50 best 3D prints

LORA
Adding long-range radio to a project

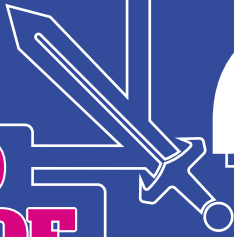**Essential builds for your home, office, and workshop**

ALSO INSIDE

TOUCH INPUT
Add capacitive touch to Pico

DIY GAMES CONTROLLER
Build your own input device

Oct. 2021
Issue #47 £6

9 772515 514006

47

Raspberry Pi PRESS

CARDBOARD PICO SMD PURE DATA

It's official! Name change announcement!
Same team, same company - just a new name!

PiShop.ca

Legendary **BuyaPi.ca**

is now **PiShop.ca**

The same great products and services
From the growing PiShop family!
Now serving all of North America

PiShop.ca • PiShop.us • PiShop.mx

BuyaPi.ca

Welcome to HackSpace magazine

Like many people who get a 3D printer, I first got very excited to test it out and print things. The usual novelty things – and some were useful, (I still have some custom-made knobs on my oven that I'm very pleased with). I made some Christmas decorations that year, and had great fun with flat-pack 'card'-style gifts. For a while, it was one of my go-to tools, whether I needed a jig to help me hold things in place while fixing them, or for creating new designs to solve a problem. However, recently I've not used my 3D printer much. That's partly because I got so exhausted trying to print face masks 24 hours a day, seven days a week during the early months of the pandemic, while also running a magazine and managing a toddler who could no longer go to nursery.

I often look at my 3D printer and think of its potential, but where should I (re)start? I'm sure I'm not the only person who has a desktop fabricator that sits idle most of the time. We've picked 50 great designs to spruce up your home office and workshop, so let's get back into it. Let's level those beds, heat up those hot ends, and squirt some bioplastic. It's time to get 3D printing.

BEN EVERARD

Editor @ ben.everard@raspberrypi.com

PAGE **46**
FREE PICO
WHEN YOU
SUBSCRIBE

Contents

Cover Feature

50 best 3D prints

Practical prints for better living

34

Tutorial

Surface-mount soldering
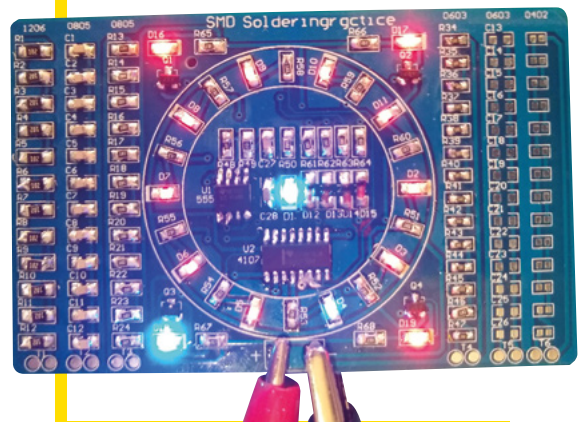
92 Level up your soldering skills with a crash course in surface-mount

80
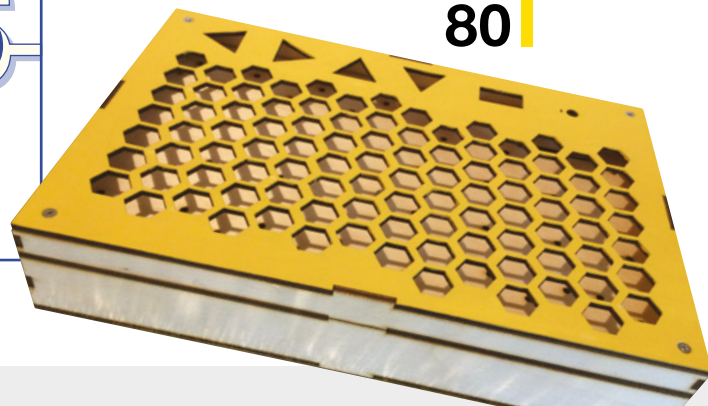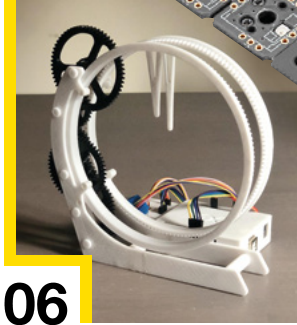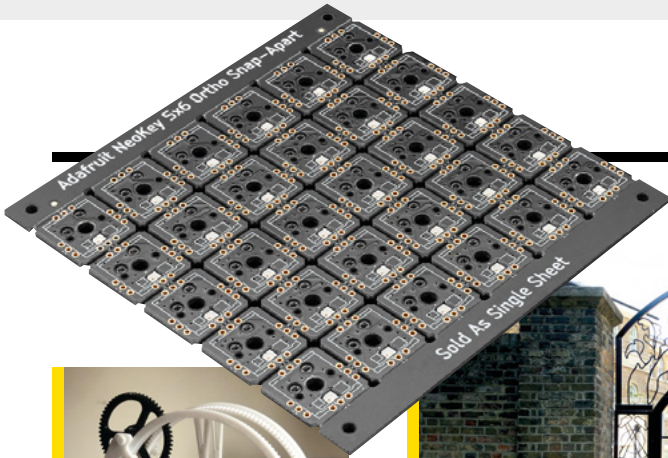
06

20

96

Direct from Shenzhen

4050 laser engraver
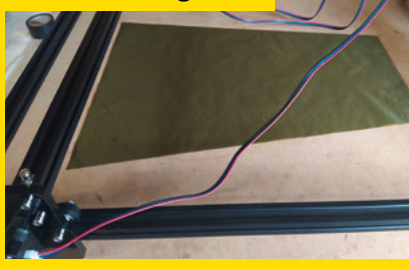
110 This laser engraver is fully armed and operational

Interview

Tommy Marshall

56 The unifying power of wibbly-wobbly sound-waves

112

RetroPie Cyberdeck

By **Willie McBoner**   **hsmag.cc/CyberDeck**

We've only just finished printing a series on building a games cabinet using the RetroPie games emulator on a Raspberry Pi… and now something comes along that makes our plywood, full-size arcade machine look old hat.

This mostly 3D-printed cyberdeck features a 5-inch 800 × 480 touchscreen display, as well as the usual ports available through the Raspberry Pi 3 Model B+ that powers it. Quite how useful the screen's portrait orientation will be for Sonic The Hedgehog is anyone's guess, but if you're playing any sort of top-down shooter, you're laughing. The maker describes this project as a "video game emulator with some edge" – we think it's pretty impressive for a project that began as an excuse to learn 3D design.

Right ↗
Just the ticket for a quick game of Xenon 2
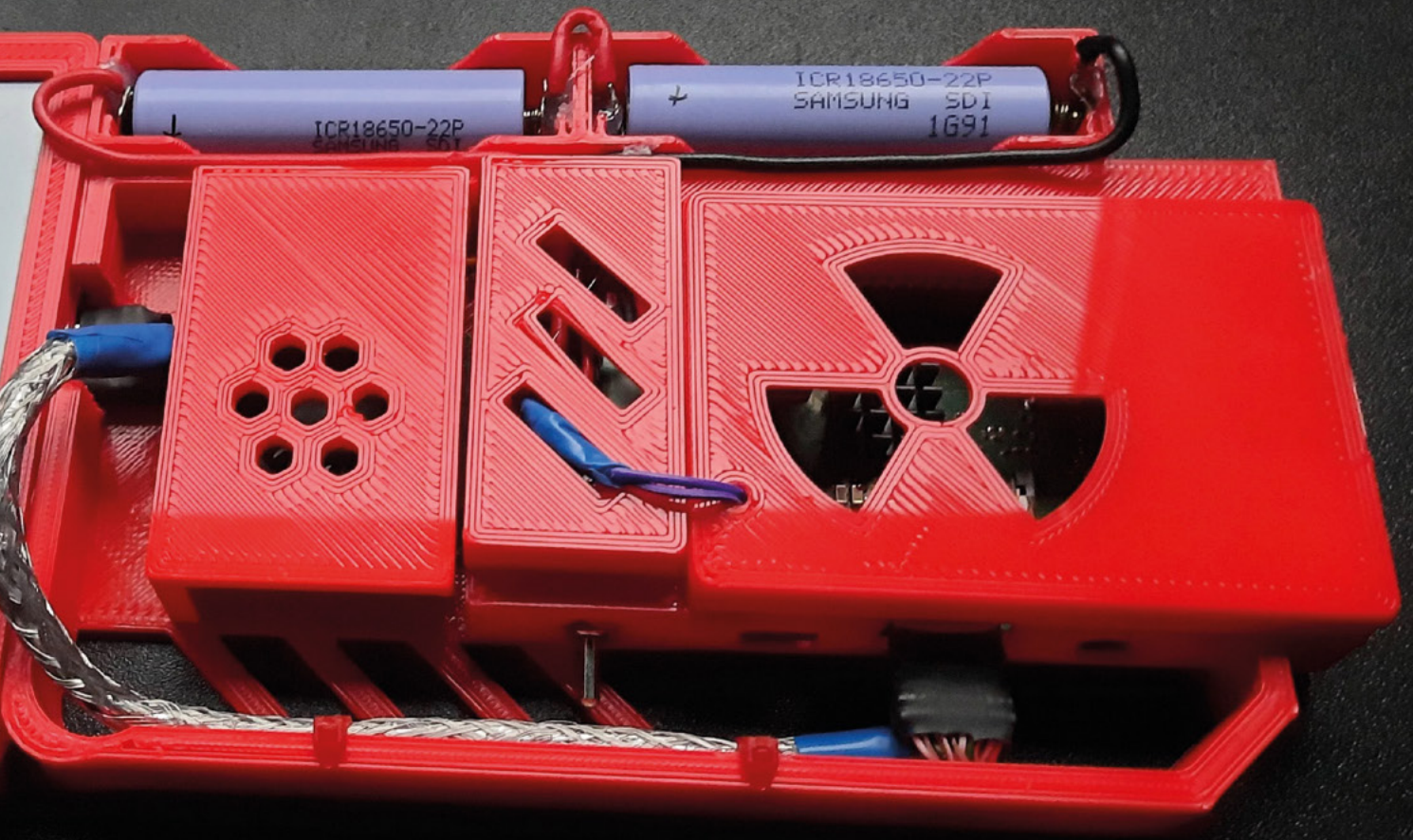
3D-printed Holo Clock

By **saulemmetquinn**

hsmag.cc/HoloClock

Analogue-display clocks have been around in one form or another for centuries. Now, with 3D printing, it's possible to design things that just wouldn't have been possible even 20 years ago. With this circular design (the maker describes it as a halo clock), maker saulemmetquinn has turned clock design upside down. It's still recognisable as a clock, with a round face, a big hand, and a little hand, but there's something refreshingly different about this design that we like.

The gear shafts are brass rods, cut to size using a Dremel, and the current design has a few exposed jumper wires. The whole thing is driven by an Arduino Uno and a stepper motor, with the frame and gears printed in white and black PLA.

Right ↗
It's easy maths to make sure that the large hand moves 12× faster than the small hand
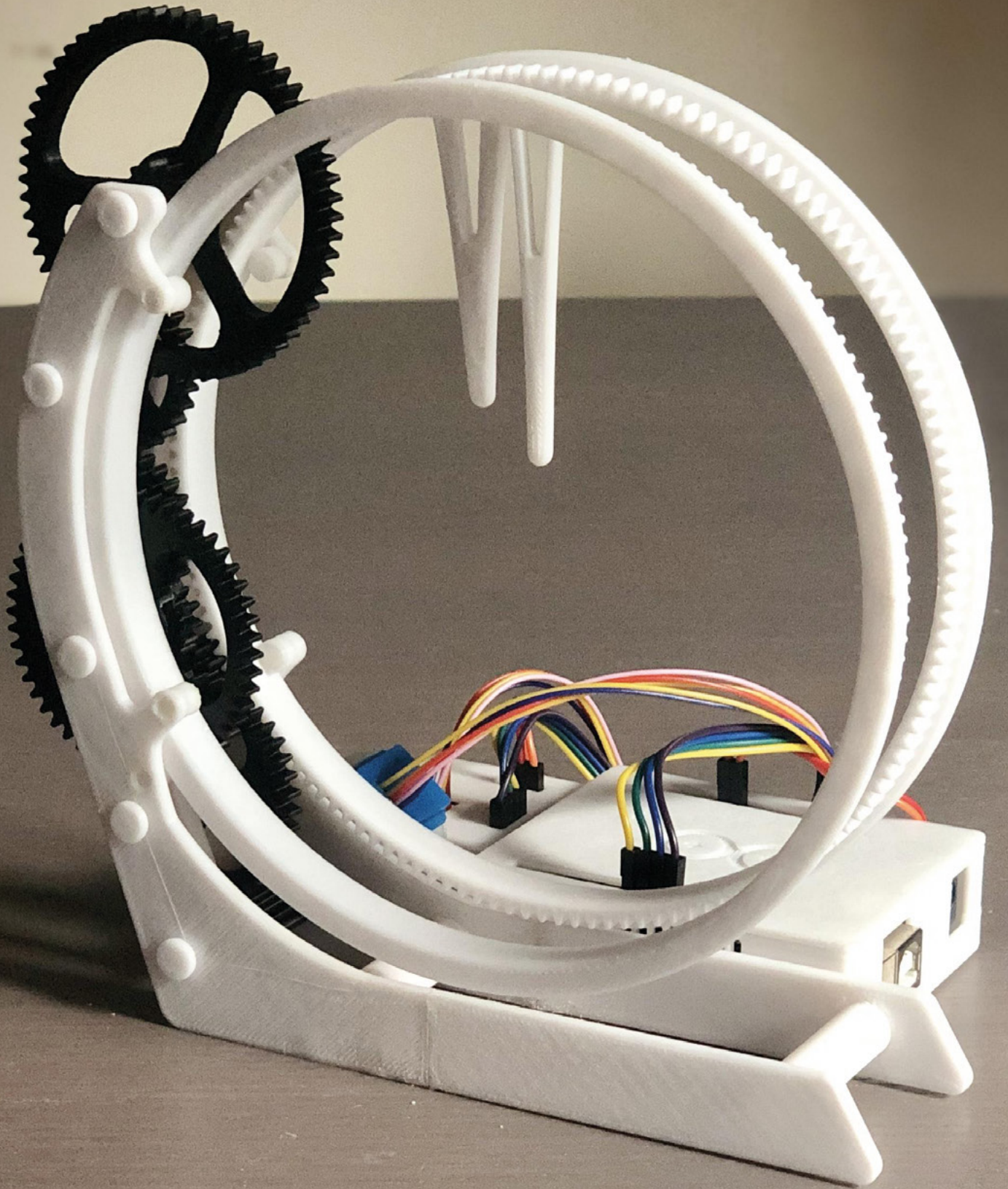
Pico audio

By **Sergio Blanco**

"I have several audio sources that I need to route into a single auxiliary input in my amplifier, so I used to have to unplug one to plug another source in." For example, if I was listening to a CD and wanted to switch to a DAC/Volumio source, or if I was listening to a vinyl record and then wanted to hear a song from a Bluetooth source, I needed a device to help me with this.

"I knew that I needed the project to be able to power up and power off without waiting for an operating system to boot – this made the Pico a far better choice than a Raspberry Pi Zero.

"So, to the point; for this project I used a Pico to control two 8-relay control boards, so I got eight stereo inputs that I can switch by means of a joystick, although I use it only from left to right to select the different audio sources. I also used an I2C OLED display to show my family logo at the beginning and the selected source after a couple of seconds, which is the time needed by the amp to power up and be ready. After a power-off, the last selected source gets saved, so on power-up, it will always come up with the last selected source."
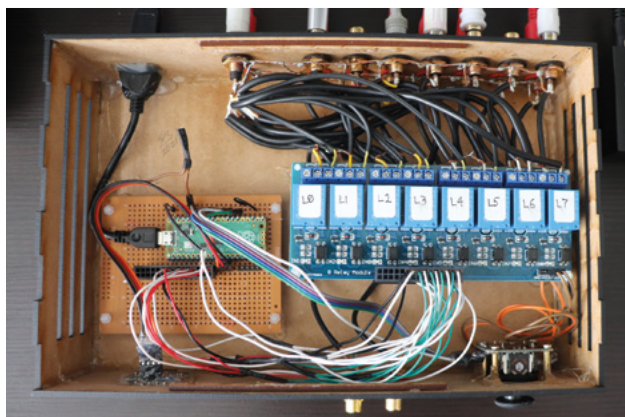
Right ⟋
Sergio's audio setup, now managed by a Raspberry Pi Pico, has a total of seven inputs: two phono inputs, cassette player, CD player, Volumio, keyboard, and a Bluetooth input

MiG-29K soldering kit

By **Phyx**   hsmag.cc/MiG-29K

This tiny soldering kit, with only five components (the PCB, battery holder, switch, and two 3 mm LEDs), is available on Tindie, at the bargain price of just $3.25. It's your way of getting your hands on a MiG-29 fighter jet – which was introduced in the USSR in 1982 and is still in service around the world – at pocket money price.

The kit is 6 cm long when assembled. The instructions on the PCB are in Dutch, as it's designed in Belgium for use in Flemish-speaking schools.

