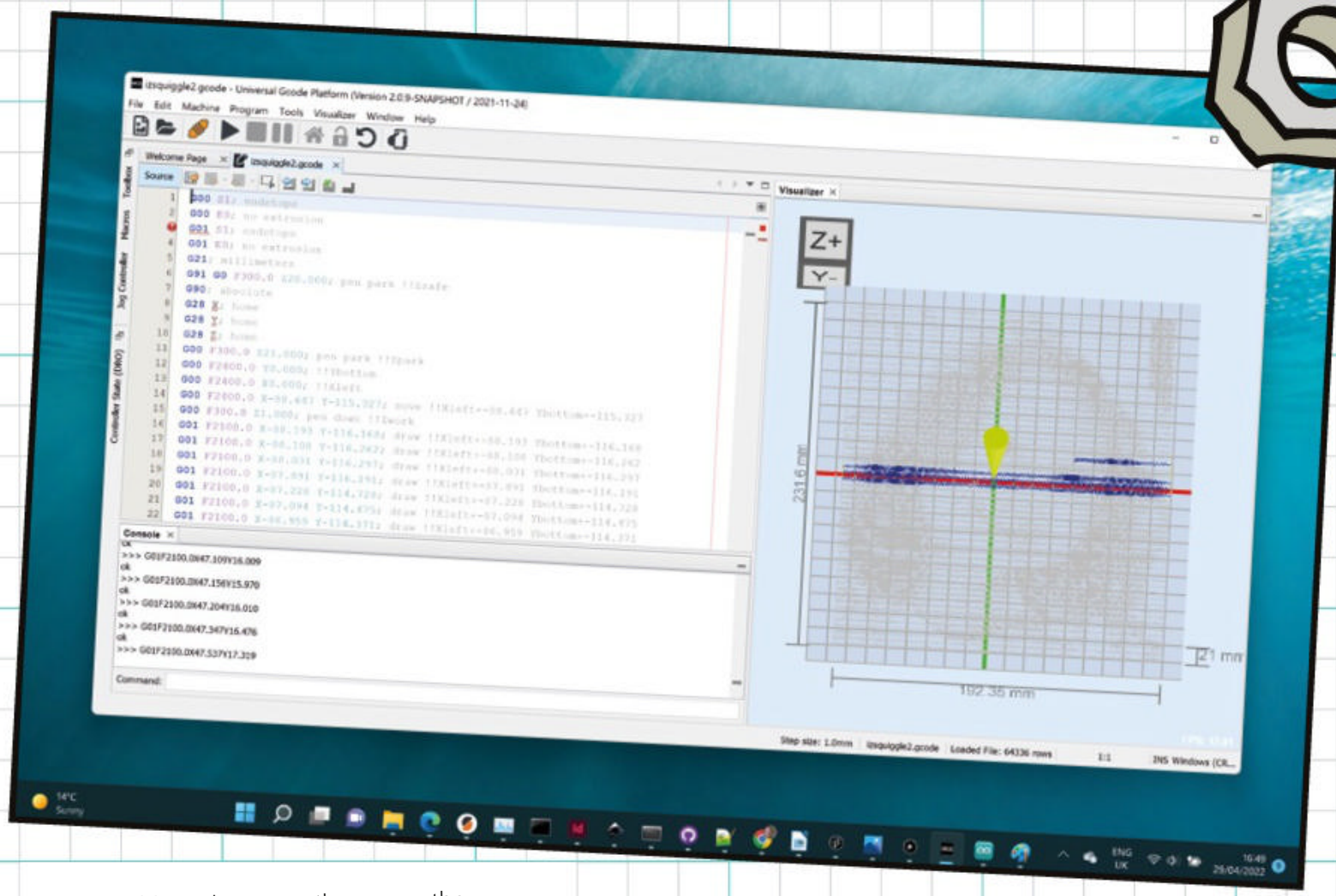
Download the zip file from that website and extract it. Inside the directory, you need to find the **bin** directory for your platform. For 64-bit Windows, this is UGS-platform-win > bin > ugsplatform64.exe. →

We need some software to convert the vectors

Motor turning the wrong way

If you find a motor turning the wrong way, this is really just that the string is wound around the spool the other way. After powering on the plotter, you can use the jog buttons to unwind all the string and then wind it back in the correct way.

FEATURE



Right Universal Gcode Sender is an easy-to-use program for feeding G-code to the plotter

Open this and you can then open the G-code file you created earlier (or the one we built that's in the **gcodeplot** directory).

Now is also a good time to add a pen to the gondola and secure it in place with the grub screw. You need to position it so that it just pokes out of the gondola plate, but not so much that it touches the paper if the servo is in the pen-down position.

When you first power on the plotter, you can move the motors using the buttons to get the gondola in the 'home' position. This is exactly halfway between the two motors (21.5 cm from each one), and 20 cm below them. It's important that the gondola is exactly here because the maths that control the motion depend on it. If it's not, then you'll get distorted prints. We'd recommend taking some time to measure out the exact place the first time, and then marking the string at the point it enters the motor housing so it's easy to find the next time.

Now place some paper so that the gondola is in the middle of the paper, and click on the Connect button in UGS. If it connects properly (you might have to ensure you're connected to the correct serial port), you're almost ready to go. Press and hold the Start button for about a second and you should see the lines:

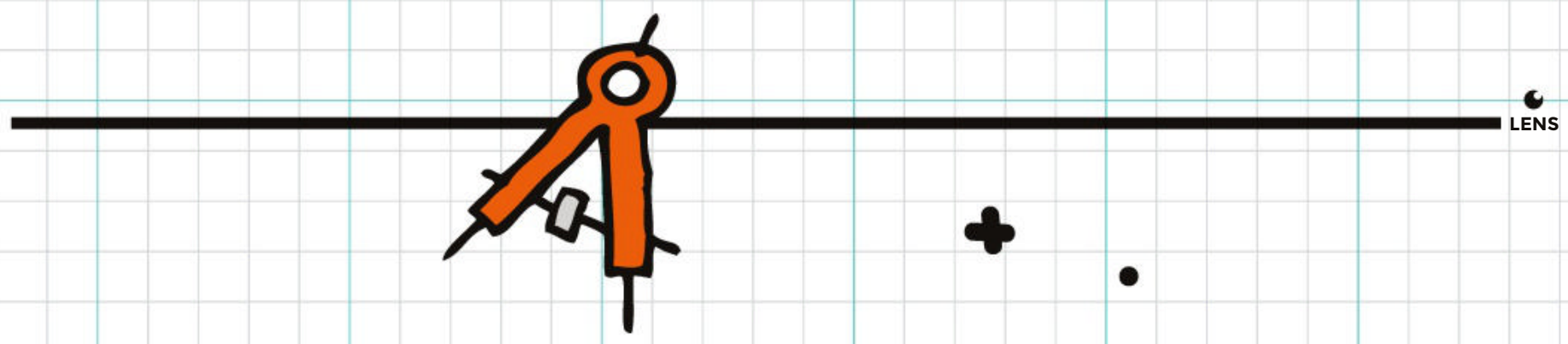
```
Grbl 1.1h ['$' for help]
<Idle|MPos:0.000,0.000,0.000|FS:0,0|
Ov:100,100,100>
```

These are all lies – it's not running GRBL, you can't press $ for help, and none of the information in the second line is correct. However, this doesn't matter: it just ensures UGS knows that it's ready for action.

If you've made it this far, congratulations, it's all about to happen. Click the play arrow to start sending your G-code to the plotter.

It should draw the image and then pause. You have to reset it (power it off and on, and manually rehome the gondola) each plot.

Congratulations, you now have a working plotter, but what are you going to plot? Plotters draw lines, but many of the digital images we produce are made up of pixels. If you want to plot a photograph or other pixel-based image, you first need to convert it to lines. This isn't a particularly straightforward process, and it won't work with all images. In fact, you'll probably only get a satisfactory result from a few images. Before we look at techniques, let's take a look at a few things that can make an image vectorise well or badly.



- **Simplicity** – The more going on in an image, the harder it is to vectorise. This is especially true for things that aren't the subject of the image – if there's a busy background, this can make things particularly challenging. You could preprocess the image to remove the busier parts of the background using a normal image editing tool.

- **Colours** – It's possible to plot multiple colours by creating different images and changing pens, but it isn't particularly easy. If the image relies on colours to make it work, it's going to be a challenge. If you're not sure, convert the image to black and white first to see how it looks.

- **Bold lines** – If the image is made up of bold lines, then it will probably work well. If it's reliant on more gentle contours, then it might not.

There isn't a simple 'right way' of converting a pixelated image to a vector, because they're not directly comparable. Instead, you have to use some artistic licence to do it.

There are a few approaches you can use.

- **Bitmap tracing**
  In Inkscape, you can load a bitmap and go to Path > Trace bitmap. The results can vary a lot and there are some parameters to adjust to try and get the effect you're looking for.

- **SquiggleDraw**
  (**github.com/gwygonik/SquiggleDraw**)
  This draws a lot of horizontal lines with the same separation but, in darker areas, the lines start to oscillate as sine waves to colour the area in.

- **Stipple Gen**
  (**wiki.evilmadscientist.com/StippleGen**)
  This turns a drawing into a series of dots in a way that is reminiscent of, but slightly different to, pop art.

- **LineDraw**
  (**github.com/LingDong-/linedraw**)
  This creates a sort of pencil sketch effect where darker areas are filled in with a slightly haphazard cross-hatch effect.

Now, it's over to you.

Compiling the firmware

Before running the plotter, you need to make sure that the software is correctly configured for your setup. This is done in the Arduino IDE. You can download the sketch from **github.com/benevpi/Walldraw.**

To compile this, you need to have the Raspberry Pi Pico board definitions installed. Unfortunately, this doesn't include a compatible version of the Servo library, so you also need the Raspberry Pi Pico build from **github.com/earlephilhower/arduino-pico.**

This might lead to a slight confusion if you already have a Raspberry Pi Pico definition installed, because you'll have multiple board options all called Raspberry Pi Pico. The easiest way is to pick one and if you get an error that Servo.h doesn't support the platform you're using, pick a different one.

We found the Arduino IDE quite unreliable for directly programming Pico. Instead, we exported sketches using Sketch > Export Compiled Binary. This will save a UF2 file inside the Arduino sketch directory that you can use to program Pico. Unplug Pico from your computer's USB, then hold down the BOOTSEL button as you plug it back in. You should get a new USB drive appear in your computer's file manager and you can drag-and-drop the UF2 file directly onto it.

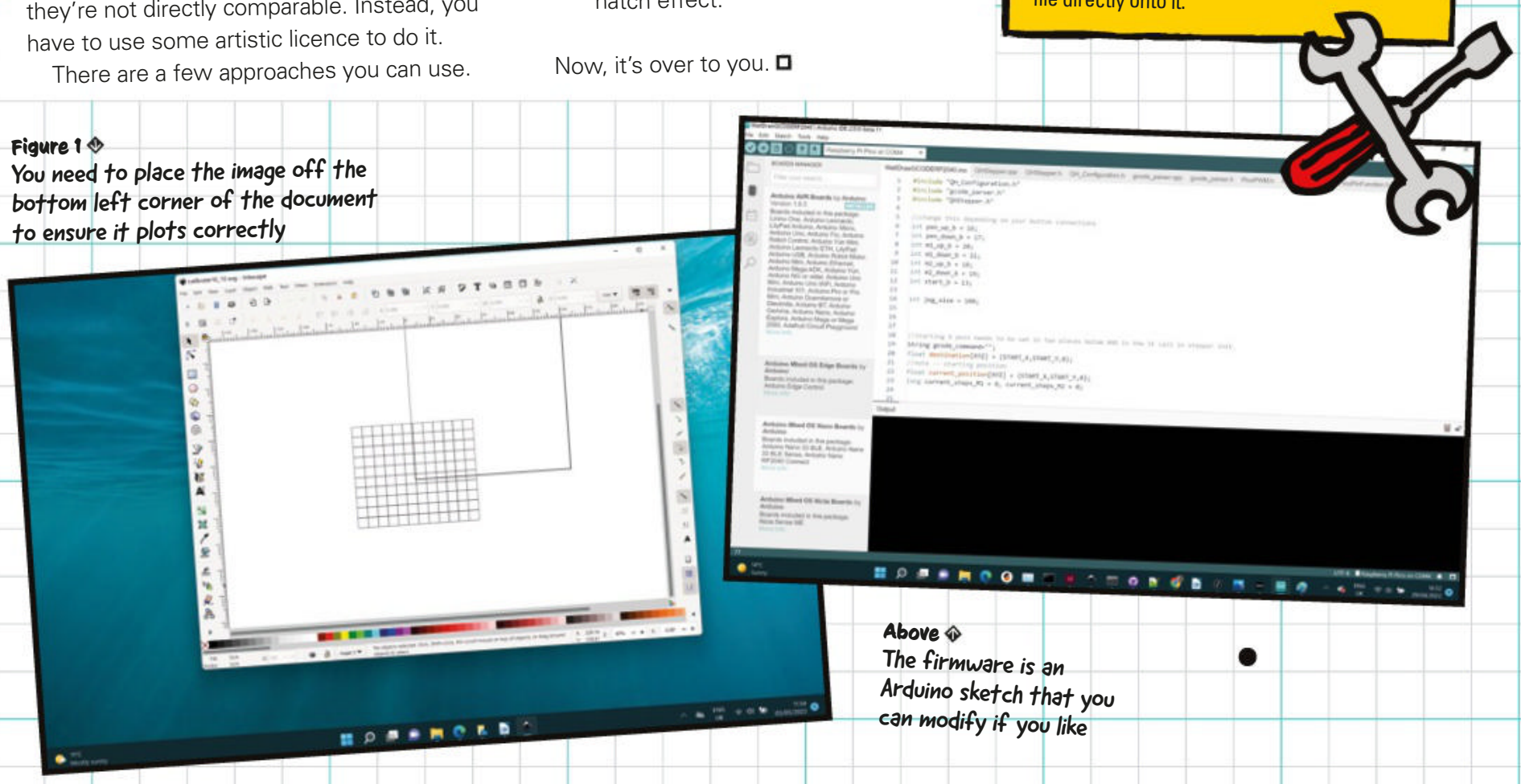


Figure 1
You need to place the image off the bottom left corner of the document to ensure it plots correctly

Above
The firmware is an Arduino sketch that you can modify if you like

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

Buy now: hsmag.cc/subscribe

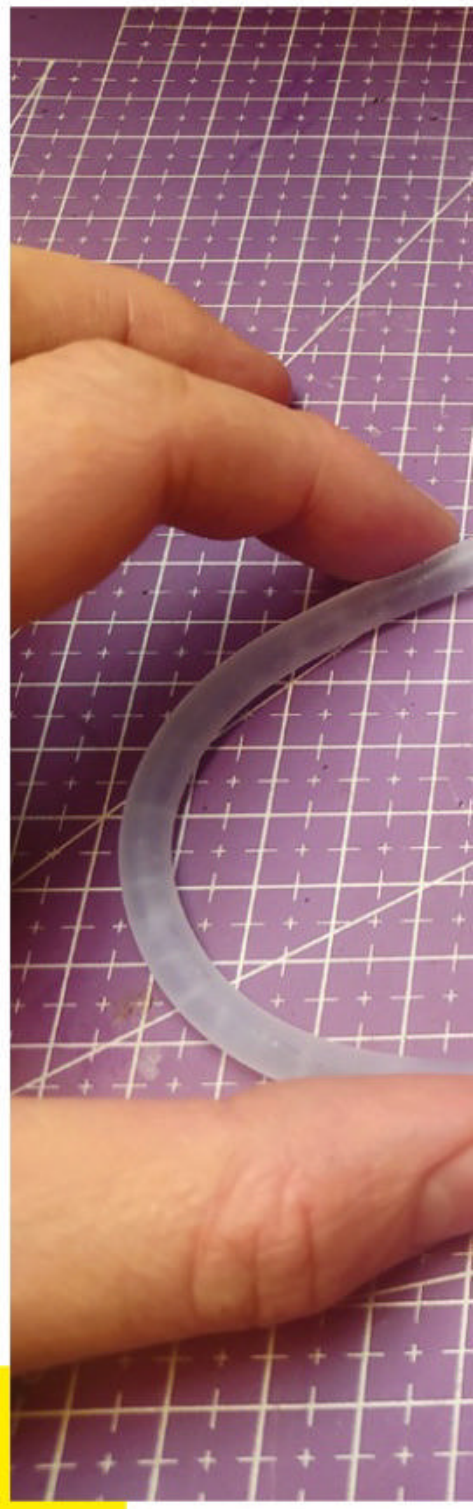
Free Pico with print subscription only

SAVE 44%

FEATURE

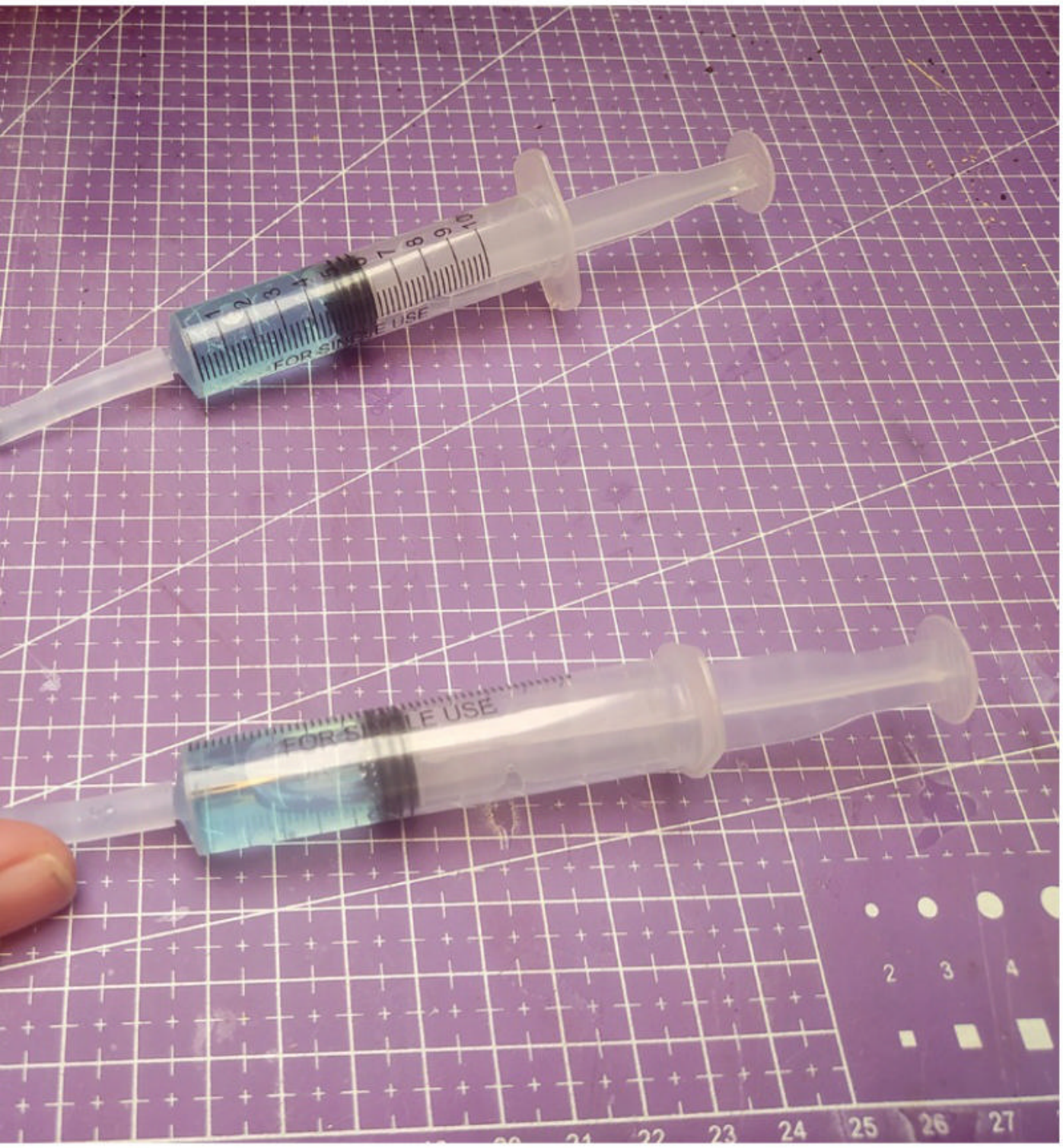
HOW I MADE

By Jo Hinchliffe



BUDGET PNEUMATICS AND HYDRAULICS

Using cheap and widely available syringe drivers and some aquarium tubing, let's create some simple pneumatic and hydraulic system experiments



What I used

- 10 ml syringes
- Aquarium tubing
- Aquarium valves; one-way and control types
- Aquarium tube T connectors
- Tongue depressors
- Barbecue skewers
- Nuts and bolts
- Scraps of plywood, or similar

Figure 1
Simply connecting two syringes with a short length of tubing creates a basic pneumatic or hydraulic system

When it comes to creating motion in making, we are spoilt for choice. In systems driven by electronics, our go-to these days are servos which are cheap and available, and can easily create a range of motion. Outside of electronics, there are options as well – mechanical linkages and gears, pushrods, and cams are all great fun to explore. Then, outlying a little from the centre ground, are pneumatics and hydraulics.

A really simple and affordable way to play and experiment with pneumatics or hydraulics is to pick up a pack of syringe drivers and some tubing (**Figure 1**). Syringe drivers are the plastic bit of a medical injection system, and you don't want or need the pointy needle end! Widely available online, or possibly from your local pharmacist, syringes are quite useful objects in your hacker lab, even before you want them for hydraulic experiments. They see regular use for moving liquids, applying light oils, refilling printer cartridges, small suction devices for picking up difficult objects, and, of course, measuring fluids.

Syringes come in a range of sizes and we picked up a pack of ten 10 ml syringes as well as an enormous 350 ml syringe for our experiments. The smaller syringes all tend to have a similar size part for the output that is around 4 mm outer diameter, with around a 2 mm internal hole. Even our giant 350 ml syringe had an adapter which, when push-fitted, meant it had this size nozzle. This uniformity in nozzle size is excellent, as it means we can easily buy tubing to connect up our pneumatic and hydraulic experiments.

A great source of not only compatible tubing, but also other compatibles like valves and connectors is aquarium supplies (**Figure 2**). Lots of the airline systems for filtration and aeration in aquaria happen to use similar 4 mm internal diameter tubing, and these work just as well in our experience when passing air or fluids.

As a first experiment, grab two syringes and push on the syringe plunger all the way down, and move the other one all the way out. Connect the two syringes with a short length of tubing (**Figure 1**). This is the most simple example of pneumatics. Push down the syringe plunger and observe the other syringe moving upwards. It's so simple, but yet somehow very pleasing. So, if pneumatics are so easy to create,

FEATURE

QUICK TIP

It's worth working over a tray, in a sink, or even outside whilst setting up your hydraulics experiments.

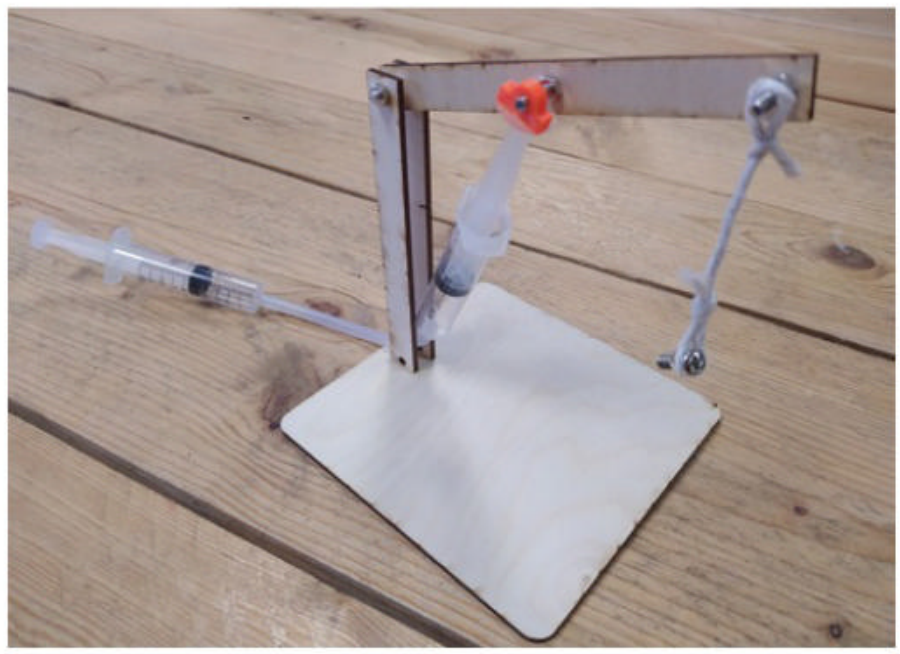


Figure 3 ↑
A simple crane example made from a few plywood scraps

why bother with hydraulics? Well, repeat the experiment but, this time, add some resistance to the other syringe. You should find that you can force the syringe to stay closed, despite compressing the other one. This is due to air being pretty easy to compress. Hydraulics exist as, usually, adding liquids instead of air into a system makes the system harder to compress and you can transfer more force. In industrial hydraulics, special oils fill the systems that greatly resist compression and expansion and can result in very high-pressure systems. Whilst you're welcome to experiment filling your systems up with oil, we went with regular tap water! We've added a little food colouring, just to make it easier to see in the images.

> *" WE BEGAN WITH A SIMPLE CRANE IDEA"*

Repeating our simplest experiment with two syringes and a tube, we filled the syringe and tube with water. It's now harder to compress both ends of the system, and it's capable of applying a lot more pressure.

With just this simple setup, there are lots of interesting fun and, dare we say, educational projects we can build. We began with a simple crane idea (**Figure 3**). Whilst we have used a laser cutter to make the pieces for this project, it would be really easy to make the small wooden pieces by hand with a small saw and a drill. The crane assembly is pretty straightforward. We made

ACCESSORIES

One of the great benefits of using syringe drivers for hydraulic and pneumatic experiments is that they are cheap and available and have pretty regular dimensions. We've mostly used 35 ml capacity syringe drivers and we bought a pack of twelve online. The body of the syringe, conveniently, isn't tapered and, on our pack of syringes, was 16.5 mm in diameter. With such regular dimensions, it is easy to create various accessories. One useful idea, if you create an experiment with numerous syringes, is to create a handheld controller. A simple pair of pieces of wood with 16.5 mm holes can suffice, and slotting in three syringes makes a super-quick handheld control device.

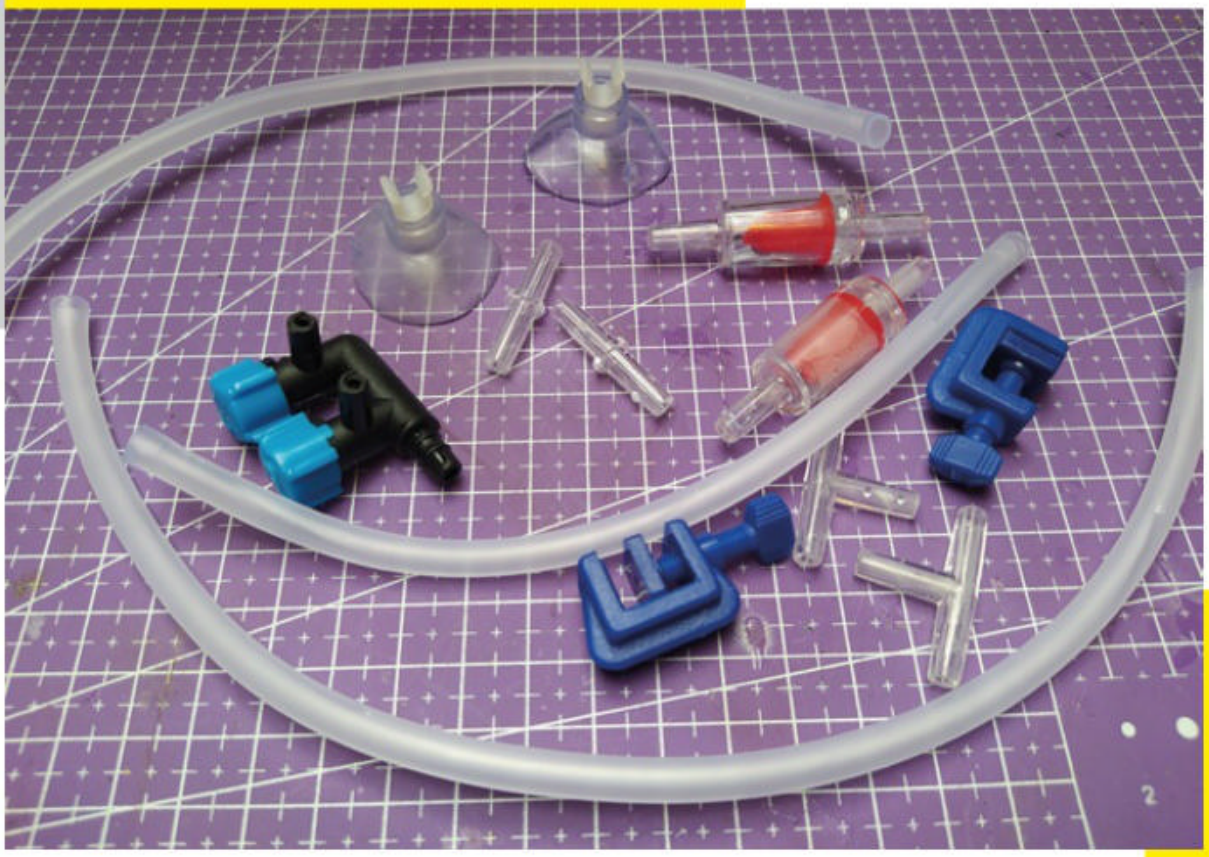


Figure 2 ←
The aquarium section of a local pet store will probably have lots of useful accessories including tubing, one-way valves, control valves, and more

two uprights that we glued into a small base. Using a few M4 nuts and bolts, we added a pivoting arm that would be our articulated crane boom or jib. Part-way along the boom arm, we placed a hole for another nut and bolt and quickly designed and 3D-printed a glue-on hole attachment for a syringe plunger. Again, you could easily make this type of plunger adapter with a scrap of wood and some glue if you don't have access to a 3D printer.

Having positioned and attached our syringe at the arm end of the crane, we then placed the tube onto the syringe and passed it through the legs of our crane. This loosely traps the syringe in position, whilst allowing it to pivot freely as the crane boom moves. We attached the controlling syringe to the other end of the system, having removed its plunger. This design, similar to all our other experiments, works well as either an air pneumatic or a water-filled hydraulic system.

Another great example of a mechanism that works well with a small syringe system is a small scissor lift table (**Figure 4**). A scissor lift table has this great attribute that you can amplify a small amount of movement on the syringe actuator into quite a large movement in the system. We went low-tech with our scissor lift build, using the wider tongue depressor-type lollipop sticks, some M3 nuts and bolts, and some bamboo barbecue skewers.

To begin, we taped a stack of eight of the tongue depressors together, making sure that they were all lined up well with each other. We then drilled three holes through all the sticks – one hole in the centre, and a hole at either end. We don't need to be massively accurate with this. We drilled all the holes with a small 1 mm bit as a pilot before stepping up to 3 mm drill bit.

Next, add some nuts and bolts to connect the tongue depressors into pairs. Then, in turn, connect two sets of two pairs to create the two sets of scissors that will be either side of our scissor lift (**Figure 5**). It doesn't make much difference how wide we make our scissor lift, so we cut some lengths of the barbecue skewers to around 80 mm. We needed to expand the four holes from 3 mm to 3.5 mm for our skewers, but then they were a pretty tight fit and didn't need any glue. You could easily make this by gluing the four skewer cross members into place, avoiding the need to accurately size the holes. Once you have the cross members in place, enjoy playing with the assembly – it's an addictive mechanism to play with!