Right ⏩
Though a mainstay of Soviet Air Forces throughout the 1980s, the MiG-29 was officially secret at the time *Top Gun* was filmed; the producers named the enemy aircraft the fictional MiG-28
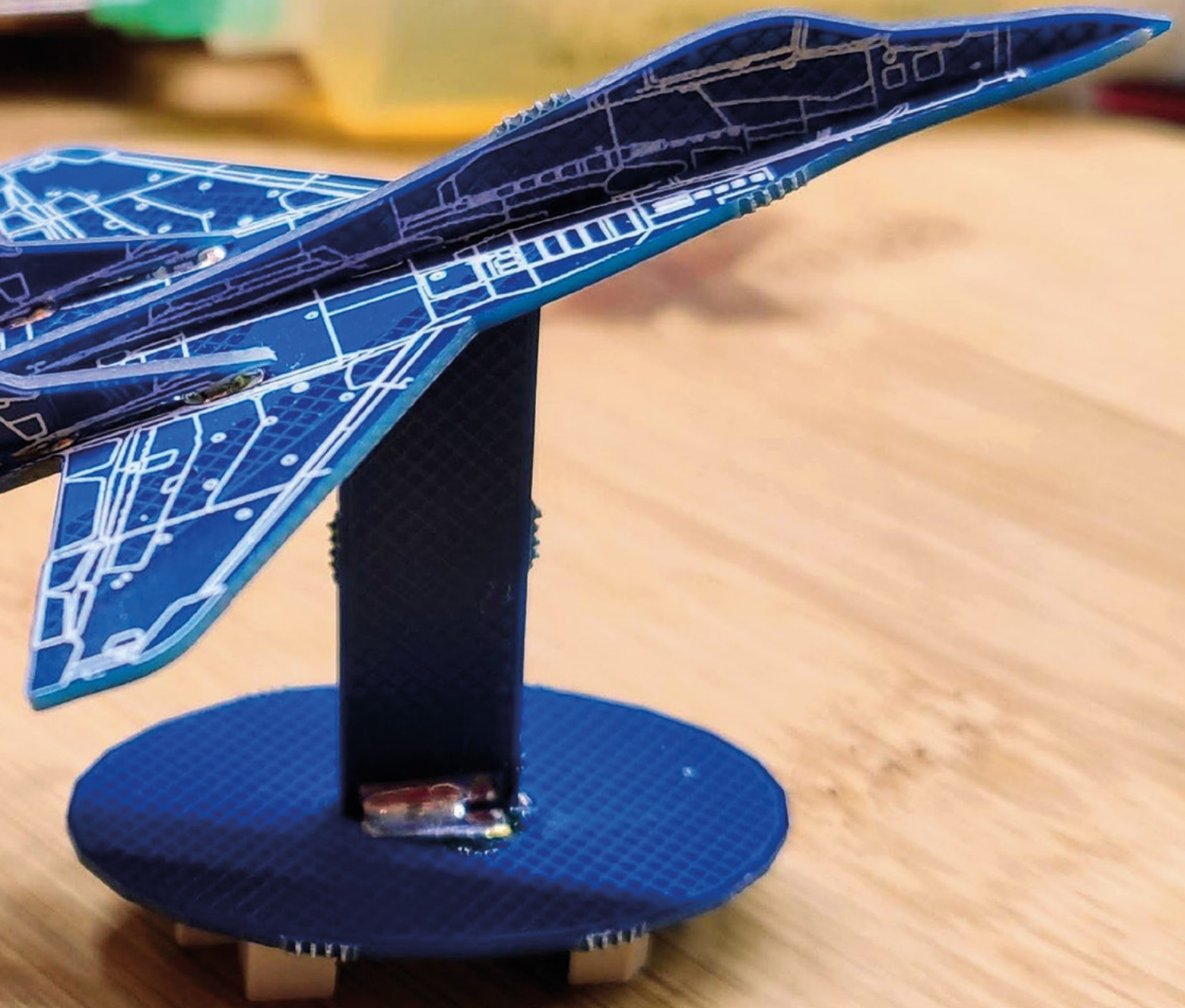
VFD Tube Clock

By **Neon Things**   ➤ **hsmag.cc/IV3TubeClock**

This clock is an updated version of an older kit by Cambridgeshire-based Neon Things. The update isn't to the VFD tubes, which are ancient bits of Soviet display kit now found on internet auction sites. The update is instead to the software that runs the clock. Whereas the previous version of this clock ran on an Arduino, this uses CircuitPython, making it considerably more user-friendly for anyone with Python as a first language. Even better, thanks to I2C, you can connect this clock to anything you want, turn it into a temperature readout, stock price monitor, or anything else that uses four seven-segment displays.
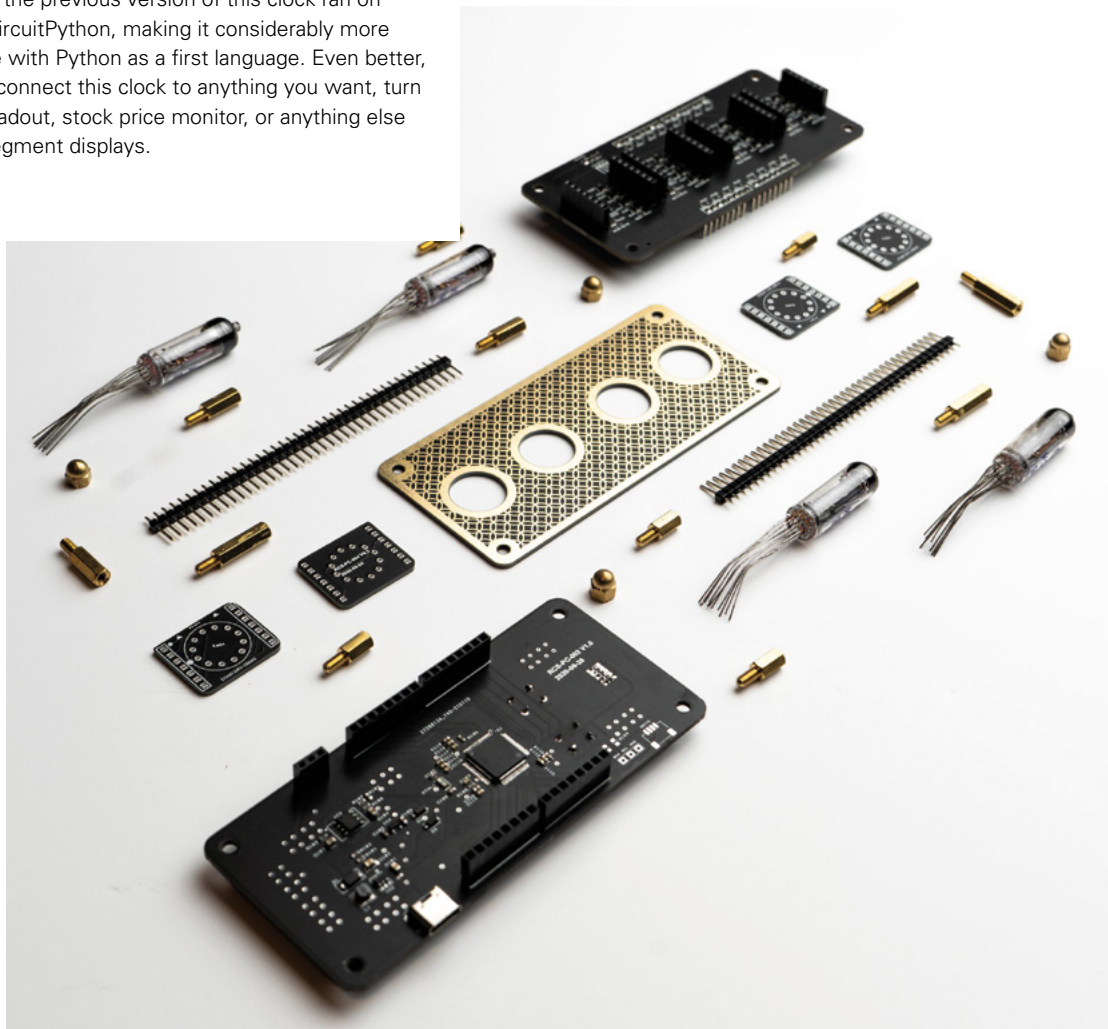
Teasmade 2.0

By **Malcolm Splendiff**   **hsmag.cc/Teasmade**

Robot servants are still a long way off – sadly, all the R&D money is going into Boston Dynamics' creepy parkour bots. We do, however, have a robot that will make you a cup of coffee when you awake, thanks to Malcolm Splendiff. This build is based on a Goblin Teasmade, a contraption from the 1970s that combined an alarm clock with a kettle. With a relay to bypass the alarm, a Raspberry Pi and a Python script, as well as a device called a Clever Dripper that combines water with coffee, Malcolm's made a smart IoT device that polls a Google Calendar once a minute for a configurable trigger word. When it detects the word, it waits until there's an eight-minute window in your schedule (that's how long the process takes) to make you a pot of coffee. ◻
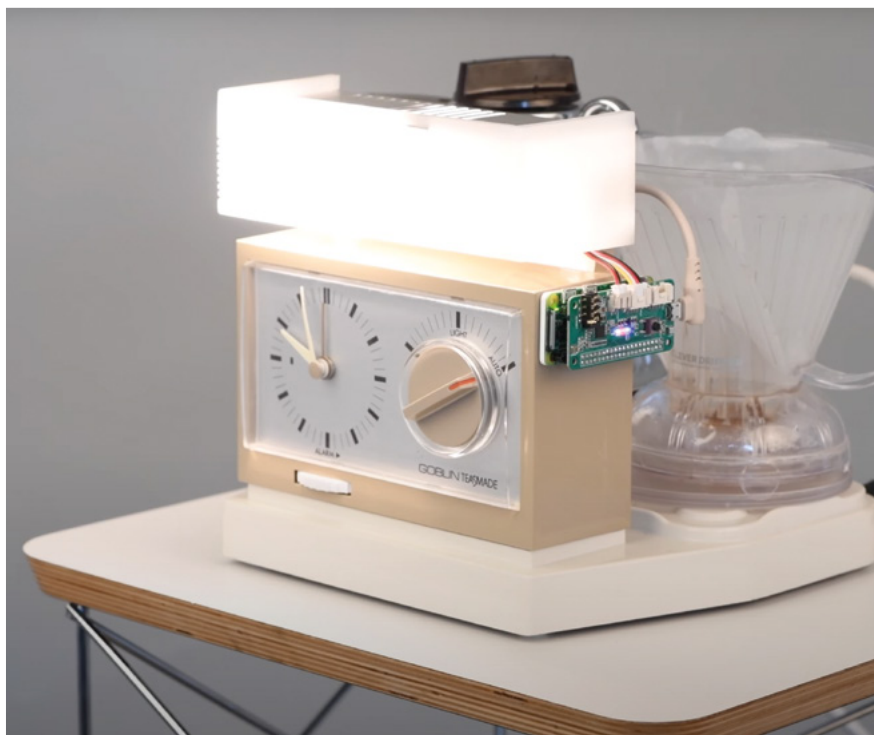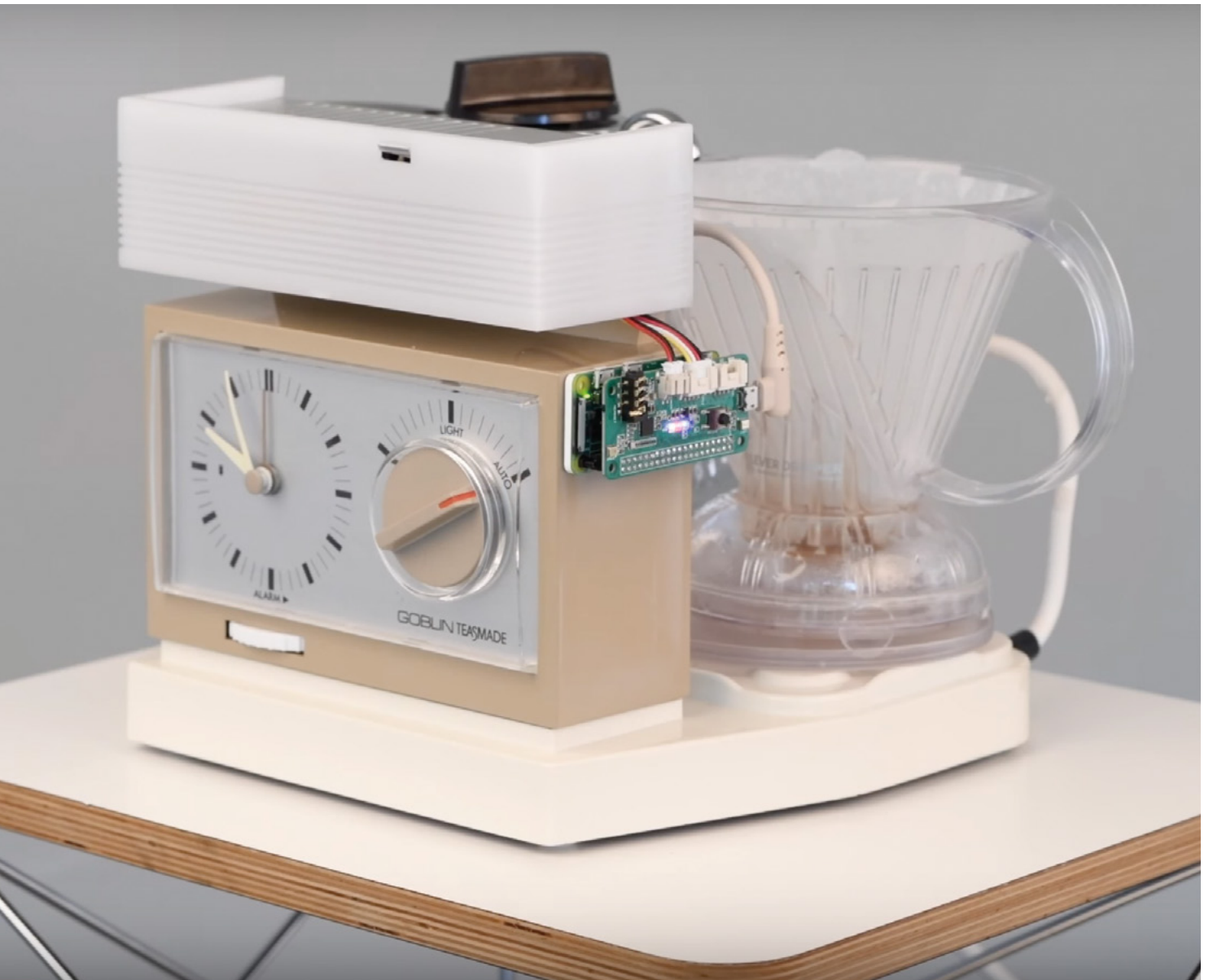
Objet 3d'art

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

Nicolaus Copernicus was a real Renaissance man – he was (among other things) an economist, diplomat, translator, and most relevant for us, an astronomer, who put forward the theory that the Earth went around the sun, rather than the other way round.

The Copernicus crater was named after him by lunar cartographer Giovanni Riccioli in the mid-1600s, when we were still burning witches. If you'd like a physical reminder that great scientific enlightenment existed in the world at the same time as widespread ignorance, there are few better ways to do it than to emulate this 3D print of the Copernicus crater, printed by daxpatel from information made freely available by NASA. This model was printed on an Ender-3, with a nozzle diameter of 0.4 mm, wall thickness of 0.8 mm, and a 20% infill. ◻

↗ **hsmag.cc/Copernicus**
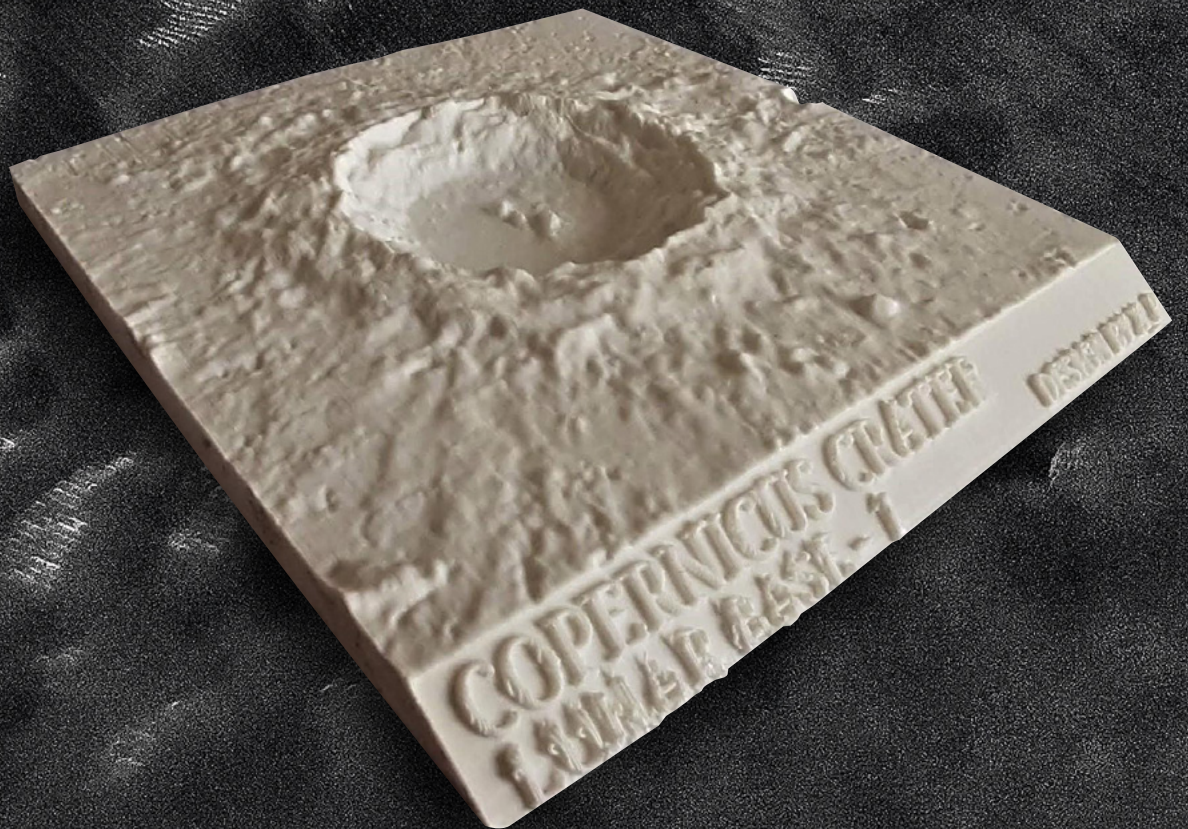
Meet The Maker:
Agnes Jones

The art of hitting things with a hammer

Photo: **Kevin J Thomson**

Agnes Jones is a sculptor and a blacksmith. Put away the image you have of a villager making horseshoes: she makes a combination of useful and purely decorative things, but the thing they have in common is they are designed to look good. We spoke to Agnes to find out what's it like being a blacksmith now, rather than in the 18th century; what it's like making things beautiful rather than utilitarian; and how do you go about making hard stuff look soft?