To finish the scissor lift example, we need to trap one of the cross members in a way that it can still rotate, but doesn't move. After considering numerous ideas (gluing tabs to a base with holes in to act as bearings to receive the cross member), we realised that simply cable-tying the cross member loosely to a base gave us good results! We then added a syringe connected to the free cross member, and attached it to the base plate. Again, we've used some 3D-printed parts here, but this would work

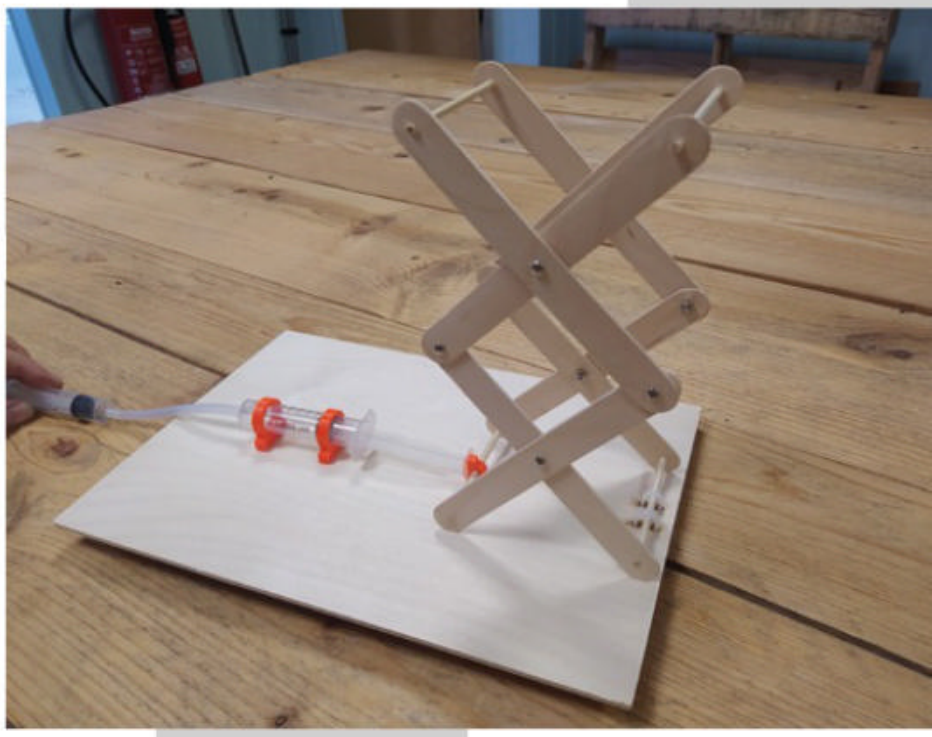


Figure 4
A fun project, with a fascinating movement, is to make a scissor lift system

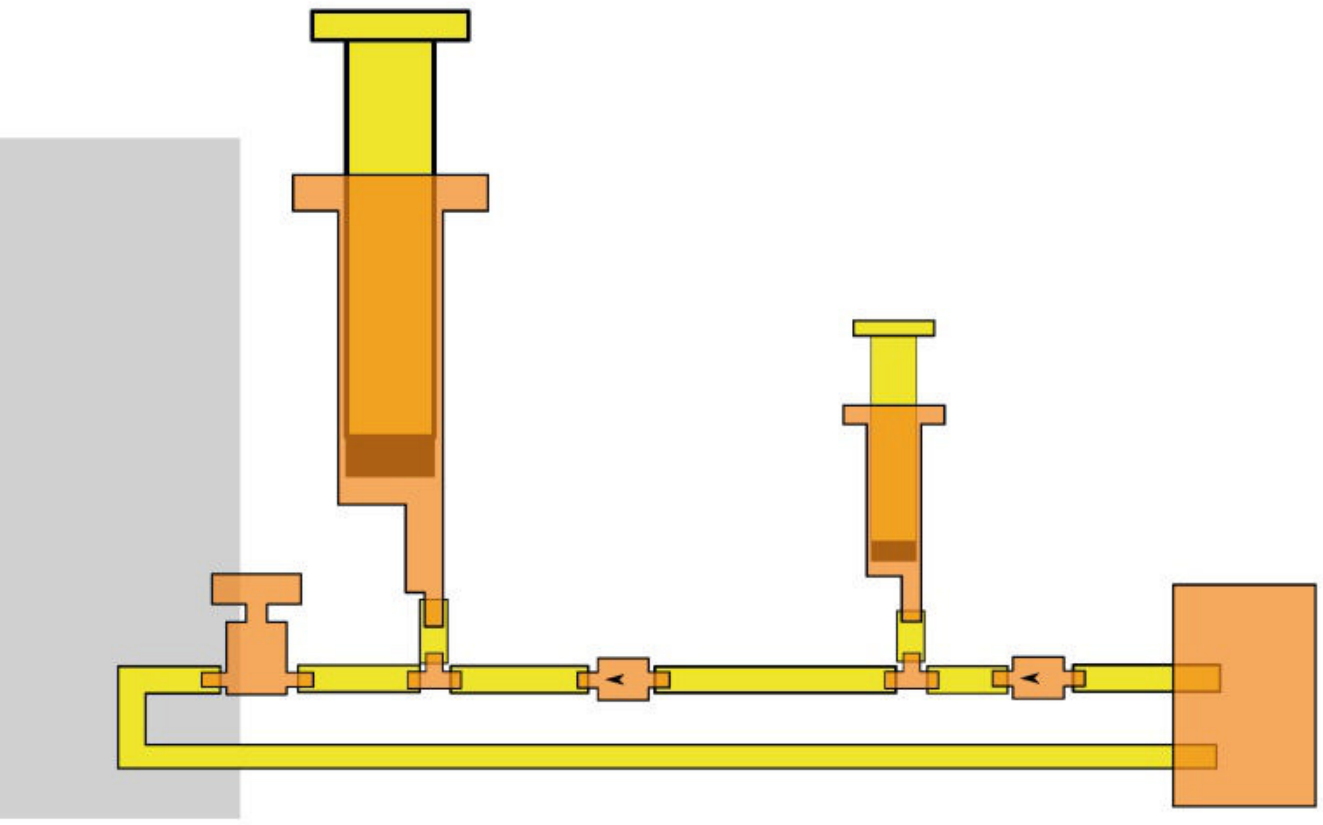
Figure 5
Drilling all the tongue depressors at once makes this a quick build process

Figure 6
Simply using M3 nuts and bolts to create the pivot points in our scissor lift

QUICK TIP

Blowing through aquarium check valves can help you identify which way they allow air or liquid to flow.

FEATURE



just as well if you hot-glued the syringe to the project's base. Our scissor lift worked well pneumatically, but was slightly better with hydraulics. When running only with air, the scissor lift sometimes didn't retract fully, as the friction in the system overpowered the pressure in the syringes. This effect disappeared when we used water. It would probably work a little better with a weight on top of the scissor lift, and it would be fun to add a little table top to the model at some point. We imagine a plate with two slots underneath it, that allow the cross members to travel back and forth, would work well.

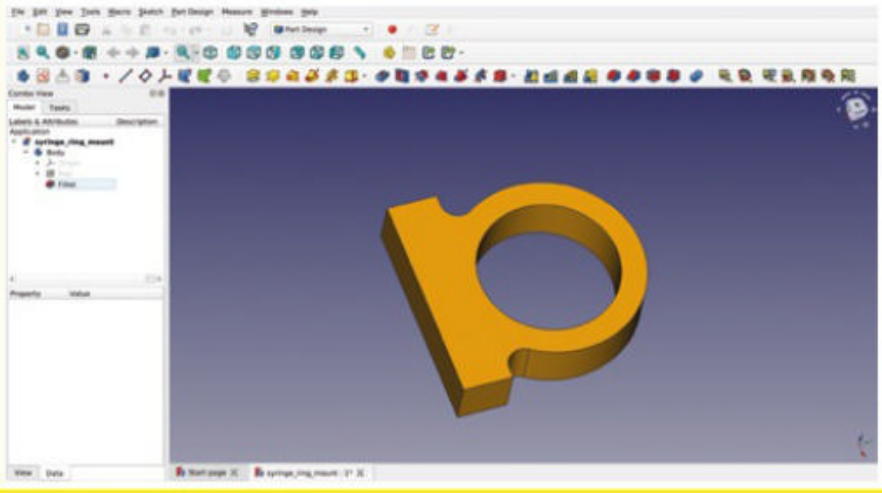
So far, we've stuck to simple direct connections of same size syringes. Adding more lengths of tube and some different components, you can create an interesting hydraulic jack-type system where you can create a much larger movement, in a larger syringe, using multiple pumps of a smaller syringe. See the general layout of our system in **Figure 7**.

Figure 7
A more complex experiment – creating a multi-stroke hydraulic system that uses a small syringe to pump up a huge syringe

Figure 8
Our large 'jack' system, set up using syringes and aquarium accessories

3D DESIGN

You'll notice that, in a couple of places, we have used 3D-printed accessories in our example hydraulic models. Whilst you definitely don't need to do this, 3D design and printing is a really useful approach to make custom parts that fit onto syringes. We, of course, have used FreeCAD to do this and our long-running series of FreeCAD tutorials have been collated into a free-to-download book. If you work through the first couple of chapters of the book, you'll have all the skills you need to design the accessories we have made, as well as any others you can imagine. The book will be available soon on the HackSpace website, **hsmag.cc**.



QUICK TIP

When drilling tongue depressors, drill a small pilot hole before stepping up to a larger hole, otherwise they may split.

Looking at **Figure 7**, moving from right to left, the first thing you see is a reservoir for our 'hydraulic' fluid. This can be any container with some water in it – we simply used a plastic bottle. Into this is inserted a tube and, moving to the left, we can see that water brought into this tube meets a one-way valve through which it can flow into the system, but not back into the reservoir. After the valve, the next item is a T piece. This T piece then has a short section of tube which couples it to the small syringe. Moving further left, the T piece has another short section of tube which couples it to the next valve. Again, this second valve allows the water to pass through it, moving to the left, but doesn't allow water beyond it to travel back toward the small syringe. If you set the system up to this point, you should be able to pull back the small syringe which will pull some water from the reservoir into the system, and plunging the syringe will push the water through to the left – essentially we have created a small pump. When the control valve after the larger syringe is closed, water pumped through the system gets pumped into the large syringe, causing it to rise. Then, to reset the system, you open the control valve and push the large syringe back down, sending the water back into the reservoir.

We assembled our system outside as there is a larger amount of water in play and, after checking that the pump part of our system worked, we connected another T piece and tubes to attach the larger syringe. On the output, we used one of our aquarium control valves that allows us to seal and, in turn, open this part of the system. With this valve closed, we could pump our hydraulic fluid into the larger syringe, using multiple strokes as described. It's a great experiment

plunger has a rubber piston on the bottom that creates the seal needed for the syringe driver to work. We removed the plunger and removed the rubber piston, and then measured the moulding of the end of the plastic plunger to then try and replicate this fitting into the piston. We cut a disc from some 1.5 mm plywood on the laser cutter, and then a thicker 3 mm stand-off disc. This replicated the bottom of the original plunger and gave us a platform to build onto. We glued the long rack gear to this platform and then we could refit the rubber piston. We then checked that this worked as a plunger back in the syringe body manually inserting and retracting it.

At the other end of the syringe body, we created a small wooden plate to glue to the top of the syringe which has a slot in that is just wide enough for the rack gear to pass through. We needed to position this wooden plate so that the rack gear remained vertical and centralised inside the syringe to enable it to travel the full length of the mechanism. The back of the slot acts as a support to keep the rack gear in position. We then glued a few shims onto the wooden plate to place an N20 micro metal gear motor into a reasonable position for our 16-tooth pinion gear to engage with the rack. For this prototype, we just used a small spot of hot glue to hold the motor in position for testing. Whilst this is slightly crude, it's a good working prototype that we can carefully measure and find some dimensions from to create a 3D model of a motor mount that we can 3D-print (**Figure 10**). Wiring up the motor to a power supply, we simply ran the motor backwards and forwards to test how well the system works. Of course, in future prototypes, we could opt to use a stepper motor for positional control, or we could use some kind of limit switch to stop the system at either end. Or, perhaps, an encoder system to enable accurate positioning. As an aside, motor-driven syringes are being used in DIY submarine rover builds where they act as chambers that can be flooded and emptied to change the buoyancy of a vehicle – a very tempting project rabbit hole!

We've really enjoyed playing with these simple ideas – it's an affordable way to get some powerful movement into mechanisms. We hope to include some of these ideas in future projects, and we hope we've tempted you to experiment with them too.

" WE ASSEMBLED OUR SYSTEM OUTSIDE"

and when you make this larger syringe rise, you have essentially created a much larger actuator for your experiments (**Figure 8**). Whilst we pushed the larger syringe plunger back down, you could rig this system into an assembly where you could balance weight on the larger syringe, or lift an object (don't try this with your car, but think in similar terms to a car jack). We imagine you could create a system where, on opening the control valve, the larger syringe will be pushed back into the start position by the weight of the load. It's a very satisfying and fun system to play with – getting around 230 mm of movement from an approximately 40 mm stroke length syringe is great.

We started these experiments enjoying the idea of creating actuated mechanisms without using electronics, but we found ourselves considering how cool it would be to have the power of pneumatics and hydraulics built into systems using microcontrollers. As a final experiment, therefore, we prototyped an idea for a motor-driven syringe system. We used the excellent open-source Inkscape to generate a rack gear and a small pinion. The syringe

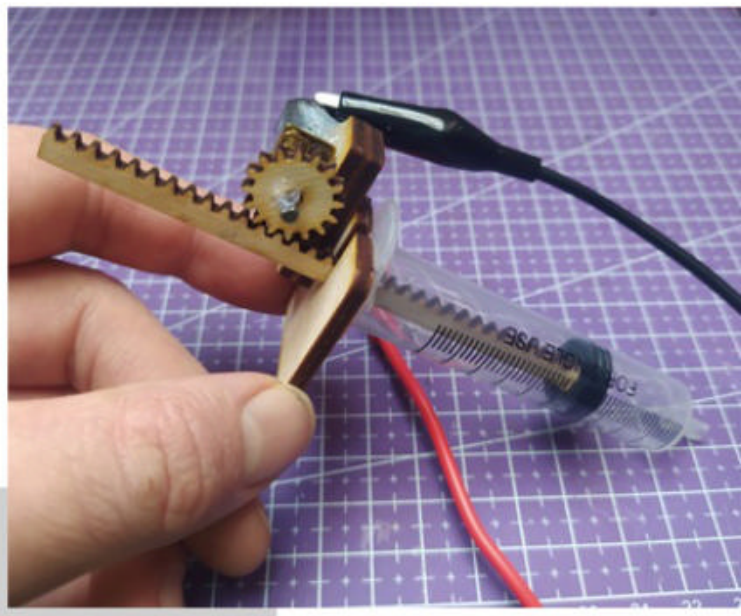


Figure 10
Our finished prototype using an N20 micro gear motor to test the rack and pinion syringe system

Figure 9
We created a stack of small discs that emulated where the plastic plunger inserted into the rubber piston and attached a rack gear

QUICK TIP

We covered using Inkscape to generate rack and pinion gears in a tutorial on gear principles in issue 54.

Interview: Jason Coon

He crafts bits into atoms and photons

Jason Coon (aka EvilGeniusLabs.org) makes delightful objects. His work with LEDs is, on the surface, really simple – they're just components arranged on to a pattern, after all – but that hides an ever-changing complexity that evolves before your eyes, creating new, organic shapes all the time, until before you know it you've lost half an hour to the glory of procedurally generated Fibonacci animations. Here's why and how he does what he does…

> I really didn't do anything with electronics or hardware for quite a long time, **until the advent of Arduino and Raspberry Pi**

"By day I'm a software developer. By night, I make stuff. And I have for a long time: I got started with Arduino, probably seven or eight years ago or so, after being away from electronics for a long time. I actually took four years of electronics in high school; I went to a vocational technical high school that had a four-year electronics programme. I graduated from that and started to go get my electronic engineering degree, but ended up dropping out and switching to computer science.

"I really didn't do anything with electronics or hardware for quite a long time, until the advent of Arduino and Raspberry Pi, which sparked my interest to get back into it and get closer to bare metal. So I started making stuff and eventually people started wanting to buy some of the stuff that I made, and that's when I started Evil Genius Labs.

"I actually started working with Louis Beaudoin, who runs Pixelmatix and Embedded Creations. He wanted to do a Kickstarter for RGB LED matrix displays, and the Kickstarter was called Smart Matrix. Originally it was like a shield for the Teensy from PJRC, to drive the non-addressable RGB LED matrix boards, and he still sells those I think – Adafruit and Pimoroni and several other places still stock and sell those. But we did a Kickstarter that was less of a kit and more of a fully featured RGB LED matrix display, with an SD card slot so it could play animated GIFs right off the SD card; it had a microphone and an audio line-in jack and it could do music visualisations. He was the hardware guy and I mostly stuck with the software, writing code for it. But that really kicked me off down my descent into Arduino and LED madness. →

REGULAR

"With Louis's help, I designed my first printed circuit board – it was all through-hole components, and it was just a level shifter shield for driving addressable RGB LEDs that want 5V logic level from microcontrollers that use 3.3V logic level – for example, the ESP8266 is one of my favourite boards that I use. That was the very first thing that I started selling on Tindie.

"At the time, I was an active member of the FastLED community. And at the time, they were on Google Plus, which tells you how long ago it was. FastLED is a library for driving addressable RGB LEDs like NeoPixels from a wide variety of microcontrollers; it focuses on super-efficient discrete integer math as opposed to floating-point math, which you just can't do on some of the smaller, less powerful microcontrollers.

"At the time on the FastLED community, I saw a post by Jim Bumgardner, who had made a Fibonacci LED display. Actually, he had simulated it in Processing, and then printed it out on a sheet of paper and drilled – I think – 100 holes in a piece of particle board and then pushed 8mm WS2811 addressable RGB LEDs through, and that was the first display with LEDs laid out in a Fibonacci distribution that I'd ever seen. My head just exploded.

"I had to build my own, and so I did the same thing, drilling holes and writing some scripts in Python. This first one I drilled in just a thin piece of sheet metal, with 8mm RGB LEDs, and it has a Teensy in the back for driving it. That was my first Fibonacci display.

> I'd call myself a tinkerer or a maker maybe, but I've had **no formal post-secondary or college training in math or art**

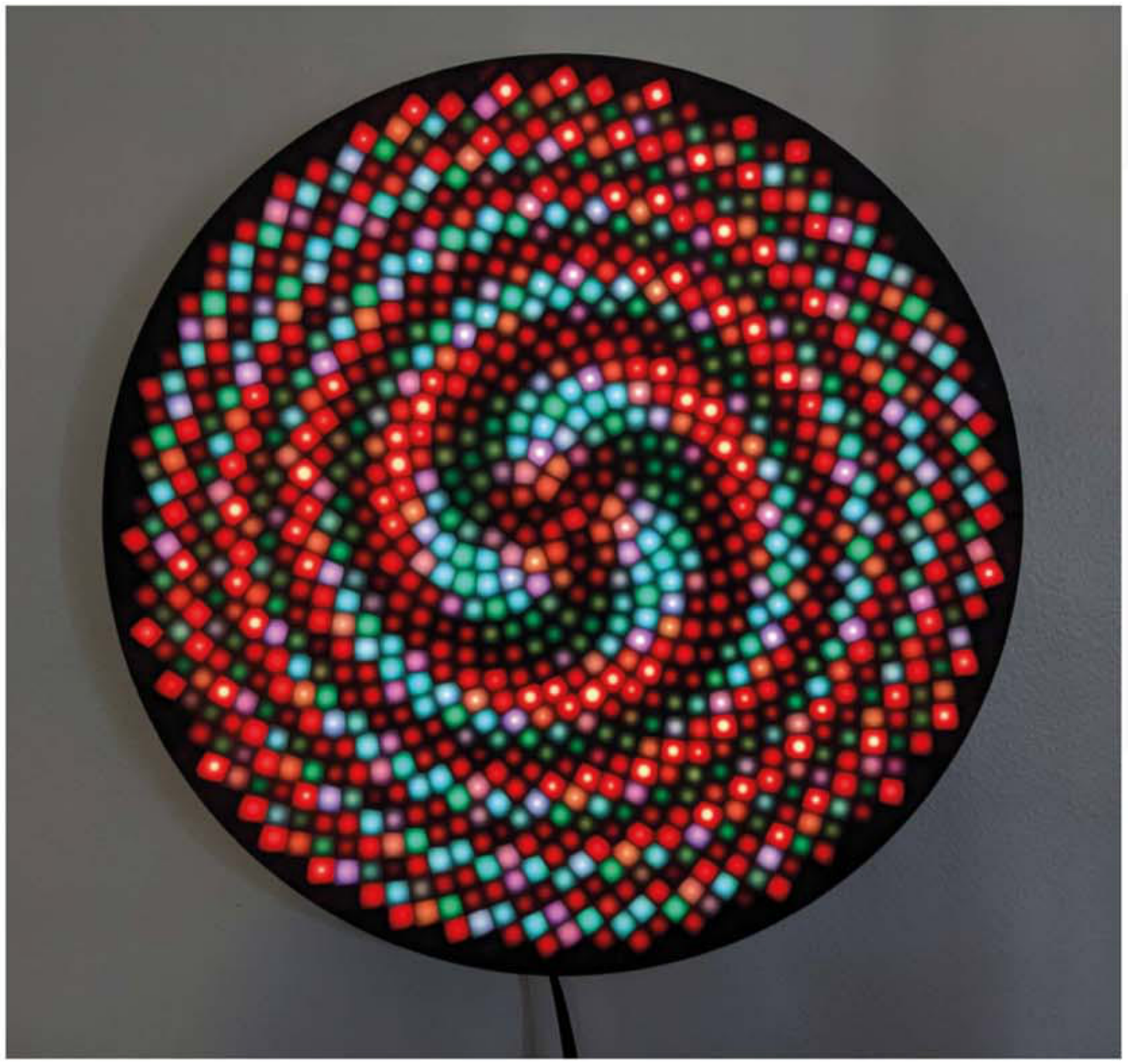
"With Louis's help, I started trying to do this but in a printed circuit board, just because placing and drilling holes by hand doesn't scale very well.

"I used SparkFun's tutorials to learn how to use EAGLE and make my very first printed circuit boards. Usually I would order the first batch prototypes from OSH Park here in the US and then, stencils from OSH Stencil; I learned how to use solder paste and place the components and reflow them in my oven. Later, I got a hotplate for doing rework and a hot air station rework, which is tricky with little plastic fragile LEDs.

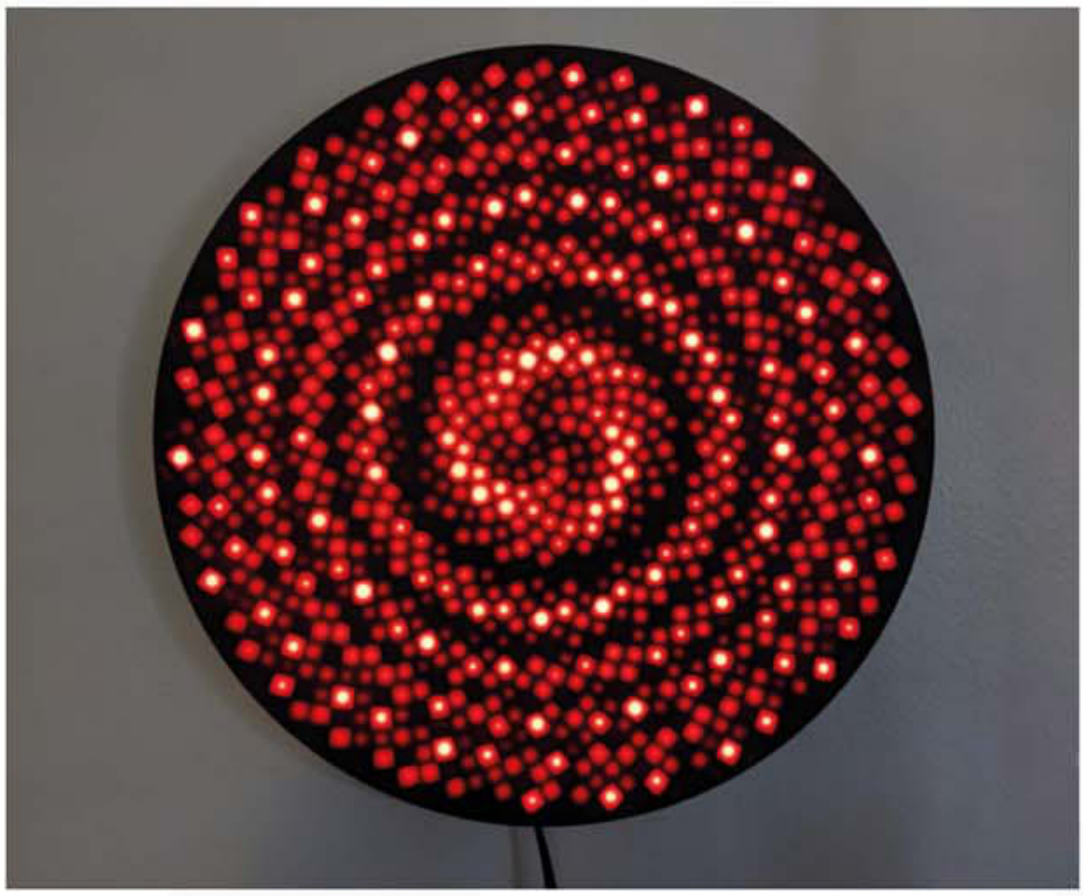
"I've been called a mathematician and an artist; I wouldn't call myself either of those, absolutely not. →

Above Chameleons change colour; so do WS2812 LEDs

REGULAR



Right
What you can't see in a still photograph is the mesmerising movement that's animated on these panels



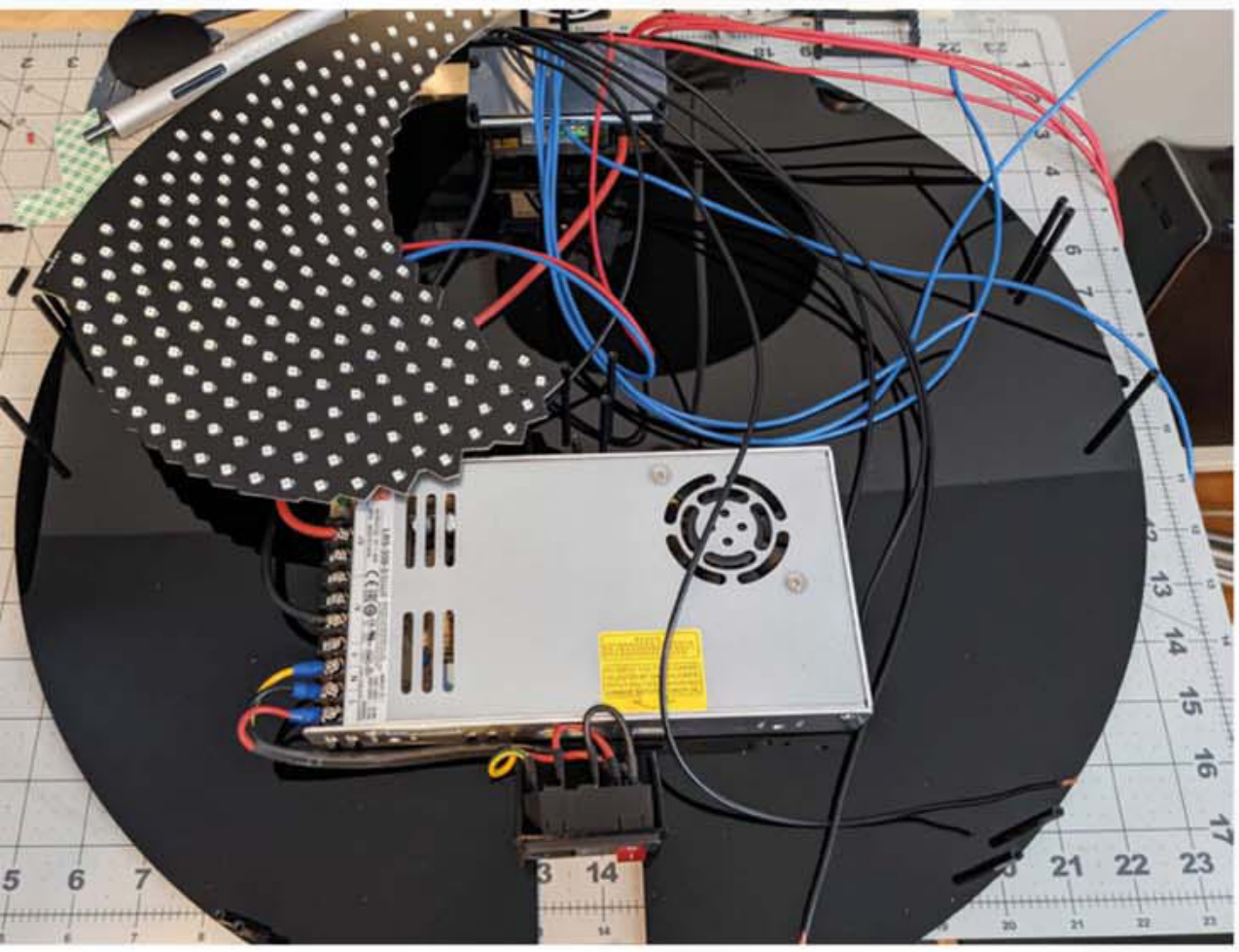
I just like tinkering. I would call myself a tinkerer or a maker maybe, but I've had no formal post-secondary or college training whatsoever in math or art.

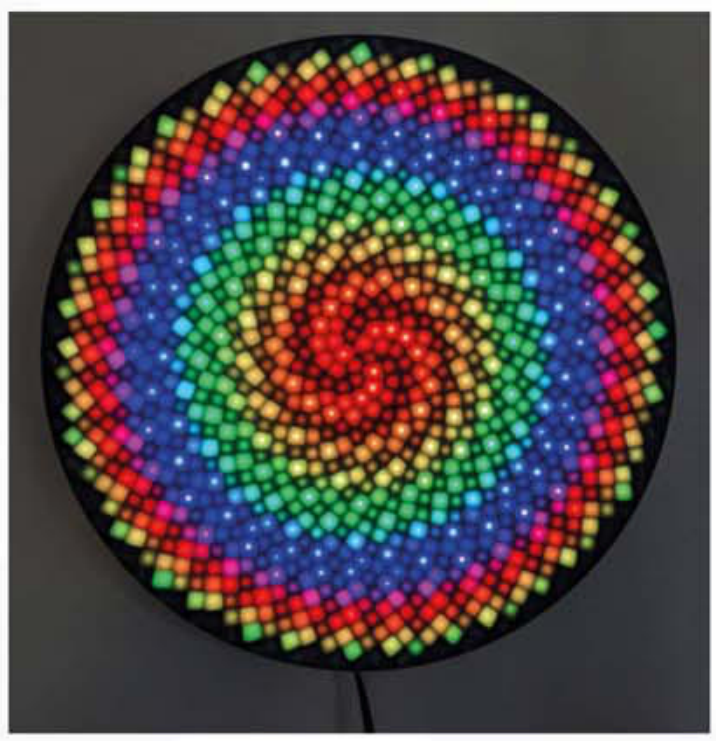
"My constant source of inspiration is other makers. It's like this amazing feedback loop of inspiration and making and creativity.

"In addition to electronics in high school, I took art and drafting on paper with a mechanical drafting machine. And I always loved making drawings, even just with a pencil, freeform on a blank sheet of paper, but especially on graph paper, and with a compass and all of the cool procedural shapes that you can make with just circles and a ruler. I've always, absolutely always been drawn to geometry, and especially nature-inspired geometry.

"And then, also books like *The Nature of Code* by Daniel Shiffman, which is another constant source of inspiration.

"Debra of Geek Mom Projects, we talk constantly, and I'm constantly inspired by the things that she makes. And you know, Sophie Wong and Becky Stern,





Left
The displays Jason's working on now are so large that the PCBs have to be assembled like a jigsaw

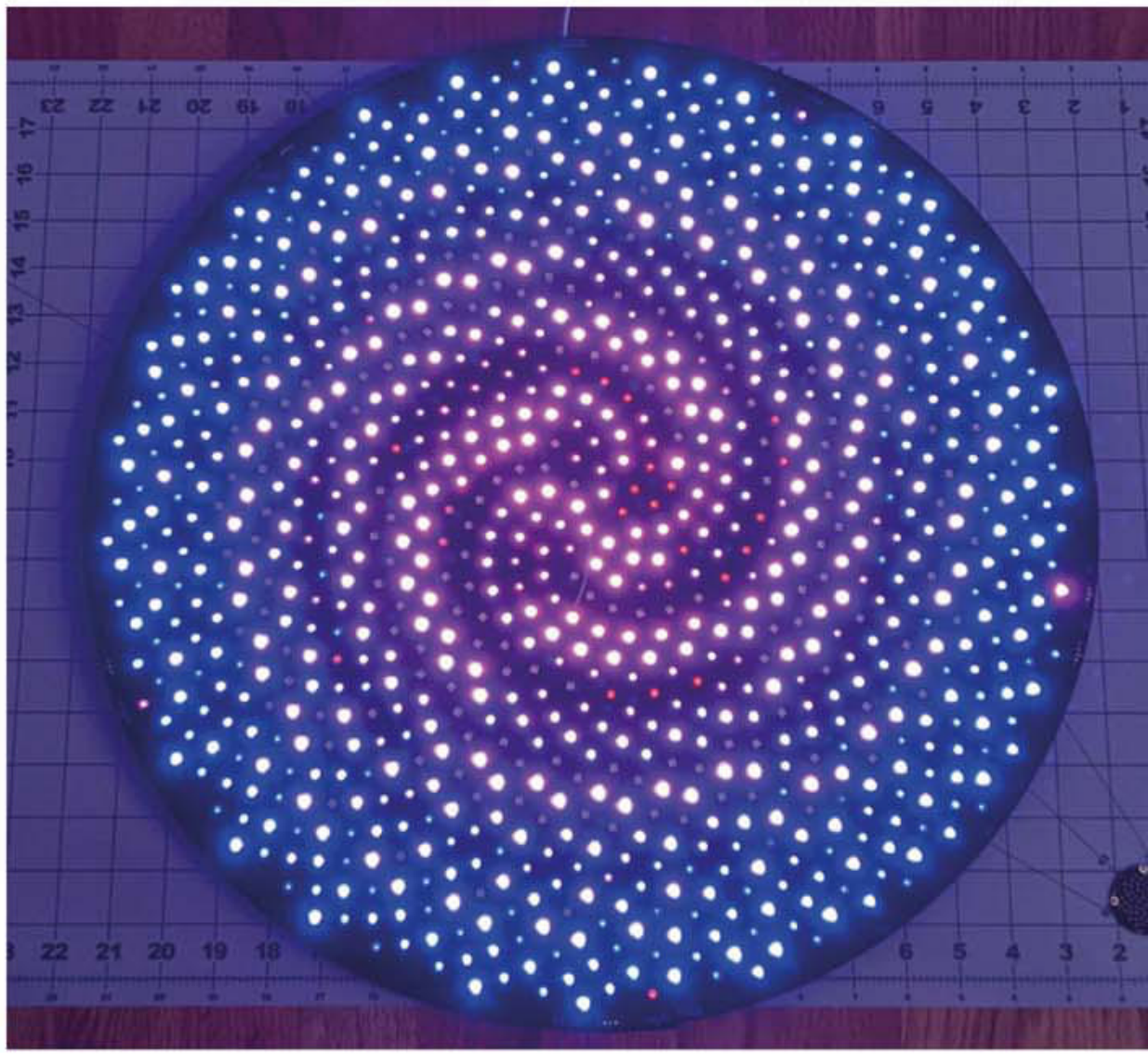
and Jorvon Moss for his little robotic creatures – the amount of inspiration out there is amazing.

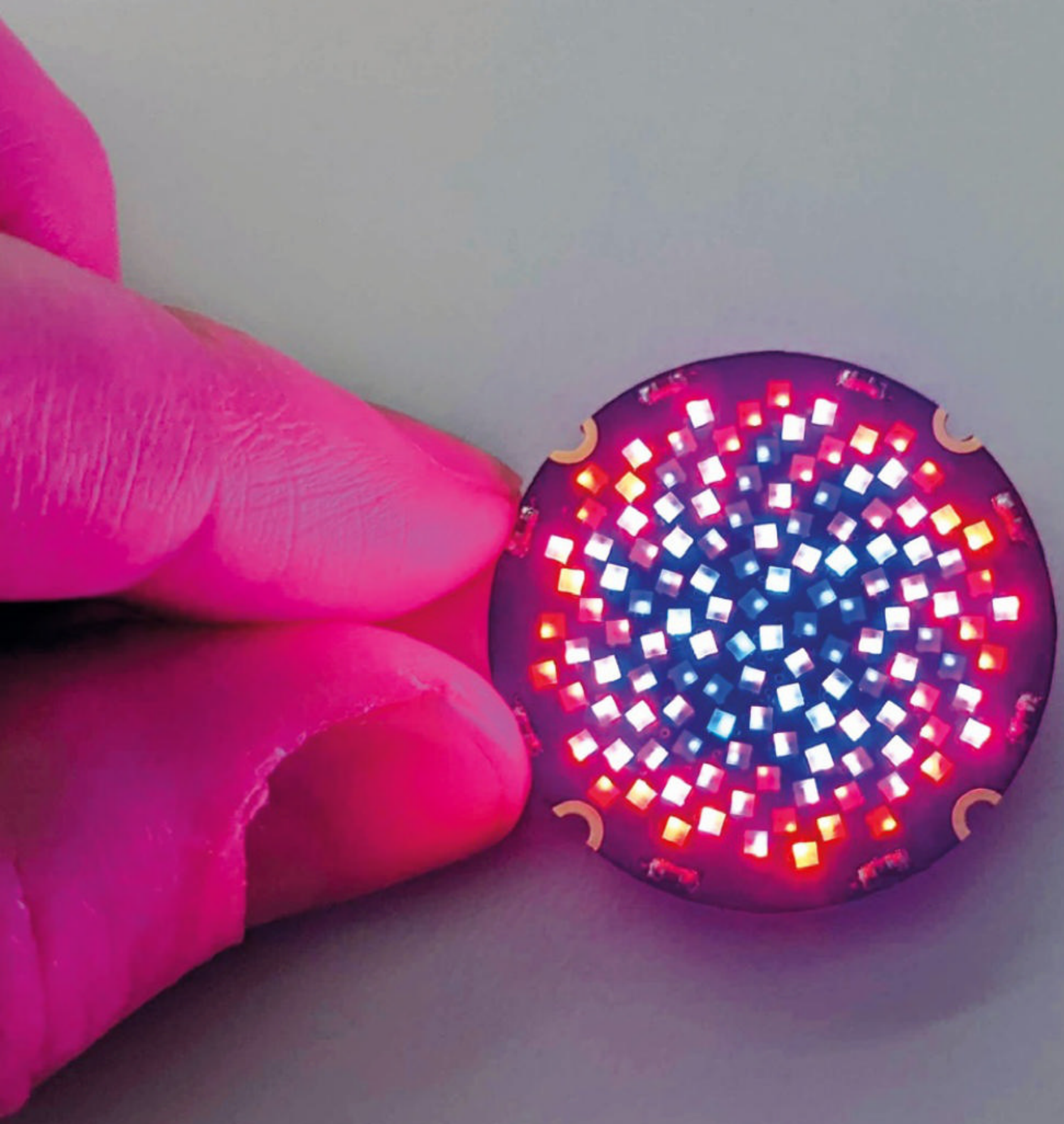
"I sell my boards as just LED displays, and you can do whatever you want with them. And one of the things you could do is play back pre-rendered pre-generated animations. You can definitely do that; I don't do that.

"I really enjoy writing procedural animations – patterns with code. I've been accused of being an artist, but really, the only way that I've ever been able to do anything remotely close to art is with graph paper or rulers or compasses, protractors, whatever, and then with code on the computer using trigonometry – cosine, tangent, and drawing circles and lines – all the way back to fourth grade in the computer lab in the library, writing Logo commands, moving the turtle around the screen to draw lines.

"I can remember sitting down with graph paper and drawing spaceships and things with my older brother. Give me a blank sheet of paper and maybe I can draw something, you know, while looking at it. But give me graph paper and I'll be much happier.

"All of my code is open-source, I definitely share all of my code. Until recently, I hadn't released anything of mine as open hardware, and I'm extremely conflicted about it – I have a lot of respect for people that are passionate about open source. Lately, I've kind of abandoned my early beginnings with PCBs – →





you know, the level shifters – I just decided, I didn't want to bother with the time and expense to create, to make and produce, and sell them. So I released all of those on OSH Park – you can just buy them directly from OSH Park and the board files are up there.

"Once I've abandoned something, I have no problem making it open-source. And my maybe selfish justification for my stance on it is just that this is a self-funding hobby for me – it's not my full-time job, and it's not just a hobby, it's somewhere in between. I use the proceeds of selling the things that I make to make new things, and there's a feedback loop of that which enables me to buy tools to make cooler things. I've been able to use the proceeds from the sales of Fibonacci boards and my other projects to buy a 3D printer and a laser cutter. I have dreams of having a garage or shop with a pick-and-place machine and a larger laser cutter – you can never have too large a laser cutter.

> My maybe selfish justification for my stance on it is just that this is a self-funding hobby for me **– it's not my full-time job**

"One of my most recent projects was to pack 128 RGB LEDs into a 33mm diameter. So each of those LEDs is only 1.2mm^2 – tiny. I assembled it by hand and I could not have done it without my microscope. I have a stereoscopic microscope and I use that a lot. I'm getting older and my eyes aren't what they used to be. And the LEDs don't have any visible indicator of polarity or anything on the top, so you have to turn them over, or assemble them on a microscope on a mirror so that you can see the marking on the bottom of them. So yeah, it's challenging. I had to do lots of rework; I think I had to rework eight or nine LEDs on the first one that I tried to assemble.

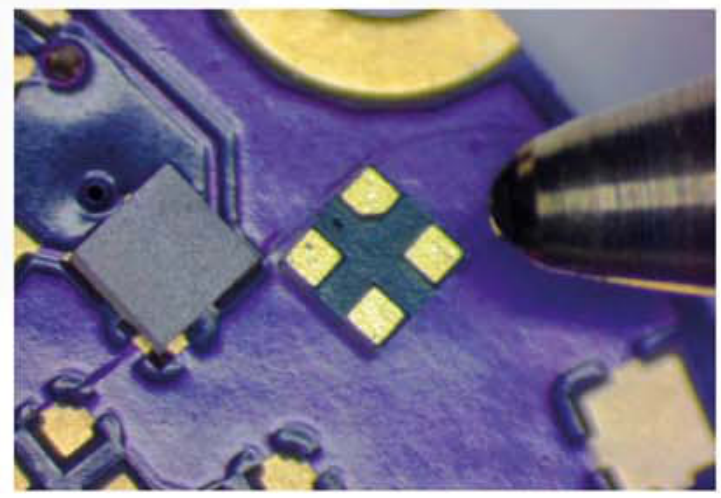
"And every component is at a unique angle… they're all rotated, which is the beauty of the Fibonacci distribution: no two LEDs are at exactly the same angle, and never will be. Irrational numbers rotate by phi, or the golden angle, between each LED, so they know to never completely overlap. And you can always add one more LED, just like a plant can always add one more petal or one more leaf.

Above
Jason makes these slate macro chips as custom artworks – the chip requested is a decent guide to the age of the customer

"I got started with LED matrices, and I still love them. I just got to the point where I felt like I was making smaller and smaller monitors; monitors with really big pixels. I still love them, but I wanted to try to find other forms.

"And that's been my driving purpose ever since. I have to say that the majority of the patterns that I ship on my boards are actually slightly modified examples from the FastLED library. I absolutely would not be able to do the things that I do without the FastLED library. They have implementations of sine wave generators, and Perlin noise generators and things like that, that just make this all so easy.

"One of the two remaining founders of the FastLED library is still around, Mark Kriegsman. And a lot of the code that runs the patterns – and, in fact, even some of the patterns themselves – was written by him. (David Garcia was the other guy that started the FastLED library and unfortunately, he's not with us any more). So a huge shout out and kudos to them for inspiring and making all of my work possible. That's another reason why I was inspired to share every line of source code that I write – because they did the same."



FEATURE

CORRUGATED IRON

IMPROVISER'S TOOLBOX

Ridged and rigid, corrugated iron has a range of practical uses

Rosie Hattersley

@RosieHattersley

Rosie Hattersley writes tech, craft, and life hacks and tweets **@RosieHattersley**.

Many [Keith] Moons ago, The Who's Roger Daltrey was a sheet metal worker – a fact that was hammered home to readers whenever Smash Hits magazine ran a quiz or crossword. Another fun fact is that steel is actually the metal at the core of corrugated iron. The misleading moniker has stuck, but now you know the truth, you won't get fooled again [sorry].

Corrugated iron was invented and patented in 1829 by engineer and architect Henry Robinson Palmer, who used it to build wide-span sheds at London Docks. The material was widely used in construction and adopted in overseas colonies such as Australia, since the rigid but lightweight material can be supplied rolled up, making it easy to transport as well as hard-wearing and flexible. In the UK, it found favour in rural settings, particularly for roofing.

These days, corrugated or sheet metal is no longer simply a practical alternative to a thatched roof. With sustainability and circular use firmly on designers' minds, corrugated iron has become a popular choice for interiors. One of the more intriguing designs is the spiral staircase, reminding us pleasingly of an unpeeling tin can: **hsmag.cc/SpiralStaircase**.

Architects and DIYers are using the crinkled stuff for walls, kitchen and bar splashbacks, on ceilings, and to achieve the all-important inside-outside connection beloved of home improvement shows.

Worshippers got there about 150 years ago, however, when builders began using the metal to construct congregational churches across the UK and beyond, in an early example of using improvised materials. The large volumes that could be constructed were important as urban populations and congregations increased. Five such examples still exist in London and are now heritage-listed, with

> "The material was widely used in construction and adopted in overseas colonies such as Australia"

celebrated blogger ianVisits delightfully relating the tale of Kilburn's Tin Tabernacle (as such structures came to be known), which became the Training Ship Bicester when it switched use to become a Sea Scouts headquarters and was fitted out accordingly: **hsmag.cc/TinTabernacle**.

A CONVERSATION WITH THE SEA

Uncompromising artist Maggi Hambling was determined to memorialise her hero, and former Aldeburgh resident, Benjamin Britten and raised the funds herself for her scallop sculpture, formally known as *A Conversation With The Sea*. The 4 m-tall steel form mimics that of a scallop shell, of course – perhaps the original inspiration for the distinctive ridges of corrugated iron? The artwork, which also functions as a shelter and seat, can be found on the Suffolk beach. It is made from four tonnes of heat-treated steel, with a polished silver interior, and a non-reflective brown exterior. A line from Britten's opera *Peter Grimes* is etched onto the outer flanges.

> "The artwork, which also functions as a shelter and seat, can be found on the Suffolk beach"

Project Maker
Maggi Hambling

Project Link
hsmag.cc/ConversationWithSea

Left
You can take cover from the sea in this shelter if you head to Aldeburgh in Suffolk

FEATURE

TWO-CAR BACKYARD WORKSHOP

Project Maker
GIZMOLOGIST

Project Link
hsmag.cc/TwoCarWorkshop

Above
We really like the curves you can get with this style of building

In some ways, this Instructable takes corrugated iron full circle by using the same construction principles and rationale as that of the makeshift churches of late Victorian times and, more obviously, the aircraft hangars of World War II. Maker gizmologist needed a garage that could house two vehicles, but she and her husband didn't really have the money to replace their trusty wooden ramada shelter when it succumbed to termites and a storm. They didn't want to embark on a time-consuming project, but the idea of a workshop in which to build dream boats, aeroplanes, and the like appealed. There were plenty of nearby examples of steel huts, but many of them ended up flattened or transported wholesale by the area's frequent strong winds.

The Backyard Shop is a half-size version of a US Navy Quonset Hut (see **hsmag.cc/Quonset**), itself based on the hemicircular construction method of Nissen huts. Essentially, you need prefabricated, galvanised corrugated iron curved over a semicircular frame. This frame consists of 1 inch metal pipes, while corrugated iron was chosen for its resistance to termites, and because it can be painted and made waterproof. The 20 ft-long building took two months to construct and used 40 to 50 sheets of 24 inch-wide corrugated iron, but didn't require a costly concrete base. Sited in the desert in southwest USA, the structure has withstood significant storms thanks to its rounded shell and the subtle flexing of the steel. The whole thing cost less than $2500 to build, leaving more money for the couple to put towards designing and building their own aeroplane.

OUTDOOR SHOWER

Embracing staycations and glamping over the past couple of years, it's been hard to avoid corrugated iron in its al fresco shower guise. A timber post and metal roof and walls is the usual approach, though the aforementioned flexibility means we've also showered in a door-free, curved wall walk-in version in which metal brackets are simply bolted to the sides and into the ground to prevent the shower structure from toppling over. DIY videos and tutorials are surprisingly scarce, but the famed hardiness of corrugated iron makes it a sensible option if you're thinking of embracing outdoor living for the summer months. If you'd prefer a plan to get started, head to **MyCarpentry.com** for downloadable plans.

Project Maker
MYCARPENTRY

Project Link
MyCarpentry.com

Left
The easy way to make curvy walls

CORRUGATED IRON MIRROR

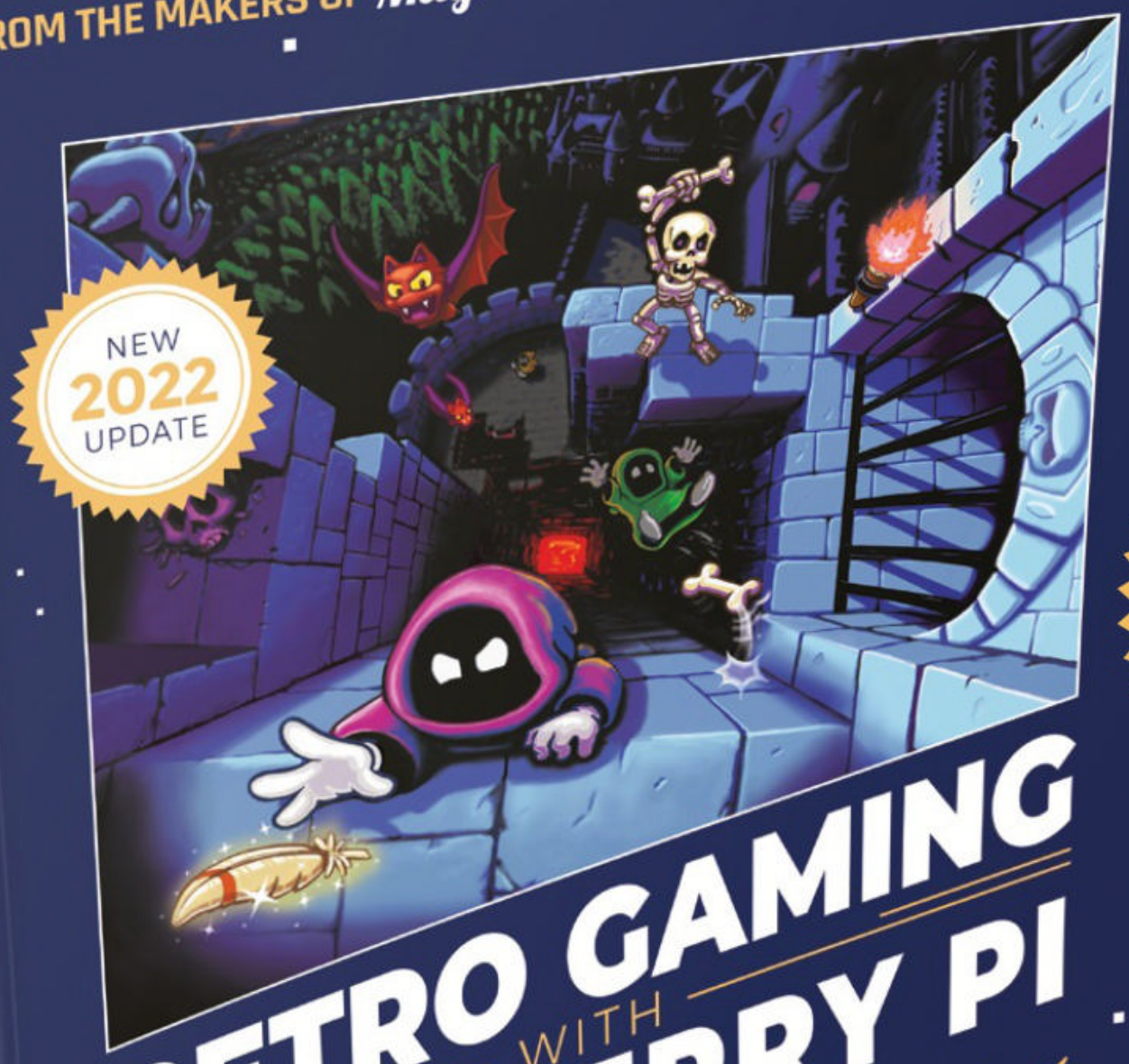
Salvaging items that would otherwise be discarded and finding new uses for them is the whole purpose of SARRU, which specialises in reclaiming and reusing apparent junk. The mirror surrounds, boxed in by teak, are simply corrugated iron strips that are often used for the flashing on roofs. Similar lengths of metal can be found in garden and hardware stores, where they are packaged as useful edgings for lawns and garden borders. If you're thinking of making a similar mirror, look out for reclaimed wood or discarded offcuts of timber to frame and enclose the corrugated metal strips.

Project Maker
SARRU

Project Link
hsmag.cc/CorrugatedMirror

Left
Add a shabby chic note to your home

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE



NEW 2022 UPDATE

RETRO GAMING WITH RASPBERRY PI

2ND EDITION

164 PAGES OF VIDEO GAME PROJECTS

BUILD AN ARCADE MACHINE

PLAY & CODE GAMES!

RETRO GAMING WITH RASPBERRY PI

2ND EDITION

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games. Build your own games console or full-size arcade cabinet, install emulation software and download classic arcade games with our step-by-step guides. Want to make games? Learn how to code your own with Python and Pygame Zero.

- ***Set up Raspberry Pi for retro gaming***
- *Emulate classic computers and consoles*
- ***Learn to code your own retro-style games***
- *Build a console, handheld, and full-size arcade machine*

BUY ONLINE: **magpi.cc/store**

FEATURE

IN THE WORKSHOP: Making good use of the things left behind

By **Ben Everard**

This month we've been repairing and upcycling bits and bobs that have fallen into our possession

Right
This sun lounger is made of three pieces that just slot together. The back rests in the gap between two planks in the bottom, and there's a ridge that the back-support slots into to hold it in place. No tricky joinery involved

Above
I found that joints could be weak, so included plenty of cross-braces to keep the table nice and strong

I've got an allotment – well, half an allotment. Well, actually, I'm sharing half an allotment with some of my neighbours, so really it's a quarter of one, but the point is that I now have some overgrown land on which to start growing fruit and vegetables.

In Bristol, where I live, allotments are pretty hard to come by and usually mean a wait of several years before one comes up. When it does, it's usually been unloved for a bit of time, which means it's covered in weeds and grass – ours is no exception. They also come with whatever things the chain of previous tenants have left on them. In our case, that's a pair of compost bins, a water butt, and a tumble-down shed filled with gardening equipment in various states of repair. The first thing we need is to clear at least a bit of the ground, and there's a battery-powered strimmer in the shed. However, there is no battery or charger.

We could buy a new battery and charger – they are available, but they're not particularly cheap, and it's barely more economical than buying a new strimmer. We could do it by hand, but it's a big area and, well, I'm lazy. Unfortunately, the local library of things (Share Bristol, **sharebristol.org.uk** – check them out if you're in the South West) only has mains-powered strimmers. Eventually, an idea presented itself – I don't have the right 18V battery, but I do have a couple of 18V batteries for other power tools. Would it be possible to 3D-print a battery holder and wire this up to the strimmer?

There are loads of designs for power-tool battery holders

It turns out that on Thingiverse, there are loads of designs for power-tool battery holders. There was one for the Milwaukee set I have, and the chances are there are already designs for whatever batteries you use. I just needed to print it out, glue in a couple of wires with blade terminals crimped on, and I was good to go. →

Above
They still need sanding down, but this bench-and-table combo makes our garden ready to once again invite people round

This is one of those things that I wish I'd done years ago. The battery holder is solid and was easy to print. I've got the batteries anyway, so they're no additional cost, and the batteries are rugged and have a built-in charge indicator. The only downside of this is that the batteries aren't particularly light, so it won't work for all projects.

Unfortunately, a nasty cough of unknown provenance has meant I don't feel comfortable going to a place frequented by retired people, so I haven't been able to fit it to the strimmer yet, but the battery holder itself is working and sending out 18V on demand.

GARDEN FURNITURE

I've had some building work done on my house and, just like a child that's more excited by the box their Christmas present comes in, I've had my eye on the ever-growing pile of wooden pallets in the garden. With the weather getting warmer and the building work coming to an end, it's now time to convert my haul into something more useful. In this case, some garden furniture.

The thing I really like about working with pallets is how quickly you can prototype your builds. Generally, I found I could get about 50% of the way towards a build very quickly by piling pallets on top of each other, clamping them in place, or throwing in a few screws to hold the build temporarily in place. This makes it very easy to play with designs, get the ergonomics right, and generally make sure everything feels right in real life as well as on paper. This is particularly important for people like myself who don't make much furniture.

However, the bad thing about pallets is that the wood is generally in a pretty shocking state. It's prone to splitting (and often already has several cracks by the time it arrives on your doorstep), and you have to think carefully about how you can join it together in ways that will be strong enough.

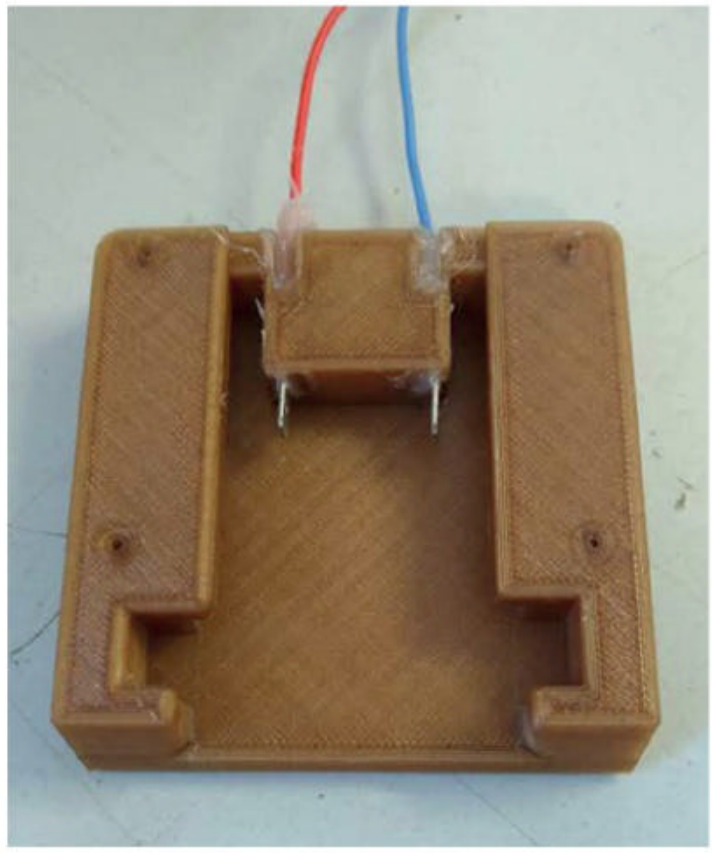
The lengths of wood need space to expand when they get wet

I made a few pieces. Firstly, a ground-level sun lounger. If you use pallets in their raw form, they are reasonably sturdy, but quite hard to join together in ways that aren't at 90 degrees to each other. However, sun loungers are one bit of furniture that absolutely needs non-right angles. After plenty of humming and hawing, I sidestepped this problem entirely by not joining the three parts together. The back-rest sits in a groove in the base (which does have an extra bit screwed on to make it recline). A support sits in a notch in the back-rest, and the whole thing is held together by gravity and friction. So far things are working well, but I might add a pair of hinges to make the back-rest to base join stronger if this proves necessary.

Secondly, I wanted a bench. This was made a bit easier by the fact that I had two extra-large pallets that were almost tailor-made for the back and seat of a two-person bench. With a bit of bent wood and crossed fingers, I was able to join these at right angles, and use a cut-off of one of the pallets to make the legs at the front. I stripped off all the lengths of wood that weren't needed and used these to make the slats on the front and back. One tip I had from a friend who made pallet furniture was not to place the lengths too close together as they need space to expand when they get wet. So, with a loose space between everything, I screwed it all together.

Thirdly, I made a garden coffee table. It's low enough for my young children to play at, but also high enough for it to be useful for grown-ups. This is simply a pallet with additional slats on the top to make it solid enough to be a surface. To this I added four legs. The hardest part was making sure that the legs were strong enough. One thing I quickly learned while building this is that it can be hard to join bits to pallets securely. In some directions it's easy because there's a nice surface to screw things to, but unless you're very careful with your design, this won't cover all the joints you need to make. In this particular case, I had to use some blocks from another pallet to make spacers to get everything to fit together. It's not the most elegant solution, but it was a quick way of making a very solid garden table.

So far, I'm really pleased with the furniture I've made. While it might not quite be perfect, it is designed for exactly our needs, and there's more than a little joy from the fact that it cost nothing but some time and a bag of pick-and-mix screws. ◻



Left
A simple 3D print lets me reuse drill batteries in other projects

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

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

18+

WHAT IS AVAXHOME?

AVAXHOME-

the biggest Internet portal, providing you various content: brand new books, trending movies, fresh magazines, hot games, recent software, latest music releases.

Unlimited satisfaction one low price
Cheap constant access to piping hot media
Protect your downloadings from Big brother
Safer, than torrent-trackers

18 years of seamless operation and our users' satisfaction

All languages
Brand new content
One site

AVXLIVE.ICU

AvaxHome - Your End Place

We have everything for all of your needs. Just open https://avxlive.icu

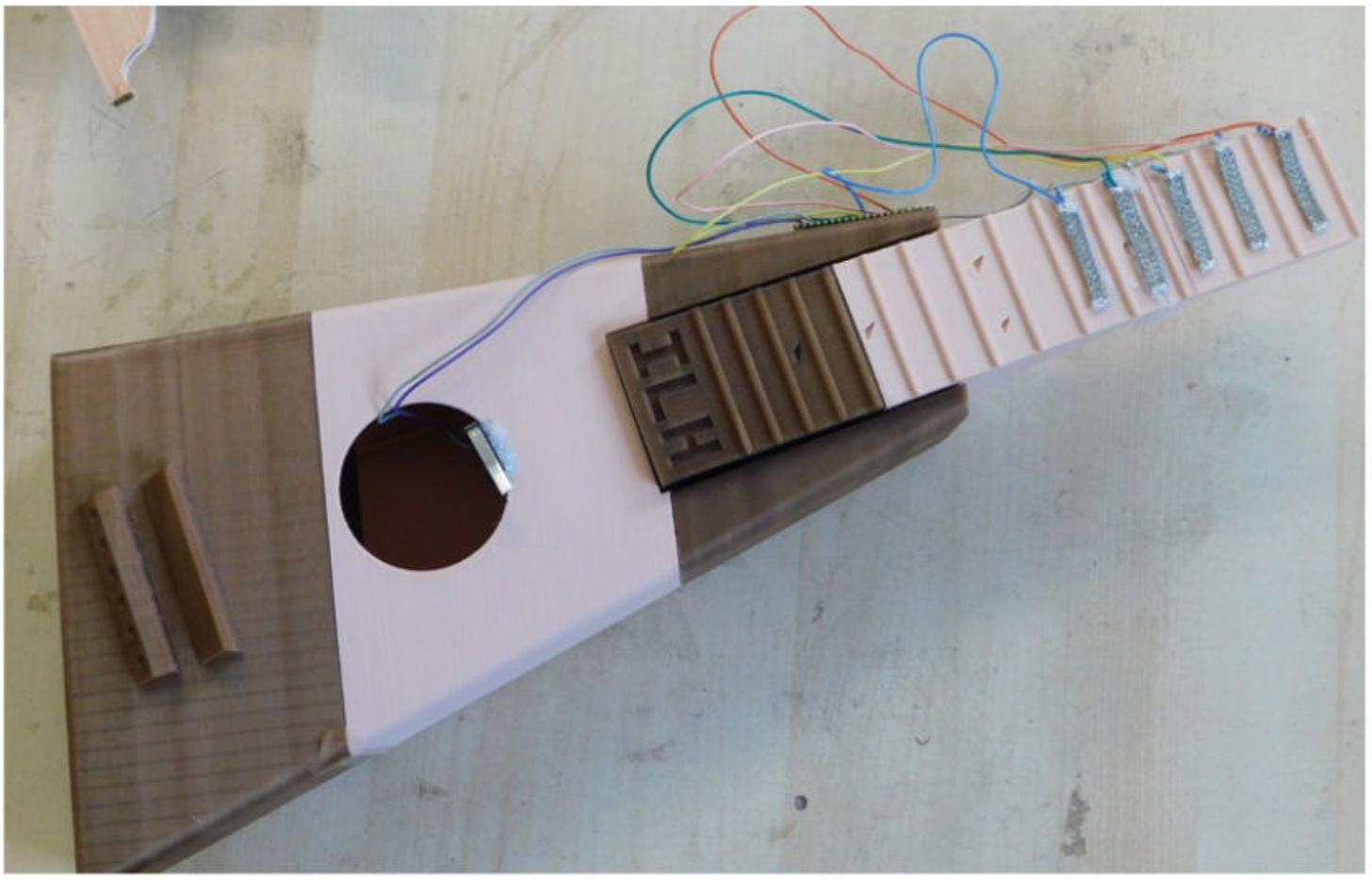
Guitar games controller

We turn failure into success by re-purposing a failed 3D print

Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

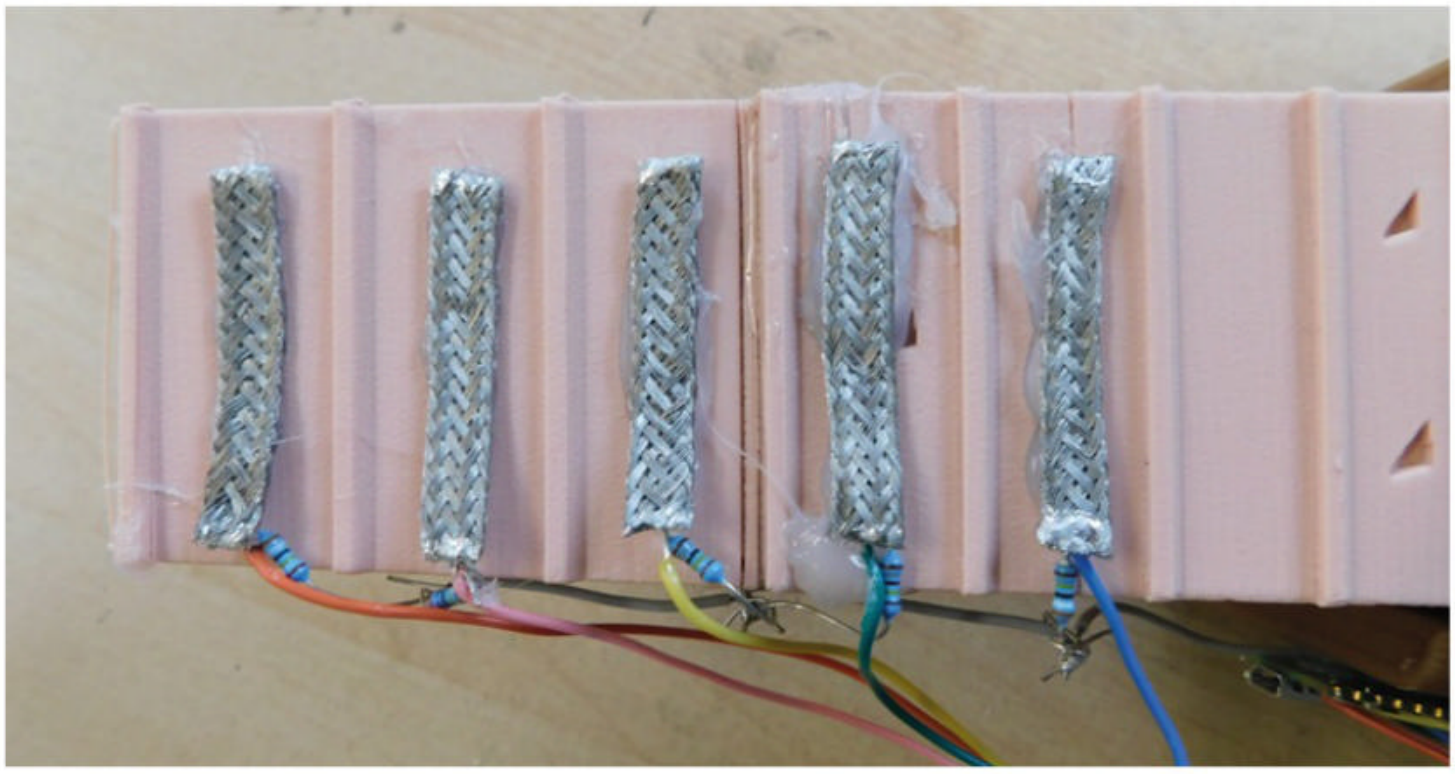


Above
From the broken parts of a failed guitar, we built a games controller

We tried to 3D-print a guitar, but misunderstood the stresses on the neck and oriented the neck in the wrong direction on the print bed. This was made worse by us printing in a cold room that may have affected the layer adhesion strength. It may not have worked even if we had done everything right, since there are a lot of forces on the neck of a guitar and it has to hold up to them very well in order to work. The end result was that the neck snapped in many places. This tutorial, however, isn't about making musical instruments, or even 3D printing. It's about what we did after we ended up with a non-functional, guitar-shaped object made of the pieces glued back together.