"There's a lot of talk right now of robots and things taking over people's jobs. My job was sort of taken over by the Industrial Revolution! It's freeing – I don't really have to do it for any particular reason; I'm doing it because it's a really enjoyable thing to do. And it means that you get to use the steel to do things that maybe seem a little bit surprising or different.

> **The big difference between shaping a line out** of steel and drawing on paper is that you've got absolute control of where the pen goes

"When I was being taught at the University of Brighton, I had one blacksmithing tutor who was a woman who was about five foot tall and really slim. And the two metalworking technicians were women as well. And so it just seemed like a really like a normal job for a woman to be doing. It wasn't until after I graduated and went to work for another blacksmith up in Glasgow that I realised that most blacksmiths are still these big, muscly men.

"I've always done drawing; that's something that I've done since I was like a child. And when I was at university, I did a lot of life drawing. Most of the drawings I do are line drawings. And it seemed obvious to make a jump from drawings that you're doing on a piece of paper with a pen, to translate that into metal.

"I can't remember when I started doing wire drawings. But that might have been at Art Foundation year or university, and then seeing the work of Alexander Calder. He worked around 100 years ago, making an incredible range of different things, but he did a lot of really comedic kind of drawings in wire of people, quite often of acrobats.

"I remember seeing them when he had an exhibition at Tate Modern a few years ago. I remember thinking that he was doing what I love. But he did it like 100 years ago; like how depressing that something that I really wanted to do had already been done so long ago!

"The big difference between shaping a line out of steel and drawing on paper is that you've got absolute control of where the pen goes. Whereas when you're using the metal, you're kind of shaping it to your idea of what it should be, but it already exists as a line, so you've got to think about where you're directing it.

"I use steel. It's really hard to get iron any more. I mean, you definitely can do it. And there are people who create iron, but if you're going to any kind of supplier, they'll be supplying you with mild steel generally.

"The amazing thing about steel is that you think of it as being this incredibly solid, structural piece of material. But if you heat it up in a forge, it turns into this incredibly pliable thing that you can bend to any →

Above ◆
Agnes Jones, at work in
her studio in Glasgow

Photo: **Kevin J Thomson**

different shape: you can tie it in knots; you can do anything with it.

"And then it solidifies back into this really solid structural thing again, so I think people think that being a blacksmith is, like, all about lots and lots of strength. But as soon as it's hot, it's so wibbly-wobbly and easy to use. It's amazing. It does have this really magical quality.

"The course I did at university was very much materials-based. So it was wood, metal, ceramics, and plastics. And it was such an exciting space. All of the other workshops were kind of messy and had things going on in them. But the metalworking one just felt completely different. It had this big fire in the middle. It was half open to the elements.

"It just felt much more gritty. I think there were 40 of us in our year, and there was me, and maybe one or two others, that decided to use the forge, and they were barely ever in. So, I felt like it was sort of my space. I remember the first couple of days I wasn't really able to do very much. And then there was a sudden moment at which just everything clicked in my head, and I just understood a lot more how to work and how to use it. I just felt like I'd found my material.

"Over the last ten or so years, I've been teaching other people on a one-to-one basis. And there's often a moment where you're teaching someone, and then they get that moment of realisation. That's a really fun thing to see. I teach anyone who wants to learn at the moment; I remember there seemed to be quite a lot of ex-policemen coming in to learn a while ago.

"Lately, because people have been spending more time at home, I've been doing a lot of practical work. There have been a lot of gate commissions and also railings. I've really enjoyed that as a process. You go and meet someone, you have a look around their space, and have a lot of discussions about what they're inspired by. I'm living in Glasgow and quite a lot of people are interested in Charles Rennie Mackintosh. So there's quite a lot of requests along the lines of that kind of thing.

"Your job is to kind of try and tease out the interesting bits and turn it into a design that they are happy with, and that you're also going to be happy with, because there is nothing worse than making a design that you feel isn't really worth your time.

"And so recently, I've made something very Charles Rennie Mackintosh-inspired, quite a lot of plant-like forms. [The client] before that wanted something in the garden that reflected the plants in the garden, which meant a lot of grass-like forms, so it became something that felt like it was growing upwards. And

then there was someone who's got this incredible house in Lanark with a walled garden, with lots of incredible different flowers inside. And that, combined with a Robbie Burns poem, ended up being this poppy gate. That was really, really fun to make.

"At the moment I'm on a residency in Copenhagen. It's always useful to be able to take yourself out of what you're doing and think about something different for a while. For example, the life drawing pieces: they were something that came about because I took a few days to do something that I enjoyed doing, instead of doing something that was like a project for someone else. And they've led on to quite a lot of interesting work, because as soon as someone sees what you want to make, it's easier for them to imagine other kinds of versions of that.

"And so, I think it is really important to take time to kind of do something that maybe is a little bit different to your normal way of working. So I'm hoping that something interesting will come out

> **It's always useful to be able to take yourself out of what you're doing and think about something different for a while**

of these pieces, because it means that it will be a chance for me to make something that's a bit different.

"The work I'm doing at the moment is all based on Piranesi. He did lots and lots of etchings around 300 years ago – loads of drawings of Roman ruins. Escher took a lot of inspiration from his work, because he did all of these things to do with dungeons and stuff.

"And anyway, I've had to respond to Piranesi, and I got really interested in the kind of ruins that he did. And so, I'm making some sculptures to do with ruined objects. →

Photo: Agnes Jones

"I've just got really interested in the idea of broken sculpture. It's kind of taken away from the original purpose that it had; it's been stripped of all its narrative. And so the idea that you could come across that in the middle of the street, made me really excited. At the moment, I'm doing lots of large-scale drawings of kind of slightly broken, Romanesque sculpture. I'm really finding it fascinating at the moment.

PROTECTING AND LEARNING

"The difference in terms of how well work will last inside or outside is mostly down to the finish that you put on it. So, if it's outside, you have to make sure that you're either galvanising or putting an undercoat and then top-coats on top. A lot of people ask for you to powder-coat things, which is sort of seen as 'the way' to protect metal, but it's horrible. It coats it in this plastic evil. But if it's going indoors, it can just have a really nice layer of iron paste, which is what you would put on a fireplace or something.

"The other thing is about the thickness of the steel. So obviously if it's quite thin, it's easier for it to corrode. A friend once put one of my indoor pieces outdoors, and it was all rusting through. And it was kind of rusting more at various points because of the thinness of the steel at those points. Also, with Glasgow being quite a wet city, you've got to maybe be a little bit more careful about things that are going outside than if it was somewhere else.

"I think it's really, really easy to get into blacksmithing. You've got to have a certain set of tools in order for you to do anything – woodworking, 3D printing, whatever. Also, if you are trying to become technically, really, really able, then that would take a very long time. But I think that it's actually really easy to play around with something like blacksmithing, as long as you're careful and use the proper protective equipment.

"Anyone can hit something with a hammer or bend a bit of steel. And yeah, I think the problem is having access to the tools to do that. Whenever I'm doing teaching, someone will come in and they will be a little bit scared of the fact that you've got all of these tools and techniques. These are techniques and tools that people have been using for so long that there's not really anything scary about them.

"I've got a lot of friends who are really good at ceramics. And I'm always a little bit wary of trying to get into that because everyone else has these incredible skills. But as long as you've got someone there who can teach you a little bit, then the whole thing is about learning in the way that makes you enjoy it.

"It's really frustrating at the moment, because I know that the government is cutting the amount of funding that goes into craft and art degrees. And it's really frustrating to see that happening, and seeing that other people don't value these skills. I know that it's not something that necessarily drives forward the economy [actually the arts in the UK are worth over £10bn per year, far more than, for example, fishing], but it does have a massive part to play in society.

"So, I did a research master's [degree] looking into how to use my metalwork in the street and how it's used in railings, and how often railings are used. Obviously, they've got to be a barrier. But whether that barrier needs to be something that makes people feel kept out of something, or whether it can be used to be much more welcoming is a different question. You can create spaces that feel much more welcoming and engaging. You need people who are good at making and good at designing in order to create an environment that people actually want to spend time in." □

Photo: **Agnes Jones**

Above ◈
As soon as steel is hot, it's so wibbly wobbly and easy to use… it really does have this magical quality"

Photo: **Liam Prior**

Build a Makerspace for Young People

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

Sign up today: rpf.io/makerspace

Letters

FINISH YOUR PROJECT

I had a swath of mixed emotions reading your guidance on finishing off projects. I have a load of half-finished stuff cluttering up my workbench, and none of it is getting done any time soon unless I take some of your advice on board. So, instead of waiting for a day off so I can go to the timber merchants, buy wood, cut all the dovetails, join it, glue it, plane it, oil it, then fit the electronics, I'm putting them in an ice cream tub. No, it doesn't look as good as if I'd handmade the enclosure myself, but it does have the benefit that it's finished. And as you said, a finished project is a good project. Maybe I can go back and do things better later on, but for now, done is done.

David
Leeds

Ben says: The best advice of all time is that done is better than perfect. A finished project not only clears your physical workspace, but it also clears part of your brain. All those half-done projects are in your head, churning up RAM (or whatever equivalent our amazing flesh computers use). When you finish something, you get a bit of your mind back for more important things; I even find that I sleep better when I've finished something off.

CITIZEN SCIENCE

I've read Anna Ploszajski's book, *Handmade*, and can confirm that it's excellent. One of my constant bugbears of chemistry at school was that it was so dumbed down at O level, once you got to A level, you found out that it was all just oversimplified and you had to learn everything about electrons, bonds etc. all over again. By comparing subatomic particles to bits of cake floating in the English Channel, Anna's given me an image that will stay with me (NB. the cake bit resonates far more strongly on a personal level than the feat of swimming the Channel). Also, on the theme of ordinary mortals understanding science: when did Luftdaten change its name to Sensor. Community? It's like the Marathon/ Snickers debacle all over again!

Dan Cohen
Bournemouth

Ben says: You should listen to the podcast too (also called *Handmade*). It's full of the sort of things that you don't know you're interested in until you're exposed to them, in the same way that you don't know what you want in the middle aisle of cheap, German supermarkets until you get there. Have a listen, and fill some of the empty space you've freed up by finishing old projects?
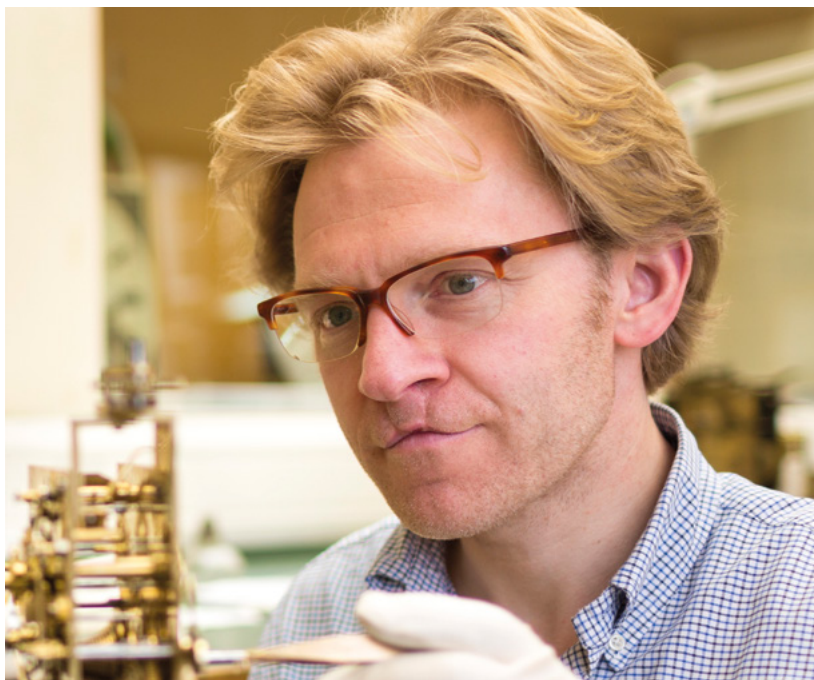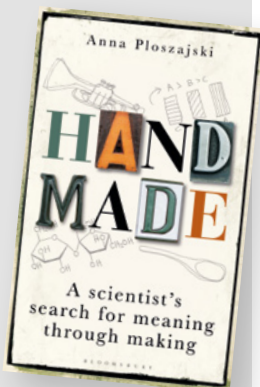
Christopher Ison: West Dean College

MAKING IS CARING

Your chat with restorer Matthew Read (in HS46) got me thinking. I saw a thread maybe a year ago where people were talking about how they got into making. It was (quite rightly) pointed out that if you ever took things to bits as a kid, you were incredibly lucky that you grew up in a household where taking things to bits was allowed! Things are expensive, and if you break them, you get in trouble with your mum. Or, in the case of historical objects, you get in trouble with the spirits of the people who made them, the people in the future who won't get to enjoy them, everyone alive now, and even the animus of the object itself. Far better to get some brass rods and try to make something new than to clumsily destroy a piece of heritage. Anyway, thanks for printing his wisdom – it made me think!

Margaret Baker
Basingstoke

Ben says: You are indeed a fortunate child if you can get away with disassembling functional things for fun (if you have inquisitive offspring who like to take things to bits, I recommend the broken stuff that you can pick up cheaply at car boot sales). And I take Matthew's point. If all we did as a species was go back to old things, all we'd have is progressively rounder versions of the wheel. Or maybe slightly hotter fire. Luckily for us, we're part of a rich line of human beings who like to try building new things. Do it for the process as much as the end result – you learn far more making something new than you do fixing something old.

CROWDFUNDING
NOW

Home Assistant Amber

A PC for your home

From $149 | crowdsupply.com | **Delivery: from Q1 2022**

Home **Assistant is one of the most popular DIY home automation systems.** It lets you connect a myriad of different smart devices together, control with them from a single interface, and even get devices to interact with each other. A huge range of both commercial and homemade device types are supported – you can see the full range at **home-assistant.io/integrations**.

While you can run Home Assistant on most Linux-capable hardware, Amber is here as a purpose-built platform to host the system. It's based on a Raspberry Pi Compute Module 4, and adds in a Matter-compatible Zigbee radio module and M.2 expansion slot. There's optional Power over Ethernet and an attractive polycarbonate case.

For $149, you get everything needed to run your own smart home (just add IoT gadgets). If you've already got a Raspberry Pi Compute Module 4, you can save $50.

Bringing everything together into one custom-built system (rather than relying on add-on USB devices and Raspberry Pi HATs) means the system is tidier and should be more reliable as there are no risks of

cables coming unplugged or contacts oxidising. For less technical users, having it come with the software set up and ready to go should significantly ease the learning curve.

We've not had a chance to test this out, but it looks like an easy way to get up and running with the maker-friendly smart home system. □

HackSpace

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 34

50 practical prints

Improve your home, your office, your entire life with 3D printing

50 best 3D prints

By Glenn Horan

Best stock up on filament: there's a lot of printing in these pages

The 3D printing community have been a busy bunch imagining, designing, and sharing their creative output for years. So much so that the most popular 3D model repository, Thingiverse, now has over two million (and counting) 3D models for you to choose from. This abundance of choice does have one minor setback: where do you start when looking for inspiration for something new to print, something useful, something that will improve your life in some way? We present to you, 50 functional 3D prints for your perusal.

To whittle down two million prints to a meagre 50, we decided upon a number of factors: how useful is it really? Is it practical to print on common 3D printer sizes (200×200 mm)? Does it only require minimal non-3D-printed parts? Does it actually work? Is it fun? We also gave priority to objects that were fully 3D printable. We avoided anything that came into contact with food or liquids to be consumed, and steered clear of anything that relied on 'exotic' printing materials (TPU and other flexible materials for example, are difficult to print with Bowden-type 3D printers).

There are a couple of minor exceptions to these rules for things that were just too good to pass up on and/or are worth the extra effort. We've also included an effort rating: 1 out of 5 is 'trivial' – a short 1–2 hour print and does its job right off the build plate with no modification; 5 out of 5 is 'significant effort' – multi-part print taking dozens of hours, with additional parts needed and significant assembly/printer calibration involved.

Finally, a shout out to the awesome 3D printing community. This list wouldn't be possible without the countless hours of work put in by Thingiverse contributors who have generously chosen to share the fruits of their labour online for people to print at home. →

FOOD SAFETY AND 3D PRINTING

Regardless of what material you use for your 3D-printed objects, they're generally not considered to be food-safe. This is because the nature of fusion deposition 3D printing means there will always be plenty of microscopic nooks and crannies for all sorts of microbes to hide in. This means even if you can expose the object to the high temperatures and chemicals which will kill off any nasties, and even if the print survives this process, there could be a few stragglers holding out somewhere. There are techniques to make printed objects food-safe, and we'll look at these in a future issue.

Tools

Turn your printer into an infinite tool-box

The biggest benefit of 3D-printing tools is that if you don't have the right tool for a job, you can simply pull down the files and get printing. This is great for things like spanners, wrenches, and even hammers, but where 3D printers really shine is in the ability to print things that are a little more specialised.

Sanding blocks

Web: thingiverse.com/thing:2404850
Designer: mightynozzle (remix of a design by workshopbob)
Effort: 2/5

Sanding blocks like these will help distribute force around the area being sanded and will save your hands some grief in the long term. These fully 3D-printed sanding blocks are a remix of a design which uses M5 nuts and bolts to hold everything together. They come in a range of shapes and sizes, but the contributor has also uploaded the base SCAD file so you can customise the dimensions of the print to your own needs.

Fractal vice

Web: thingiverse.com/thing:4904044
Designer: Michael Laws (Teaching Tech)
Effort: 5/5

A fractal is a pattern that repeats itself infinitely. You can see examples in nature in leaf patterns, sea-shells, even in the alveoli of the lungs. The fractal element of this design is the four layers of increasingly smaller semicircles, eventually finishing in small gripper pieces. The result is a clamp which can grip onto objects of virtually any shape with several points of contact so as to spread the pressure from the vice over the surface of the entire object. You can see a video of it in action on the designer's YouTube page here: **hsmag.cc/FractVice**.

While this is an especially complex print, we think you'll agree the result is amazing. Not only will it make an eye-catching conversation piece, but in many situations, it'll function much better than a run-of-the-mill desk clamp. It goes without saying that a well-calibrated tolerance and high infill are a must for this print.

Profile copy tool

Web: thingiverse.com/thing:3712856

Designer: PSomeone

Effort: 3/5

When trying to copy an irregular shape, or make something to match up against an irregular profile (cutting linoleum to go around a toilet, for example), it can be difficult to take accurate measurements, sometimes resulting in a 'measure once, cut ten times, and waste a lot of material' approach.

This design has minimal non-3D-printed parts, just a couple of nuts and bolts to keep the frame together, but the sheer number of 'feeler' parts means that this ends up a pretty beefy print. In terms of assembly, bolt one side in and start stacking the feeler pieces in one by one until all 40 are in.

Hex bit holder

Web: thingiverse.com/thing:4703894   **Designer:** SnakeP   **Effort :** 1/5

This design is so compact, it should allow you to access screws in the most awkward of places, while still being able to get a decent grip on them.

In terms of ease of print, really the only thing to worry about here is that your printer tolerances are well calibrated so that your hex bits will fit in securely, but not so tight that you can't remove them again. With a print of this size, we'd also recommend using high infill for added strength and feel.

Vacuum former

Web: thingiverse.com/thing:1611996

Designer: BallardBandit   **Effort:** 4/5

If you're unfamiliar with vacuum forming, basically you heat up a sheet of thermoforming plastic until it's soft and pliable, set it on top of an object you want to replicate the shape of, and then suck all the air out from between the object and the hot plastic so it forms around the object.

We found this design very straightforward to print, with the caveat that you really need to be using PETG or ABS rather than PLA due to the high temperatures it'll be exposed to. In the file list, you'll also find a template for a frame, which at a push, you can make out of thick cardboard. See our boxout for details on how to use this nifty tool. →

MORE COOL TOOLS

G-clamp
Web: hsmag.cc/gclamp
Designer: forrozza
Effort: 3/5

Resistor leg bending tool
Web: hsmag.cc/resbending
Designer: dnewman
Effort: 1/5

3D-printed wrench
Web: hsmag.cc/wrench
Designer: barspin
Effort: 2/5

Helping hands
Web: hsmag.cc/helphands
Designer: bn3dee
Effort: 4/5

One-handed bottle opener
Web: hsmag.cc/bottleopen
Designer: Kart5a
Effort: 1/5

Household

3D prints for an easy life

Our household section includes a bunch of mercifully easy prints that will both enhance the look of your home and/or enhance your day-to-day life in some way. As far as categories go, this was quite a big one since it covers office, kitchen, garden, storage, etc., but despite fierce competition, we've managed to pull together this shortlist of just ten objects.

Food bag clips

Web: thingiverse.com/thing:330151
Designer: MasterFX
Effort: 1/5

You'll always have at least one more bag that needs to be sealed than you have clips to seal them. That is, of course, unless you can generate an infinite number of them on demand!

There are a few clip designs online, but we picked this one due to the selection of clip sizes they have available and the fact that you can print them in place in one piece. The small size of this print means that you can print with 100% infill for something really strong and durable.

'Nonlethal' door-stop

Web: thingiverse.com/thing:200571
Designer: ChrisMUC
Effort: 1/5

There are a lot of design considerations squeezed into this door-stop, especially considering the job can be done with a simple wedge of wood. The designer has made sure that we're well aware of each of these on the description of the object on Thingiverse, as well as a number of alternative uses for the object.

This is a very easy object to print, and as a bonus, it's fairly well optimised in terms of how much material it'll take. It works great right off the print bed when using it in rooms with carpet and can be scaled if you've not much clearance under your doors.

Vertical wall planters

Web: thingiverse.com/thing:1716323
Designer: UptonAK
Effort: 2/5

These vertical wall planters look great both indoors and outside and scale really well to fit a wide range of different plants. You could have a small army of succulents on your wall indoors, or cascading trailing plants like ivy or trailing petunias outdoors. If the time of the year is right, a wall of strawberries would be irresistible once they're ripe.

These are very easy to print and require no supports. Even scaling them up 120%, it only took around three to four hours to print each one. The designer has uploaded a variety of models with different mounting options, so you can choose to screw them into a wall, mount them onto a backing plate, or just use a command strip. A great feature of these planters is that there are spouts on the bottom and holes in the top, so excess water from those at the top drains into the planters below.

Laptop stand

Web: thingiverse.com/thing:1330829
Designer: Thmsdc
Effort: 1/5

Using a laptop sitting flat is fine for short periods of time, but if you're working long hours, you really should have the screen at a comfortable height; your back will thank you for it. We chose this particular laptop stand for its simplicity and because it was exceptionally aesthetically pleasing.

The listing on Thingiverse states that it's specifically for MacBooks, but it should work with any laptop with a fairly thin body.

Self-watering plant pot

Web: thingiverse.com/thing:903411
Designer: parallelgoods
Effort: 2/5

This is a practical and very attractive self-watering flowerpot. The pot is split into two parts: the first is the actual pot, which has a funnel shape on the bottom filled with holes, and the second is the reservoir. Once you've put some soil and a plant in your pot, they will suck water up from the reservoir as the soil dries out.

This was a really easy print and had virtually no assembly. It did take around twelve hours for both pieces to print, but you could easily split this into two separate prints over a couple of days. The design scales really well, meaning you can use it for houseplants of virtually any size, from tiny succulents to beefy yucca plants. →

EXTRA HOUSEHOLD STAPLES

Hexagon shelves
Web: hsmag.cc/hexshelf
Designer: Guinlet
Effort: 1/5

Egg cup
Web: hsmag.cc/eggcup
Designer: joo
Effort: 1/5

Spice rack
Web: hsmag.cc/3dspicerack
Designer: MouZ304
Effort: 2/5

Plot clock
Web: hsmag.cc/plotclock
Designer: smily77
Effort: 5/5

Moon lamp
Web: hsmag.cc/moonlamp
Designer: humphrey_b52
Effort: 4/5

Gifts

Good-looking gifts that are cheap to make
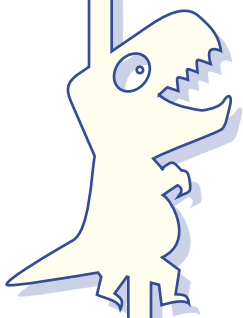
3D-printed objects can make great gifts, be it as a small token 2–3 hour 'flat pack' print, or something a bit more special in the form of a multi-day, multipart print that you've lovingly sanded down and hand-painted. If nothing else, the novelty of it being 3D-printed is an added bonus, particularly to those who don't own their own 3D printer, or are still very new to the concept.

Flexi animals

Web: thingiverse.com/thing:2738211
Designer: DrLex (remix of design by Kirbs)
Effort: 2.5/5

Forget fidget spinners; there's something strangely addictive about the satisfying way flexible prints flop around when you're interacting with them. We picked this T. rex designed by DrLex as it was the most popular, but we'd highly recommend checking out the other flexi prints on Thingiverse.

We put the effort of this print at a mid-level 2.5 because they require a fairly well-calibrated print for the satisfying free-flowing links to work. This is because of its 'print in place' design, which leaves the links vulnerable to being seized together if you're over-extruding. Print using 100% infill for extra weight and a more satisfying feel in your hand.
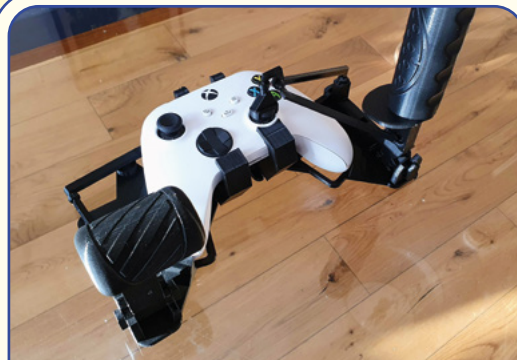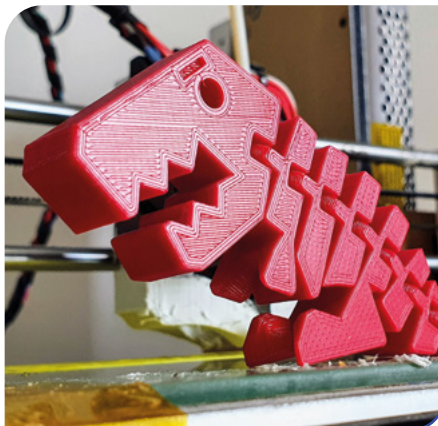
HOTAS controller conversion kit

Web: thingiverse.com/thing:4605273
Designer: Akaki Kuumeri
Effort: 3.5/5

Using a Hands On Throttle and Stick (HOTAS) setup is undoubtedly the best way to play flight games, from hardcore flight simulators to much more casual space fighter games. Thingiverse contributor Akaki Kuumeri has come up with an ingenious means of converting your plain old Xbox controller (there are also versions for PS4 controllers and even the Switch Pro controller) into a HOTAS rig.

The conversion kit is made with 100% printed parts (with an optional elastic band for a better action on the joystick), and with a sensible layer height (we used 0.2 mm), you can comfortably print the dozen or so parts over a couple of days. In terms of assembly, everything should just click into place. Once we worked in the joints a little bit and added a drop of grease, everything worked great. You can see the kit in action here: **hsmag.cc/HOTAS**.

Touring bike gift card

Web: thingiverse.com/thing:714484
Designer: CyberCyclist
Effort: 2/5

'Flatpack' models like this one make perfect little stocking fillers, or even just additional little tokens to slot into a birthday card. Like the flexi animals, there are many different gift card models on Thingiverse (velociraptors, trebuchets, TIE fighters, etc.).

These cards require a fairly well-calibrated printer with good dimensional accuracy in all three axes due to the interlocking parts. Once you have that dialled in, we recommend you print the first layer very slowly: around 25% of your regular print speed.

Collapsing pirate sword

Web: thingiverse.com/thing:3597069
Designer: 3DPRINTINGWORLD
Effort: 3/5

Make them walk the plank during pirate playtime with this 'print in place' pirate sword. We were impressed with the size of it once it's fully extended and were happy to see a nice blunt tip on it for increased safety during pretend adventures on the high seas.

This is a great design in that it requires absolutely no non-3D-printed parts. The biggest challenge we had was printing a sword where the interlocking parts weren't fused together, but the designer has thoughtfully uploaded a small test print, as well as a troubleshooting video, which helped us get it printed.

EVEN MORE PRESENTS

Cork pig
Web: hsmag.cc/corkpig
Designer: uauproject
Effort: 1/5

Fidget cube
Web: hsmag.cc/FidgetCube
Designer: acurazine
Effort: 2/5

Useless machine
Web: hsmag.cc/useless
Designer: pcdangio
Effort: 4/5

Mjolnir fridge magnet
Web: hsmag.cc/mjolnir
Designer: PRIma
Effort: 2/5

Cybertruck
Web: hsmag.cc/cybertruck
Designer: wov
Effort: 3/5

Page holder

Web: thingiverse.com/thing:1503676
Designer: Cruele
Effort: 1/5

Sometimes the simplest things can be the most useful. This page holder will come in handy for the bookworm in your life when they're lying in bed reading, or chilling on the beach and need to have a hand free for an ice-cold beer/piña colada.

Due to the fact that this is such a small print, we recommend printing it with 100% infill and with very small layer lines (~0.12 mm). It'll take a little longer, but it'll still only be a few hours and it will drastically increase the quality of the item if it is to be a gift. Just make sure to print a couple of test prints first to get the scaling right to make sure you (or the recipient) are likely to comfortably put your thumb through the hole. →

3D printing

The self-improving printer

Among the first things people print with their 3D printers are often upgrades for the printer itself, or accessories which will have the potential to vastly improve their day-to-day printing experience. We've tried to stick to general 3D printing accessories that complement pretty much any (or at least most) fusion deposition style 3D printers – and where that's not the case, it should be easy to find remixes for other models.

Noise-dampening feet

Web: thingiverse.com/thing:3520264
Designer: regularlabs
Effort: 1/5

Do your ears a favour with these noise-dampening feet for your 3D printer. It has to be heard to be believed just how much these reduce the noise from cheaper printers. This specific design is for the Ender 3, but there are noise-dampening feet designs for virtually every printer on Thingiverse.

Another easy print, just make sure that you have thick enough walls so that these are strong enough to hold up your 3D printer.

Belt tightener

Web: thingiverse.com/thing:1780636
Designer: supasorn
Effort: 2/5

Making sure you have appropriately tight belts on your 3D printer can help you avoid a whole host of issues – including, but not limited to, layer shifting, banding, dimensional inaccuracy, and malformed circles. This is a small and easy print, but it does benefit with some tweaks to your slicer settings: printing it with thick walls, or even with 100% infill, will ensure that it's strong enough to handle the belt tension.
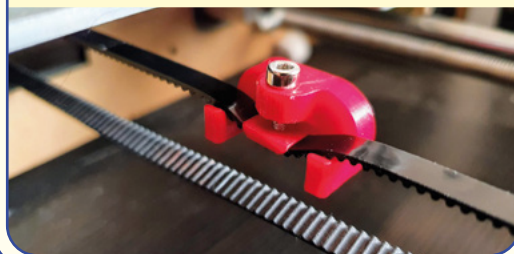
Filament sensor

Web: thingiverse.com/thing:2665162
Designer: stylesuxx
Effort: 4/5

Filament sensors are a means for a 3D printer to know when the filament has run out so it can pause the print until the user is able to replace their empty spool. For this design, you can choose to either plug the microswitch into your mainboard and get the functionality from the firmware, or into a Raspberry Pi running OctoPrint.

There's nothing to this print really; just make sure that you have the correct size of lever microswitch for whichever filament runout sensor you choose to print.
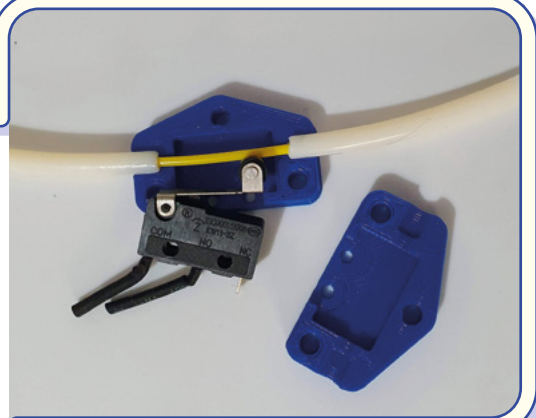
ADDITIONAL UPGRADES

Filament dry box
Web: hsmag.cc/drybox
Designer: crashdebug
Effort: 3/5

V-slot nuts
Web: hsmag.cc/vslotnuts
Designer: icimdengelen
Effort: 1/5

Nozzle box
Web: hsmag.cc/nozzlebox
Designer: KevinShinyDann
Effort: 2/5

Filament clip
Web: hsmag.cc/filamentclip
Designer: Walter
Effort: 1/5

Time-lapse camera holder
Web: hsmag.cc/tlcamera
Designer: Needitmakeit
Effort: 2/5

OctoPrint Raspberry Pi case

Web: thingiverse.com/thing:3852672
Designer: ArtieH
Effort: 2/5

OctoPrint is one of the best things you can add to your printer, so why not house your Raspberry Pi in an OctoPrint-themed case?

With this print, everything fits together snugly with no sanding or additional work needed. It can be installed onto any printer built using aluminium V-slot extrusions; you'll just need a couple of short M5 screws and T-nuts. You'll also need four M2.5 screws for mounting the Raspberry Pi into the case, but it's pretty secure without them.
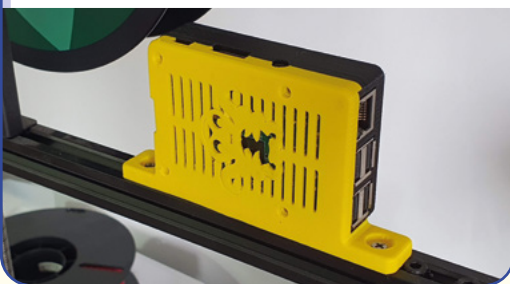
TUSH spool holder

Web: thingiverse.com/thing:2047554
Designer: filamentry
Effort: 2/5

A poorly designed spool holder not only looks ugly, but can have a negative effect on your prints. The 'TUSH' spool holder looks great, and your spool will roll smoothly over it while your printer is doing its thing. It also works with virtually any common size of spool, takes up a minimal footprint on your workspace, and uses very little material.

We put the effort of this print at a 2 due to the requirement of needing additional parts. The type of bearings you need for this to work well will need to be free-spinning (think of the type you'd get in a fidget spinner). Once you have your bearings, the parts simply click together. →

Aesthetic
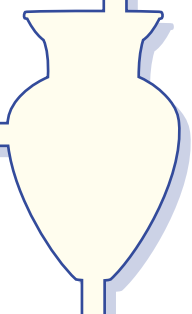
In this final category for functional **3D prints, we showcase what we think are some of the best-looking designs on Thingiverse.** In many cases, the designers have combined their love of art with their love of technology, and the fact that these are 3D-printed means they'll make a great conversation piece wherever they're on display.