Our colleagues over at Wireframe magazine – a mag about making and playing games – have written a tutorial on how to make a Guitar Hero-like game. If you've not played Guitar Hero, the basic premise of the game is that you have to play a song on the controller. One button selects the note, and another strums. Strum at the right time while holding the right note and you score points. In the original game, this is done with a guitar-shaped controller, but in Wireframe's re-implementation, it's done with keys on the keyboard. This game is in Wireframe issue 62 – you can download it from **wfmag.cc**. This will also show you how the game works and how to extend it.

Of course, we couldn't let this stand. A guitar-playing game needs a guitar-shaped controller. We had a guitar (well, about three quarters of a guitar to be honest), a



Raspberry Pi Pico, and enough gubbins to build a replica controller for replica Guitar Hero.

In this tutorial, we'll build our own controller. If you don't have a broken 3D-printed guitar, don't worry: this is just what holds everything together. You could use almost anything. The simplest would be a wooden plank, but you could be creative. An old tennis racket would give classic air-guitar vibes, or you could cut a guitar shape out of sheet material (such as plywood or MDF) with a jig-saw. If your really want to go the whole hog, you could also 3D-print your own guitar – the design files are at **thingiverse.com/thing:486731**, but we wouldn't recommend using the same alignment that we used.

> **An old tennis racket would give classic air-guitar vibes,** or you could cut a guitar shape out of sheet material

Since our game takes input from the keyboard, we just have to set Pico up as a keyboard, then we can get our program to send keystrokes to the computer and the game will interpret them as though they'd come from any normal keyboard. It's actually very easy for some microcontrollers to behave like keyboards, and you can do this in a range of languages, but we find CircuitPython to be just about the quickest and easiest option for this. Since Wireframe's game is also written in Python, this means that both the game and controller are in a single language, which helps keep things simple.

We'll look at the code shortly, but we can use the adafruit_hid module's `keyboard` object which exposes `press()` and `release()` methods that let you press and release buttons. Let's first look at the hardware.

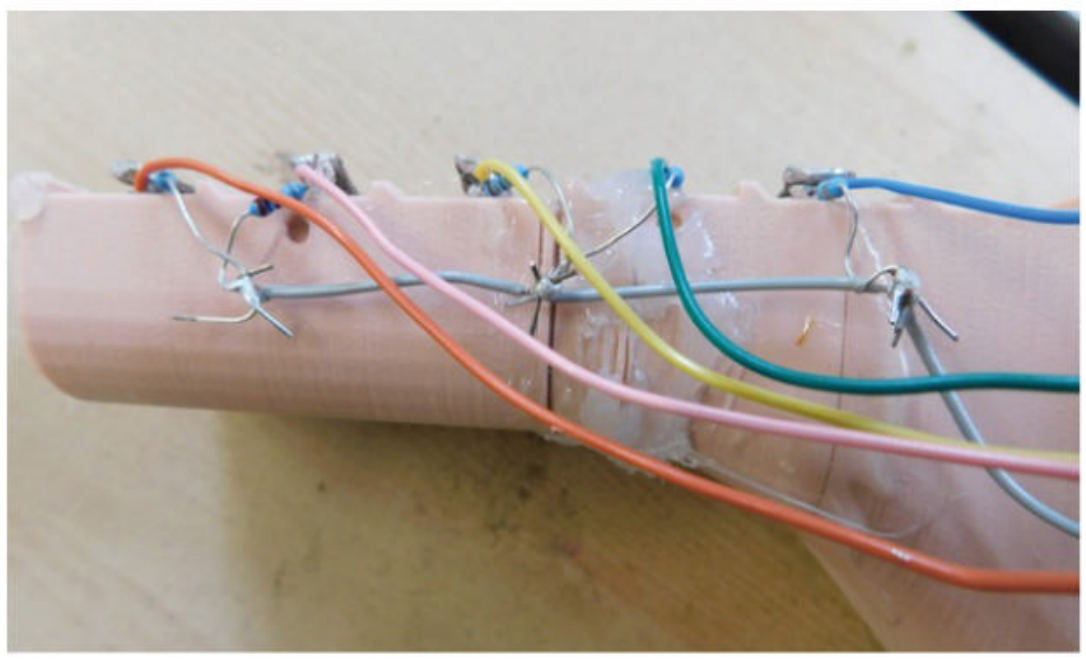
We'll treat the two inputs (note and strum) a little differently. For the notes, we'll use touchpads, and for the strum we'll use a microswitch. This arrangement fitted our guitar well, but you could use all buttons for the notes if you prefer.

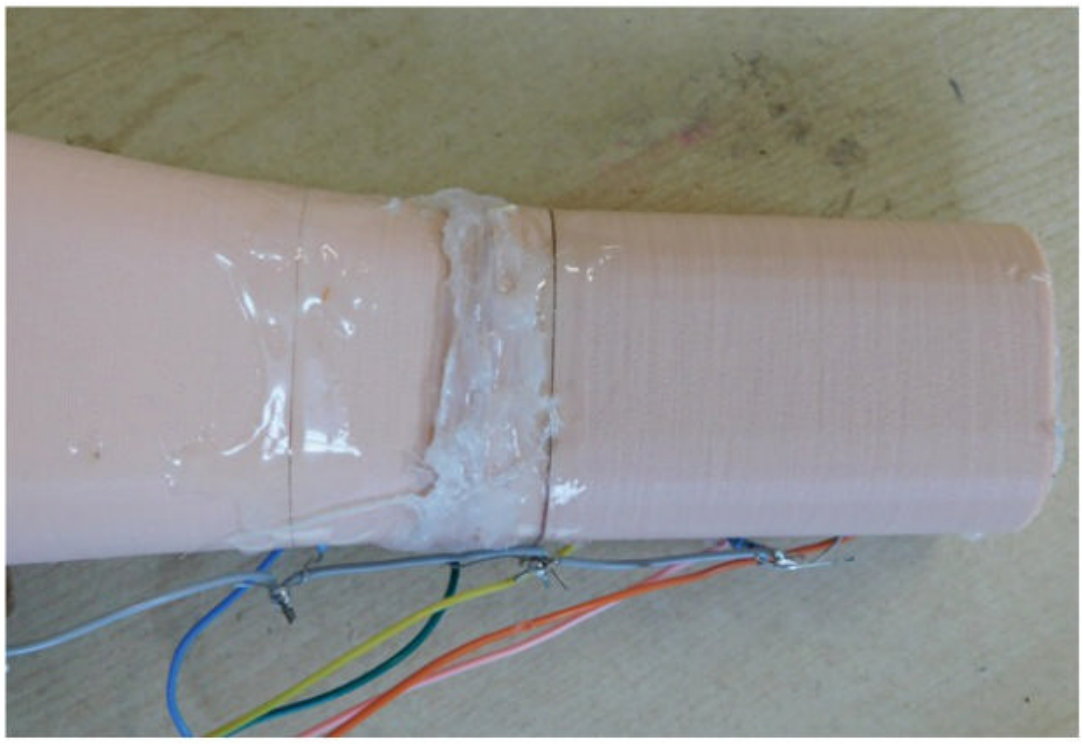
A touchpad is just an exposed conductor. Wire, tinfoil, screws, and sheet steel all work, but we used a wiremesh designed for protecting cables because we had some to hand. A bead of solder along each cut edge stopped it fraying, and also protects fingers from the stray strands.

Raspberry Pi Pico doesn't have in-built capabilities for touch sensing, but that doesn't matter, we can still use the TouchIO CircuitPython module, as long as we connect the touchpad to ground via a 1MΩ resistor. Other than that, the touchpad just needs to be connected to any IO pin. →

Above
The pads can be anything conductive – we've used cable shielding hot-glued on, but you can use whatever you have to hand

Below
All the pads are connected to a common ground wire (the grey one) via 1 MΩ resistors





When working with touch-sensitive circuits, you need to be a little mindful of the whole system. Although we said that a touchpad is an exposed connector, you'll still get some reading if you touch the wire, and you can get some cross-talk between pads. We've addressed some of this in the code, but it's better to minimise it in the circuit as well, so keep the wires as short as possible and don't route them where they'll come into contact with your hands or body.

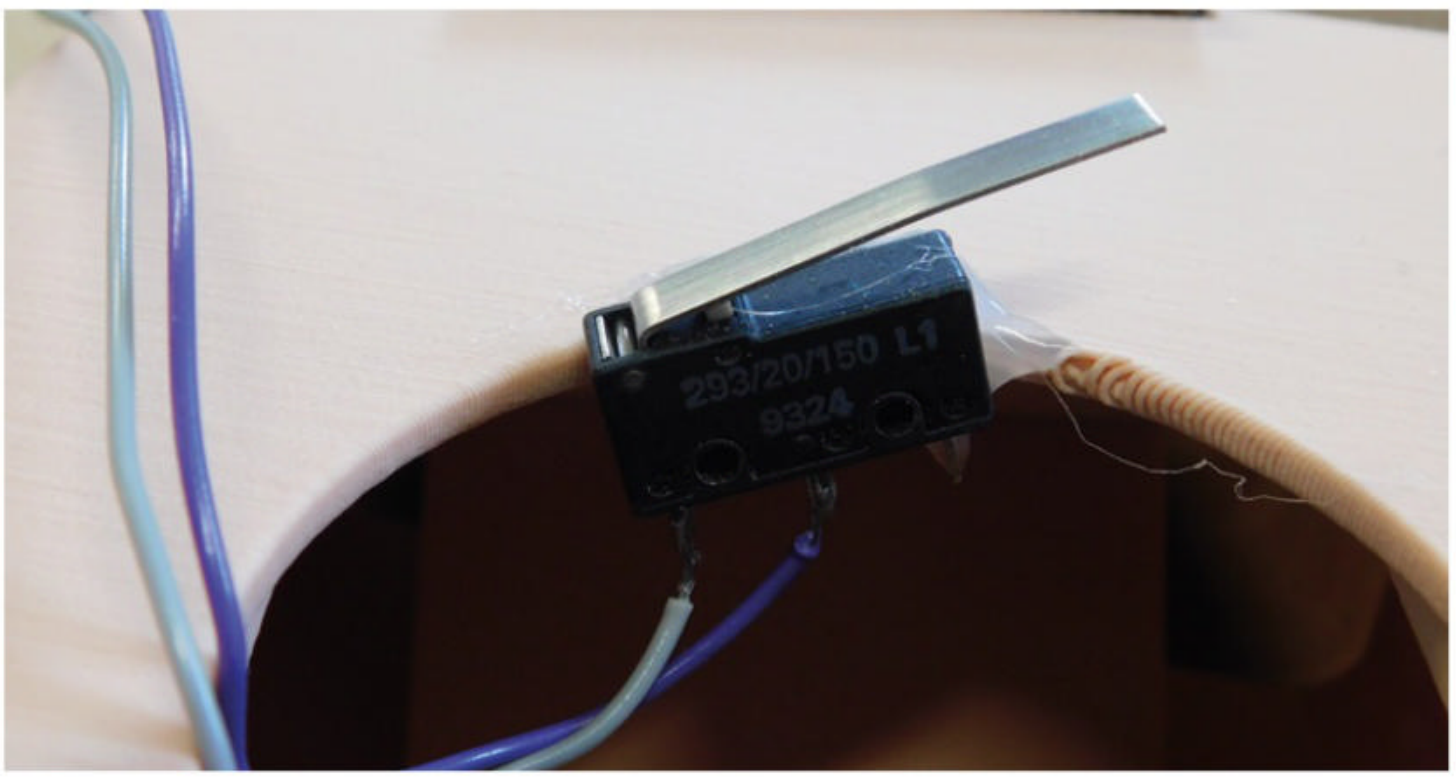
The microswitch will have three connectors: one common, one normally open, and one normally closed. We need two of these – the normally open and the common. These may be marked on your microswitch – or you can find them with a multimeter, set it to its continuity setting (where it beeps when the two probes are electrically connected), then find the pair of terminals that don't beep when the switch isn't pressed, but do when it is.

You need to attach one of these connections to ground and the other to any unused IO pin.

That's all there is to the circuit; you just need to physically attach everything to the part of the controller you want. There are probably some clever ways of doing this, but we just hot-glued it all in place.

Let's now take a look at the code (this should be saved as **code.py** on your device):

```
import time
import board
import touchio
from digitalio import DigitalInOut, Direction,
Pull
import usb_hid
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keyboard_layout_us import
KeyboardLayoutUS
from adafruit_hid.keycode import Keycode

touch_pad_z = board.GP2
touch_z = touchio.TouchIn(touch_pad_z)

touch_pad_x = board.GP3
touch_x = touchio.TouchIn(touch_pad_x)

touch_pad_c = board.GP4
touch_c = touchio.TouchIn(touch_pad_c)

touch_pad_v = board.GP5
touch_v = touchio.TouchIn(touch_pad_v)

touch_pad_b = board.GP6
```



Above
For the record, this is a terrible way to 3D-print a guitar, but that doesn't mean we should let it go to waste

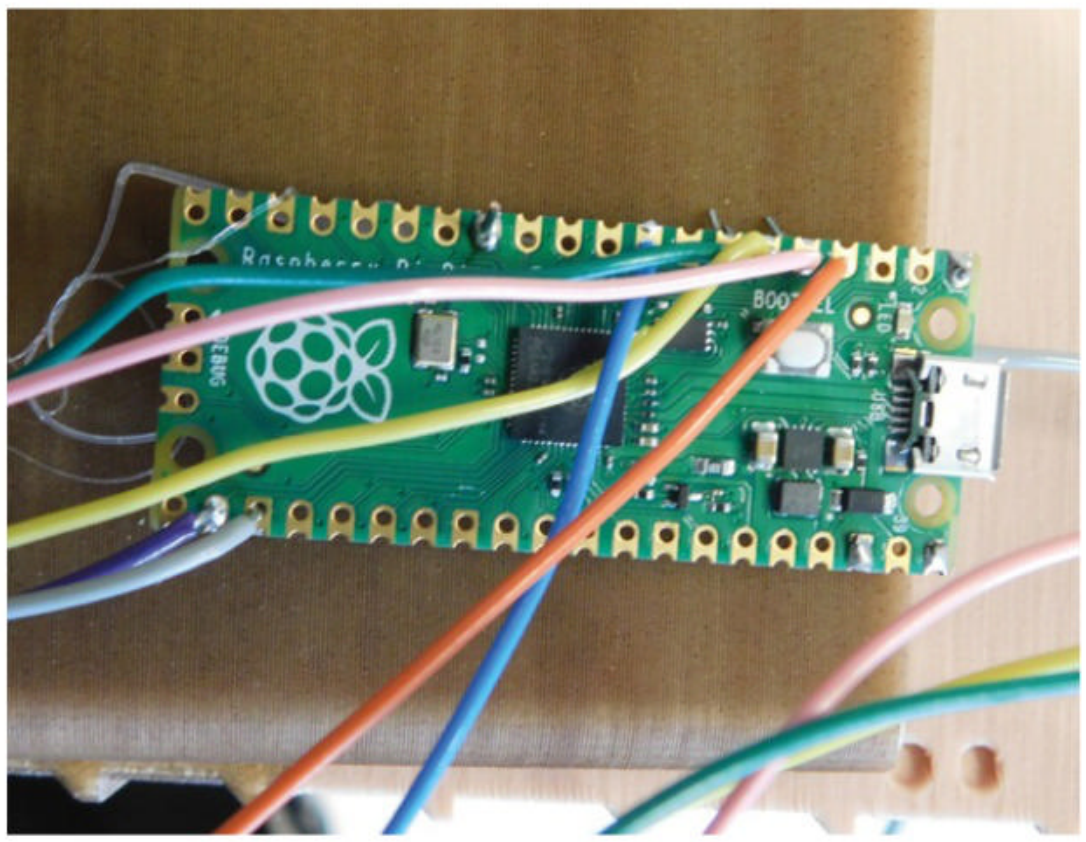
Right
You can use any type of switch as the strum switch, but we like the twangy-ness of a microswitch

```
touch_b = touchio.TouchIn(touch_pad_b)

touchpads = [touch_z, touch_x, touch_c, touch_v,
touch_b]
keycodes = [Keycode.Z, Keycode.X, Keycode.C,
Keycode.V, Keycode.B]

thresh_jump = 2500
for pad in touchpads:
    current = pad.raw_value
    pad.threshold = current+thresh_jump

switch = DigitalInOut(board.GP17)
switch.direction = Direction.INPUT
switch.pull = Pull.UP

keyboard = Keyboard(usb_hid.devices)
keyboard_layout = KeyboardLayoutUS(keyboard)

touch_thresh = 3500
currently_pressed = False
currently_pressed_index = 0

while True:

    pressed = False
    max_value = 0
    max_index = 0
    last_key = ""
    for i in range(5):
        value = touchpads[i].raw_value
        if value > touch_thresh:
            pressed = True
        if value > max_value:
            max_value = value
            max_index = i
    if (pressed):
        print(max_index)
        if (not currently_pressed) or currently_
pressed_index != max_index:
            keyboard.release(keycodes[currently_
pressed_index])
            keyboard.press(keycodes[max_index])
            currently_pressed = True
            currently_pressed_index = max_index

    if (not pressed) and currently_pressed:
        keyboard.release(keycodes[currently_
pressed_index])
        currently_pressed = False

    time.sleep(0.05)
    if not switch.value:
        keyboard.press(Keycode.SPACE)
    else:
        keyboard.release(Keycode.SPACE)
```

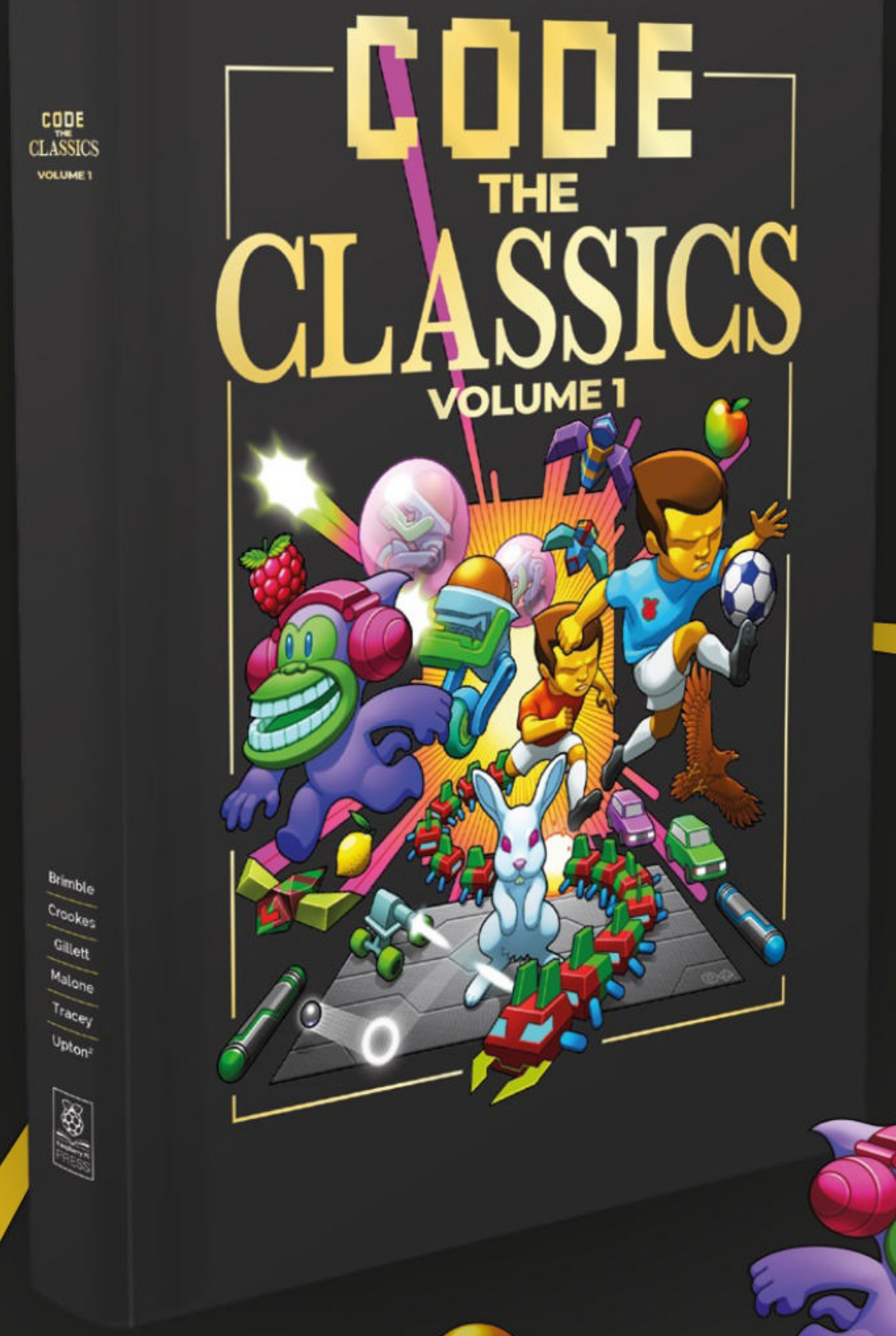
Normally, with TouchIO, you can just use the **value** property (which will be true if the pad is currently touched and false if not) to see if the pad is currently touched, but this was triggering a lot of false positives for us. Instead, we set a fixed threshold and tested to see if any of the pads exceeded this threshold – if they did, then it checks to see which of the pads had the highest **raw_value** (which is basically an indication of the capacitance of the pad, which will increase if you touch it), and it treats the highest one as touched. Using this, we were able to accurately detect which pad was touched, but the downside is that it can only detect a single note press at a time. This isn't a problem in the game Wireframe created, but could potentially limit future songs. You may find that you have to tune the threshold value for your build. Increase it if you're getting false presses, and decrease it if it's not registering presses at all.

In order to run this, you'll first need to download CircuitPython and flash it to your device – you can find it at **circuitpython.org**. You'll also need the libraries bundle, and copy across the adafruit_hid module to the **lib** folder on your Pico. Finally, you'll need to save this code to your device as **code.py** (you can download it from **hsmag.cc/GuitarHero**).

With that all in place, you should be able to fire up your controller and rock out to a Guitar Hero-like game with your own controller, both coded in Python.



Below
Raspberry Pi Pico does have mounting holes, so you can use screws to hold it in place, but as we're retrofitting it, we've just hot-glued it in place


CODE
THE
CLASSICS
VOLUME 1
CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²

Part 01

CDP Studio: Trigger LED patterns with a web GUI

WRITER

Phil King

Long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

Use the low-code, block-based environment of CDP Studio to create and deploy a Raspberry Pi project

An 'out of the box' software development tool, CDP Studio is used by numerous companies to build industrial control, automation, and edge systems. Yet, it's fairly easy to get to grips with its low-code programming environment. Better yet, you can deploy projects to a Raspberry Pi – this is often used for testing and prototyping purposes in industrial settings.

In this introductory tutorial, we'll guide you through the process of creating an app to flash LEDs connected to a Raspberry Pi in different patterns, selectable on a web GUI. We'll mainly be using CDP Studio's Block Diagram mode, connecting various preset block components, although we'll also do a tiny bit of hard coding in C++ to create a new custom function.

You'll Need

- Linux or Windows PC
- CDP Studio **cdpstudio.com/getstarted**
- Raspberry Pi
- Raspberry Pi OS (Bullseye or Legacy version, 32-bit)
- 3 × LEDs
- 3 × 330 Ω resistors
- Jumper wires

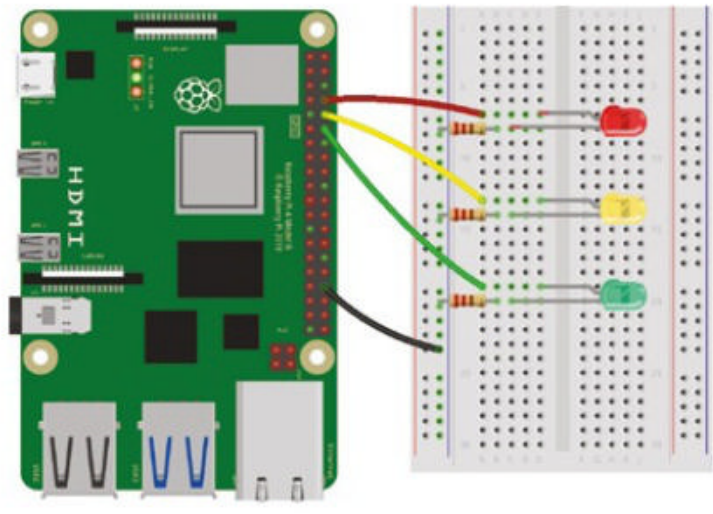


Figure 1 The wiring diagram for the LEDs circuit

01 Install the software

Visit **cdpstudio.com/getstarted** and download the free non-commercial version for Linux or Windows. During installation, make sure you select both Raspberry Pi ARMv8 32-bit (Debian 11) and Raspberry Pi ARMv6 32-bit (Debian 10) components, along with the one already ticked for your host PC. You will then be able to deploy projects to any Raspberry Pi model, using Raspberry Pi OS Bullseye or Legacy version, by selecting the appropriate toolkit in CDP Studio.

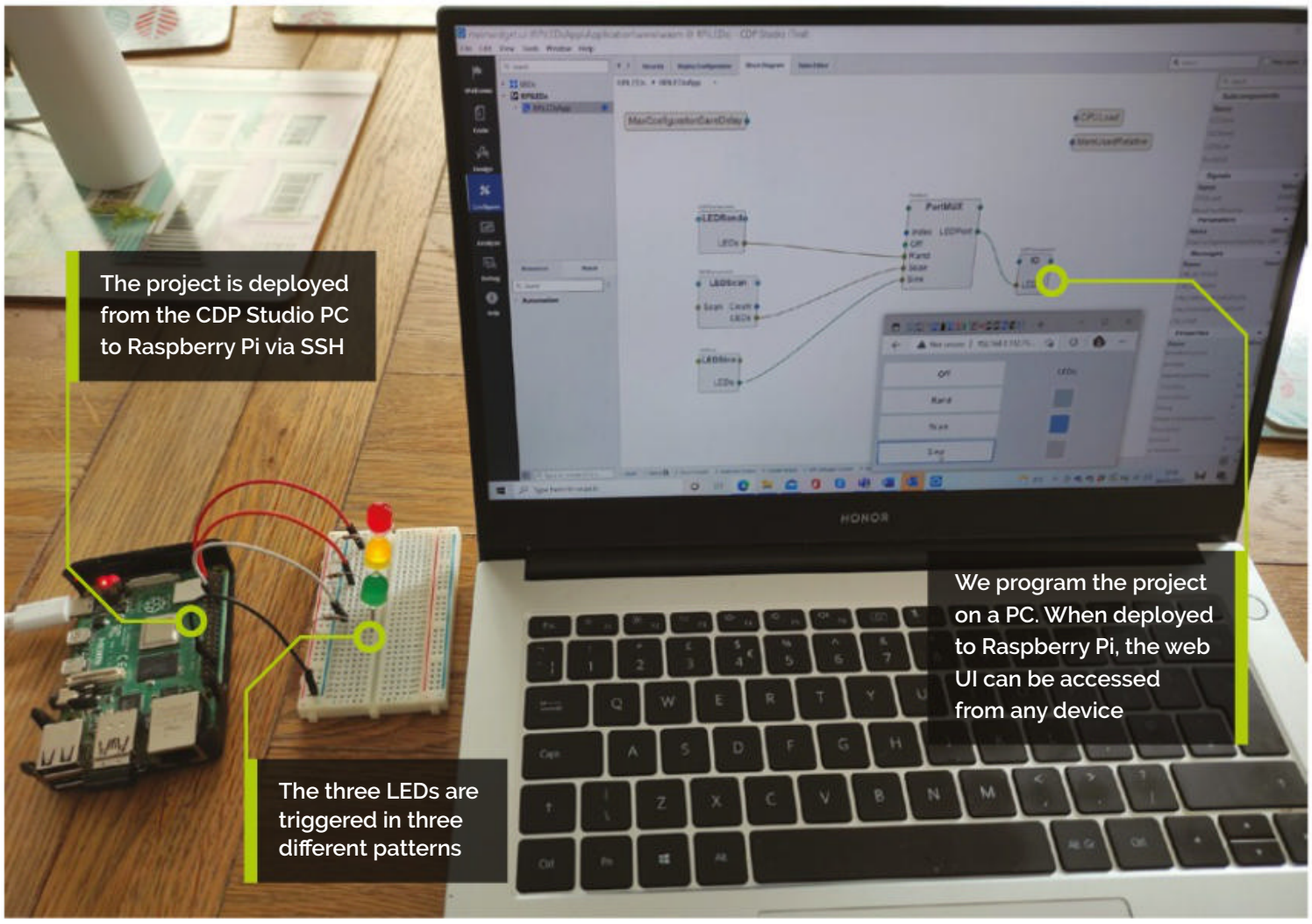
02 Create a library

For this project, we'll create a library to house a custom function. Components created in libraries can be used easily in other projects.

Go to Create New > Library and enter a name for it (we called our library 'LEDs'). Right-click your library in left panel and select Add New. Select the CDP Port Model in the dialog and click Choose. Click Next, then enter name for it (LEDPort in our case), then click Next and Finish.

At the bottom of the **LEDPort.cpp** C++ code file that appears, right-click `UserProperty` and select CDP > Change, then add a name for it (LEDPort). Change the Type to bool, then click OK. This will auto-create an **LEDPort.h** header file.

In **LEDPort.cpp**, we need to add extra properties for two more LED patterns. Add a new line under



`LED1.Create`, then right-click and select CDP > Add > Property. Name it LED2 and change its Type to bool, and Routing Type to Periodic. Tick the box for Create Another. Click OK, then name this one LED3; untick Create Another, then click OK.

Finally, we need to build the library. Right-click the LEDs library name in the left panel and select Build, then Save All.

03 First LED pattern

For our first pattern, we'll create a custom component. Switch to Configure mode and right-click your LEDs library name in the left panel, then Add New > CDP Component Model. Click Choose, then Next; in the Component Model parameters, give it a class name (LEDSine), and change the Frequency to 2 (cycles per second). Click Next, then Finish.

The **LEDSine.cpp** file will be auto-created. Right-click in its code, then select CDP > Add > Port. In the Port type drop-down, select LEDS::LEDPort (the one you created). We'll name it 'LEDs'.

Under `/* Write your code here */` in **LEDSine.cpp**, add the following (indented) code:

```
LEDs.LED1 = (counter==0);
LEDs.LED2 = (counter==1);
LEDs.LED3 = (counter==2);
counter += increment;
if (counter==0 || counter  ==2)
    increment*=-1;
```

In the **LEDSine.h** header, under `LEDPort LEDs;`, add these two lines to create the `counter` and `increment` variables:

```
int counter = 0;
int increment = 1;
```

Build the LEDs library again. In Configure mode, under LEDs in the Resources panel, we now have the LEDSine component available.

04 Main block structure

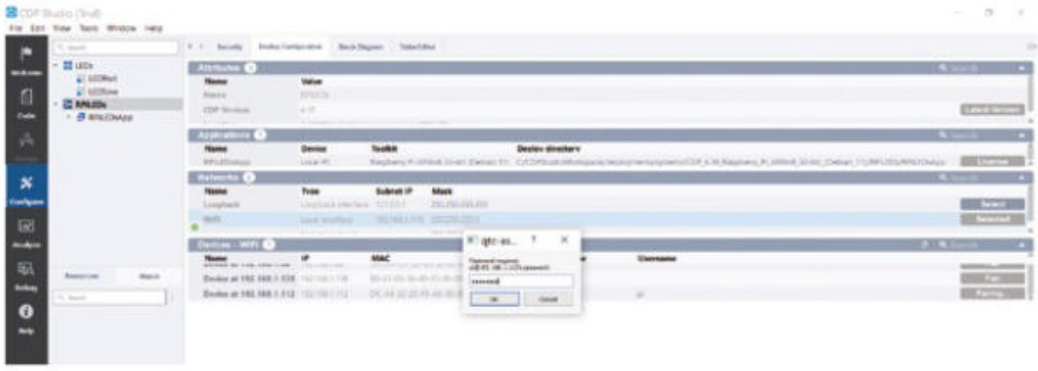
Right-click in the left panel and select Create New > CDP System. Give it a name (RPiLEDs), click Next twice, then Finish. In the Block Diagram tab of Configure mode, you'll now see the main block for the project. Double-click →

The main block diagram for the LEDs app in CDP Studio, with the three LED functions on the left. The web GUI is also shown here in a browser window

Top Tip

Random issue

When the project is deployed on a Windows PC, the three AddRandom blocks will set the same value at the same time, turning all three LEDs on/off together. When deployed to Raspberry Pi, however, each individual LED will toggle randomly.



Pairing CDP Studio with a Raspberry Pi on the wireless network via SSH. It will also need its **/etc/security/limits.conf** file modified – see **magpi.cc/cdprpisetup**

it (or select RPILEDsApp in the left panel) to see inside. We can now start adding program blocks.

In the bottom left Resources panel of Configure mode, open the CDPCore category and drag a CDPComponent into the diagram. Drag another CDPComponent underneath it. Then open the LEDs category in Resources to find your LEDSine component; drag it under the two CDPComponent blocks.

Click on LEDs on the LEDSine block and untick Input in the right-hand panel, to make it an output. Select LEDSine in the LEDs library. In the Table Editor tab, click the right arrow next to LEDs (under Ports) and untick Input there too.

Back in the Block Diagram view for RPiLEDsApp, open CDPCore in Resources and drag a PortMUX block into the diagram. This acts as a multiplexer for the three LED patterns, avoiding complex, messy individual wiring.

Drag another CDPComponent to the right of PortMUX. Right-click and rename it IO. This block will house our GPIO server.

Now drag over a CDPPort (under CDPCore) into both CDPComponent blocks. Select it in each block and untick Input in the right-hand panel, to change it to an output.

05 PortMUX inputs

Now to add more inputs to our PortMUX block. From LEDs in Resources, drag an LEDPort into the PortMUX block. Select it and, in the right-hand panel, rename it 'Off'.

From CDPCore, add three CDPPorts to the PortMUX block – these are for our three LED patterns. Rename CDPPort to Rand, CDPPort1 to Scan, and CDPPort2 to Sine. From LEDs in Resources, drag an LEDPort into PortMUX; untick Input in the right-hand panel.

It's time to make our first connection: drag a wire from the LEDs output of the LEDSine block to the Sine input on PortMUX.

06 IO block

Add a CDPPort to the IO block and rename it LEDs, then connect the LEDs output of PortMUX to it.

We need to add a few lines of C++ code for our custom function's **LEDSine.cpp** file

Now to add our GPIO server for Raspberry Pi. Double-click the IO block, then open GPIOPinIO in Resources and drag a GPIOServer into IO. With GPIOServer selected, right-click GPIOPin in Resources and select Add Multiple. Change its name to LED1 and increase Count to 3. Click OK to create three outputs – LED1, LED2, LED3; change them all to inputs.

Select the LEDs block in IO. Open CDPCore in Resources and drag three PushConnection outputs into the LEDs block. Rename them LED1, LED2, and LED3. Now connect the LED1, LED2, LED3 outputs of LEDs block to the LED1, LED2, LED3 inputs of the GPIOServer block.

We need to enter the GPIO pin numbers for our circuit's LEDs (**Figure 1**). Select each of the LED1, LED2, LED3 inputs on the GPIOServer in turn and set its Nr value (right panel) to the GPIO pin number: 14, 15, and 18 respectively.

07 More LED functions

Back in RPiLEDApp, rename the top CDPComponent to LEDRand, and the other CDPComponent to LEDScan. Rename both their CDPPorts to LEDs. Connect them up to the Rand and Scan inputs on PortMUX.

We'll create both of those functions using existing elements and no hard coding. Open LEDRand, select the LEDs block, and add three PullConnection inputs. Rename them LED1, LED2, and LED3.

From Automation in Resources, drag an AddRandom <unsigned char> component to the left of the LEDs block. Change its Upper value to 2 in the right-hand panel – it will generate values that will be interpreted as a 1 or 0, to turn an LED on or off.

Connect its Out output to the LED1 input on the LEDs block. Let's create two more AddRandom blocks. Right-click the existing AddRandom block and Copy and Paste into a blank area of the diagram, then select Add. Repeat this process to add a third AddRandom block. Connect the two new blocks' Outs to the LED2 and LED3 inputs respectively on the LEDs block. In the right-hand

panel, change the fs (i.e. frequency) of LEDRand to 2 (or whatever you prefer) to slow it down.

08 Scan function

Our LEDScan function flashes the LEDs in sequence, using bit shifting. We don't have room here to describe its creation in detail, but you can recreate it or copy it directly (by holding **SHIFT** while dragging over it) from the downloadable project (**magpi.cc/github**). If you do copy it over, make sure to copy the three LED1, 2, 3, connections (right panel) from the LEDScan block into the LEDs block you created earlier for your LEDScan function, then delete the copied-over LEDScan block.

In your LEDScan function, select Count on the left and untick Input, then connect the Out of the MUX block to it. We need to loop it back to the start, so right-click Count and Copy Path. Select the In of the ShiftLeft block and paste the path into the Routing field.

Finally, check the values (right panel) are the same as those in the downloaded project version for the ShiftLeft, GT, and MUX blocks. Change the LEDScan function's fs property to 2 to slow down the sequence.

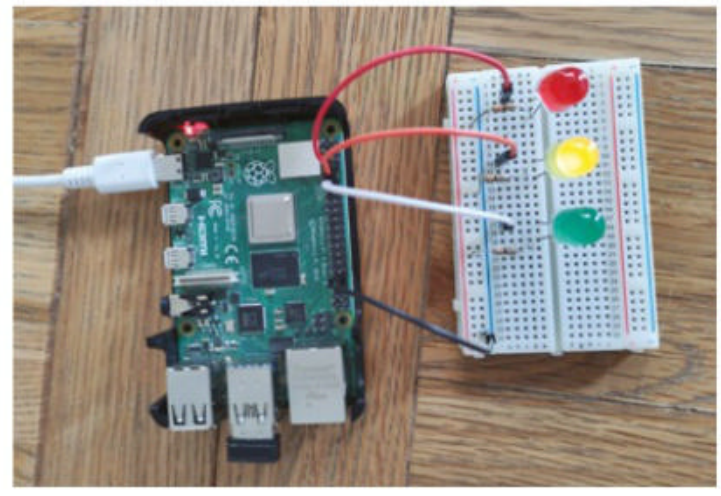
09 Web GUI

Let's add a web GUI to control the program. Right-click RPiLEDsApp in the top-left panel and Add New > Add Web UI. Use the default options in the dialog.

Switch to Design mode to start designing the UI. Delete the existing text box, then drag a Grid Layout element into the canvas. Drag four Buttons into it. Drag a Lamp (representing an LED) to the right of the second button from the top. Then drag two more Lamps to the right of the lower buttons. Drag a Label to the right of the top button; double-click it and change it to 'LEDs'. Double-click the buttons and label them: Off, Rand, Scan, and Sine.

Select the Off button. In the bottom-right panel, scroll down to cdpCheckedRouting. Here you'll need to input the routing for the button so it works. To do so, return to the RPiLEDsApp diagram, right-click Index on PortMUX, and Copy Path. Back in Design mode, paste it into the cdpCheckedRouting field for the Off button. Tick checkableByEmitValue.

For the Rand button, tick checkableByEmitValue and change valueToEmit to 1; paste the path (as before) into cdpCheckedRouting. Repeat this for the other buttons, with valueToEmit set to 2 and 3 respectively.



A simple LED circuit is connected to Raspberry Pi's GPIO pins

Now to route the Lamps. In the Block Diagram, select the LEDPort output on PortMUX, then right-click LED1 (right panel) and Copy Path. In Design mode, select the top Lamp and paste the path into its cdpStyleRouting field. Select each of the other Lamps, then paste the path into cdpStyleRouting and change the last digit to 2 and 3 respectively.

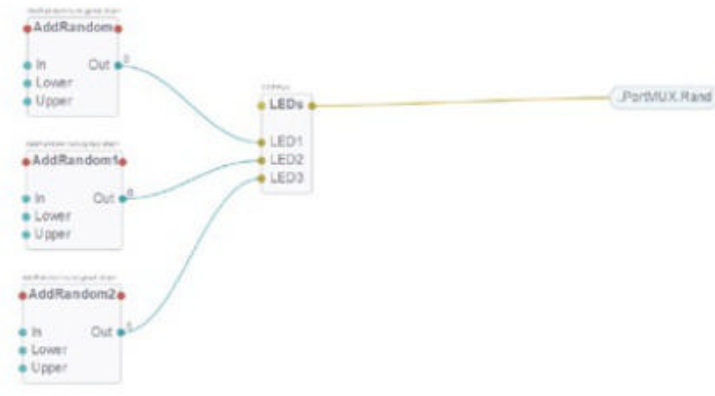
10 Ready to deploy

In Configure Mode, select the LEDs library and the Deploy Configuration tab. Tick the box for Raspberry Pi ARMv6 and/or ARMv8 under Toolkits. Rebuild the library.

Make sure your Raspberry Pi has SSH enabled and you've modified its **/etc/security/limits.conf** file – see **magpi.cc/cdprpisetup** for details. Wire up the LEDs circuit as in **Figure 1**.

To find your Raspberry Pi from CDP Studio, in Deploy Configuration for RPiLEDsApp, click Select next to WiFi under Networks to show devices. Enter **pi** as the Username and click Pair, then enter Raspberry Pi's password.

Choose the appropriate Toolkit for your Raspberry Pi OS version; under Device, select your Raspberry Pi's IP address. Right-click RPiLEDs in the left panel and select Run & Connect. The web UI can be found at your Raspberry Pi's IP address followed by **:7869/index.html** – for example, ours was **192.168.1.112:7689/index.html**. You can now click the UI buttons to change the LED pattern.



The LEDRand function comprises three AddRandom blocks, one for each LED

Top Tip

Deploy on PC

Before deploying the project on Raspberry Pi, you may want to test it by running it on the local PC, in which case the web GUI is at **127.0.0.1:7689/index.html**.

Talking lube

Most of the things you ever wanted to know about lube, but were afraid to ask

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

There's an old engineering axiom: if it moves and it shouldn't, use gaffer tape. If it doesn't move and it should, use lube. Occasionally acetylene is also mentioned, because something can't be stuck if it's liquid, but melting metal is usually the tool of absolute last resort when it comes to getting parts to move. But what exactly does 'lube' mean? For that matter, what do 'grease' and 'oil' mean? This article should help you make the right choice when you have to apply the slippy stuff to a project.

Lubricant is one of those big umbrella words that covers a whole range of different products. Chances are that some of you will be thinking about metal surfaces covered with oil and grease when you see the word lubricant, but metal grease and oil are only part of a much wider range of products that cover practically every possible material. To break your pre-existing mental image of what lubricant is, let's start by talking about sewing and knitting. Cotton thread and yarn aren't perfectly smooth. They have some areas that are more likely to tangle or snag on the weave of materials, or on themselves. Threads and yarns are often treated with special anti-static lubricants called coning oils to stop this from happening and ensure that they can be worked smoothly without tangles.

Right
There's a world of lube out there just waiting to be discovered. Mineral oils, synthetic oils, silicones, dry lube, specialist applicators, waxes, and thermal grease. Each type of oil, grease, or wax has a special purpose that makes it valuable

QUICK TIP

Sometimes, the best lube is none at all. Certain products like drylin® linear bearings are self-lubricating. Applying external lubricant isn't going to help much, and may even ruin the bearings.

If you've ever had to work with musical instrument valves or locking mechanisms, you'll know that oil and grease aren't the only types of lubrication. Dry lubricants like graphite powder or PTFE are used in situations where the natural stickiness of oil or grease would cause a problem. In a lock mechanism, an oil or grease would work like a glue, sticking any pieces of dirt, dust, or other contaminants into the mechanism. A dry lubricant like graphite powder doesn't have that problem.

IF IN DOUBT, FLUSH THE OLD OUT

So why do people specify the use of grease rather than oil? In the simplest terms, grease is sticky and stays where you stuck it for longer than an oil would. Oil is less viscous, and it will follow gravity relatively quickly if it isn't restrained in some way. You might expect that the lubricity of an oil is always better than a grease, but that isn't necessarily the case. At a molecular level, lubricants like oil tend to stick to →

THE **LOWDOWN**

If you're wondering about the difference between grease and oil, it all comes down to temperature. Grease is generally a mix of oils and additives that make it thicker at room temperature. The consistency of the oil at a given temperature will depend on the type and amount of additives, as well as the viscosity of the base oil. As a blended product, the properties of the ingredients used to make it will affect the properties of the end product. Additives like bentonite clay or lithium-based complexes are the general favourites for thickening a grease, although other additives may be used for specialist greases. As an example, copper may be used to make a high-temperature grease with anti-seize properties. Copper grease is handy to keep around the workshop, as it has several high-temperature uses – particularly in engines and battery terminals. The principal disadvantages of copper grease are that it can damage rubber seals, and is electrically conductive.

Above
The gearbox on this lathe isn't enclosed, so the gears need regular lubrication. You can see three brass lubrication points in the top of the gearbox on the left of the image. A sprung ball bearing in each fitting prevents contamination from falling through the holes

TUTORIAL

Right
These grease-fittings are designed to screw into a threaded hole and accept the standard grease-gun fitting. This makes it easy to add grease points to a machine. The ball bearings prevent debris from entering the system, and also act as a one-way valve to prevent oil being squirted comically back out of the hole if it is placed under pressure

Below
Having perfectly smooth machine ways isn't the best idea from the perspective of smooth running. Machine ways are often scraped and levelled with a slightly textured finish that gives pockets of lubricant somewhere to pool. If you had a perfectly smooth surface, any lubricant would quickly squish out of the sides, leaving the two metal surfaces without any effective lubrication

> **When it comes to working with wood, tallow and** beeswax are the lubricants of choice

themselves more than they do to other surfaces. They are viscous, but also have a low friction level, which can be modified with a chemical additive to make them more or less slippery. With a grease, an additive is used to increase the thickness of an oil, but it's still oil that's doing the lubrication. The grease essentially releases some of the oil it contains, like a sponge being squeezed. When grease is under load, this liberated slick of oil is what does the job of lubrication. However, if you aren't using the right grease for the job, then you might find that the grease acts more like glue than a lubricant, and for effective thermal transfer (getting heat away from something hot, quickly) oil is usually a much better choice.

Some situations just don't need a lubricant with the consistency of grease. Cars and machine tool gearboxes, for example, are often sealed units with a bath of oil. The oil is contained within the gearbox, and it can be drained or replaced easily. Grease would be much more difficult to remove in this situation.

When it comes to working with wood, tallow and beeswax are the lubricants of choice. Adding a light coating of beeswax or tallow to wooden drawer slides, or even to screws, nails, or saw blades will make things run much more smoothly. There are even non animal-based tallows available if the thought of using an animal product isn't to your liking, and even dry PTFE lubricants make a suitable alternative in some cases.

Penetrating oils like the classic WD-40 (yes, WD-40 is an oil, despite what the old mechanic in the pub might have told you – check their website if you don't believe me) deserve a special mention in this article. Penetrating oils have a very low viscosity, which allows them to slide down into the tiny cracks and pores in a surface and displace what's already there. Not only are they lubricants, they're also very good at cleaning away thicker greases, loose rust, dust, and varnished oils.

If you're a machinist or metalworker, then you're probably familiar with cutting oil or cutting fluid – which is a specially formulated lubricant used to protect your cutting tools by cooling and lubricating them. If you find yourself in a corner, pretty much any light oil (like penetrating oil) will work as a cutting or tapping fluid, although dedicated cutting fluid will give better performance.

This article has made a whistle-stop tour of some of the more common forms of lubricant, but it isn't an exhaustive list. Some items have been left out because not every grease or oil is primarily a lubricant. Packing grease, for example, is incredibly useful for getting a watertight seal around a marine driveshaft or tiller coupling – but, in these cases, the grease's primary purpose isn't lubrication. Thermal greases and oils have been left out for similar reasons. Take a look around the web and figure out which greases and oils you're missing from your collection. Most aren't too expensive, and it's handy to have them in stock when you need them.

VISCOSITY

The viscosity of an oil is often referred to as its 'weight', even when it shouldn't be. It's one of those hangovers from the old days that just doesn't want to go away, like imperial units. In the most basic terms, the higher the weight of an oil, the thicker it is. If you've ever done an oil change on your car, then you might notice that most of the oils you can buy have two measurements on the bottle, separated by a W (like 15W40 for example). You'd be forgiven for thinking the W stands for 'weight', but it actually stands for 'winter'. Oils rated this way are known as multi-grade oils, and the explanation of what that means is actually quite simple, once you think about it.

The weight of an engine oil is assigned by the Society of Automotive Engineers, based on how the oil performs at 210° Fahrenheit. That's roughly the temperature that most ICE car engines perform. In the winter, cold-starting your car in sub-zero temperatures, a simple oil would be too thick to pump through the engine. An SAE 15W40 oil behaves as though it's a 15 weight oil while it's cold, but as a 40 weight oil when it's at 210 degrees.

Weight isn't the only measure of viscosity, and when it comes to things like hydraulic oils and modern machine oils, the ISO standard is probably the most popular measure. An SAE 10W oil is equivalent to ISO 32, while SAE 30W is ISO 100. As with all metric to imperial conversions, the transformation doesn't scale as predictably as you might want.

QUICK TIP

Some oils will melt or degrade certain types of plastic and rubber. Check before you melt your project into goo.

Above
Lots of modern machines don't have greasing or oiling points because bearings are sealed with lubricant inside. Once a bearing or bushing loses its lubrication, it's expected that the part will be replaced. However, this scary-looking fitting for a grease gun lets you inject sealed rubbers with grease, extending the life of the part. Obviously, the part should still be replaced if the bearings are worn, but a little bit of extra lubrication is generally a good thing until that can happen

Left
WD-40 makes a good cutting oil for acrylic, and in a pinch it can also be used as a cutting fluid for other materials

TUTORIAL

Communication by light beams

Never one to abide by convention, Mike Bedford spurns his mobile phone in favour of communicating by light. Here, he shows you how

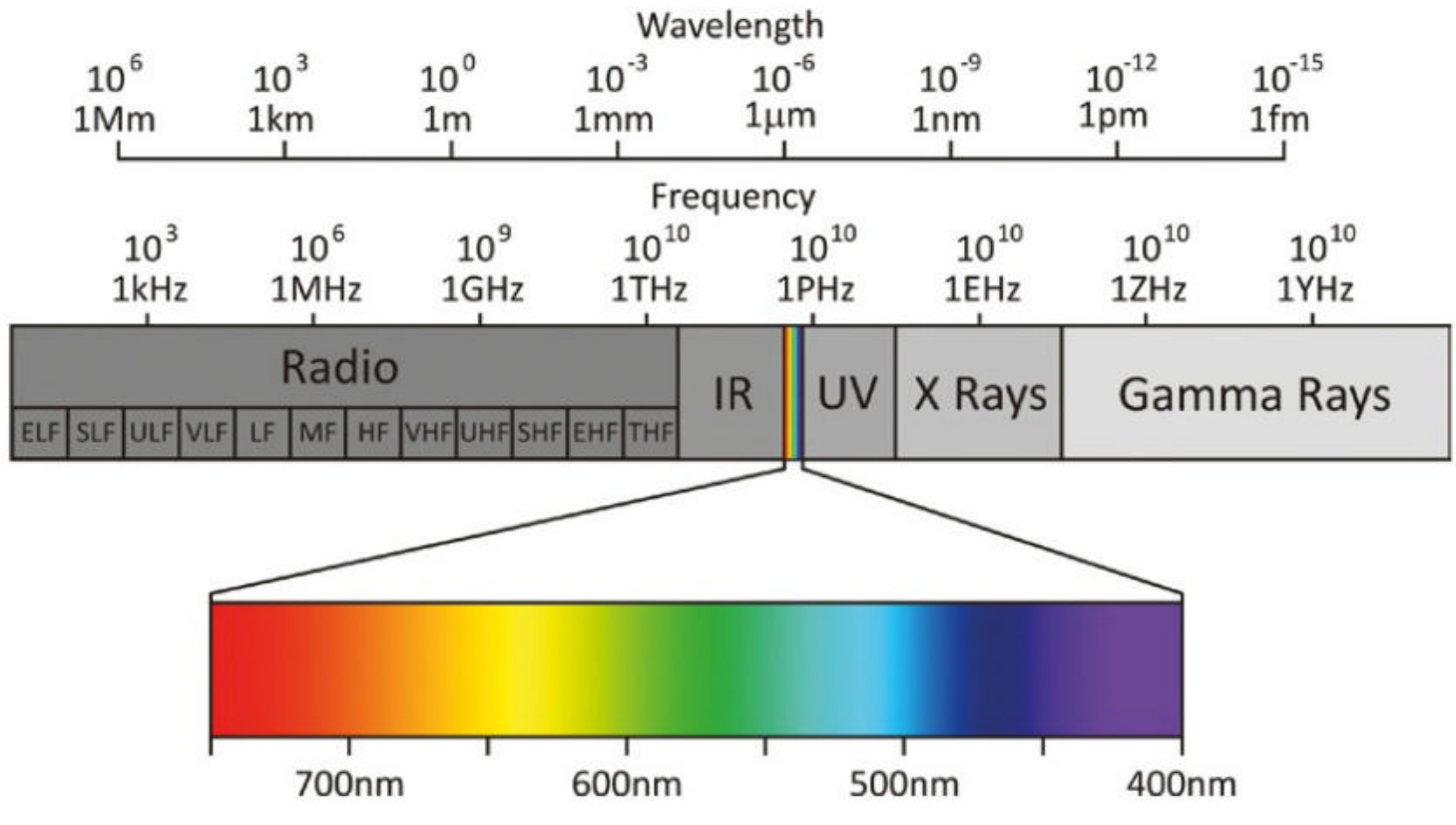
Mike Bedford

Despite loving all things digital, Mike admits to being a bit of a Luddite, vinyl records and all.

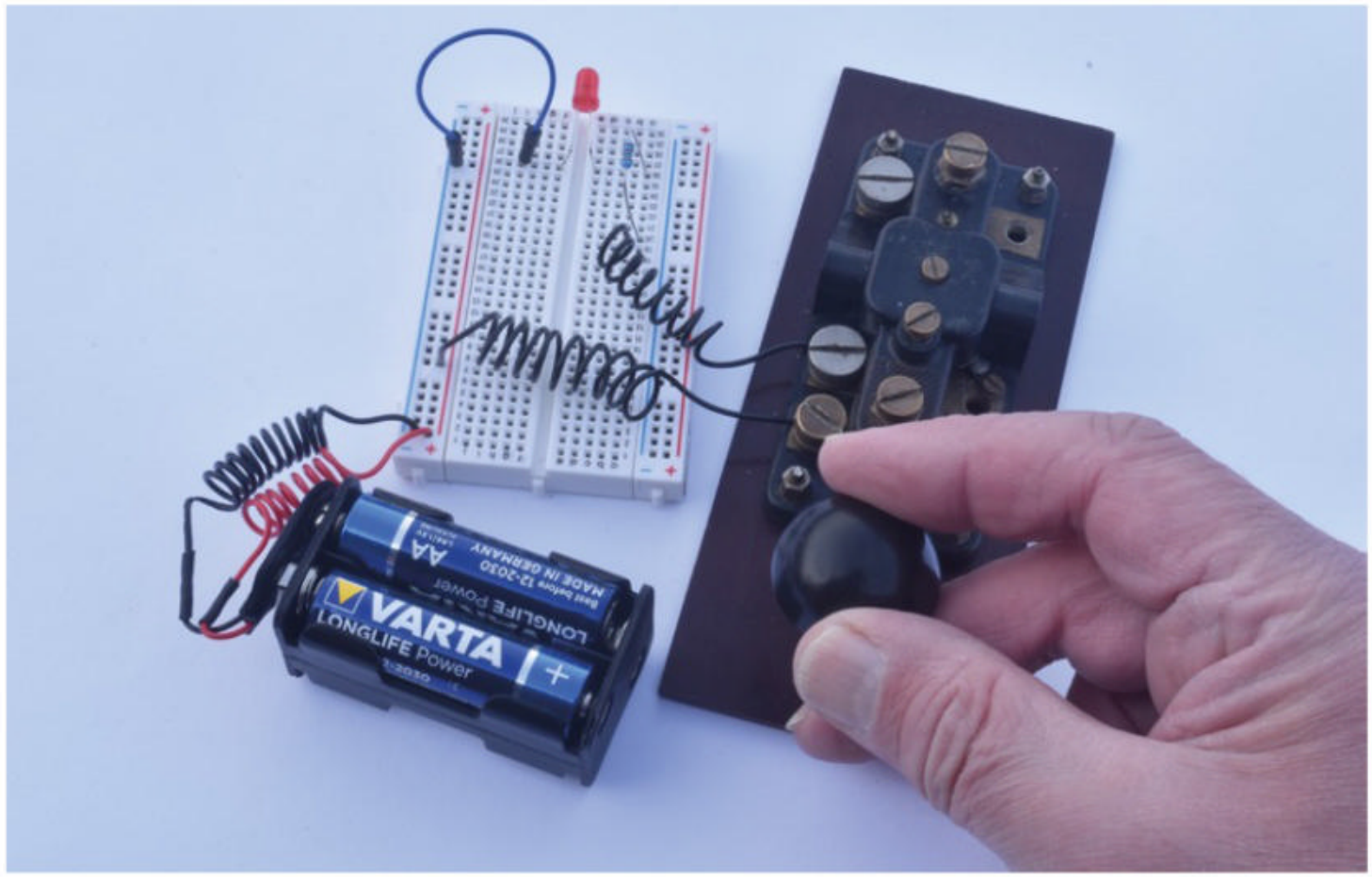
We're used to radio being defined by its frequency; what might not be as well-known is that radio ends at 3 THz – that's 3000 GHz. This is the top end of the part of the radio spectrum known as THF, that's Tremendously High Frequency, not nearly as familiar as VHF, UHF, and a couple of steps up from SHF and EHF. There's nothing magic about 3 THz, though – it's just a convention that this is where radio finishes – and electromagnetic radiation goes beyond this frequency, well beyond. This begs the question of whether frequencies above 3 THz can be used for communication, just as radio is. The answer, it transpires, is yes, and this is our subject here. In particular, we're going to see how light can be used to pass messages through the air and, at its simplest, this couldn't be much more straightforward. You're certainly not going to be covering the thousands of miles that short wave radio can span, but it's quite an eye-opener – quite literally – to communicate using a visible beam of light.

LIGHT PRIMER

Light covers the range of frequencies from 400 THz to 750 THz, with red at the bottom end and violet at the top end. As an alternative to frequency, electromagnetic radiation can be defined by its wavelength, a measure that decreases as frequency increases. For radio, talk of wavelengths died out decades ago, but with light, it's the most commonly used measure. So, we're talking of 650 nm for red



Right
Just like radio waves, visible light is part of the electromagnetic spectrum, but it has a much higher frequency



Left
Communicating by Morse using a beam of light couldn't be simpler. And if you choose a high-brightness narrow-angle LED, you might span a few kilometres

light through to 400 nm for violet, and you'll probably have encountered similar figures for specifying LED colours. Oh, and just in case you were wondering, the gap between the top end of the radio spectrum and red light is where infrared is found.

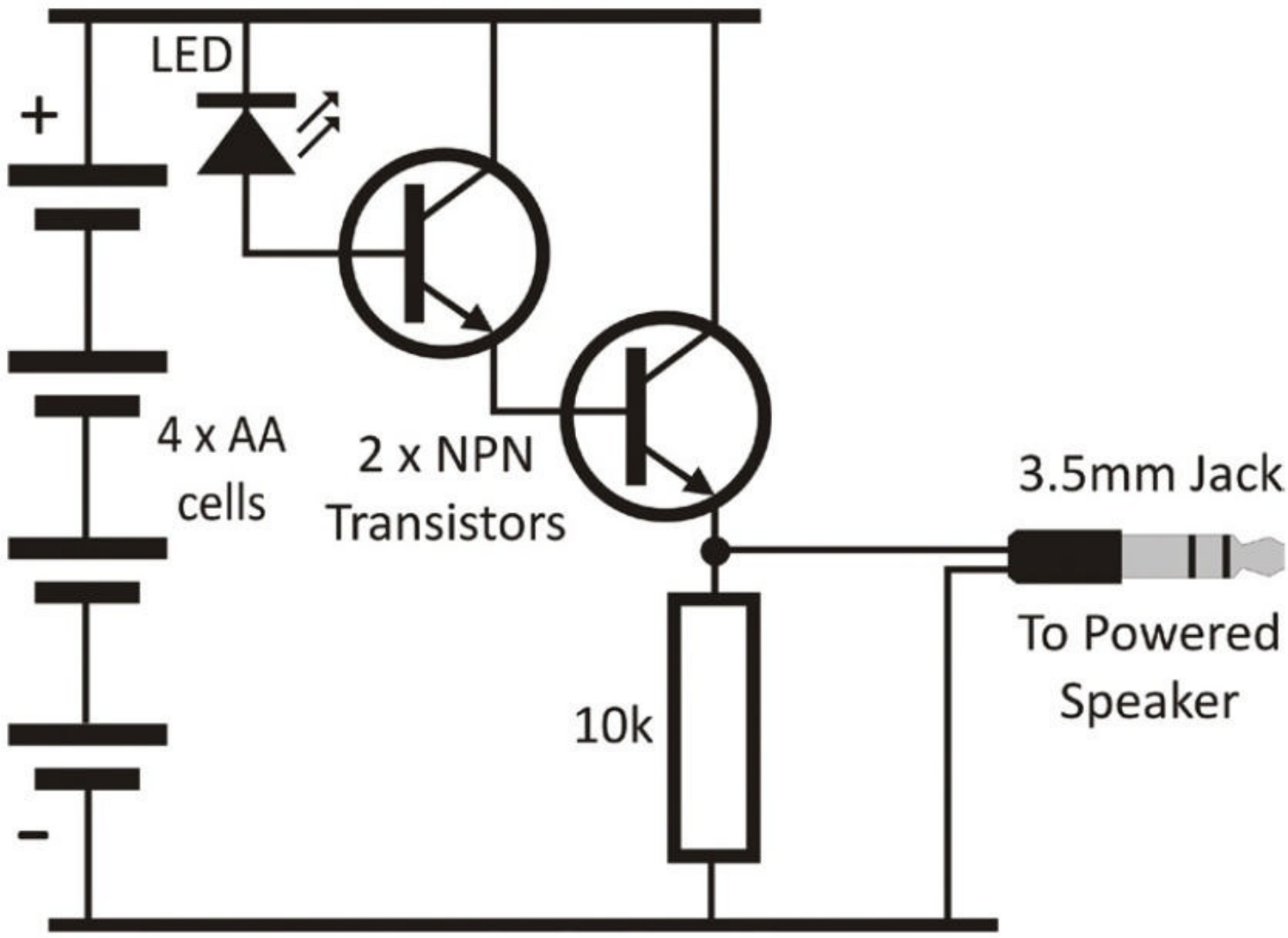
In thinking about light beam communication, we need to bear in mind that light is absorbed by the atmosphere. What's more, the absorption depends on the wavelength, with red being absorbed least, and violet most. This explains why even a hugely powerful source of white light – the sun – looks red at sunrise and sunset, when it's low in the sky so its light passes through more of the atmosphere. So, the bottom line is that using red light will maximise the range.

> In thinking about light beam communication, we need to **bear in mind that light is absorbed by the atmosphere**

MORSE BY LIGHT

Our first method of sending a message via a beam of light is trivially simple, yet it provides plenty of scope for learning about how various factors affect the range. It involves nothing more complicated than turning an LED on and off with a Morse key, so all you need is the LED, a current-limiting resistor, a battery, and the Morse key, all connected in series. Morse code provided the first ever method of transmitting text via telegraph, radio, or even light, and we delved into it in HackSpace #52. We used an ordinary red LED and managed to communicate over more than 100 m in the dark, receiving the signal purely by eye, but we ran out of space so it's not clear how much further we could have got. The distance can be increased significantly though, as we'll see later. If you're experimenting on your own and don't have someone to send Morse, because it's much easier to see a LED if it's flashing rather than continuously on, we suggest you use a simple bit of circuitry (or a Raspberry Pi) to turn it on and off every second or so. Also remember that an LED fires most of its light from the domed end opposite the leads, so be sure to point the business end in the direction from which you're going to receive it.

The brightness of an LED, which obviously affects the range, is defined by two different measures in datasheets, although many manufacturers don't provide both. The first is called the luminous flux, which is measured in lumens (lm), and specifies the total light output. The second is called the luminous intensity – it's measured in candelas (cd) or milli-



candelas (mcd), and it specifies the brightness in a particular direction. So, if two LEDs have the same luminous flux but different beam angles, the one with the narrower beam will have a greater luminous intensity because it concentrates the light. You can find the formula and calculators online but approximately, for typical LED angles, halving the angle will increase the luminous intensity by a factor of four for the same luminous flux.

Returning to our rather modest range, the LED we used was a 5 mm panel indicator, as opposed to one intended as a light source, so it didn't have a particularly high luminous intensity, yet that's the figure that affects the range. The datasheet showed a luminous intensity of about 10 mcd at 20 mA, and a beam angle of 60 degrees. What this means, of course, is that if you chose a similar LED with a 20 degree beam angle, it would have a luminous intensity of about 90 mcd for the same power of 40 mW (20 mA × 2 V). That's just with ordinary LEDs, though, and if you were to use high brightness LEDs, you can find as much as 100 cd and, although it'll draw more power, perhaps not by a vast amount. That would be 1000 times more than our ordinary wide-angle LED, but that certainly doesn't mean you could communicate over 1000 times the distance. Light follows an inverse square law, which means that the luminous intensity decreases with the square of distance. The range depends on the square root of the luminous intensity therefore, so by using that 100 cd LED, our 100 m could, perhaps, be increased to several kilometres, if you can find a sufficiently long line-of-sight path. If you experiment with narrow-angle LEDs, though, be sure to take care in pointing it in the right direction because if it's misaligned by just a few degrees, the range would be severely impacted. So how far can you get? Astonishingly, radio amateurs have achieved up to 278 km. But they used high-power LEDs and some serious external optics, and they located both stations on mountain tops. Oh, and they also exercised more than just a little bit of patience.