Add a bit of flair to your home and office

Light panels

Web: thingiverse.com/thing:4686921
Designer: Akyelle
Effort: 4/5

These are a beautifully inventive way to use NeoPixels to transform the lighting of any space you decide to put them. While you can buy similar lighting panels, this project allows you to use whatever brain you want to control your LED strip of choice. Keep things simple (and cheap!) with a Raspberry Pi Pico, or connect it to the internet with an ESP32 to make IoT light panels.

While a quick look at the print files may have you asking some questions, the designer has uploaded a useful assembly PDF as part of the list of files on the Thingiverse page which will help you see which parts (and how many of them) you need to print.

Lithophanes

Web: 3dp.rocks/lithophane
Designer: Mark Durbin
Effort: 2/5

The image quality and resolution that you'll get from a 3D-printed lithophane is shockingly good, and they're a great unique way to share treasured moments. They work well when held up to a natural light source, but there are a load of designs out there for lithophane holders if you want something a little more permanent.

3D printing enthusiast Mark Durbin created his own custom lithophane creation tool and has generously shared it (and the source code) with the world. All you've to do is upload an image, select the shape of your lithophane, and download your custom STL file.

2D art

Web: thingiverse.com/thing:2618266
Designer: mightynozzle
Effort: 1/5

You may think it's ironic to use a 3D printer to make a 2D design, and you'd be right, but actually, the effect is genuinely really good. There are loads of beautiful pieces of 2D geometric art on Thingiverse, but we particularly liked this little sausage dog. As a bonus, the fact that it will stick out about 5 mm from the surface means that it'll cast some beautiful shadows (depending on how dramatic your lighting is, of course).

Although your build plate might restrict how big you can print these in one piece, they make exceptionally easy models to split and then reassemble when you're mounting them onto a frame, canvas, or even straight onto the wall.

Vase

Web: thingiverse.com/thing:2376777
Designer: eggnot
Effort: 2/5

Vases are so common because they make a great first 3D design project (check out HackSpace 28 if you want to design your own). We chose to showcase this design as it would be fairly difficult to recreate and we hope you'll agree it is a thing of beauty.

Of the two designs featured in this link, the thick one will take much longer but is watertight, so can be used as a functional vase. The other design is for use with 'vase-mode': it'll print fast and look great, but the shell will be riddled with fairly large holes.

ADDITIONAL ART

Deer wall art
Web: hsmag.cc/deerwall
Designer: UnpredictableLab
Effort: 2/5

Skull
Web: hsmag.cc/3dskull
Designer: LeFabShop
Effort: 2/5

Tensegrity Planter
Web: hsmag.cc/tensegrity
Designer: Gen0idea
Effort: 2/5

Wall Flower
Web: hsmag.cc/wallflower
Designer: 3DPVDB
Effort: 3/5

Solar Beehive Light
Web: hsmag.cc/solarhive
Designer: 3ddruckqueck
Effort: 2/5
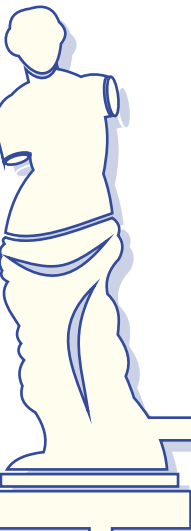
Various priceless pieces of art

Web: thingiverse.com/thing:83781
Designer: CosmoWenman
Effort: Varies

This collection of models really is amazing. Each of them is created by scanning a sculpture created from a mould of the original piece of art. We've chosen to showcase the Venus de Milo as we figured it'd be among the most instantly recognisable models featured. As these (obviously) haven't been designed with 3D printing in mind, you'll need to have well-dialled-in support settings so as to not ruin the surface of your sculptures. □

MAKE | **BUILD** | H

Hack

TECHNOLOGY IN Y

SO

NEE

HOW CITIZEN S
BOUNDARIES OF

NAILS
Repurpose spiky
steel for your makes

**CRYSTAL
RADIO**
Build a retro
audio system

Sept 2021
Issue #46 £6

FREECAD PICO **T-**

LORA
**Adding long-range
radio to a project**

0best
3D

INSIDE

**TOUCH
INPUT**
Add capacitive
touch to Pico

**DIY
GAMES
CONTROLLER**
Build your own
input device

Oct 2021
Issue #47 £6

**Essential
builds for your
home, office,
and workshop**

prints

CARDBOARD PICO **SMD** PURE DATA

HACK | **CREATE**

Space

hsmag.CC | **August** 2021 | **Issue #45**

**ERRY PI
LDS**

g projects you can make at home

+ **DIY Gaming**
Build your own
games machine

+ **Circuit
Sculptures**
Building electronic
artwork

+ **3D Design**
Build models
from sketches

E: 3D PRINTING IN COSPLAY

How I Made

LORA MICRO:BIT RADIO NETWORK WITH SENSORS

Connecting microcontrollers to the internet

By **Răzvan T. Coloja**

LoRaWAN and IoT devices **are relatively new in the IT scene.** LoRa technology is barely five years old. So I thought, why not use my LoRaWAN gateway to push data to The Things Network (TTN) using a micro:bit radio network? And, at the same time, build an IoT network of micro:bit devices that talk to each other and upload data to ThingSpeak – the IoT data network.

The more micro:bit devices you use for this, the better. I used six micro:bit V1 microcontrollers and some add-on boards. I also used a LoRaWAN gateway (a Dragino LPS8), but you do not have to use one – you can just stick to IoT and send the data to ThingSpeak. Or you can use another micro:bit and read the data on another microcontroller.
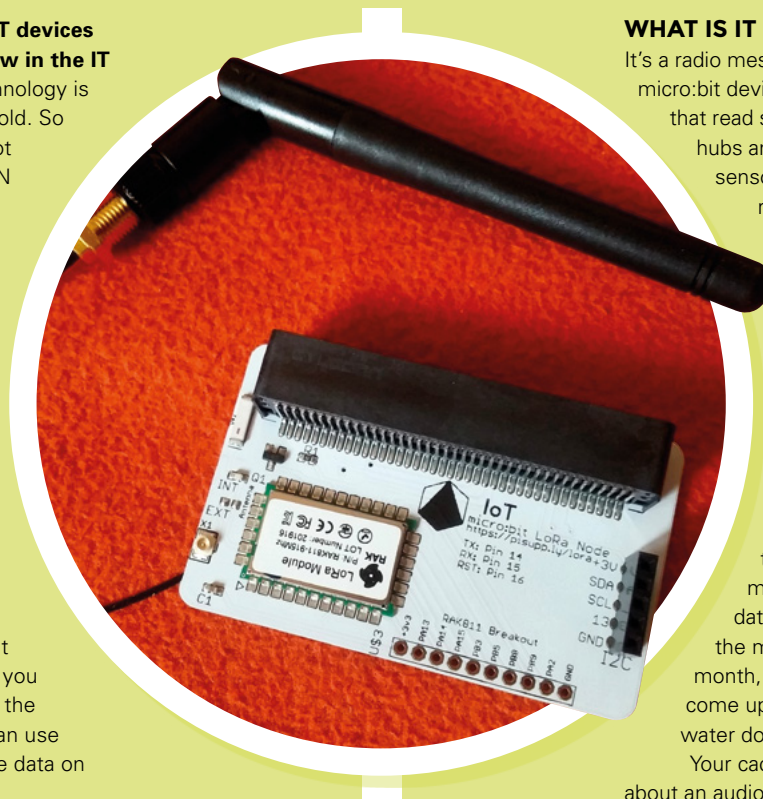
WHAT IS IT EXACTLY?

It's a radio mesh network made out of micro:bit devices. Some act as end nodes that read sensor data; some as network hubs and upload stations that gather sensor data via the micro:bit's radio and send that data to TTN and ThingSpeak. You can build an IoT home using just micro:bit devices. You can thus monitor your entire home: temperature, plant water levels, air humidity, light conditions, pressure, noise levels, and a lot more. Using TTN and/or ThingSpeak, you can gather that data, create graphs, and make statistics out of that data. For example – what was the median temperature of the month, at what hours did the sun come up this summer, how much water do your house plants consume? Your cactus needs watering? How about an audio and visual signal when the

What you need

- Six micro:bit 1.0 units
- ElecFreaks IoT:bit board
- Pimoroni enviro:bit board
- Kitronik Prong soil moisture sensor
- IoT micro:bit LoRa Node
- Monochrome OLED display (0.96 in, 128 × 64, IIC) (optional)
- Monk Makes - 7-Segment for micro:bit
- Keyestudio micro:bit 4-Digit Tube Shield
- Four-port USB hub (optional)
- WiFi router

pot is dry? How about an email when a red light is detected?

SENSOR – PLANT WATER LEVELS

Connect one micro:bit to the soil moisture sensor, and use a micro USB cable to upload code to it. The code is simple, as seen in **Figure 1**, overleaf.

Set whatever radio number you want, as long as it's not already used by one of your other projects. I used 7 for this micro:bit network.

Create a new variable and name it 'moisture', then set the new variable to use analogue pin P1 to gather data from the soil. If the soil humidity level goes below 100, then display a sad face on the micro:bit LEDs. If not, display a happy face.

Send soil moisture data via radio channel 7. The data is identified as 'Moisture' (capital M).

Now connect the micro:bit to a power source and stick it in the soil of your plant. Whenever the plant needs watering, the device will display a sad face. Data is sent via radio every 2.5 seconds.

ENVIRONMENTAL SENSORS

For this I used Pimoroni's enviro:bit. This is a nice little add-on board that has a lot of sensors. Install the micro:bit extension for it and arrange the code blocks as in **Figure 2**, overleaf. You can set the on-board LED to Off if you wish. Change the house icon

> Set whatever radio number you want, as long as it's not already used by one of your other projects. I used 7 for this micro:bit network

block to whatever you like. Be sure to also use 7 in the radio block.

Now define each of the enviro:bit blocks with a new variable. Create a new variable named `Light`. One named `Temp`. One for `Pressure`. Do the same for the other sensors and use the radio block to send each variable name through radio with its corresponding sensor reading.

Program a micro:bit with the overleaf code and connect it to the enviro:bit. Plug in a USB hub and connect the micro USB cable to this hub to power the microcontroller.

DISPLAY PLANT MOISTURE LEVELS

We will use the Keyestudio micro:bit 4-Digit Tube Shield to display water level data from the plant sensor. Connect another micro:bit to the 4-Digit Tube Shield, install the Keyestudio extension for it, and make the code look like **Figure 3**, overleaf. The LED blocks just show a nice animation of a graph bar moving down. It's just eye candy. Use the same channel 7 for radio.

This basically reads moisture data from the plant sensor and further transmits it on radio channel 7 after displaying water levels on the 4-Digit Tube Shield. This is so that you can have this micro:bit in another part of the house to display the data from the room where the plant is. Connect this new micro:bit to the USB hub. →
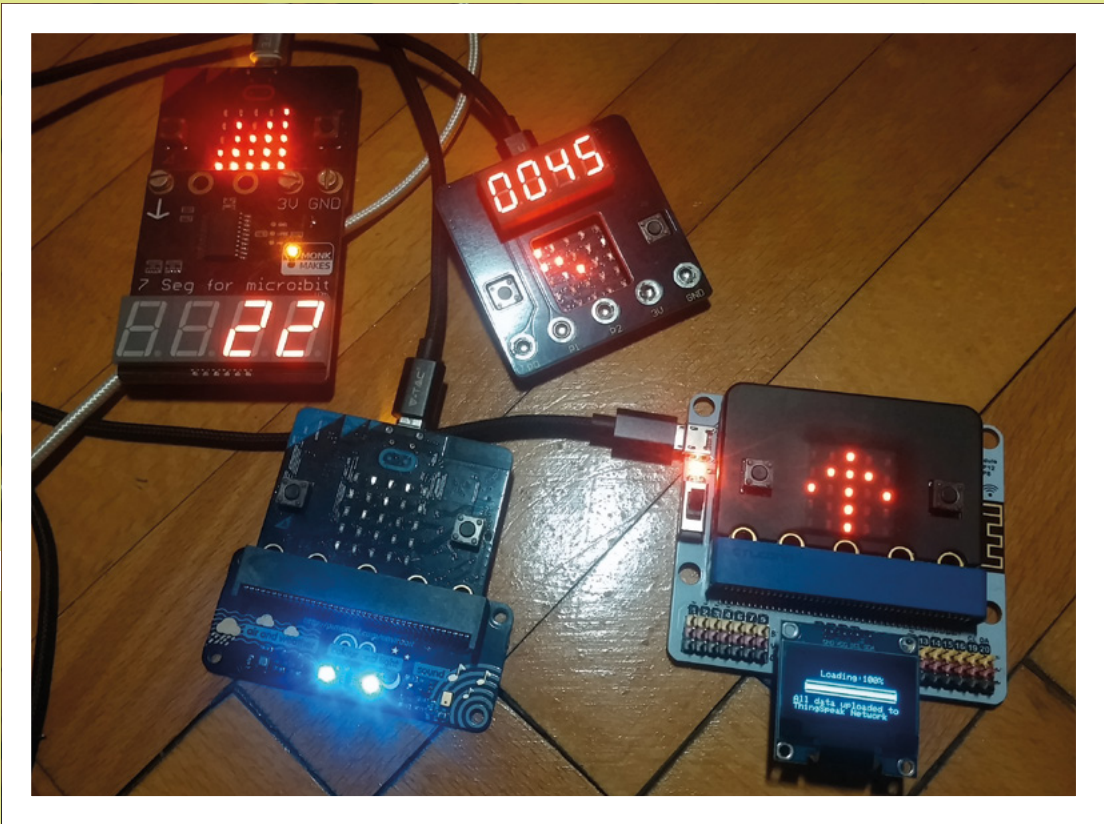
DISPLAY TEMPERATURE

Take another micro:bit and make it display
the house temperature using a Monk
Makes 7-Segment for micro:bit shield.
This is another four-digit display shield
that has larger numbers. We can set
it to display the temperature using the
micro:bit's built-in temperature sensor or
to use the temperature readings from the
enviro:bit board.

Make the code, upload it to a micro:bit
and connect it to the seven-segment board.

Then connect these to the USB hub to
power them.

GATHER THE IOT DATA

We were using radio channel 7 – remember?
All that data has to go somewhere – this will
be to another micro:bit that is connected to
the IoT:bit board. This board has WiFi and
can upload it to ThingSpeak. We'll make
a ThingSpeak account later and configure
eight channels to receive and display data
from this micro:bit.

We'll also use the small
monochrome OLED screen
here just to be visually
alerted if the connection
to ThingSpeak is broken or
when the data is uploaded to
the cloud.

Set the radio group to
channel 7. Add the IoT:bit
extension to be able to use
its blocks. The IoT:bit also has
an on-board real-time clock
powered by a coin cell battery.
Set the year, month, day, and
time. Next, use the OLED
block to initialise the OLED

Figure 1 ◈
Moisture levels
are reported
though the simple
medium of happy
or sad faces
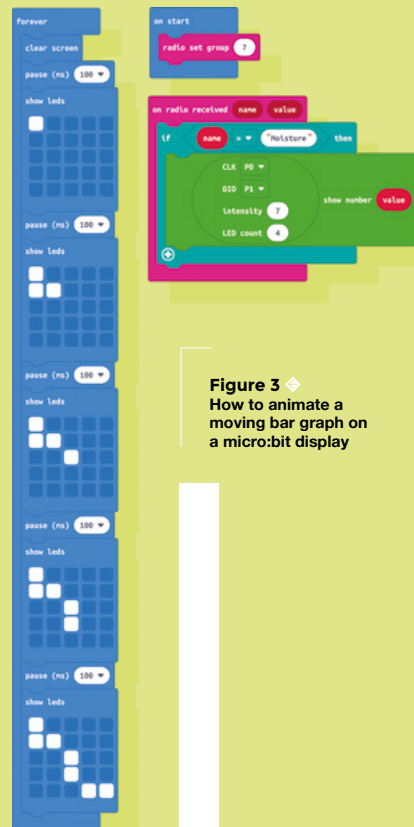
Figure 3 ◈
How to animate a
moving bar graph on
a micro:bit display

screen. Be sure to specify its dimensions in pixels: 128 × 64.

The board has an ESP8266 WiFi chip that has to be set to a baud rate of 115200. Enter your router's WiFi name and password in the 'connect WiFi' block. We'll play a note and display a tick on the micro:bit's LEDs when the board successfully connects to the internet via your router. Use the 'show string' block of the OLED screen to display something when this connection happens: 'Connected to WiFi'. Insert a 'newline' block to insert a newline character on the OLED, then display the current date and time on the OLED.

Now you have to make a few new variables: **name**, **value**, **Light-IoT**, **Pressure-IoT**, **Noise-IoT**, and the rest. We will name these with the -IoT suffix to differentiate them from **radio** variables and – later – **LoRa** variables. Each time a text string named like the ones sent through the radio channel 7 is received by this micro:bit we are currently programming, it is transposed through its equivalent **-IoT** variable.

And these **-IoT** variables are the ones we will send to ThingSpeak via the IoT:bit.

In the 'forever' block, add a 'clear display' block to clear the OLED screen. Input your ThingSpeak API key (we will configure this later), then add the variables with -IoT at their end to each of the eight fields. Use the 'upload data to ThingSpeak' block to make sure this data is sent via WiFi to the cloud.

The rest is just eye candy to be displayed on the OLED screen: a progress bar and information regarding a successful data upload. Finish with the 'connect ThingSpeak' block.