Above
The audio receiver uses an ordinary LED as a photodiode, amplifying its weak output with two transistors in a Darlington pair configuration to drive a powered speaker

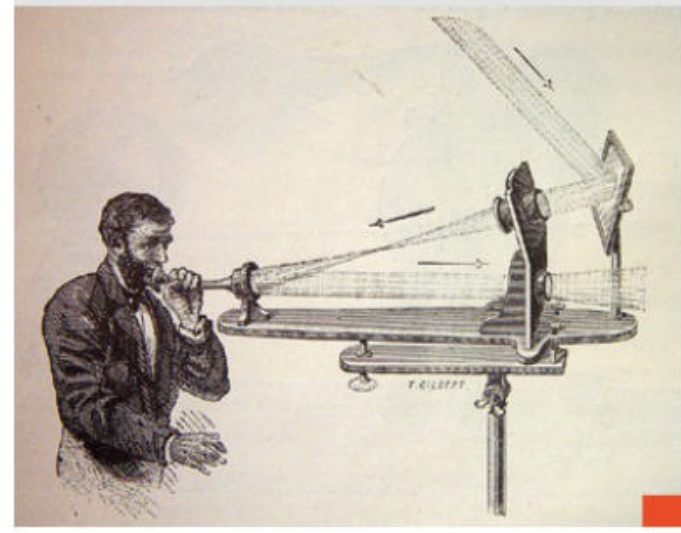
LISTENING TO LIGHT

If Morse is good, then surely speech is better, so let's now look at how to superimpose audio on a light beam. And because our audio setup contains a receiver – something we didn't need for Morse communication, since we were receiving visually – we'll start by looking at methods of detecting light electronically.

There's not just one type of electronic component that acts as a light sensor, but several. One of the best-known is the photovoltaic cell, otherwise known as a solar panel. Commonly, these are used for generating electricity from sunlight, but since the available output current depends on the intensity of the light, it also works as a sensor. Next up is the light-dependent resistor, or photoresistor, the resistance of which is inversely proportional to the light intensity. Then we come to semiconductor devices. There are both phototransistors and photodiodes, but we'll concentrate on the latter here. A photodiode can be used in two ways: photoconductive mode, i.e. light-dependent resistor-like, and photovoltaic mode, i.e. like a photovoltaic cell. And here's an interesting fact – although proper photodiodes are designed to maximise their photo-sensitivity, LEDs will work as photodiodes, but with reduced efficiency. Because you're likely to have spare LEDs at hand, we're going to use one in our receiver. It won't work as well as a proper photodiode, but there's plenty of scope to improve the performance once you've seen speech-by-light operating at short range. In addition to using a dedicated photodiode, and perhaps improving the associated circuitry, you could also employ the →

A HISTORY LESSON

Although LEDs didn't appear until 1960, the first optical device for transmitting voice dates back to 1880. Called the photophone, and invented by Alexander Graham Bell, it featured a mouthpiece that directed the sound onto a reflective membrane which vibrated in step with the spoken word to modulate the reflected sunlight. A photo-resistive selenium cell – at the focus of a parabolic mirror – was wired in series with a battery and an earpiece to act as the receiver.



Above
Ideally, so you don't risk breaking contacts when you move the receiver around, build it (and the transmitter) on stripboard instead of a breadboard

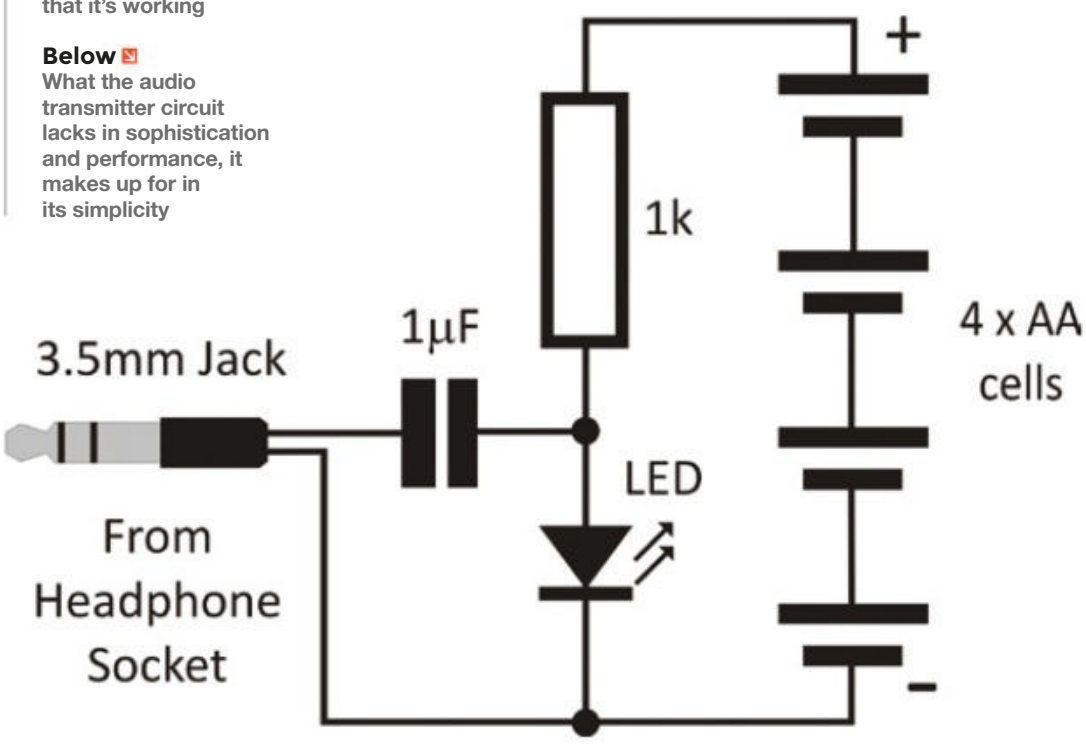
Left
Back in 1880, this remarkable contraption allowed communication over a distance of 213m using modulated sunlight to carry a voice signal

Above
Hearing a buzzing noise when you move your receiver close to a fluorescent tube or halogen bulb proves that it's working

Below
What the audio transmitter circuit lacks in sophistication and performance, it makes up for in its simplicity

various techniques that we saw for increasing the range of the Morse code setup.

To keep things simple, the circuit uses a powered speaker, so we only need to add some electronics to adequately amplify the signal from the LED. We used a powered speaker with just a line-in socket, but a cheap Bluetooth speaker with a line-in socket or a PC speaker would also be suitable. In the schematic, although it's being used as a photodiode, we've shown the photosensitive component as an LED, which is exactly what it is. Unlike a proper photodiode, LEDs are mainly sensitive to the colour they emit, so we suggest using a red LED, both here and in the transmitter. We're using the LED as a photodiode in its photoconductive mode, which requires it to be reverse polarised: the positive supply voltage is applied to the cathode so a current will flow in the presence of light. This current isn't nearly high enough to drive the powered speaker, so amplification is needed. To do that, the current is applied to the base of the first transistor, the two being wired as a so-called Darlington pair which provides much higher gain (i.e. amplification) than a single transistor.



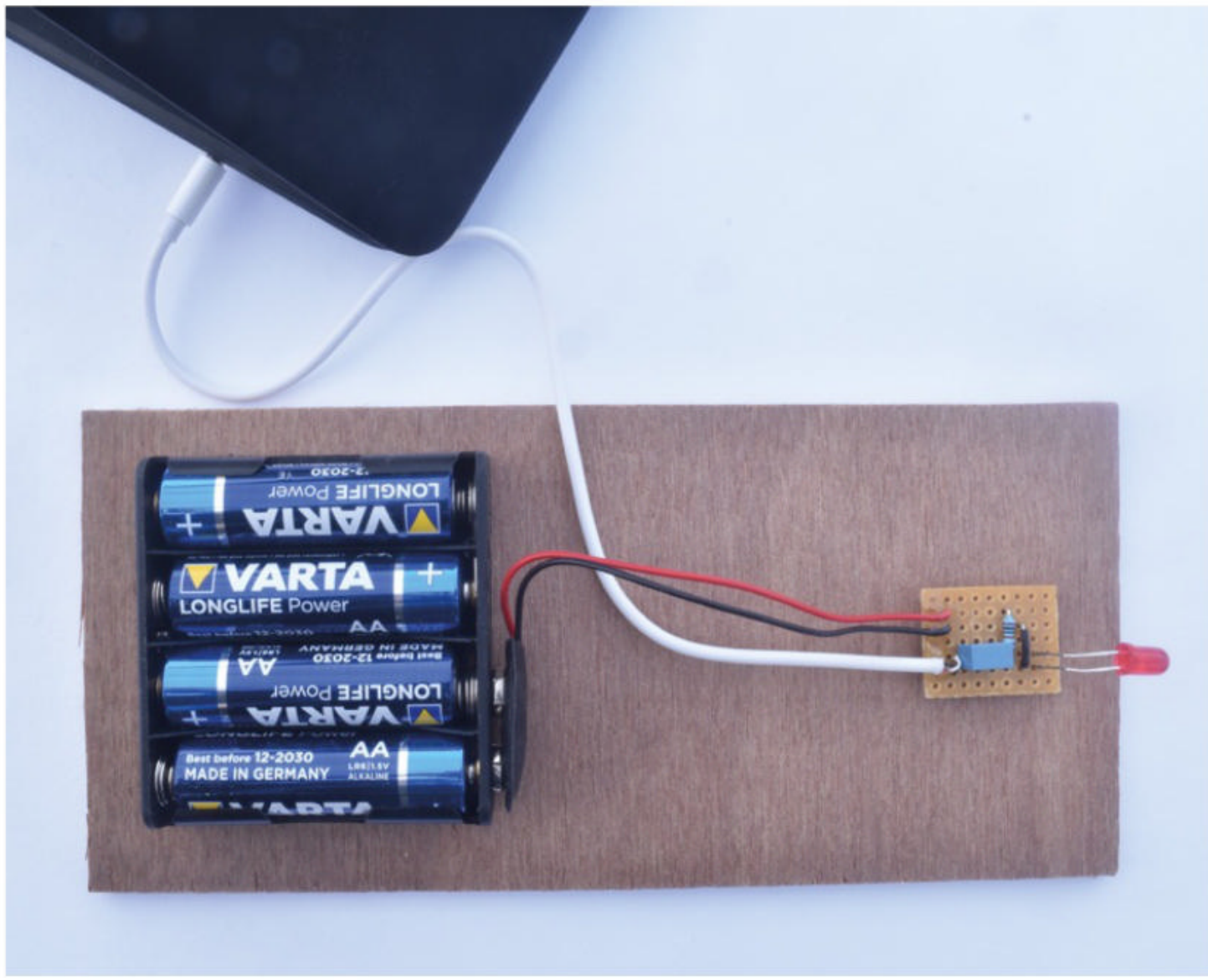
> **A benefit of building the receiver first is that you** can try it out before you make the transmitter

The transistors can be any small-signal, general-purpose PNP bipolar transistor, like a 2N2222N or a BC184L. When the current from the LED turns on the Darlington pair, a voltage appears on the emitter of the second transistor, and this provides a suitable input to the powered speaker.

A benefit of building the receiver first is that you can try it out before you make the transmitter. If you hold the receiver close to a mains-powered filament lamp, including a halogen bulb, or a fluorescent tube, because their light fluctuates rapidly, you'll hear an audible noise. We found that you don't hear nearly as much with the receiver close to a LED lamp, although you should hear a noise if you move your hand rapidly between the bulb and the receiver.

THE TRANSMITTER

For the transmitter, we used an audio source with a headphone socket to modulate the light generated by the LED – or, in other words, vary its brightness according to the level of the audio. For the audio source, you could use a cheap portable MP3 player or a laptop, although we suggest the former if you're not confident of your electronic constructional skills, to prevent any risk to your laptop, however remote. The voltage output of a headphone socket isn't high enough to drive an LED, and we had planned to use an amplifier. However, having noticed a circuit similar to the one presented here, despite our initial scepticism, we decided to go with this super-simple



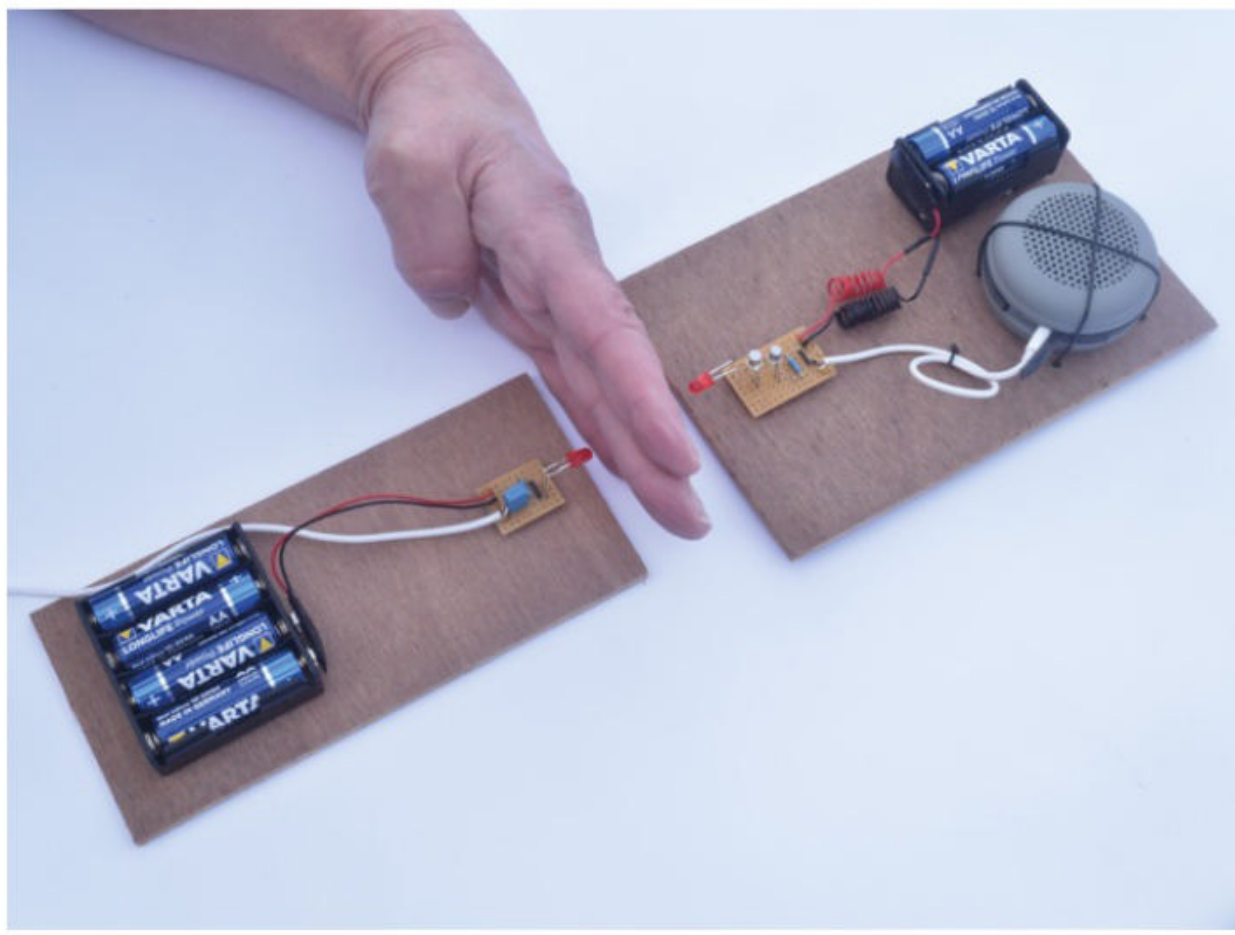
Left
The transmitter uses an audio signal from a laptop or portable MP3 player to modulate an LED

Below
Now you hear it, now you don't. If you doubt that the signal from the transmitter to the receiver is really carried by light, just put your hand between the two LEDs and hear the sound of silence

approach. The 6V supply and resistor illuminate the LED but only dimly, and the audio signal increases the voltage across the LED to modulate it. Depending on the LED, you might need to fine-tune the resistor value.

When you turn on the transmitter, the LED will light, but you'll probably struggle to see it flicker with the audio. However, if you move your receiver close to the transmitter, with the two LEDs facing each other, you should hear the audio from the receiver and, if you place your hand between the two LEDs, the sound will stop. You're not going to cover long distances, though, and that's an understatement: we're probably talking a few centimetres at the most. But using a proper amplifier in the transmitter will improve things a lot, as will choosing a high-power narrow-angle LED, as we saw earlier.

As a parting shot, don't dismiss this exercise as little more than a fun way of learning something about optoelectronics. A similar method is now being promoted in the form of LiFi as an alternative to WiFi. Several benefits are on offer, including the ability to transmit data at up to 100Gbit/s.



TUTORIAL

Moving beyond the K40

Running your laser cutter too hot for too long can cause burnout. Let's explore the best ways to cool it

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Cooling your laser is a hot topic on the internet, and it's one area where Heath Robinson-inspired solutions seem to flourish. The focus on cooling is understandable, because proper temperature control is important for prolonging the life and effective power of your laser tube. Running a laser tube too hot for a long time is a sure way to burn it out. In this article you'll take a look at some of the options available to cool your laser, and learn which cooling solution best suits your need.

If you own a K40, then you probably remember the slight confusion that you felt when you opened up the box and removed what appeared to be a common garden pond pump and some clear tubing from the packaging. The K40 is a relatively low-powered laser, but it probably seemed unlikely that a small submersible pump would be enough to keep things cool for any length of time. In true K40 laser style, the pump and hose provided in the box represent the absolute bare minimum amount of kit required to get the laser up and running without instantaneous catastrophic failure. If you want to do anything more than run the laser for a few minutes at a time, you're probably going to need a better solution than a garden centre special offer and a bucket.

It's important to understand that your laser cutter exists as part of a larger environment. The cooling system you use will be sympathetic to and part of that environment. If your laser cutter is housed in a cold workshop during a Canadian winter, then you'll have less need to dissipate excess heat than you would if it were running

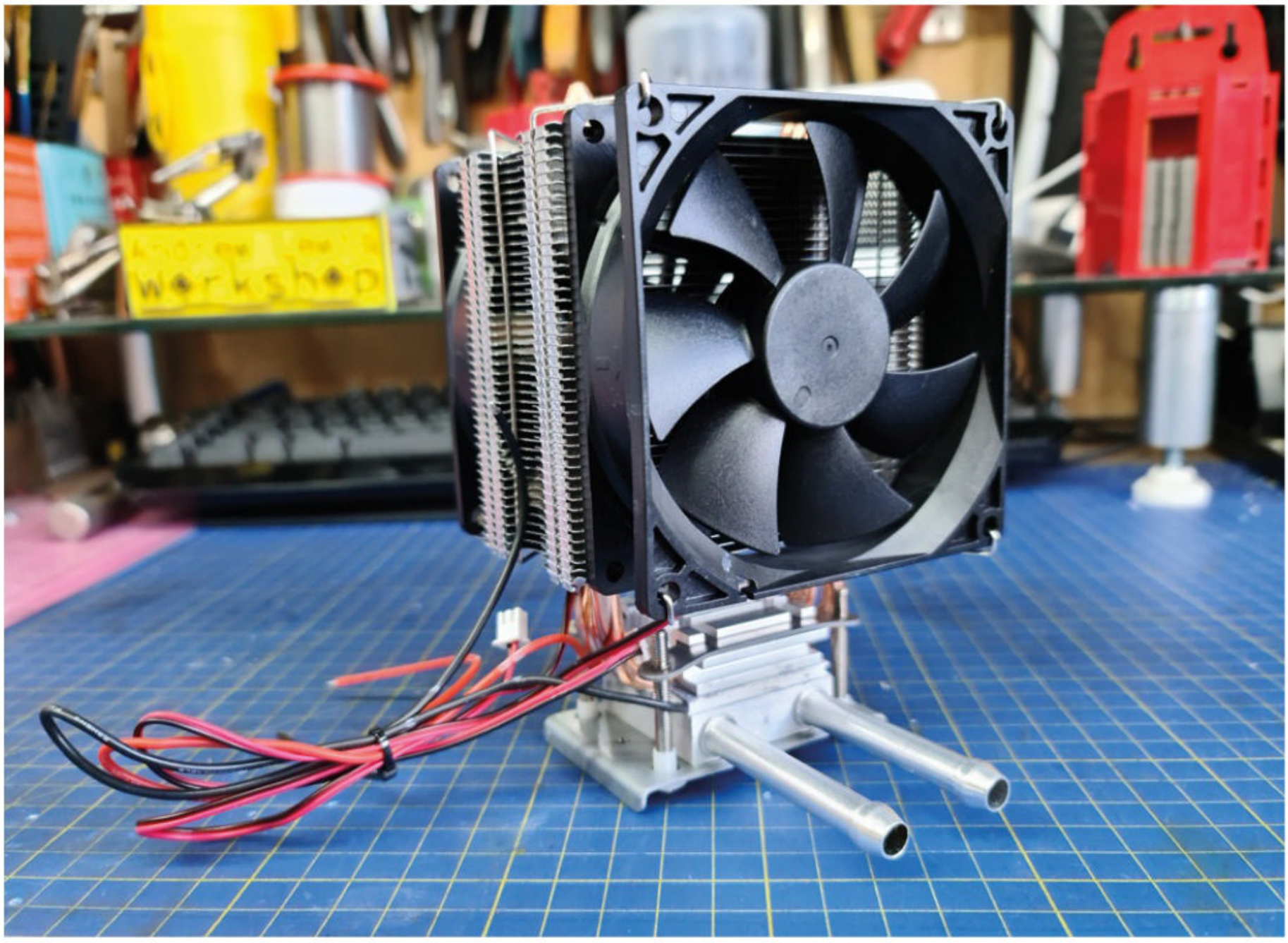
Right
There are multiple solutions available for cooling a laser, but which one is the best for you? For professional users, an active cooler like this is the most effective choice. For the hobbyist, a more passive solution might be suitable

on a tropical island in the middle of a bright summer day. The surrounding environment always has an effect on the ability of a cooling solution to keep up with the excess heat generated by the laser. Now that we understand that, let's look at some of the solutions available for cooling the laser.

The simplest cooling solution is the one provided with a K40 laser. A pump recirculates a reservoir of coolant through the laser and back into the cooling tank. Every time the water recirculates around the system, it gets a little bit warmer. If you only need to use the laser for a few minutes a day, this is probably OK as a cooling solution, with a few caveats. The more coolant you have, the longer it will take the coolant temperature to increase. However, the temperature of the coolant will start out at the ambient temperature of the environment. If your workshop is a steady 20°C, then your coolant tank will be at an uncomfortably high temperature from the beginning. Some people mitigate this problem by throwing frozen items into the tank to cool it down. This might be OK in an emergency, but every time you introduce items into the pure coolant, you risk contaminating it and eventually you will need to flush the system through (probably with acid to remove mineral deposits), replace the coolant, and start the whole process over again.

> Every time you introduce items into the **pure coolant, you risk contaminating it**

Above
Instead of directly dropping frozen drinks bottles into your coolant, you could use coils of pipe as heat exchangers, circulating coolant around the outside of frozen bottles and cans to lower its temperature



Above
When it comes to cooling power, the Peltier cell isn't very efficient at all. But, a Peltier cell can be useful in some situations where spot cooling or the lack of moving parts are an important feature. This cooling unit uses a metal heatsink with heat-pipes and powerful fans to draw excess heat away from one side of the Peltier cell, while liquid coolant is drawn across the other side of the cell. Reversing the voltage to the cell reverses the action, and the liquid cooler becomes a liquid heater

To reduce the amount of coolant needed and extend the maximum runtime of the laser, another common solution is to use a radiator and fans in addition to a coolant reservoir. Again, this solution is limited by the ambient temperature, and the best that fans can do is slow down the increase in temperature of the coolant. This is very similar to the cooling system for most ICE cars and lorries, where the radiator and fans are mounted at the front of the vehicle and the hot coolant circulates around them from the engine. The configuration is no different for a laser cutter, with the radiator connected to the outlet of the laser, cooling the fluid before it returns to the reservoir.

> Another common solution is to use a radiator **and fans in addition to a coolant reservoir**

A less common solution to cooling involves exploiting the environment outside the workshop. If the temperature outside the workshop is lower than inside the workshop, it's possible to move a radiator or coil assembly outside and use naturally occurring lower temperatures to chill the coolant to a level beyond that which would be possible with an indoor radiator. Unfortunately, it isn't always practical to dump heat into the air or through a coil into a conveniently located body of water, although this is a good solution if you can do it. Just be aware that the same caveats apply and that energy flows both ways – if it's hot enough outside, you might end up heating the coolant in the tank rather than cooling it down.

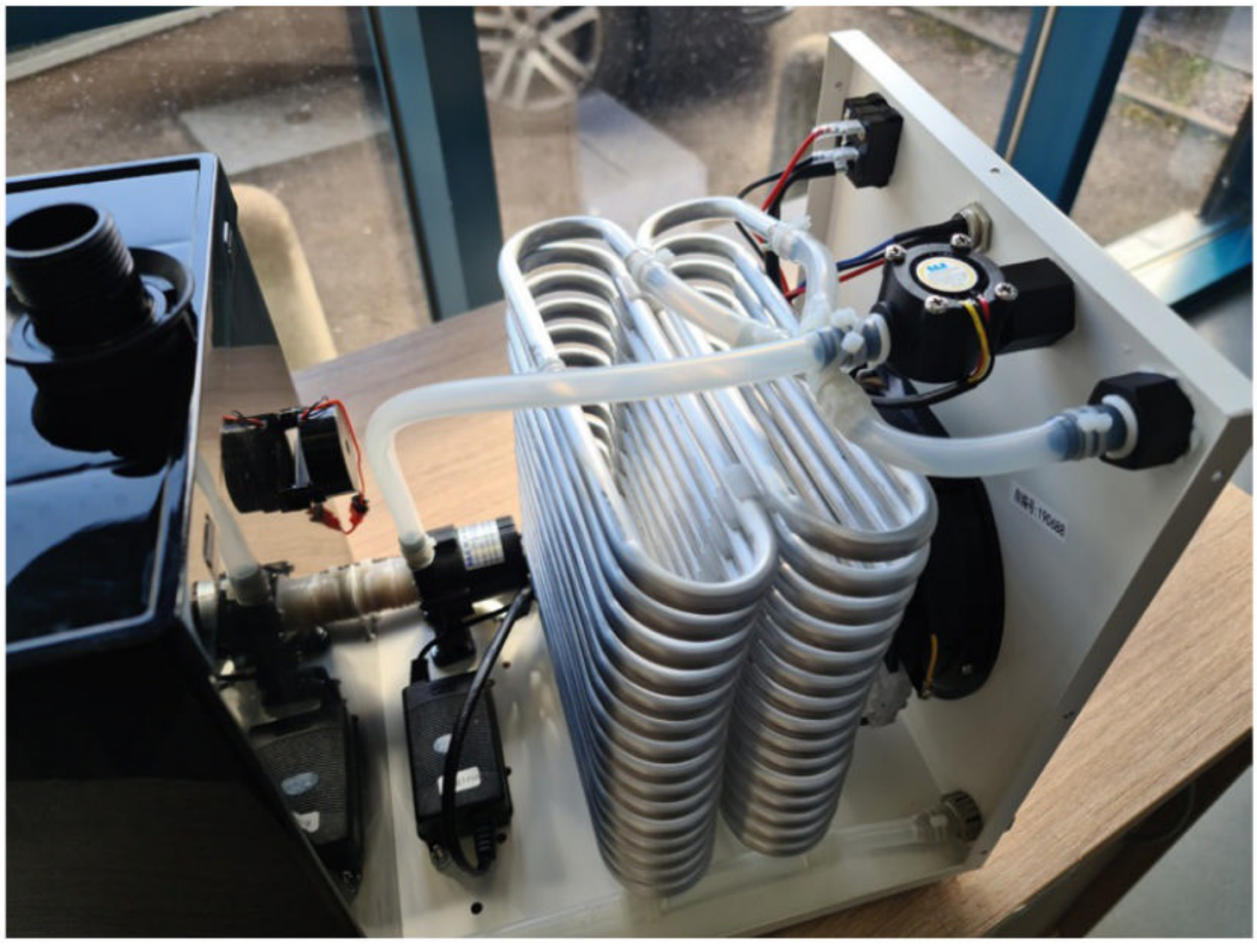
While passive cooling methods might be suitable for casual and hobby uses, professional users need something more reliable than a radiator or a big tank of coolant. That's where active cooling solutions come into play. Unlike passive solutions, active cooling systems exploit the miracle of physics to remove heat from the coolant and maintain its temperature at a level below the ambient environment. It's the same technology that gets used to keep your ice cream frozen and your energy drinks cold – refrigeration. Instead of cooling your tank by throwing in items from the freezer, why not just move the whole coolant tank into the refrigerator and cool it directly?

Active cooling solutions are more expensive than passive solutions, and that extra cost can be prohibitive for casual users. If the laser cutter isn't critical to your workflow, then you might be willing to accept that you can't use your laser cutter sometimes and save yourself some money. However, the advantages of active cooling extend beyond the obvious point that they allow for continuous use of the laser, and may actually help you to save money over time. For one point, maintaining a constant coolant temperature keeps the laser tube in better condition, meaning that it will need replacing less often. Most active systems will also use less coolant than a simple passive reservoir, saving space and saving money when it comes to changing the coolant.

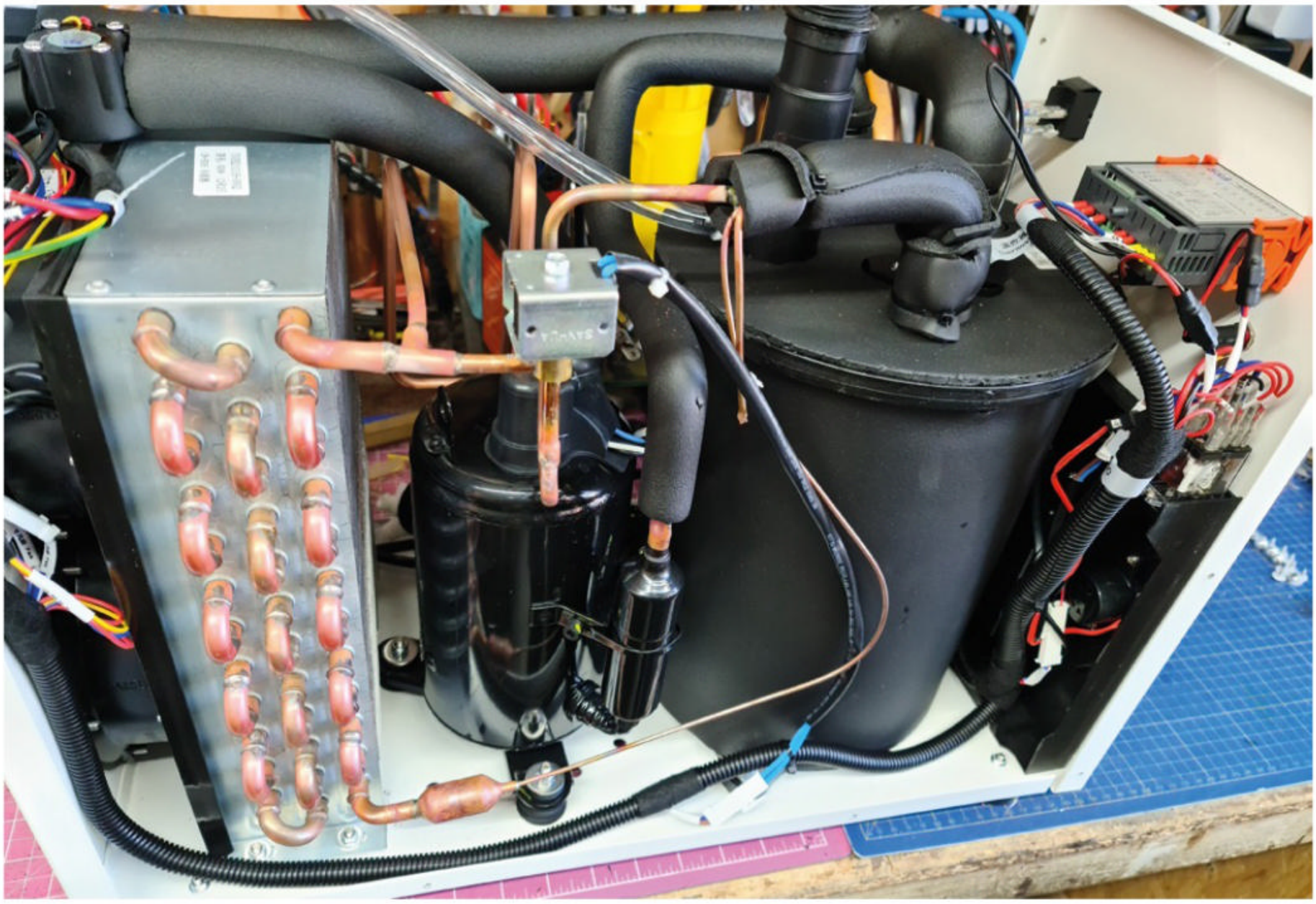
There are lots of examples of homespun active cooling systems on the internet, some of which are very creative. Repurposing air conditioning units and water chillers are common themes, running cooling coils directly into the coolant, or running the coolant through the water chiller. However, some of these solutions might not be effective enough to support a laser running for very long periods of time. While an air conditioning unit should be able to cool a reservoir →

ELIMINATE ALGAE

The coolant tubes provided with a K40 laser are usually transparent. This isn't good, because allowing sunlight to shine on the coolant will promote the growth of algae. Although some chemicals exist to prevent algae growth, they are generally aimed at water cooling systems in computers, where conductivity of the coolant at high voltages isn't an issue. Combat algae by using UV light-fast tubing, or enclose all of the tubes in a light-fast conduit to reduce the possibility of organic contamination.