Insert the OLED into one of the three I2C slots on the IoT:bit, and the micro:bit into the IoT:bit board. Connect them to the USB hub. →

Figure 2 ◈
The code for setting the temperature, pressure, and humidity parameters

ThingSpeak is a cloud platform allowing one to upload data in various ways using its API (application programming interface)

Above ◆
You'll need to get an API key in order to allow devices to talk to one another

CREATE A THINGSPEAK ACCOUNT

ThingSpeak is a cloud platform allowing one to upload data in various ways using its API (application programming interface) and display it using nice widgets. You can make graphs and display them publicly, or embed them in your blog. Once you create a free account, go to Channels in the top bar and create a new channel. You can name it whatever you want.
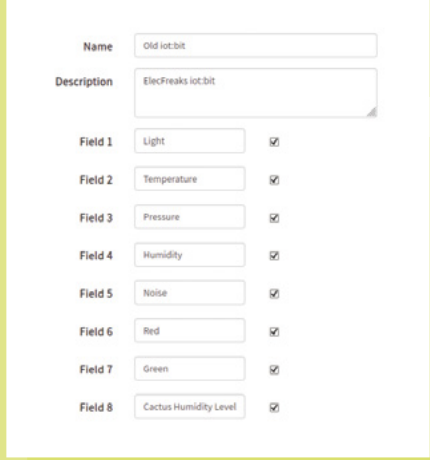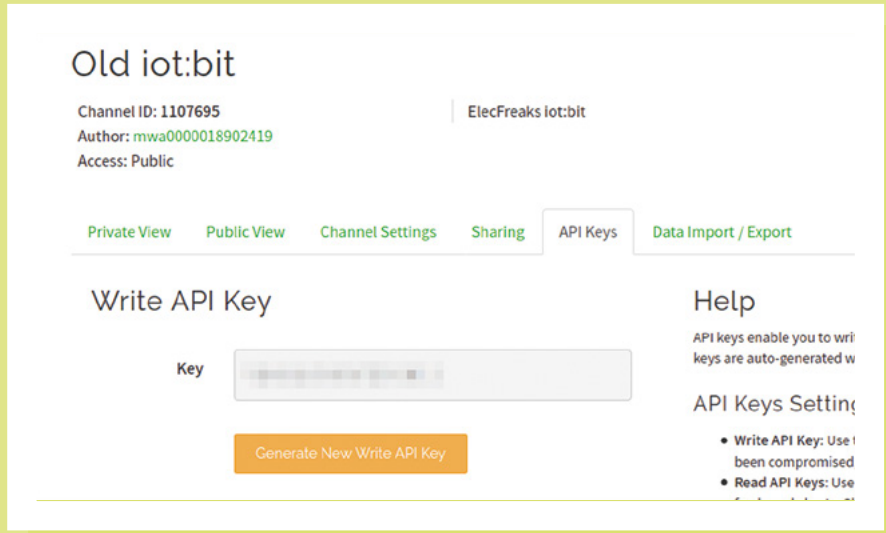
POPULATE THE THINGSPEAK CHANNELS

Input text labels in eight of the fields according to the eight fields you created in MakeCode in the previous step. They must be in the exact same order.

GET YOUR API KEY

Click the API Keys tab and get your Write API Key. This is the one you need to paste into your IoT:bit API Key field.

ADD WIDGETS

Still in the Channels tab, click Add Visualisations and Add Widgets, and start adding modes of displaying incoming data. My public ThingSpeak channel can be viewed here: **hsmag.cc/TS-Channel**.

So far, we have used MakeCode to add blocks, and gather and send data via radio channel 7 to an IoT:bit. The latter sends this data via WiFi to ThingSpeak, each datum on its own channel, each with a different online visualisation mode.

NEXT LEVEL: LORAWAN

If you have a LoRaWAN gateway, you can also use it to receive micro:bit data and upload it to TTN. For this you will need a micro:bit LoRa node. Connect the last micro:bit to a micro USB cable, add the

Left ◆
ThingSpeak is an IoT data aggregation platform
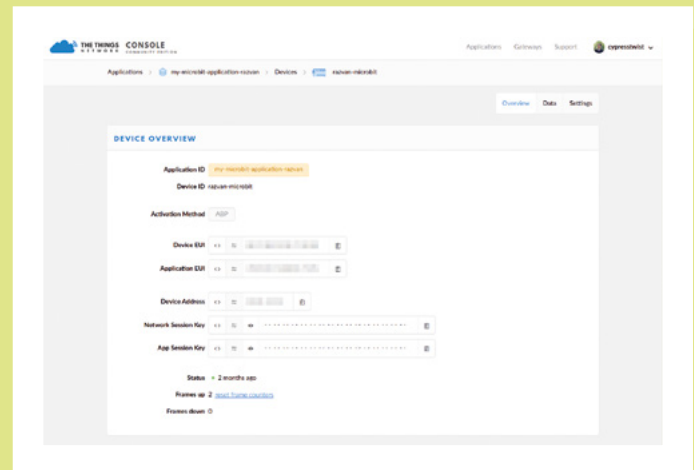
LoRa node extension, and use its code blocks to define the same variables we defined in the IoT:bit step.

You will need to have a free account to TTN. We will configure the LoRa micro:bit shield in the TTN Console in the next step.

For now you will need your Device Address, your App Session Key and your Network Session Key. Leave these blank for now. Set the radio group to 7. Set the SF (Spread Factor) to 7.

Since TTN has strict upload policies, you should test if your link works using a button on the micro:bit, and only after that, set time limits on an automatic data upload. In the example, whenever button A of the micro:bit is pressed, the LoRa node sends data to TTN. This data is defined by the variables we used before – the ones ending in -IoT. Define one for

each of the eight channels of the LoRa board. Input the data received via radio group 7 into these variables, as shown above.

TTN CONFIGURATION
This step assumes you already have a gateway configured and registered with TTN. Add your LoRa node and configure it to use ABP and not OTAA (Over-The-Air-Activation). In the device's Settings panel, set the Frame Counter Width to 16 instead of 32 and then disable Frame Counter Checks. There should be a warning stating that: 'Disabling frame counter checks drastically reduces security and should only be used for development purposes'.

Copy the Device Address, Network Session Key, and App Session Key from the TTN Console in their respective fields of the MakeCode blocks, upload the code

to the micro:bit, and insert it into the LoRa node board. You should get data when pressing the A key on the micro:bit.

CONCLUSION
You now have a mesh network of micro:bits that reads various environmental data, checks your plants for water levels, uploads it to ThingSpeak in the form of nice graphs, and also uploads it via 915MHz or 868MHz LoRa bands to TTN via your LoRaWAN router.

It doesn't get more 'IoT' than this with the micro:bit. ▢

arm DevSummit

Learn to Write Fast and Run Fast on Arm

Join Arm DevSummit
Virtual Event Live on October 19-21, 2021

OVER 140 AGENDA ITEMS ACROSS THESE SEVEN TRACKS:

- Cloud Native for the Endpoint, Edge and the Cloud

- The Future of Mobility is Software Defined

- Developing Interactive Consumer Experiences

- Practical Software Development for a Productive IoT

- AI in the Real World: From Development to Deployment

- Hardware Design Methodology

- Optimizing Arm Technology for PHC, Cloud and 5G Infrastructure

DON'T MISS THESE HOT WORKSHOPS FOCUSING ON PRACTICAL SOFTWARE DEVELOPMENT FOR A PRODUCTIVE IOT:

- Introducing Keil® Studio and Cloud-based Development for IoT and Embedded Applications

- Containers Over-The-Air: Building Your First Edge AI Device the Easy Way

- Ultra-Low Power Wi-Fi Made Easy – Develop a Battery Powered Wi-Fi Sensor in Hours

Visit devsummit.arm.com to Register

INTERVIEW

Tommy Marshall

Where 3D printing, electronics, and music overlap

Y
ou might have seen (and/ or heard) the wares of Oskitone over the last couple of years. Oskitone synths are small, brightly coloured, lovingly made analogue synths that make people smile.

Oskitone fits a very HackSpace-shaped niche. It's music (which is maths), open-source electronics, 3D printing. We sat down with Mr Oskitone, Tommy Marshall, to chat about (among other things) his new synth, the Arduino-compatible Oskitone Scout, how he got here, and why music isn't just for sensitive young men with guitars and lumberjack shirts. →

HackSpace magazine **Hi Tommy! These synths are absolutely beautiful. You must have some sort of background in music, industrial design or electronics, right?**

Tommy Marshall I don't have a background in anything! I basically have some kind of film degree. It's technically called telecommunications. I don't know if you've ever heard of that. It's an antiquated term, from the radio days, I think. OK, so I have that degree; I worked as a videographer for a little bit.

When I was a kid, I got into web development. I worked my way up – I wanted to be a software engineer. I moved to San Francisco and joined a startup – for the last ten years or so, I've been doing a lot of software stuff here in the Bay Area.

But I kind of became a little disillusioned. You know what I mean? When you're in your 20s, you think the world is your oyster. Technology can solve all problems. You get a little older, you hit 30 or whatever, and you're like, oh, technology isn't the answer to everything.

I actually don't like computers. I don't like computers at all.

HS **That seems to be a common factor among people who work with computers: hating them...**

TM And so I wanted a hobby that was still technical, but got me away from computers. And so I bought one of these how-to solder kits.

I did that for a couple of years. I just did various kits: I made various little noise machine things. I figured out that you could use a 555 timer to make circuits that make square waves. And then I figured out that if you give them the right resistors and capacitors, you can make very specific square waves. And if you do that a lot, you can make a lot of them on one thing, and you can make some kind of musical instrument.

Eventually I quit my startup gig. I don't know if entrepreneur's the word, but I

was going to see if I could scratch out something with this. So I did that, and that's when I made the OKAY synth.

HS **We reviewed that in the magazine in issue 1 or 2. Many moons ago.**

TM The first instrument I put out was just called the OKAY, because I wanted to lower everybody's expectations. When I look back at the OKAY, there are so many mistakes, so many things I wish I'd done better. But that's life.

I had to go back into software for a bit, then Covid hit and I got laid off. I thought I'd try again. That's when I made the POLY555, which is just a ridiculous thing to come out of dormancy with. Nobody would want something like this. It's the most esoteric, hilarious little

> When you're in your 20s, you think the world is your oyster. Technology can solve all problems

joke thing [NB it's currently sold out on **oskitone.com**].

Then I put out my version of the Atari Punk Console, which is a simple 555-based noise machine, and then there's my most recent project, the Oskitone Scout, which is my first foray into microcontrollers.

HS **Your stuff has a distinct look to it. What are your design influences?**

TM You ever had one of those moments when you connect the dots in your life – you see things that you do now, and you think of something that you did as a kid, and you perfectly understand how those two things are related? When I was a kid, I loved Altoids containers, little tins. I would collect all containers. I would, like, smash them and try to reform them.

Right now I love containers. Boxes with a bunch of stuff in that fit perfectly inside, like a Russian doll. When everything just sort of nestles in, it's so satisfying. To me, all the toys I loved when I was a kid, were totally just like, little figures in other little things.

When I did websites, I wrote the front end code, which, if you're making websites is the part that the user sees and is how the user interacts with things. So, I had to decide things like, if the browser window narrows, what does the website do to rearrange? And so, I tend to think about design in terms of code.

I'm still a programmer at heart, so all my 3D-printed stuff is designed in a programming language if you can believe it. It's called constructive solid geometry.

But it's basically like, you have code to make a cube, you have code to make a cylinder. Invert the two: now there's a cube with a hole through it. Now, do a bunch of other stuff. Now it's slanted; now do that 1000 times. Eventually, you'll end up with something that looks like things that I made.

That's just how I tend to think about stuff. And it works well when you're designing against PCBs.

HS **You said earlier that the Oskitone Scout is your first foray into microcontrollers. How did the previous synths work without a microcontroller?**

TM So, there's a thing called a 555 timer – it's a chip with eight pins. Just as a side note, Evil Mad Scientist make a version that they call the Three Fives Discrete 555 Timer. It's basically the same thing, but with all the transistors and everything pulled out. So you're making this chip from scratch. I met them at Maker Faire a couple years ago, and they plugged their thing into my thing. And it worked perfectly. It's very cool.

But anyway, depending on who you ask, it's an analogue chip, and it's pretty old, from the 1970s. One of the things →

it can do is make a wave. It can make a wave like this [makes the shape of a wave in the air] or a wave like this [makes pretty much an identical-shaped wave, but which for audiophiles is probably completely different]. The POLY555 does exactly that, but there are 20 of them – one for each of the notes. That's why it's so ridiculous. Somebody once told me that it was like having a two-wheel semi-truck where each wheel has its own motor. I think that's a pretty good analogy.

With the Scout, I've got rid of a load of that complexity by using an ATmega328, which is the same kind of chip used in the Arduino Uno. It lets me do a lot of the same kind of stuff with many fewer parts. It's me catching up with where I want to be.

It's a cute little thing. In startup terminology, it's a minimum viable product. I think it's a nice little rainy weekend afternoon project. I hope people like it, and I hope it will be a good foundation for me to do other stuff – obviously there's scope to add more keys and other weird stuff, but maybe things that don't even look like keyboards. There's a lot of stuff that I want to do with Oskitone that I haven't done yet. Once you've got something Arduino-compatible, your world kind of opens a little bit. It's software, and I know how to write software. It's not impossible. Stakes are low. It's not rocket science. Like, if I get it wrong, nobody dies. I can just try it, and if people like it, they'll buy it.

HS Have you got any idea of the sort of people who are buying your stuff?

TM I hear back from a lot of dads with kids. I think that might be my main demographic.

You would think a lot of musicians would like these. And they do, but not as much as parents who just want something to do with their kids. It gets them away from a glowing screen, which really harkens to why I started doing it in the first place.

I try hard to make electronics kits that don't look like electronics kits. →

Do you know what I mean? If you go on Tindie, there are a lot of things that are components on a PCB, and that's kind of it.

There's this whole joke about things designed by engineers, like it's, it's more function than form. And so, I try to balance the two things.

There was a computer called the Pocket Ship that had this little hole that you put a pen in, and it's like, you put it on a shelf, and it just propped itself up. I liked it. Because electronics projects often end up sitting on shelves, I added a pen pole to everything I make. It's just a hole that you put a pen into, and it lets you prop the thing up so that you can admire it when it's finished rather than stick it in a box out of the way.

HS How do you manage working at scale?

TM The POLY1, which I only made one of… which is why it's $800, and also why I've never sold it. It's all on perfboard. I did it before I knew how to make PCBs, and before I got into 3D printing, everything was on perfboard. Everything, like, the keys are made out of wood that I cut and glued together. And perfboard is a painstaking process.

When I quit my tech job, I bought a 3D printer. I borrowed a book on EAGLE, the design software. I wanted to learn how to make things like this on a slightly bigger scale. Instead of making one, I want to make ten, the maybe 100.

HS Hang on. So you did it that way around: you quit your job first, and then started to learn how to 3D-print and design PCBs?

TM Yeah. I'm not sure I recommend it. But there's no better fire under you than you running out of money. You know what I mean? Like, the runway is short. You gotta start running.

There's a spectrum that has industrial design at one end and craftsmanship at the other. Craftsmanship is like a table that you make one out of a redwood that fell down; industrial design is like a desk that's made out of honeycomb cardboard, and you make millions of them. The POLY1 was made by me as an artist, and I was interested in getting further along that scale to industrial design.

I'm not an industrial designer, but I'm not quite a craftsman or an artist. I'm somewhere in between. And 3D printing does a weird thing too… I can't really call it handmade, but there's the maker, the hobbyist, the business person in this weird liminal state between being an artist, a craftsperson, and a factory.

People will see my stuff and compare it to, like, Casio and Yamaha and Behringer. And I'm like, you know that they have whole factories spitting out these things. I'm making this in my house; I can't reach those prices or do any of that stuff.

Someday, I will figure out what to call myself so that people can set their expectations correctly.

HS You offer your synths as fully assembled, as kits including the 3D-printed parts, or as kits for people to print at home. How many of your customers end up printing for themselves at home?

TM I think about a third of the customers have 3D printers.

It's a very small sample size, and there's probably some selection bias here because, right now, it's on Thingiverse. I bet if I put it on Prusa Printers on their website, I'll get more people who have printers and stuff like that. I'm not advertising yet, but I think the people that I'm reaching organically are people who are more into electronics than 3D printing.

I have a Prusa that's a couple of generations old… I do it all in-house. I think 3D printing is really cool. It totally makes sense that it didn't survive the hype of five to ten years ago when everybody thought that every house would have a 3D printer in it. Of course that didn't happen.

I think it's more like having a soldering iron than having a desktop computer printer. Because you've got to know what you're doing. You can't just plug and play. I like 3D printing. I'm obviously more into functional prints rather than printing figurines and stuff like that.

But each their own. You know, honestly, it's a pretty cool community. And I'm grateful that they've mostly embraced me.

Music is like food. It's one of those things that's a great uniter: pretty much everybody can like music. I was at Maker Faire a couple of years ago, and I had a table full of these brightly coloured piano things.

And pretty much everybody, as they walked by, would stop and look. A lot of folks who didn't speak my language, or kids who were too shy to talk to anybody, would just be drawn towards these things and just want to bang on them. If you like electronics, and 3D printing and music, this is the product for you. I fit all those things. And the people who like my stuff, really like my stuff. I'm so thankful that when I launch a new thing, and I email out, I will see a lot of the same names come back and just keep buying my things. I love it. □

CODE
THE
CLASSICS

VOLUME 1

CODE

THE

CLASSICS

VOLUME 1

Brimble
Crookes
Gillett
Malone
Tracey
Upton²

Raspberry Pi
PRESS

CARDBOARD TUBES

IMPROVISER'S TOOLBOX
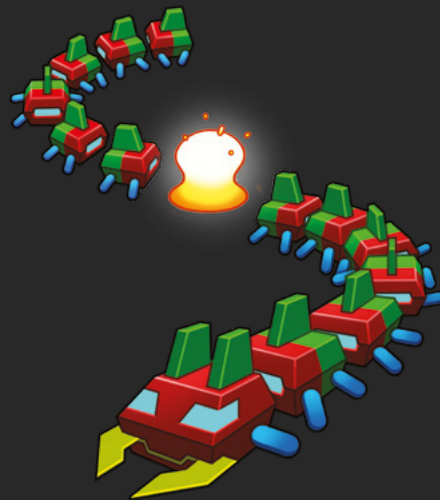
Don't dispense with household cylinders.
Rosie Hattersley suggests new uses for
discarded cardboard tubes

Rosie Hattersley

🐦 @RosieHattersley

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

You may have used the last of the loo
roll, wrapping paper, or cling film,
but there's no need to chuck out
the cardboard cylinders they came
on. Cardboard is hard to come by right
now, and there are plenty of ways to
make use of it.