Left
The CW-3000 looks externally similar to the CW-5200, but is a radically different beast on the inside. While the CW-5200 is an active cooler, the CW-3000 is passive, using only coils and fans to reduce the temperature of the coolant. This is absolutely fine for equipment designed to run hot (like a welding generator or a high-speed motor), but it is much less useful for laser cutters. In a warm environment, passive coolers like the CW-3000 can even end up working backwards, blowing warm air over the coils and heating the coolant in the tank to room temperature



Above
Active coolers like the CW-5200 contain a compressor, a large radiator with powerful fans, and an insulated coolant tank. Just like a domestic refrigerator, if left to run unattended, this system would eventually freeze the coolant in the tank solid. That's why an intelligent temperature control unit is used to monitor the coolant flow and temperature. Although these units are technically portable, you should remember that they weigh in at approximately 25kg with the coolant tank empty – you'll need your big belt and serious lifting trousers on if you are going to move one about when it's full!

of water quite easily, a water chiller is a much less powerful cooler, and may even use a Peltier cell to cool the water. Peltier cells are heat pumps that exploit the Peltier effect to heat and cool dissimilar conductors. Peltier cells generally take the form of a ceramic square with two wires connected to the side. Applying power to these wires will cause one side of the Peltier cell to heat up, and the other side to cool down, and they are frequently found in portable fridges and water coolers.

Although the Peltier cell is a useful device, it is not the most efficient cooling technology. When compared to a conventional compressor-based refrigerator, a Peltier cell is only about a quarter as efficient. To cool a laser cutter effectively, you would need to use a lot of Peltier cells, and use a lot of heatsinks with fans to get rid of the excess heat from the hot side of the cell. With the poor efficiency of the Peltier cell, you would need a huge power supply to generate enough cooling capacity. That doesn't mean that Peltier cells can't be used to help cool a laser, but it does mean that it would be expensive and inefficient to use Peltier cells to cool a laser tube completely.

ANY COOLING IS BETTER THAN NONE

Moving away from the homespun solutions for active cooling, you'll find yourself entering the realms of industrial chillers. Industrial chillers are discrete units that pack everything cooling related into one box, and usually have some sort of intelligent control to keep the coolant at a particular temperature. Like all active cooling solutions, industrial chillers are an expensive option, but they are also the best solution for cooling a laser if you don't live in a perfect environment. You would be forgiven for thinking that these types of industrial coolers were designed with laser engravers in mind, but the fact is that lots of different industrial tools need to be cooled down, and most of the chillers out there were designed with other machines in mind. CNC spindle motors and controllers often need to be liquid-cooled, and so do professional

TAKE CARE!

A laser cutter is a dangerous machine. Using an inappropriate or contaminated coolant can lead to catastrophic failure, electrocution, or a fire in the laser power supply. Invisible laser radiation can disfigure or blind you or someone else permanently. Never operate the laser in an unsafe situation, and never let the machine run out of your control. Always wear suitable protective equipment and be aware of the risks that the machine presents. Never leave the laser in a state where it could be activated accidentally by someone who does not understand the risks.

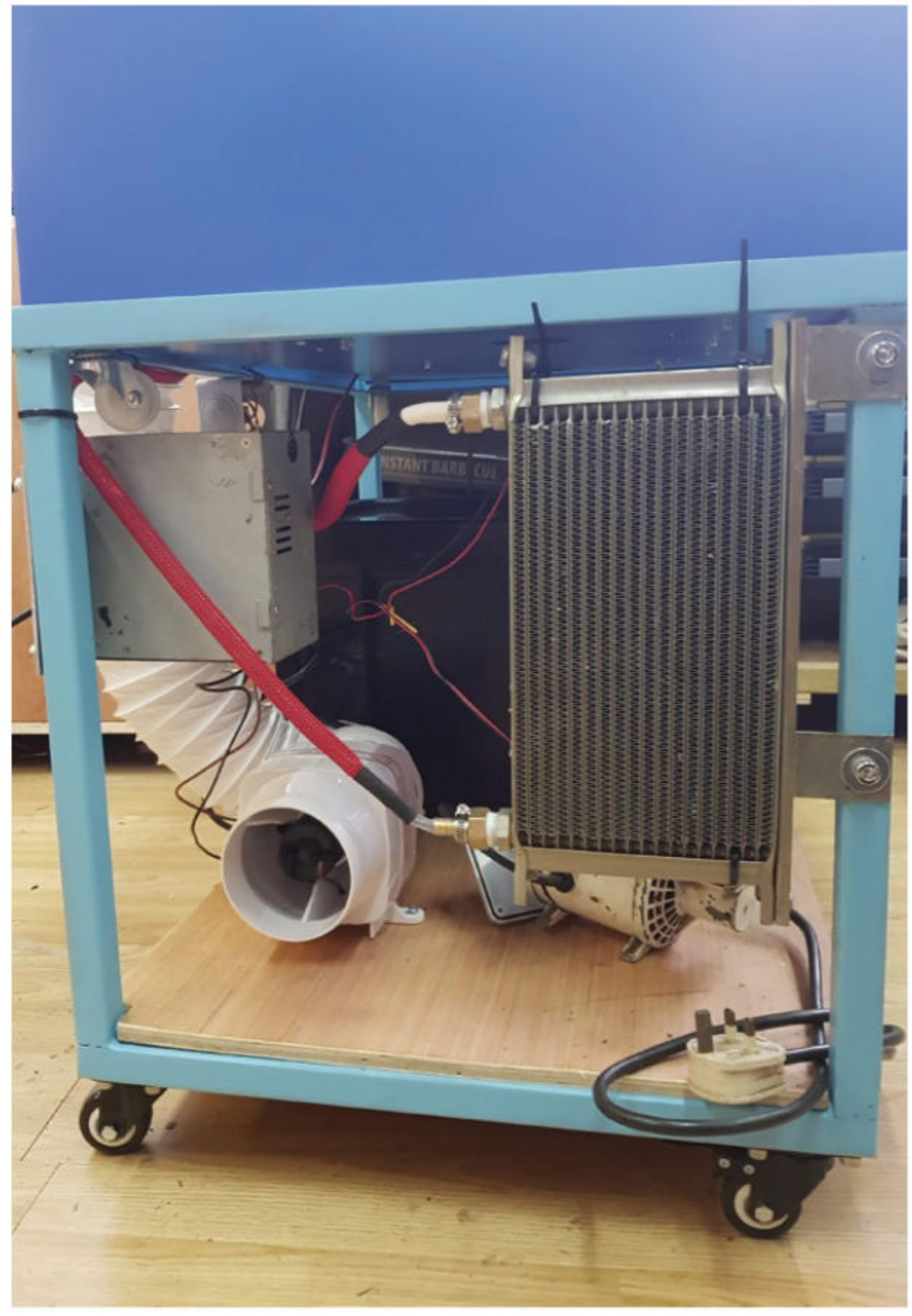
welding generators. Some MIG/TIG torches even have liquid cooling running through them to allow for comfortable use over extended periods. What this means is that choosing an off-the-shelf cooling solution isn't as straightforward as you might think. Different types of machine have different cooling requirements, so which unit is the right one for you?

COOLING IS SO HOT RIGHT NOW

An active cooler like the S&A 5200TH (distributed in the UK by **Megacold.uk.com** as the CW-5200TH) will be the best choice for most users. These types of cooler have a decent-sized tank and compressor,

> Industrial chillers are discrete units that pack **everything cooling related into one box**

a temperature controller, coolant alarm, and a flow sensor with an output that can be connected to the laser's safety interlocks. They are a refrigerated unit, relying on the expansion of refrigerant gas to lower the temperature of the coolant. They work in the same way as the air conditioning unit in a car or a domestic fridge, and can cool laser tubes up to around 150 watts. While it is possible to get similar functionality by hacking the coils of a domestic fridge or air conditioning unit into a reservoir of coolant, it's quite difficult to manipulate pipes with refrigerant gas without accidentally causing a leak and losing that gas to the atmosphere. If you're absolutely determined to go down the DIY route, you might consider using a small chest-style refrigerator as your starting point, filling the interior with a coolant tank, moving the thermocouple and cooling element into the coolant tank, and adding an external pump and flow sensor. However, since the cost of a small chest-style refrigerator is similar to the cost of a ready-made industrial chiller, this route only makes sense if you have access to second-hand parts at a reasonable price.



Above
You can use a computer water-cooling radiator and fan as a passive cooling solution, but it's a relatively expensive piece of kit. An old heater matrix from a car and some 120mm fans might offer a more wallet-compatible solution if you're on a tight budget

TUTORIAL

Using a Cricut EasyPress 2 with heat-transfer vinyl

Heat-transferable vinyl gives a project a very professional finish... let's press on!

Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor. A fan of maker gadgets, she now officially needs a 'crafting wing' added to the house!

Regular readers may remember that back in issue 50 of HackSpace Magazine, we dipped our toes into the colourful waters of vinyl cutting using a digital cutting machine. In this tutorial, we are going to investigate another fun option for using vinyl: heat-pressing it onto a base, such as fabric. We'll be using the Cricut EasyPress 2, which has a plate size of 22.5 cm by 22.5 cm – a useful size for heat-pressing vinyl, and indeed infusible ink (a whole other new area to get excited about!), onto a variety of bases.

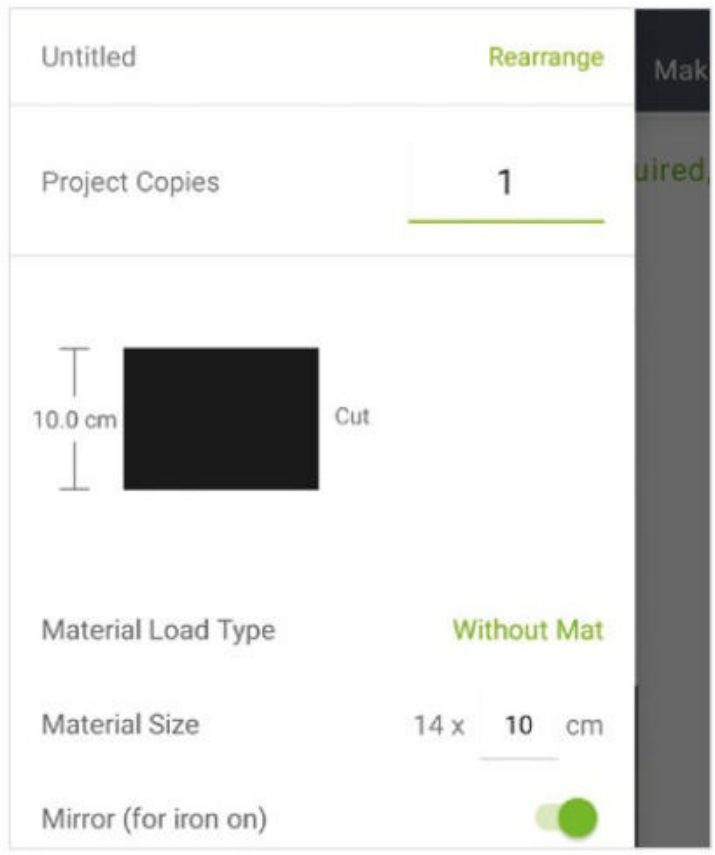


Figure 1
You need to 'mirror' your design because you are putting the vinyl face down through the cutting machine, so that the machine is cutting the back of the material with your design

The effects you can get from heat-pressing vinyl look extremely professional – we've said it before, but if you are looking for a little side-hustle, you only have to look at Etsy et al to see just how many people are using their digital cutting machines and presses to set up small enterprises making and selling vinyl-embellished products. So, in this step-by-step guide, let's look at how we can affix a beautiful vinyl design to a very basic cotton zipped pouch.

CAN'T I JUST USE A PLAIN OLD IRON?

Let's address that elephant in the room: what's wrong with a humble household iron to stick vinyl? It's got a hot baseplate, right? Well, you are, of course, more than welcome to try using a household iron to press your vinyl onto your bases but, generally, an everyday iron will not be as effective as a tool specifically designed for the task in hand. The reason for this is that a basic household iron doesn't have the same high-intensity heat over the whole area of the iron, which is what you need to ensure that your vinyl adheres properly so that it won't budge, especially if you are going to wash it.

Typically, an iron only has a very small area in the centre of its plate where the heat is highest, and our vinyl needs an even, high heat across its entire area. The whole plate of this EasyPress (and other similarly designed tools) heats up to the same temperature needed for your project – a temperature that you set and control, so you can be confident that the adherence of the vinyl will be completely even across your base. Our press is a good size, quick to heat to temperature, and this author has been very impressed with the results.

Left
Compact, lightweight, uncomplicated, and a speedy way of permanently attaching vinyl

Below
There's no end to what you can heat-transfer vinyl onto! This sentiment is ready to be pressed

QUICK TIP

Our tool came with a 'Quick Start Guide' which shares lots of hints and tips – do have a good read of any documentation that your press comes with.

Anecdotally, we've heard tales from people who have tried an iron for their projects, then tried a heated pressing tool and found the former to be far less effective overall. You probably also don't want to risk getting a sticky, gloopy mess stuck to the bottom of your iron, thus taking that important household item out of action completely! That question covered, let's now get creative…

PRESSING MATTERS

STEP 1 – DEVISE YOUR DESIGN

In issue 50, we looked at how to create a design in Cricut Design Space, so we won't go into too much detail here regarding that first step. However, one important point to note is that before you cut your design, please do make sure that you 'mirror' the design in Cricut Design Space (see **Figure 1**). This ensures that your design will be the right way round once it is eventually cut and stuck onto your project base.

GIVE IT A SECOND LIFE

We have used some blank fabric crafting bases that we had lying around here, but don't feel that you have to purchase such items in order to get going with vinyl heat transfers. We are big proponents of upcycling here at HackSpace, and we're sure that you must have a plain T-shirt, a plain zipper pouch, a tablecloth, a tea-towel, or a boring old cotton bag that needs rejuvenating. It makes sense ethically to just revitalise something you already own – it's just like getting something new, but without a large financial outlay! You just need to make sure you know exactly what you are heat-pressing onto, and the Cricut Heat Guide, (see Step 2) will tell you exactly how long, and at what temperature, you need to press your base for.

And don't limit yourself to just cotton or polyester fabrics – you can heat-press vinyl onto wood (customised chopping board anyone?!), leather, felt, wool, silk, chipboard, corkboard, cardstock… it's a long list! Here are some more ideas to ponder: **hsmag.cc/VinylIdeas** and **hsmag.cc/IronOnProjects**.

YOU'LL NEED

- **Cricut EasyPress 2 (23 cm by 23 cm)**
  hsmag.cc/EasyPress2
- **A heat-resistant mat, e.g. the Cricut EasyPress Heat Resistant Mat**
  hsmag.cc/HeatMat
- **An item to press your design onto, e.g. a blank T-shirt, pouch, drawstring bag, etc.**
- **A digital cutting machine, e.g. Cricut Joy**
- **Heat-transfer vinyl, e.g. Cricut Smart Iron-On**
- **A weeding tool to remove excess vinyl**
- **Scissors**

Using a Cricut EasyPress 2 with heat-transfer vinyl

TUTORIAL

Right
Heat-transferable vinyl comes in plenty of colours – here, we've used just two colours, but don't hold back, be bold!

QUICK TIP

Don't use your EasyPress 2 tool on an ironing board. You need a solid, flat surface, such as a solid kitchen table, to give you a safe and stable base on which to work.

Ensure that, if using Smart Iron-On, you insert the vinyl shiny side down into your machine. Once cut, 'weed' your design at approximately a 45-degree angle, i.e. remove the excess vinyl around your design, so that you are simply left with your design on its liner backing. You'll find that Smart Iron-On is quite a stretchy vinyl and easy to work with.

STEP 2 – CHECK SETTINGS

A hugely useful web page is the Cricut Heat Guide (**cricut.com/en_us/heatguide**). This is an invaluable tool when it comes to knowing at what temperature you need to press and for how long. Simply select the transfer material you are using (e.g. Smart Iron-On), and then the base material, which in our case is a cotton canvas zipped pouch. You'll then be furnished with the exact temperature and length of time you need to press. It will also helpfully tell you whether the liner needs to cool before you remove it, or if you should remove when still warm – this all depends on your transfer material and base.

Right
Ready for pressing, we're nearly done!

QUICK TIP

Check the peel-off time for the specific vinyl you are using (they all differ) – can it be peeled off while still hot or cold?

STEP 3 – PREP YOUR PRESS

Please do make sure that you are working on a flat and safe surface, such as a kitchen table, and plug your device in, power the press on, and then set your temperature and timing (per the Heat Guide mentioned above). When your EasyPress 2 is ready for action, the 'C' button will turn green. If you are using a mat, gently move the press over it for a few seconds to warm it up.

> **Make sure that you are working on a flat and safe** surface, such as a kitchen table

STEP 4 – STICK IT!

Next, take the item you are pressing onto and glide the press over it for a few seconds to remove any wrinkles and moisture. Then take your design and position it as required. When ready, take your press, start the timer (by pressing the 'C' button), and firmly press your design onto the base for the required time.

WARNING

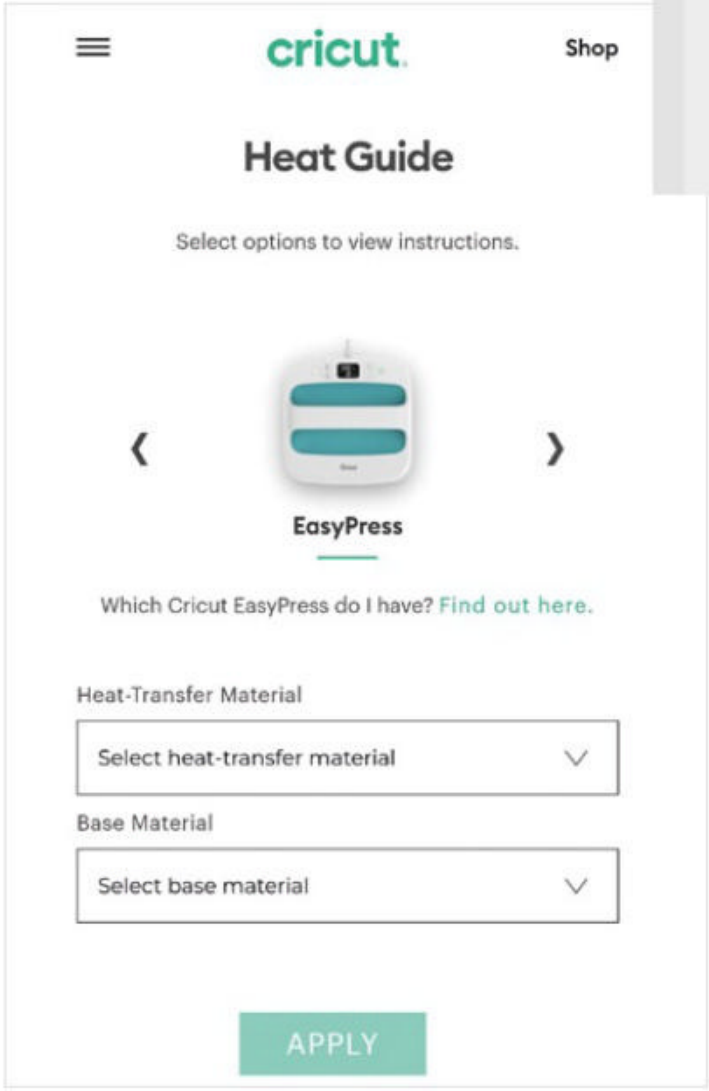
FORGE

These tools obviously operate at very high temperatures, and you can easily burn yourself if you are careless. Always exercise caution when using such a device, and please keep away from children.

GOING OFF-BRAND

Clearly, in this tutorial, we have been using a branded pressing tool from the Cricut stable of makers' tools. This is not your only option and a quick search on Amazon, for example, will throw up a wide variety of other possibilities, some of which are ultimately cheaper versions of the tool we have used here. Here's an example – **hsmag.cc/HeatPressAlternatives**. It is, of course, up to you what you purchase, and we can't judge tools that we've not used. Read the reviews and make up your own mind. There are also smaller-sized heat pressing devices available – e.g. **hsmag.cc/MiniPress**. However, this author was more than happy with her chosen gadget, and now has a myriad of ideas brewing!

A word to the wise: if you are intending to purchase such a gadget to add to your wardrobe of maker machines, this author absolutely recommends looking for a great deal – our branded press was purchased as a Black Friday bargain for under £100. Prices seem to be tumbling down on these type of gadgets all the time, so bide your time, have a good look around, and you might be surprised at the deals you can bag.



QUICK TIP

If you don't want to buy a branded, heat-resistant mat, use a folded bath towel instead… a much cheaper alternative.

Left
All the important information at your fingertips – you simply can't go wrong!

Below
An easy-to-set control panel, with a power button on the far left, temperature and timing settings in the middle, and indicator button on the right

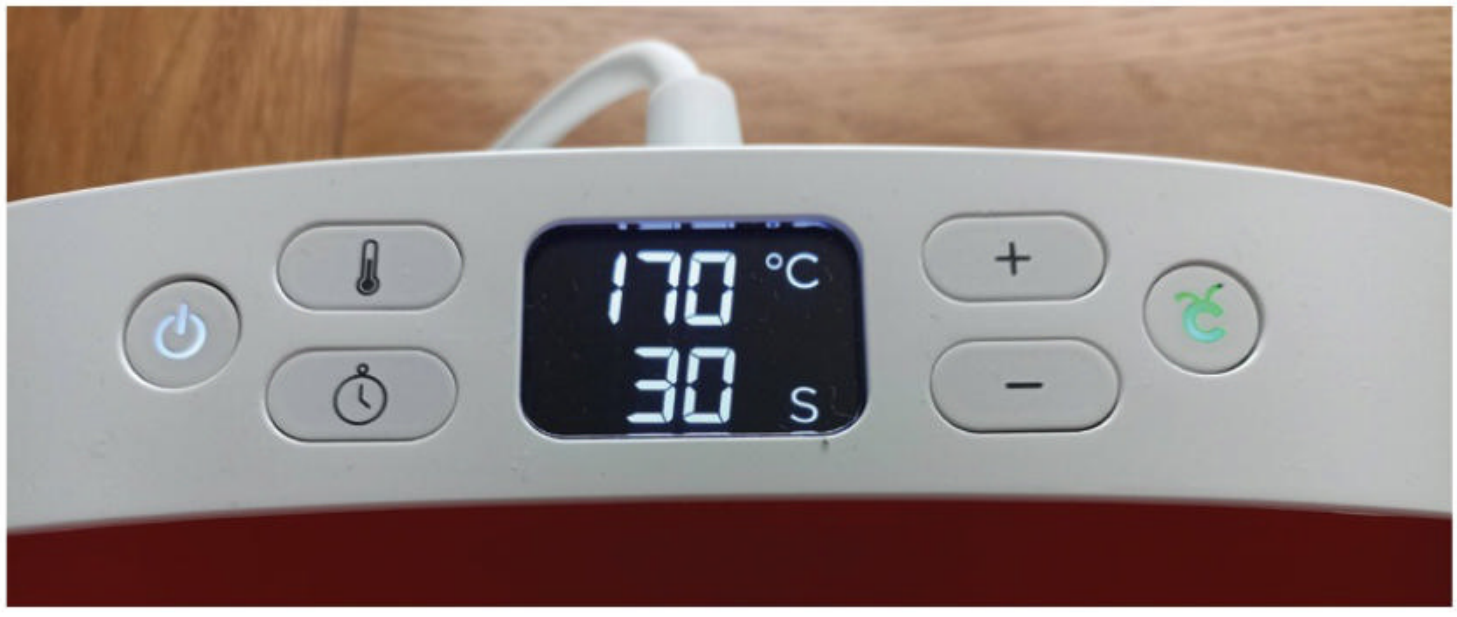
Once done, flip the item over and press on the reverse side for 15 seconds, just to add extra adherence.

STEP 5 – THE BIG REVEAL!

In our case, we needed the design to cool before removing the liner, but be aware that, depending on what your are pressing onto, you may have a cool or warm peel in terms of the Heat Guide's instructions. When ready, gently pull the liner off, but do this slowly in case the design hasn't stuck properly, in which case you'll need to press it for a little longer. Test an area, and if it's adhered, fully remove. And there you have it: one nicely embellished little pouch ready for action.

HOT OFF THE PRESS

Digital cutting is a hugely popular field of crafting, and is ever-growing. This author is hoping to get a larger cutting machine soon, as the possibilities are just vast – and, take it from someone who has tried it, it is huge fun! Using a digital cutting machine in tandem with a heat pressing tool gives you even more scope in the crafting and designing fields. It's also very accessible from a price perspective, and really lets you personalise items, enabling you to add your own unique touch. If you're a collector of maker gadgets, a pressing tool will be a positive addition to your crafting cache.



QUICK TIP

Here are some official Cricut tips and tricks: **hsmag.cc/IronOnTips**.

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 112

SIMPLY SERVOS

Kitronik's board for adding servos to Raspberry Pi Pico

PG 104

BEST OF BREED

Our pick of the finest Tindie products

A roundup of our favourite products found on Tindie

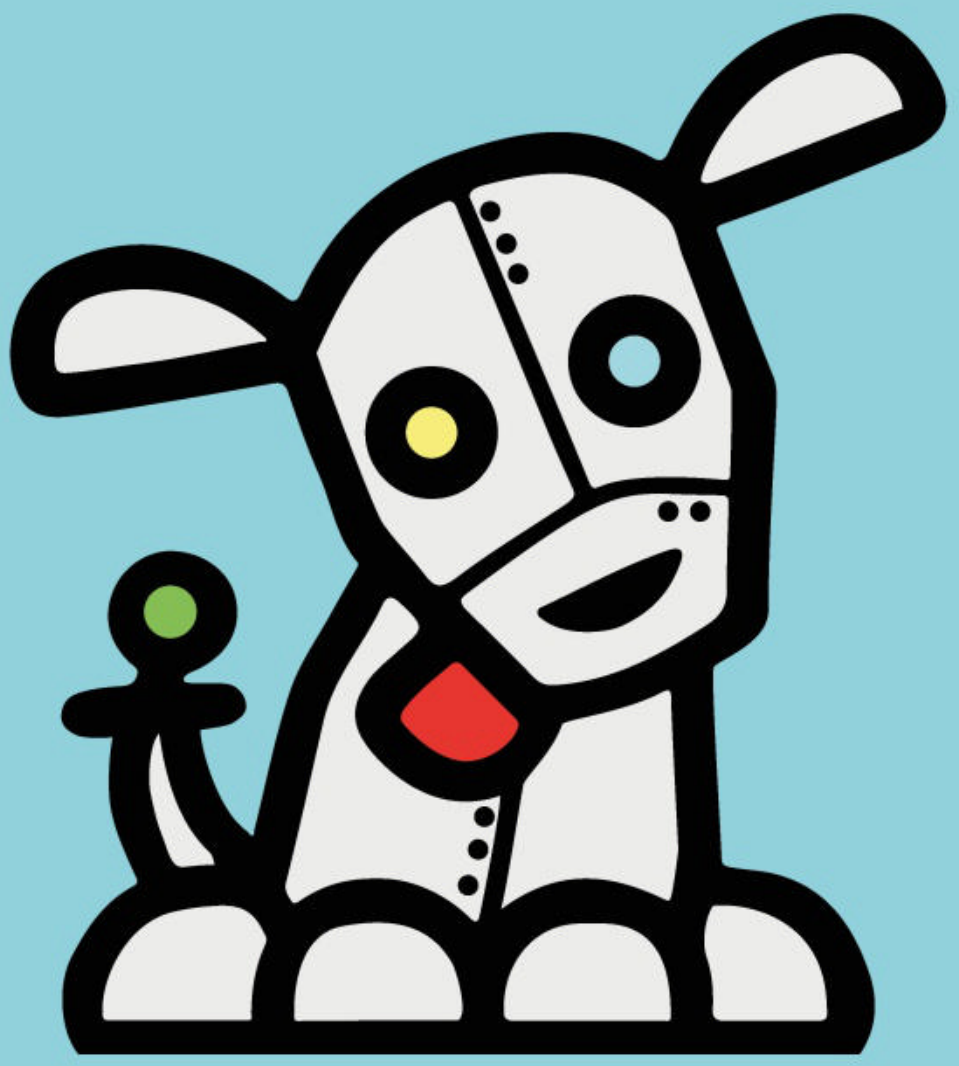
—

ONLY THE BEST

Tindie Treats

A few of our favourite projects found on Tindie, the marketplace for electronics

By **Marc de Vinck** @devinck



Since the last time I wrote about a few of my favourite Tindie products, the company has seen an additional 100,000 orders shipped and 3000+ more products added, shipping to more than 160 countries around the world. If you are unfamiliar with Tindie, well then, I have a treat for you! Tindie is an online marketplace that features mostly the small DIY technology creators. It's a great place to find esoteric kits and components for making unique projects.

I have ordered plenty of kits from Tindie and, so far, so good. In fact, I bought another kit while searching for products for this Best of Breed! The website is organised by topics like 3D printing, IoT, robots, sound, camera, and of course DIY technology, which I thoroughly enjoy browsing on a regular basis. And many of those topics include several subtopics, making browsing the Tindie website easy and fun. And a bit dangerous!

RP2040 Stamp vs CircuitMess Chatter

SOLDER PARTY $12 | hsmag.cc/stamp

CIRCUITMESS $14.99 | hsmag.cc/diywireless

The RP2040 Stamp, designed by Solder Party in Sweden, is a hand-solderable Raspberry Pi RP2040 IC module with 8MB of flash and LiPo battery charging management, featuring a status LED, a reset button, and a multi-colour NeoPixel LED. By building this module with 2mm pitch castellated edges, Solder Party has made it easy for you to add an RP2040 to your own PCB without worrying about hand-soldering the extremely small-pitch QFN integrated circuit.

The creators also provide well-documented source files including code, drawings, pinouts, and more. If you want to start prototyping with an RP2040, this is an affordable, time-saving little board. It even comes pre-flashed with CircuitPython 7.1.0-beta!

Left
The easy way to build an RP2040 board

VERDICT

RP2040 Stamp

Great, but you'll have to solder!

10/10

CircuitMess Chatter

Useful long-term application.

9/10

Chatter, designed by CircuitMess in Croatia, is a DIY electronics kit for building your own two-way text messaging communicator. The kit comes with a pair of radios because, as the creator states, "texting yourself is not as fun as you think"! They operate on the radio spectrum commonly used by LoRa devices, which is licence-free and operates in the sub-gigahertz range. It's a bit slow for modern smartphones, but it's great for small chunks of data like a simple message, and it works reliably over long distances.

You can send texts, emojis, memes, and GIFs. When building the kit, you will learn about wireless security, LoRa radios, encryption and decryption, basic code, and how to build simple apps for your communicators. This is a great way to get someone excited about DIY electronics!

Below
An easy introduction to LoRa

A roundup of our favourite products found on Tindie

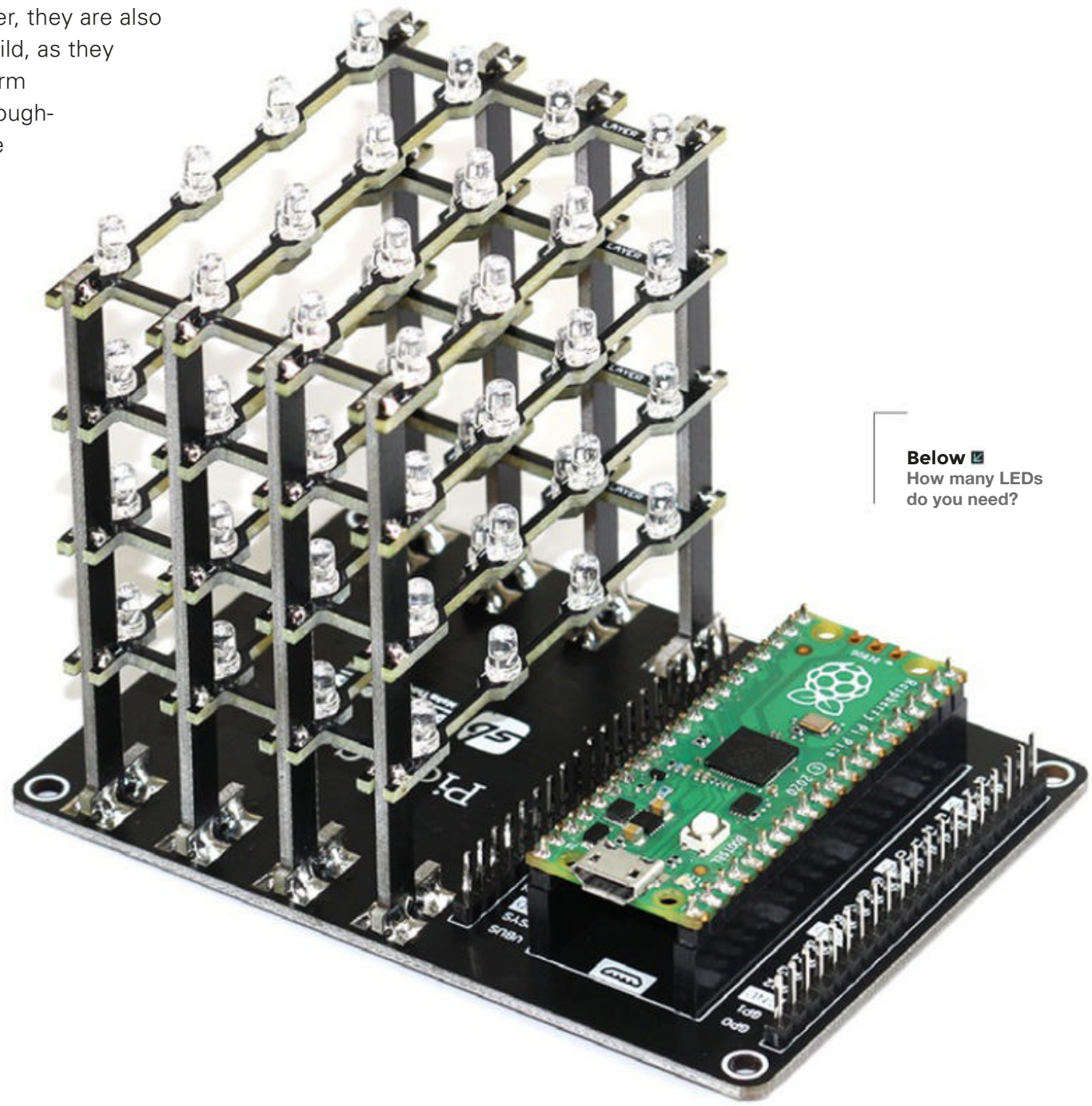
BEST OF BREED

Pico Cube

SB COMPONENTS $37.99 | hsmag.cc/picocube

3D LED cubes have been around for a long time. They are a relatively popular kit for DIYers. However, they are also typically difficult to build, as they usually rely on free-form soldering of many through-hole LEDs to form the cube. I have built a few 3×3×3 cubes, and although they came out well, they weren't exactly fun to put together. I never attempted to build a 4×4×4.