As with many easily recycled household products,
gardeners have been quick to adopt no longer
needed kitchen or bathroom cardboard tubes to
use as makeshift pods. They are ideal containers
in which to start off for your seedlings, or for
separating them when thinning them out. They can
also be planted out directly into the soil when the
plants have hardened off without the need to repot.

"THE CYLINDRICAL SHAFT PROVIDES A USEFUL GUARD WHEN YOU NEED THEM TO QUICKLY PUT DOWN ROOTS"

The cylindrical shaft provides a useful guard when
you need them to quickly put down roots – just the
thing for carrots and other root vegetables.

The speed at which cardboard breaks down also
means it's ideal to incorporate into your compost.

Rip up loo roll tubes and add them to a compost mix
of soil, vegetable matter, foliage, crushed eggshells,
and grass cuttings for a well-balanced compost.

You can gently bend and reform cardboard
tubes, or cut and splice them end-to-end to do their
duty discreetly running cables around a room, or
protecting sharp edges from soft bodies. Cut them
in half lengthways to use them to separate items
in a drawer: pairs of socks; diodes and resistors;
different coloured buttons – whichever meets
your needs.

Water, of course, is the enemy of cardboard's form
– after all, the household tube is made from pulp
(invented in Germany back in 1843 when the first
wood grinding machines were used to pulp wood
for paper), and the flimsiest versions betray their
origins as soon as they get damp. The flip side is
the malleability, which means lengths of tube can be
subtly curved and manipulated to combine with other
objects. These aspects are often exploited when
cardboard tubes are called on for an art project.

The rigidity of tubes varies a great deal, depending
on what it is designed to hold: heavy cables or
gossamer-thin tissue. Bear this in mind when
planning an upcycling project in which you need
malleable, easy-to-cut, or load-bearing tubes, as per
the hammock project featured here.

TRASHFORMATION HAMMOCK

Virgin cardboard is used in all sorts of structural art installations and architectural designs, but there's mileage in reusing what you've got.

As you'll see from the YouTube video (**hsmag.cc/TrashBoarding**), devoted to the ideas presented by attendees at TransfoLAB BCN's so-called Trashboarding Festival, thicker cardboard tubes make for excellent load-bearing makes.

A rippling wave of tubes bound side by side along their lengths is used to create a flexible structure that can be used as an impromptu wall or, as the makers in this instance have done, strung up to form a hammock or suspended chair.

The tubes used in this project are the sort that you might come by if you have something fragile delivered, if you buy lino or fabric by the roll, or order printed artworks. The tubes are secured with alternating thick woven strips of fabric, with ropes to hold up the finished furniture hammered into the tubes at each end.

Information about some of the other makes reported by Fair Companies can be found using this link: **hsmag.cc/FairCompanies**. →

Project Maker
TransfoLAB BCN

Project Link
hsmag.cc/TrashBoarding

Left ⇐
Recycled cardboard tubes form a relaxing lounger suspended from a warehouse ceiling

CARDBOARD TUBE CABINET

Project Maker
THEGUYMASAMATO

Project Link
hsmag.cc/TubeCabinet

—

T

his nifty storage cabinet came about because its maker wanted to keep his son's most-used Lego bricks on hand, but to store them neatly when not in use. Already having access to lots of thicker tubes used to mail items to customers, theguymasamato was able to sketch out a design and work out exactly how many 455 mm-long cardboard cores he'd need. In all, the project used 17 metres' worth to create a three-drawer cabinet. His detailed Instructable also reveals how to stop a row of circular tubes rolling away: skewer them with

aluminium poles that you've pre-drilled with holes. Tubes were cut at angles to slot together to form the drawer frames, with slots cut into their lengths to accommodate side panels of his box-shaped drawer base.

A masterclass in precision, the Cardboard Tube Cabinet project lives up to its billing as a low-cost, lightweight solution to show how to store small items you frequently need to hand. Offcuts of cardboard had circular bases glued to them, and became in-drawer dividers to keep items from getting mixed up and harder to retrieve.

Right ⬥
The Cardboard Tube Cabinet stores small items conveniently close by

CRAFT PUPPY PUPPET

A n ideal project to entertain a young maker, this cute marionette uses just two cardboard tubes, with one of them being cut up to form the puppy's ears and head. First, paint the outside of the two tubes and let them dry out. Cut one tube two-thirds of the way along and unwrap it. To make the ears, draw an elongated figure of '8' from the unfurled piece of tube and cut it out as a single piece. Glue them to the top of the head piece. Glue together two wooden ice lolly sticks at right angles, and make holes at each end. This will be the controller. Use coloured string knotted at one end and then fed through punch holes as legs. Pieces of plastic straws make ideal paws. Use another piece of string to attach the body to the head piece. This needs to be long enough to pass through the head piece and attach to one of the controls. Add another length of coloured string at the back of the puppy's body, attaching this to the opposite side of the controller.

Add a bendy straw as a tail, and either paint on or stick on googly eyes.
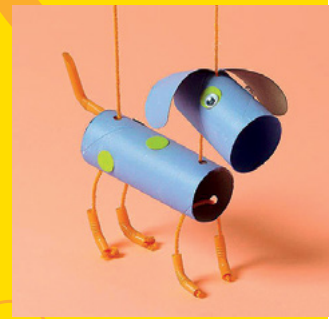
TELEIDOSCOPE

O ne of the most instinctive things to do with a newly unneeded toilet roll tube is to put it to your eye and peer through it. Crafting-for-kids website Babble Dabble Do extends this idea by making the sights to behold intriguing and memorable. Depending on whether you want to add an end cap, the result will be a kaleidoscope or, if left open-ended, a teleidoscope. Such visual trickery usually involves a triangular glass prism. Making your own would be hard to cut accurately, so crafter Ana suggests using mirrored plastic sheets or 'reflecting paper'. The interior of the kaleidoscope needs these plastic strips to be the width of the sides of an inscribed triangle or oblong, and the same length as your cardboard tube. Tape the resulting shape's sides together on the inside. Cover the exterior of the cardboard tube in paint, coloured paper, or card to disguise its origins. The pretty patterns that kaleidoscopes are known for can be made from translucent paper arranged into random patterns and tacked up against a window so that the light glows through. Create a second, matching tube if you want to view the display through both eyes at once. ▢

Building a new games controller

PART 01

Experimenting with slide potentiometers

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.
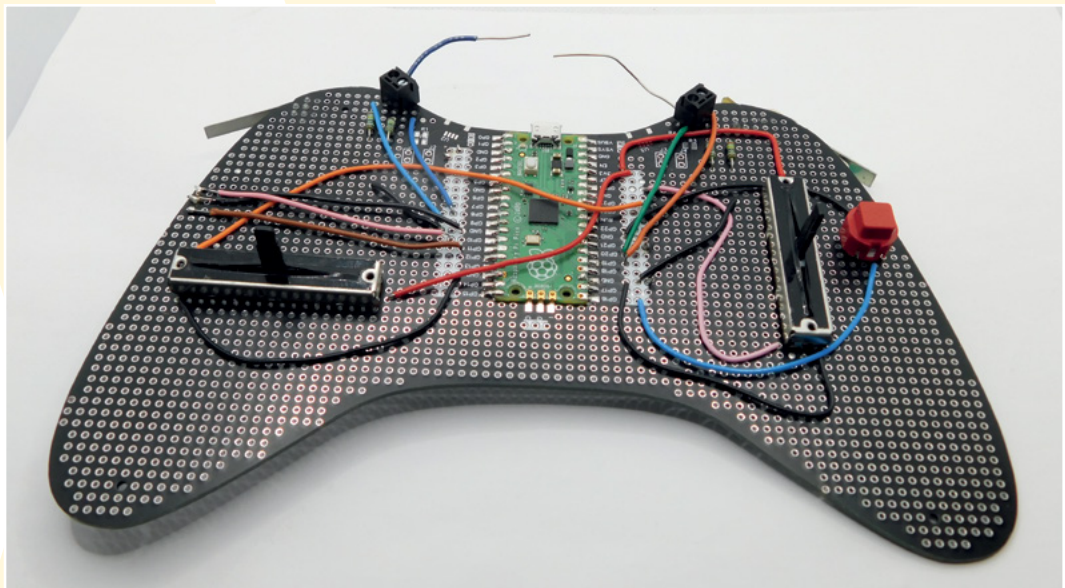
Games controllers come in many forms. Joysticks, joypads, keyboards, mice, and more can all be pressed into service. Regular readers will know that I've long been interested in how we control games. I'm actually not a huge gamer, but I find games a really fascinating area of human-computer interaction.

Thanks to CircuitPython, it's easy to create a circuit that acts like a USB device such as a keyboard, mouse, or games controller. You can add whatever inputs you like to make all sorts of strange and curious controllers. However, a problem is getting things in the right form-factor. No one likes to hold a breadboard, and I've had mixed results when I've tried to mount components in wood. It's time to get serious, so I designed a PCB which was just a matrix-style protoboard and mounting for a Pico. I got a few printed off, and set about seeing what weird and wonderful creations I could come up with.

This month, I've been playing with Jay-D – a DJing table (see issue 46 for details), and this got me thinking – why don't we use slide potentiometers as input devices? Only one way to find out!

I ordered a pair of 100 kΩ slide linear potentiometers. The linear part is important as some slide potentiometers are logarithmic (this is useful in some audio applications). I wanted to use this controller for racing games, so the plan was to use one potentiometer for throttle and one for steering.

I also needed some buttons. With the thumbs taken up with the potentiometers, the best place for buttons is on the shoulders. I hot-glued a pair of microswitches on here and also added a regular push-button next to the right-hand slider, as I thought this might be used less (just push it up to maximum, right? If the aim of the game is to get around a course as fast as possible, you don't need anything other than maximum throttle).

This seemed like it might not be quite enough inputs, so I added four screw terminals which can be used to add touch-sensitive inputs. I haven't actually

worked out the best way of using these yet. They're quite sensitive and can be used to detect how hard you're touching an object as well. I'll continue to experiment with these.

CircuitPython handles USB human interaction devices (HIDs) quite well, but by default only supports mouse, keyboard, and consumer control devices because gamepads don't always work well across different operating systems. Fortunately, there is some example code to make gamepads work. I won't go through it fully here, but you can see my code at **hsmag.cc/ControllerCode**. There are three parts: the **boot.py** file which enables the gamepad USB type (this has to go in the **boot.py** file as this is run before USB is set up); the gamepad module, which makes sure the data is formatted correctly; and the gamepad code itself.

The proof of the games controller is in the game. There are, of course, many racing games out there. Personally, I have a soft spot for SuperTuxKart (you can download it for free at **supertuxkart.net**).

The results were… OK. The strangest thing is there's a sticking point in the middle of the potentiometers. This is there to help you find the middle, but it takes a bit of a push to get over it. Initially, I had trouble with oversteering, and I really struggled to keep in anything resembling a straight line. I tweaked the code (which takes the 0–65536 value from the analogue pin and converts it to a -127 to 127 value that the gamepad needs) to be more sensitive at a lower level.

This is done with:

```
def get_range(pin):
    if pin.value < 32768:
        multiplier = -1
    else:
        multiplier = 1
    return int(((pin.value - 32768) / (2895))**2) *
multiplier
```

With this minor change, the controller is much more pleasant to use.

One thing about this controller that really stands out is how nice the microswitches feel. They're solid, positive, and have a satisfying click.

I'm going to try out some other designs for controllers with different styles of input.

I'm also really pleased with how the PCBs work. They're sturdy enough to be easy to hold. They're a bit bigger than some controllers, but that gives me space to mount more hardware and not have to worry about miniaturisation. However, there are some definite mistakes in version 1 of the PCB:

- The solder mask means you can't run 'traces' of solder joining the different pads. Having larger pads may also help this.

- I added traces for Stemma QT connectors at the top, but these haven't been useful. In fact, I haven't populated them on my boards. If I want I2C connectors, I can add them manually. Stemma QT doesn't save any hassle when you have to solder it up anyway.

- The mounting holes are too few and too small to be useful. For the next iteration, I'll go up to M3 holes and have more of them.

- More 3.3V connections. Three means one per analogue, which is a minimum, but more would definitely be useful. Maybe five or six. There's no good reason not to extend that row out.

Experiment 1 with PCBs for games controllers is definitely a success. I wanted to test out some different input types and see if a protoboard controller works. Next up, I want to explore some more different input types, including touch.

Is there anything you'd like to see on this board? Let me know – **ben.everard@raspberrypi.com**. □
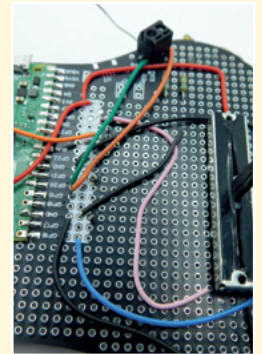
Above ◈
There are three connections for each pin (highlighted in white)

Below ◈
To make a connection, poke wire through a hole, bend it over, and solder it straight onto the component's lead
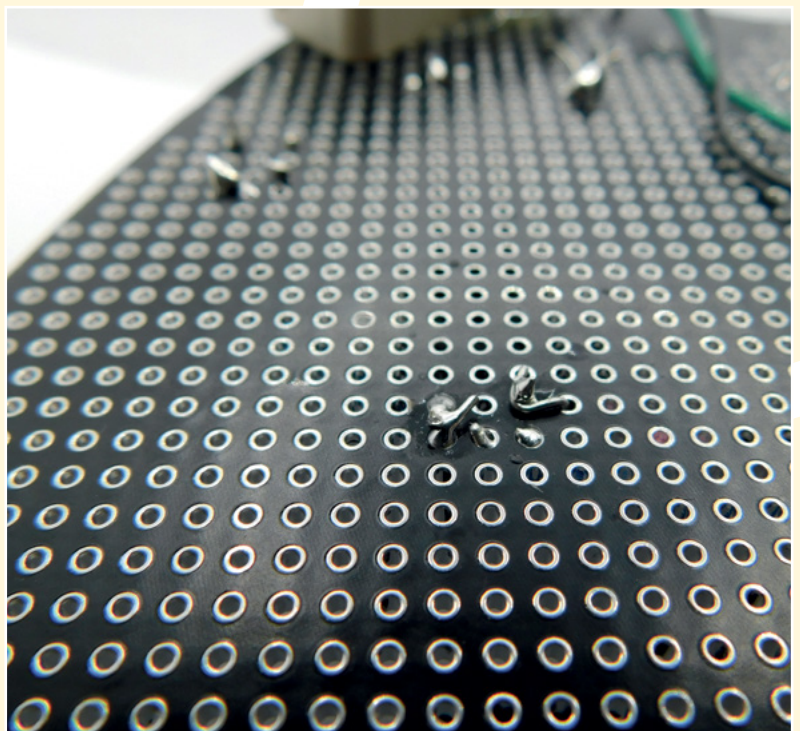
HackSpace
TECHNOLOGY IN YOUR HANDS
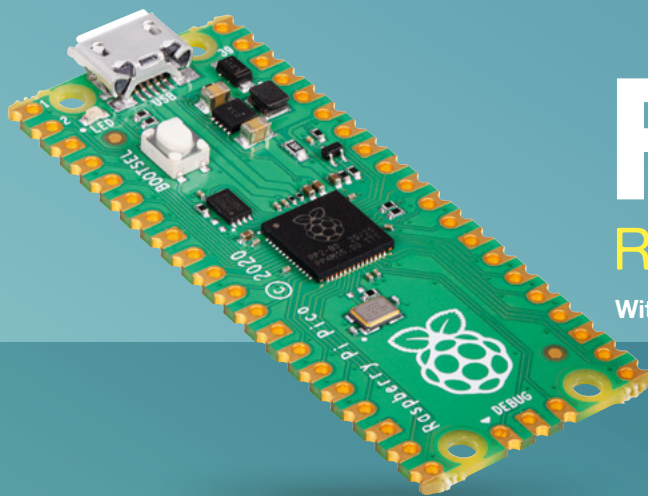
FORGE

HACK | MAKE | BUILD | CREATE

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

Touch inputs with Raspberry Pi Pico

Poke data into Pico – no buttons needed
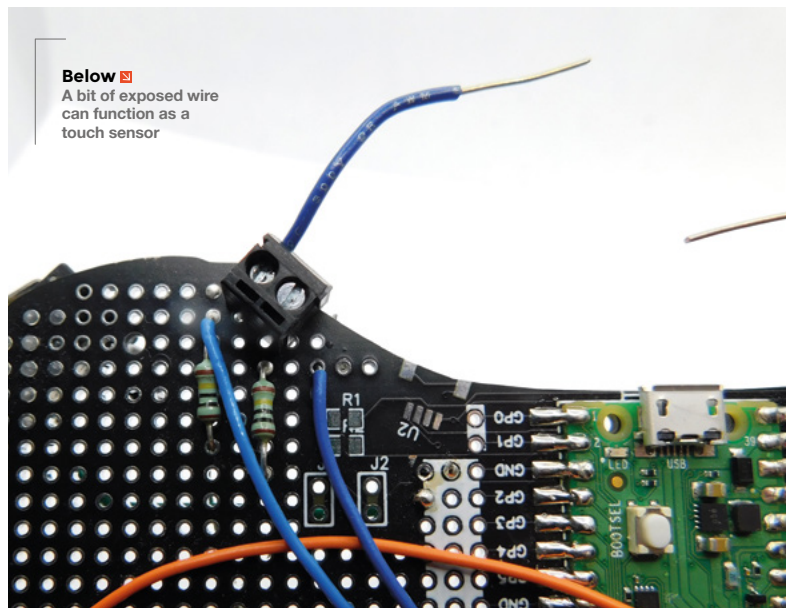
Below ↘
A bit of exposed wire can function as a touch sensor

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.