The Pico Cube, from SB Components in the UK, is a 4×4×4 cube that runs via a Raspberry Pi Pico and aims to make building a 3D LED cube actually enjoyable. They took out a lot of the variability of the building process by integrating PCBs instead of relying on the LEDs' wires for the structure. And for those of you that don't want to deal with soldering (gasp!), they even offer a fully assembled version!



Below
How many LEDs do you need?

VERDICT

Pico Cube

An easier way to build a cube.

8/10

1.28 inch Round LCD HAT

SB COMPONENTS $36.99 | hsmag.cc/roundlcd

The Round LCD HAT for Raspberry Pi, by SB Components, is a 1.28-inch RGB display HAT with 240×240 resolution and a joystick. It uses the standard 40-pin Pico GPIO interface, so it's simple plug and play. OK, plug and code, then play!

The HAT incorporates a display driver and SPI interface, which reduces the required available pins, allowing you to add more components for your project. It also features a four-way joystick with central button. It's a bit of a niche product, which is exactly what I like about the products and creators on Tindie. Something for everyone!

Above Circles are the new squares

VERDICT

1.28 inch Round LCD HAT

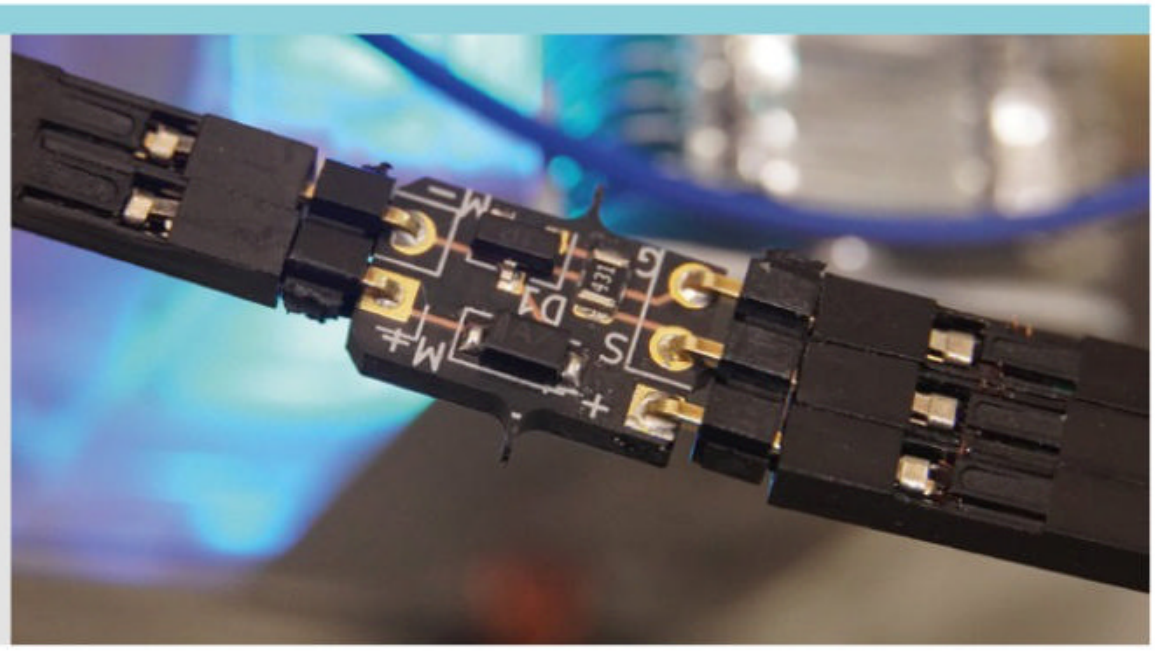
If you want a round display, this might be a good fit.

8/10

EZ FAN2

JC DEVICES $8.95 | tindie.com

The Raspberry Pi is a great computing platform. And as the technology advances, and the processor gets faster, managing the heat it produces becomes more important, especially in a small enclosure. With the EZ Fan2 by JC Devices, you can easily add an external fan (that does not have PWM inputs) to help keep your Raspberry Pi cool. It's a basic transistor-driven motor controller, not a full bidirectional driver. It can be controlled via the Raspberry Pi's GPIO fan control option, found in the latest Raspberry Pi OS.

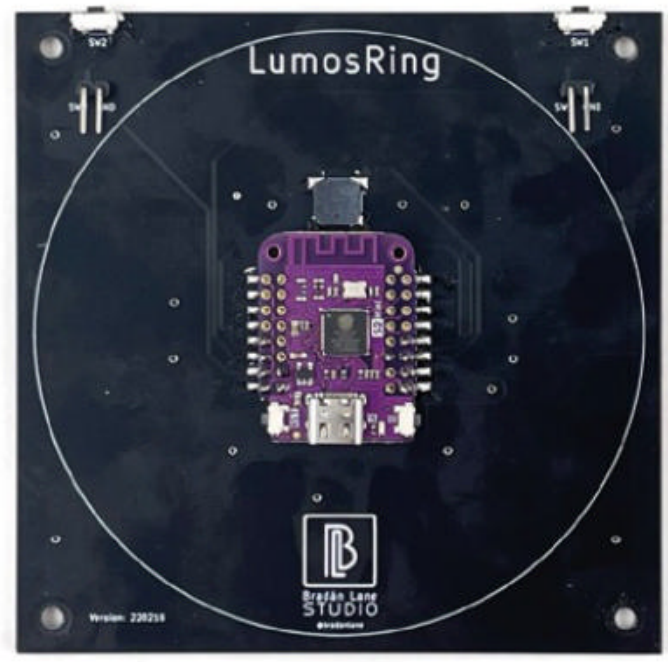


A roundup of our favourite products found on Tindie

BEST OF BREED

LumosRing

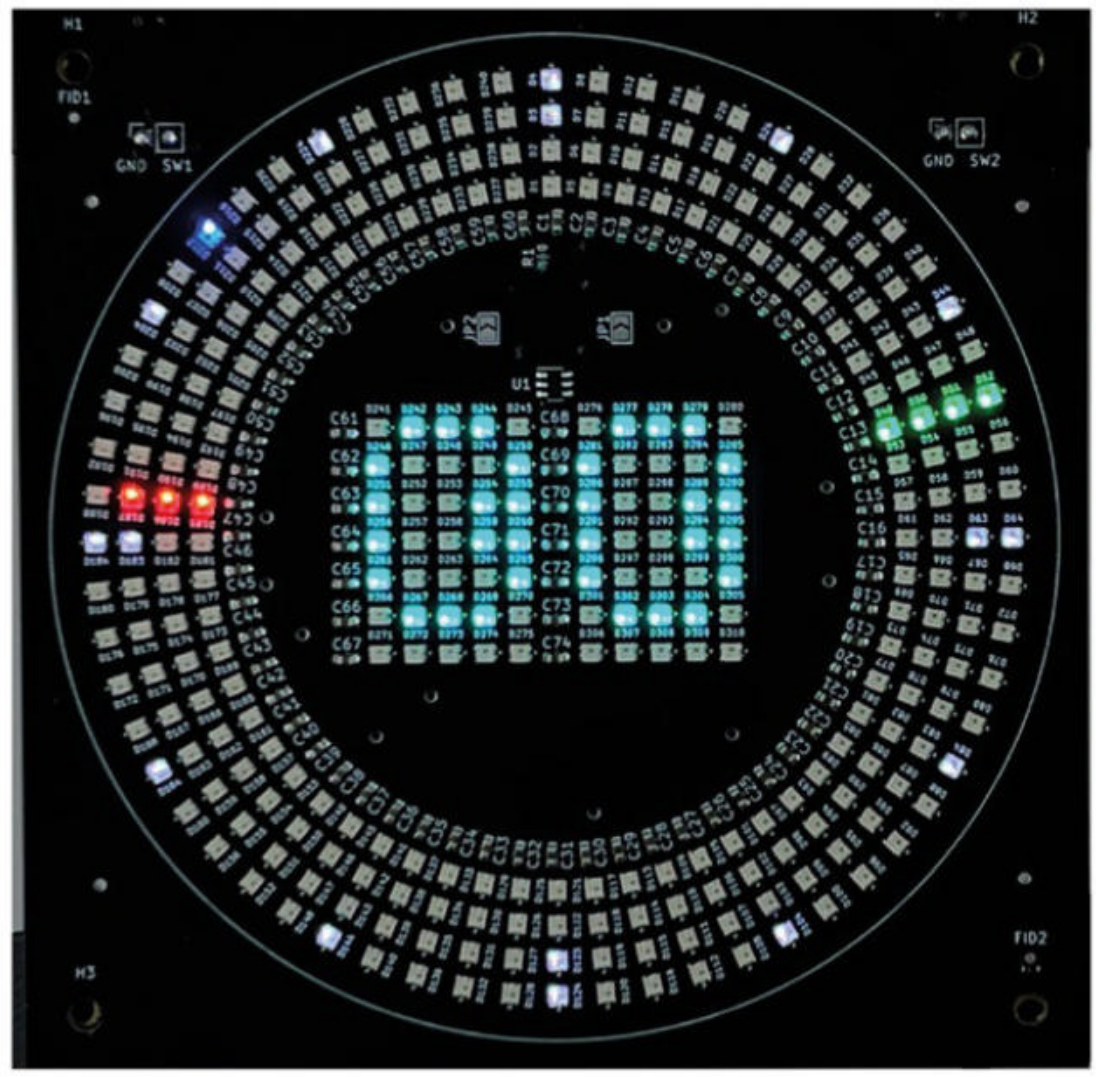
BRADÁN LANE STUDIO $65 | hsmag.cc/lr



Below
Squares or circles? Why not both?

The more LEDs the better! And the LumosRing, designed by Bradán Lane STUDIO, certainly lives up to that expectation. Featuring over 250 RGB LEDs, this beautifully designed black PCB makes for a great clock, visualizer, or you can even use it to play games. And fortunately, you don't have to solder all those LEDs, as the LumosRing comes fully assembled.

The 240 LEDs on the outer rings lend themselves to making a perfect analogue clock. Coupled with the 70 LEDs configured in two 5×7 blocks in the centre, you can add extra functionality like a temperature display, or even a scrolling message or stylised icons. The board also features an included ESP23-S2 running CircuitPython. You can add arcade buttons for extra functionality, and a diffused faceplate to smooth out all those LEDs. Check out the product page (**hsmag.cc/LumosRing**) for a video demo of all those beautiful addressable RGB LEDs!



VERDICT

LumosRing

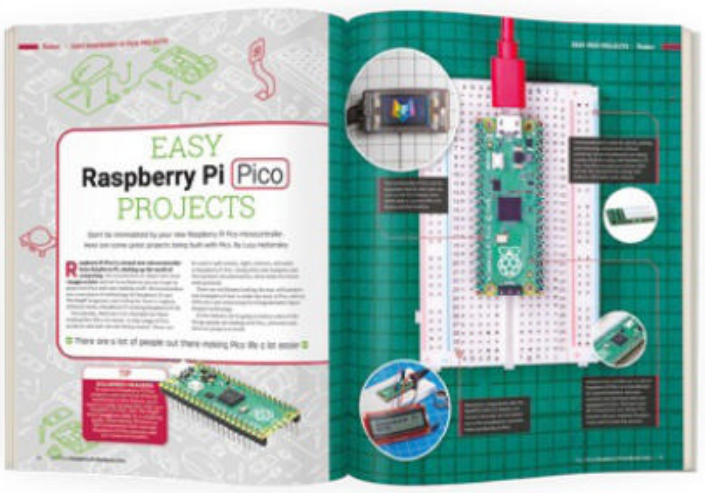
Hundreds of LEDs, on a beautiful PCB.

9/10

THE *Official* RASPBERRY PI HANDBOOK 2022

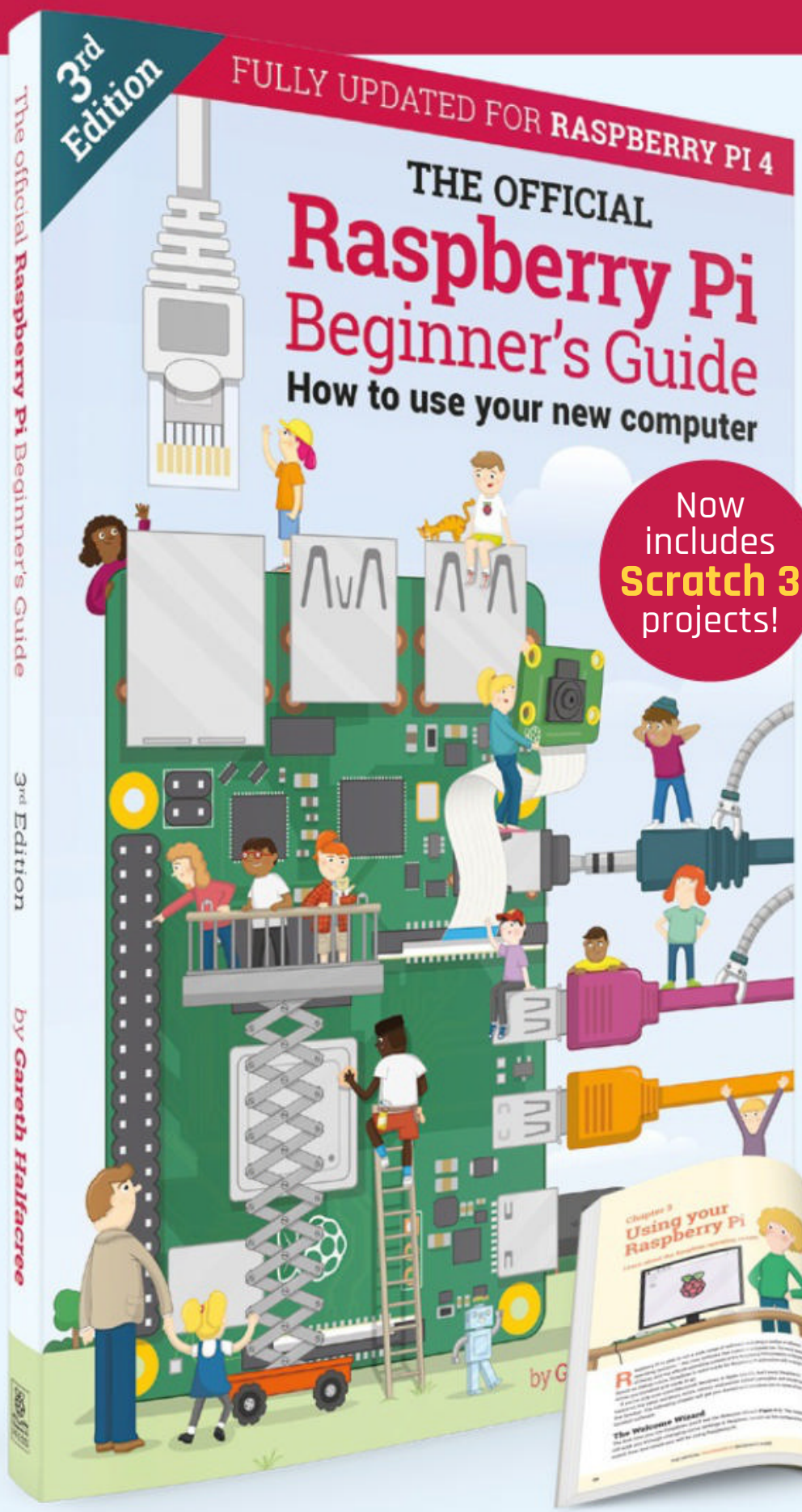
200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects



Buy online: magpi.cc/store

THE OFFICIAL
Raspberry Pi
Beginner's Guide



The only guide you need to get started with Raspberry Pi

Inside:

- Learn how to set up your Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch 3 and Python languages
- Create amazing projects by connecting electronic components to Raspberry Pi's GPIO pins

Plus much, much more!

£10 with FREE worldwide delivery

Buy online: magpi.cc/BGbook

BOOK OF MAKING

VOLUME 2

ONLY £10

WHSmith

BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

THE BEST PROJECTS FROM

HACKSPACE MAGAZINE

THE ULTIMATE SKILLS, TRICKS, AND MAKES

AVAILABLE NOW

hsmag.cc/store

FROM THE MAKERS OF **HackSpace** MAGAZINE

REVIEW

Kitronik Simply Servos Board for Raspberry Pi Pico

Power your next Pico-based robotics project

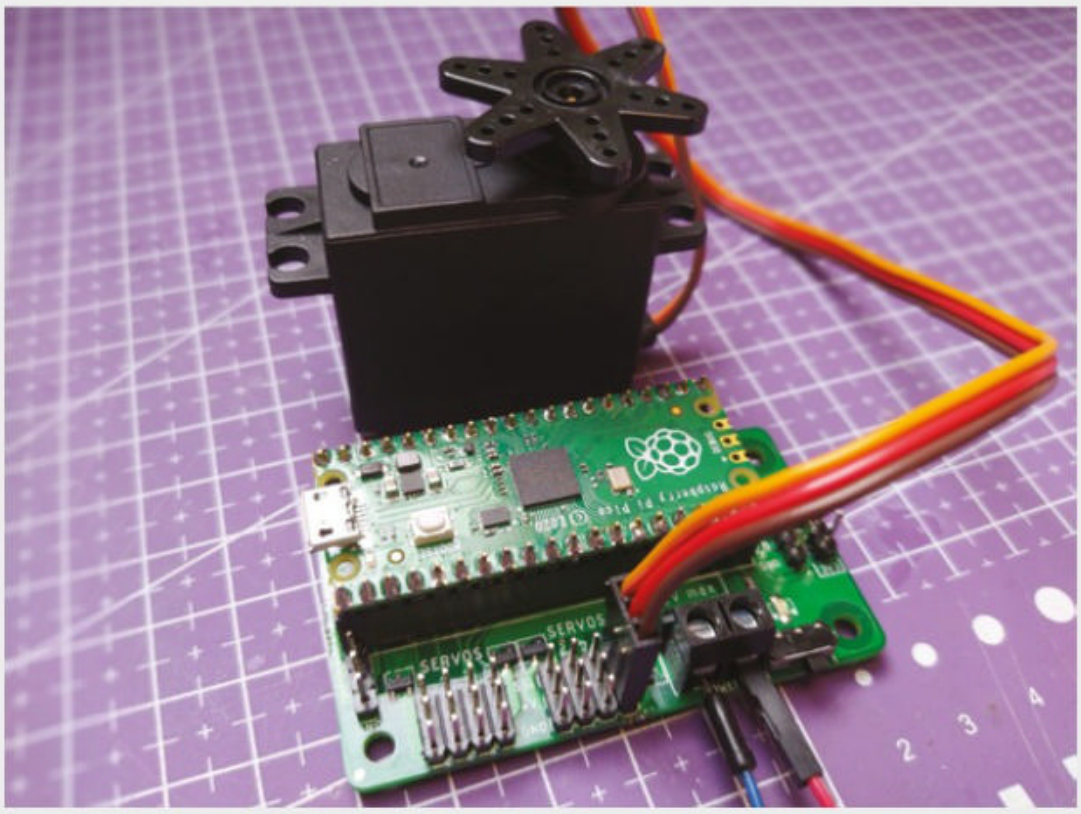
KITRONIK £6.95 + VAT | kitronik.co.uk/5339

By Jo Hinchliffe @concreted0g

Driving a servo is a cornerstone of microcontroller applications and hobby robotics, and often is more complex than it needs to be. We've certainly seen and reviewed plenty of boards that have a couple of sets of pins for servos amongst a myriad other motor driver and sensor connectors.

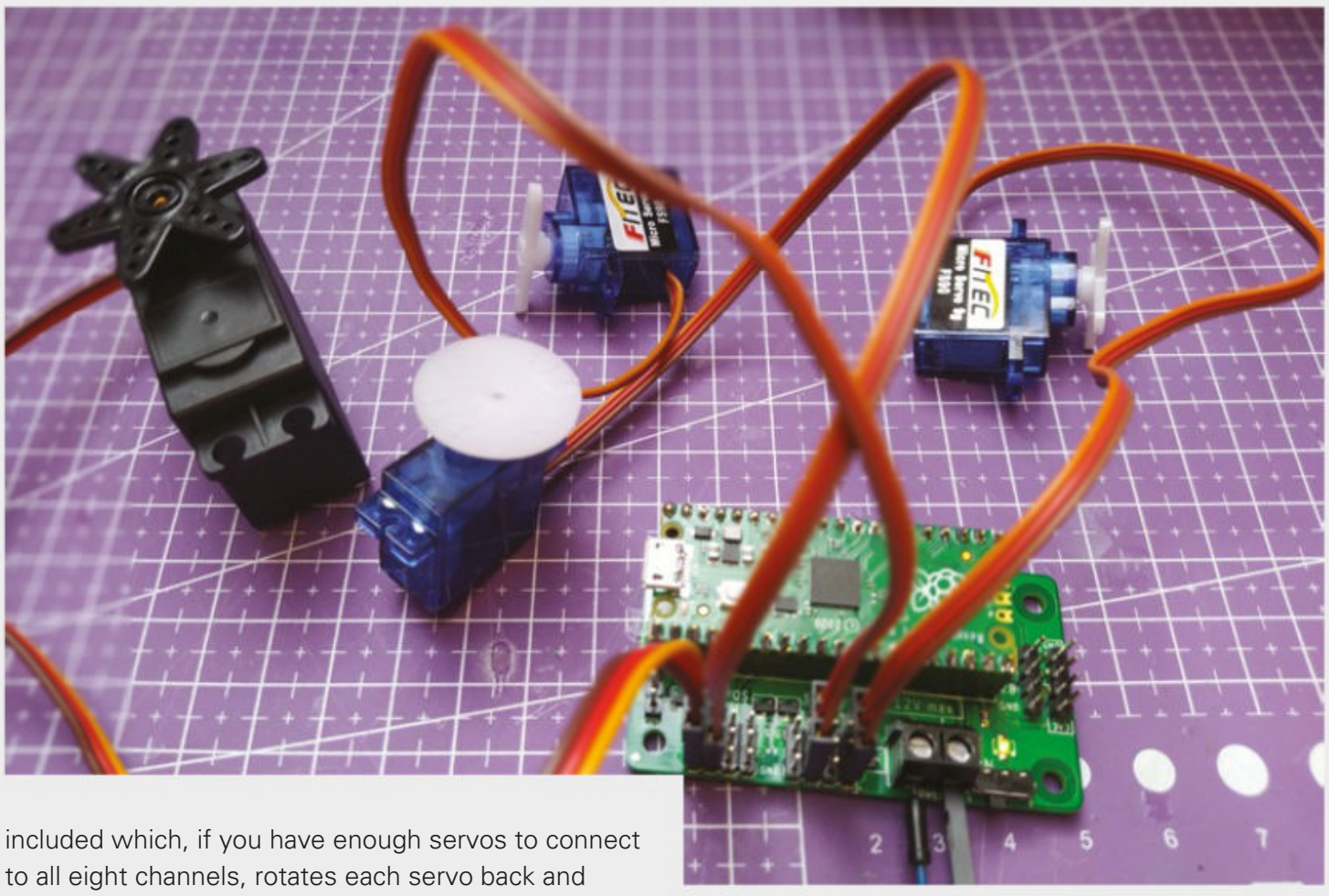
The new Kitronik Simply Servos Board has a clue in the name – it's simply about servo driving and, whilst it could do other things, servo applications are its go-to task. It's a compact 64.5 mm by 36 mm PCB board, with the familiar sight of a 40-pin header to receive a (not supplied) Raspberry Pi Pico. There are eight sets of 3-header pins that can receive standard servo connectors neatly arranged into two banks of four. On the same edge is a screw clamp terminal for power and ground, rated up to 12 V, which we will talk more about later. There's a handy on/off switch which is excellent, and negates the need for you to add one into your robot build, and there is a power indicator LED attached to the switch. Finally, there are a couple of other sets of headers that, in addition to the servos, break out the following pins: GP0, 1, 26, 27, 28, 3V3, and GND. This is a useful selection, as it breaks out three ADC pins – 26, 27, 28 – as well as the more general GP0 and 1.

Below
The Simply Servos Board happily runs larger, metal-geared servos



Power is connected into the robust screw clamp terminal, and this input is sent directly to the servo + and GND lines. This is sensible and, as such, you can power a good range of servos, from the tiny 9g micro servos through to the larger metal-geared mid-size servos. It's worth considering, though, what your servos will happily run at before connecting a full 12 V supply. Once powered, the Simply Servos Board regulates to provide a 3.3 V supply to a connected Pico. This is excellent, as it means that you don't need to think about a separate power system, and the whole board can be powered down from the switch. A Pico, once inserted, is oriented so the USB port is accessible and, in testing, the board works perfectly with the Pico connected over USB, whilst the Simply Servos Board and servos are powered on.

In the nicely illustrated datasheet, there is a link to a supplied GitHub repository where you can download a MicroPython module to get you up and running with the board with ease. There is a small example



included which, if you have enough servos to connect to all eight channels, rotates each servo back and forth. Once we copied the **simpleServos.py** module over to our Pico, the simple instructions on the GitHub repository 'read me' file gave us enough information to quickly write a small sketch to choreograph the movements of an individual servo.

> Once powered, the Simply Servos Board regulates to **provide a 3.3 V supply to a connected Pico**

One head-scratching moment, which we have had before, was the infamous 'off by one in code' mistake – we connected a servo to the servo 1 channel, as labelled on the silkscreen, and couldn't get it to work, until the slow realisation that, actually, in the code, the channels are address as 0–7. That aside, we

[test_simplyServo.py] [SimplyDrive8Servos.py]

```
1
2  from simplyServos import KitronikSimplyServos
3  from time import sleep
4
5  servos = KitronikSimplyServos()
6
7
8  servos.goToPosition(1,90)
9  sleep(1)
10 servos.goToPosition(1,0)
11 sleep(2)
12 servos.goToPosition(1,180)
13 sleep(2)
14 servos.goToPosition(1,90)
15 sleep(2)
16
17
18
```

found the documentation, the simple code examples, and the repository easy to get up and running, using Thonny as our Pico MicroPython editor of choice.

While this can run a variety of servos, you could also drive continuous rotation servos – very handy for creating simple tank-style drive systems for robot platforms. Most continuous rotation servos require a little tuning and trimming but we found, once dialled in, that you can send 90° values to stop a servo rotating, and 0° and 180° values for forwards and reverse. Aside from simple angle control of servos, there are also examples in the 'read me' of using pulse width as a means of control.

All in all, this is a quick and easy board to get up and running with and, when you have eight servos dancing and whirring on your desk, it's only a matter of time before you come up with some amazing articulated project that can do your bidding!

Above
The Kitronik Simply Servos board is compact and straightforward, and would be easy to integrate into projects with its mount holes and power switch

Left
Using the hints and tips on the Kitronik GitHub repository, it's easy to whip up a quick test script in Thonny

Below
Here we see the Simply Servos board with the Raspberry Pi Pico neatly inserted into the header sockets

VERDICT

A simple and straightforward board for building Pico projects requiring servos.

9/10

next in

issue
#56

ON SALE
23 JUNE

Create A PRODUCT

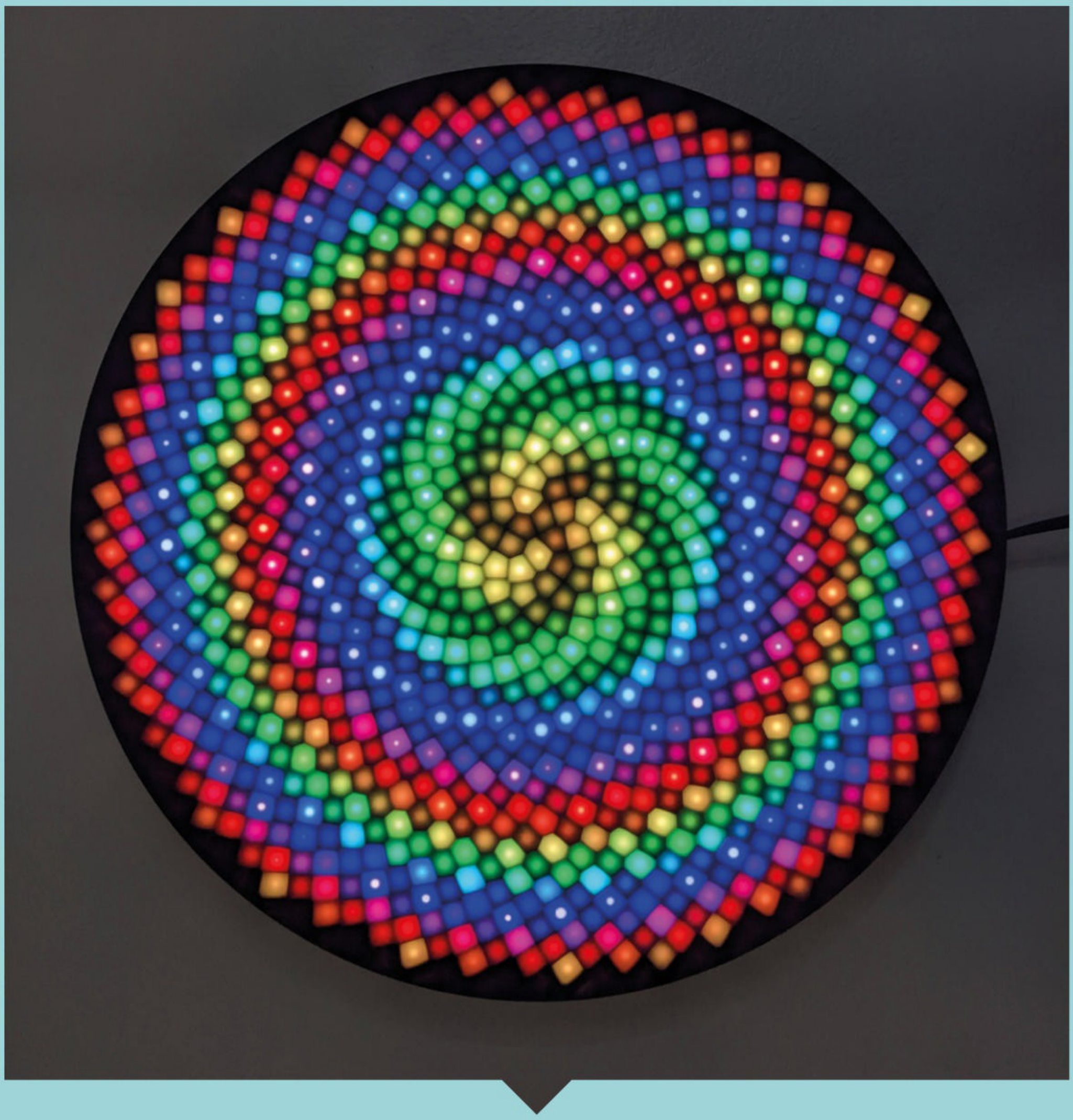
Also

- → **MUSIC**
- → **ARDUINO**
- → **PCBS**
- → **3D PRINTING**
- → **AND MUCH MORE**

—

DON'T MISS OUT

hsmag.cc/subscribe



The Fibonacci number series pops up everywhere in nature, from pine cones to nautilus shells to the arrangement of seeds in a sunflower. Jason Coon has applied this wonder of natural geometry to LEDs, with spectacular results – here's his Fibonacci512, a spectacular work of art that displays procedurally generated sequences that follow the principles of the Fibonacci series.

Start Your Concept.

Start With Digi-Key.

FREE SHIPPING
ON ORDERS OVER £33 OR $50 USD*

More Products

More Inventory In-Stock

More New Technologies

More Name-Brand Suppliers

More Technical Resources

0800 587 0991

DIGIKEY.CO.UK

Digi-Key ELECTRONICS

*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of $18.00 USD will be billed on all orders of less than $50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries.

ECIA MEMBER
Supporting The Authorized Channel