Raspberry Pi Pico doesn't have capacitive touch hardware on board. However, that doesn't mean we can't use capacitive touch. To understand how, we first need to learn a little about how capacitive touch works.

All electrical conductors have some amount of capacitance – basically, this is an ability to store electrical charge. If you put one end of a wire into a circuit without the other end attached, you might think that no electricity flows, but actually, a little does. It charges the wire and this small charge will remain stored in the wire until it's discharged. How much charge it stores depends on the size of the wire, what it's made from, etc.

Capacitive touch sensing works because your body is a conductor. If you connect a wire to a circuit that you're holding the end of, it'll be able to store slightly

more electrical charge than it would if you weren't holding the end of it. The question is, how do you read the value of this charge?

The easiest way is to discharge it and see how long it takes to discharge. For this, we need a way of discharging the touchpad – basically, a resistor connected to ground. As the amount of capacitance in the wire plus a person is very low, we want a very large resistor to slow down the discharge to a speed we can measure. We use a 1 MΩ resistor for this.

The pseudo-code to tell if a pad is currently touched is:

- Set pad to output and drive it high
- Turn pad to an input
- Continue to read pad state until the pad goes low
- Return the amount of time this took

The higher the capacitance of whatever is attached to the pad, the longer this will take (though it should only be a tiny fraction of a second). If it's above a certain threshold, then we consider the pad touched. If not, then we don't.

The good news is that CircuitPython can do all of this for us. All we need to do is attach the external 1 MΩ resistor between a pad and ground. We can then read the state of the touch sensor with:

```
import time
import board
import touchio


touch_pad = board.GP9


touch = touchio.TouchIn(touch_pad)


while True:
    if touch.value:
```

```
        print("Touched!")
    time.sleep(0.05)
```

The touchio module is built into CircuitPython, so you don't need to add it. It should work on all boards, though on some, you won't need the external pull-down, and on others, it will only work on some pins; so if you have trouble, look at the documentation for your board. On Pico, you can use it on any pin as long as you add the resistor. We've used GP9, but change this to something different if that works for you.

MAKING A GAME

Let's make a simple reaction game. In this, we'll light up an LED, then the aim is to touch the touchpad as soon as possible after the LED lights up.

This is done with the following code:

```
import time
import board
import touchio
import digitalio
import random

touch_pad = board.GP9
led_pin = board.GP25
led = digitalio.DigitalInOut(led_pin)
led.direction = digitalio.Direction.OUTPUT

touch = touchio.TouchIn(touch_pad)
led.value = False

time.sleep(random.randint(5,10))
led.value=True

start = time.monotonic()

while not touch.value:
    pass

score = time.monotonic() - start
print(score)
```

We're using the internal LED on Pico, so there's no need to add extra hardware.

We use the following line to pause the execution for a random amount between five and ten seconds:

```
time.sleep(random.randint(5,10))
```

There's then a `while` loop that does nothing but pause until `touch.value` becomes true.

This code is a bit simplistic because it just finishes at the end of the game. You could get it to loop around again if you wanted to make a game that played over and over.
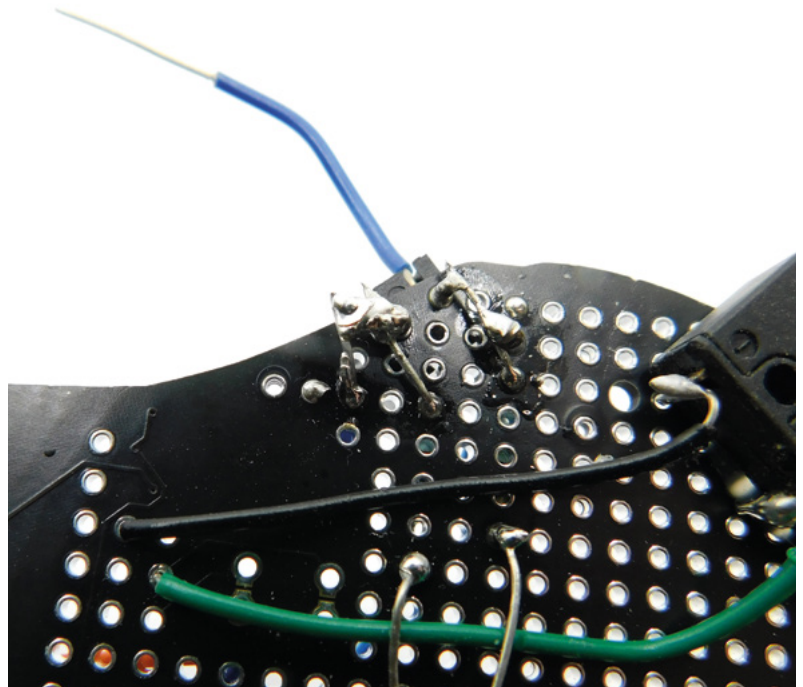
THRESHOLDS

We mentioned at the start that this is just reading the capacitance of the pad and whatever it's connected to. However, how does CircuitPython know if a capacitance is enough to be a touch? After all, if you're attaching different hardware to Pico, you might get different capacitances.

It's done with a simple thresholding. You can see the actual value returned by the touch sensing with:

```
import time
import board
import touchio

touch_pad = board.GP9

touch = touchio.TouchIn(touch_pad)

while True:
    print(touch.raw_value)
    time.sleep(0.05)
```

When you first create a touch input, CircuitPython checks the raw value and sets the threshold to that value plus 100. However, in some cases, that might be too sensitive or not sensitive enough. You can manually adjust the threshold with the `touch.threshold(value)` function.

That's all you need to know to start creating your own touch-sensitive projects. A touch sensor can be absolutely anything that's conductive. It can be a bare wire, a pad on a PCB, or something made out of tinfoil. You can be as artistic as you like. ▢
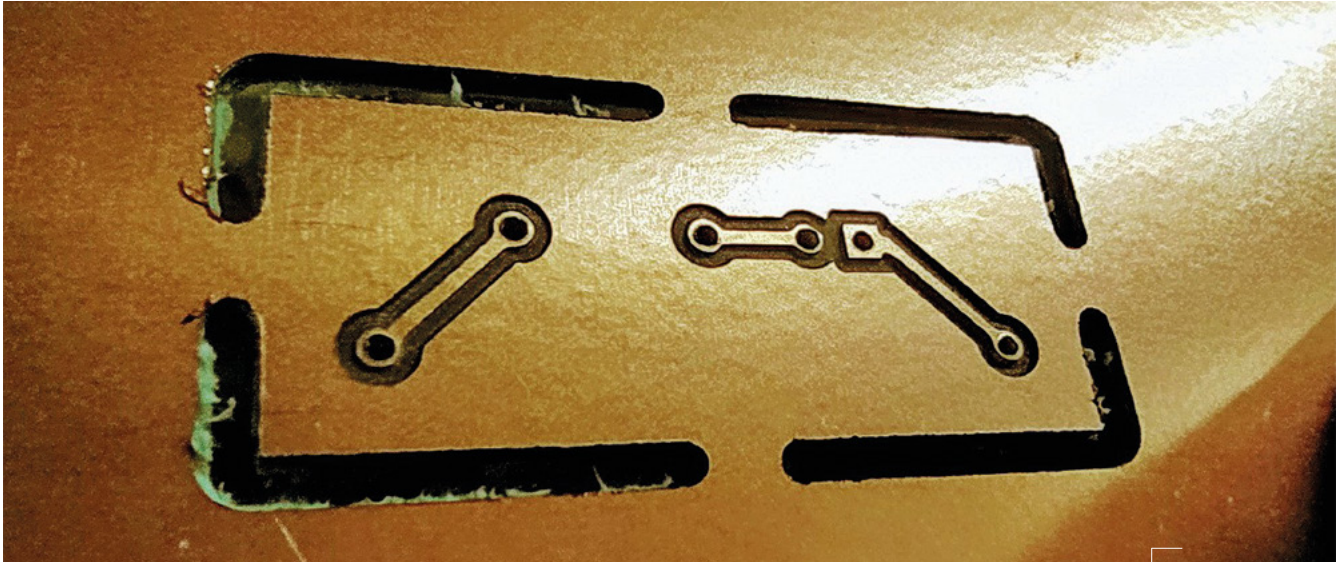
Above
Our completed project – a small milled PCB

PCB milling on the 3018 with FlatCAM

A popular use for the CNC 3018 Pro is milling printed circuit boards. In this final part of the CNC 3018 Pro series, we get started with this process

Jo Hinchliffe

🐦 @concreted0g

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

Milling a PCB has some pros and cons as an approach to making a PCB, and it's worth discussing these before we dive into the tutorial. The pros mostly centre around that it's cheap and quick, with no waiting for a PCB fabrication house to deliver your design to you. The cons are quite a longer list, with some technical limitations, most of which can be mitigated through effort, but they may put you off. As an example, it's problematic, but not impossible, to do double-sided PCBs, and so most people stick to single-sided PCB milling at first. If you want to create vias that connect the front and back layers of a board, you have no way of through-plating them, so you need to solder in a wire connector for each one. PCB milling doesn't create a solder mask. Finally, you will probably make some design changes on a board destined for PCB milling, which means if you did want to have them professionally fabricated then you may need to rework the design. With so many PCB fabrication services available these days at affordable prices, it's worth considering them over home milling! We tend to use PCB milling either when we need something small making quickly, or when we need a single large PCB that would be expensive to have fabricated. That said, let's get started!

We aren't going to concentrate on the board design in this tutorial, as we covered KiCAD in a two-part tutorial back in issues 17 and 18. As our small example PCB for this tutorial, we laid out a very simple circuit. It consists of an LED, a current-limiting resistor, and two connector pads for power and ground. Note in **Figure 1** that we have created the tracks for the PCB on the back copper layer. This is because on the end-product the components will sit on the blank fibreglass side of the board and the milled copper will be underneath. We also made the track widths wider than we possibly would for a fabricated PCB, with the tracks being 1 mm wide. Similarly, we made the pads on the components and connectors a little larger than stock;

we made the pads 2 mm wide. In our experience, this can make drilling holes more successful – with smaller pads, sometimes when we drill, the surrounding tiny amount of copper will get pulled off the board.

Having laid out our simple schematic and then the PCB, we added an edge cut layer and created a rectangular cut-out for our board. We don't need to worry about silkscreen items or layer floods, as we can't really apply these in the milling process!
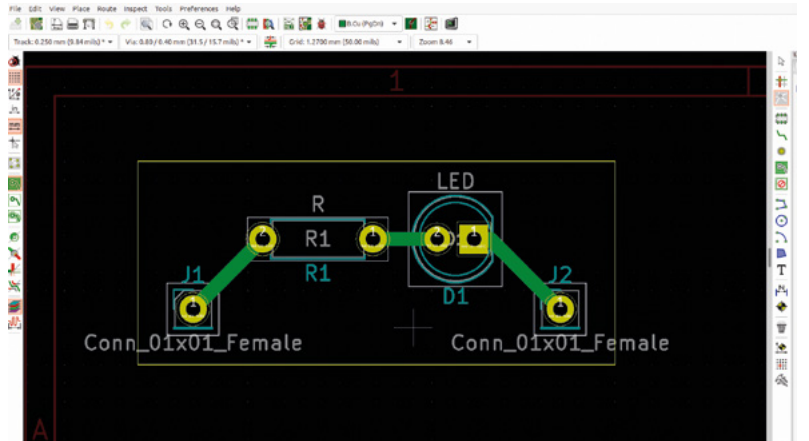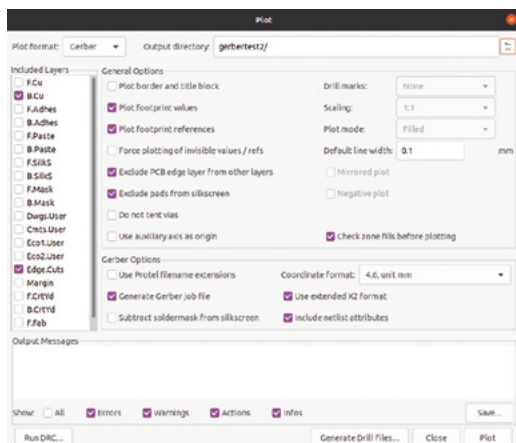
FlatCAM, the software we use for creating the toolpaths and operations for PCB milling, imports Gerber files for the board edge cuts and the board track and pad geometry, and it also uses Excellon files to create drilling operations. We'll create these files next, but we also have the example PCB files available for download here: **hsmag.cc/issue47**

FLATCAM PARTICULARS

In KiCAD, in PCBNew, go to File > Plot and in the dialog box you should see an 'included layers' list. Uncheck everything apart from the bottom copper layer called 'B.Cu' and the 'Edge.Cuts' layer. Leave all the other items at their default setting and click Plot. You should now have two Gerber files, one for each layer in your KiCAD project folder (**Figure 2**). In the same Plot dialog, you can now click the Generate Drill Files button and, in the new dialog, leave everything at its default setting and click the Generate Drill File button to create a drill file in the project directory.

We've downloaded and used the beta version of FlatCAM 8.994 on a Windows machine, as we had some challenges getting it to run on Linux. FlatCAM is

a very powerful and quite complex piece of software, and it has a huge feature range. As such, we've simplified our approach for this tutorial to hopefully help get you up and running. Beyond our simple project, FlatCAM is definitely worth studying as it has some amazing features.

The first job in FlatCAM is to import our files. Click File > Open and scroll down to Open Gerber; select this and navigate to our B.Cu Gerber file and open it. Repeat this for the 'Edge.Cuts' Gerber file. Finally, repeat this process but using the 'Open Excellon' file to import the drill file we created earlier. You should now see your three imported files rendered in the Plot Area window and listed in the file tree (**Figure 3**).

You'll probably notice that in the plot, although the files are aligned correctly to each other, they are possibly a long way from the plot area origin point. We need to set the origin point close to the files as this will translate to the zero point on the CNC 3018. On the toolbar, there are two icons that look like a cross-hair target. Hovering over them, you will see tooltips, and you need to select the icon that says 'Set Origin' as the tooltip. With that selected, click a point to the lower left of our files in the plot area to set the origin. Later, we are going to use a 1.5 mm tool to cut out the board, so make sure to leave a little room for that.

Highlight the B.Cu Gerber in the file tree and then switch to the Properties tab. In the latter, →
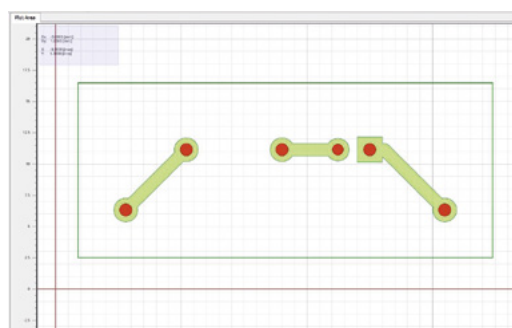
Figure 1 ◈
Our very simple PCB layout for a resistor and LED and power wires

Figure 3 ◩
We can see our imported files in both the FlatCAM file tree and the preview window where we set the origin point

Figure 2 ◈
We are using the default KiCAD settings for our Gerber and Excellon files, and these work well in FlatCAM

YOU'LL NEED

◈ **CNC 3018 Pro kit**

◈ **Z axis probe,** or some wire and crocodile clips

◈ **Some copper-clad PCB board**

◈ **1.5 mm PCB cutter, a 1 mm PCB drill, and a 30-degree V bit**
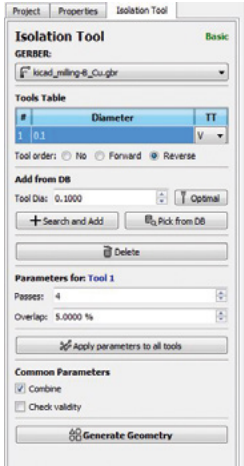
click the Isolation Routing button and then you can switch to the Isolation Tool tab. We aren't going to explore adding custom tools, so leave the tools table section set as its default, tool number 1. We are going to change the parameters for tool 1, however, so increase the number of passes to 4 and change the overlap to 5% (**Figure 4**). Finally, click the Generate Geometry button and you should see some red lines appear around the pads and tracks marking the toolpaths that the V bit will mill away in the copper-clad board.

If you now return to the Project tab, you should see that a geometry object has appeared in the file tree. Highlight this and move to the Properties tab. In the latter, look at the Parameters for the 'Tool 1' section. Notice the 'Cut Z' option; this is greyed out and will read 0. In the default tools, the depth of cut for isolation routing is calculated using the tip diameter and the V tip angle. Instead of creating custom tools, we are going to save time and cheat a little by adjusting the tip diameter value to get a depth of cut we require. Our CNC 3018 came with some 30-degree V cutters (**Figure 5**), so we set the V-Tip Angle setting to 30. We then adjusted the tip diameter down to 0.04 mm, giving us a calculated depth of cut of -0.1120 mm, which should be plenty to cut through the copper on our board.

With that done, scroll down the tab and click the Generate CNCJob Object button. Moving back to the Project tab, you will see that we now have a file in the file tree listed under the 'CNC Job' heading. Highlight this, then move to the Properties tab

and click the Save CNC Code button to save a .nc G-code file.

Next, we are going to repeat a similar process to create our edge cuts. Highlight the Edge.Cuts Gerber in the file tree and move to Properties. Click the Cutout Tool button and move to the Cutout Tool tab (**Figure 6**). Change the Tool Dia setting to match the tool you will use; we used a 1.5 mm PCB cutting tool. We checked that our Cut Z parameter was set to a value that would cut through the thickness of our board. Our board measured 1.65 mm thick, so we set this to -1.8 mm. This of course means we will cut through and slightly into the waste board underneath. We checked the tick box for 'multi depth' and set the value to 0.6 mm; this means the edge cut will be made in three passes with a 0.6 mm depth of cut each time. We left the margin at 0.1 (this adds a small offset from the edge geometry and wasn't important for our test project) and we turned the gap size down to 3 mm. Gaps are equivalent to the tabs we have added in previous parts of this series that stop the work from being ejected from the bed.
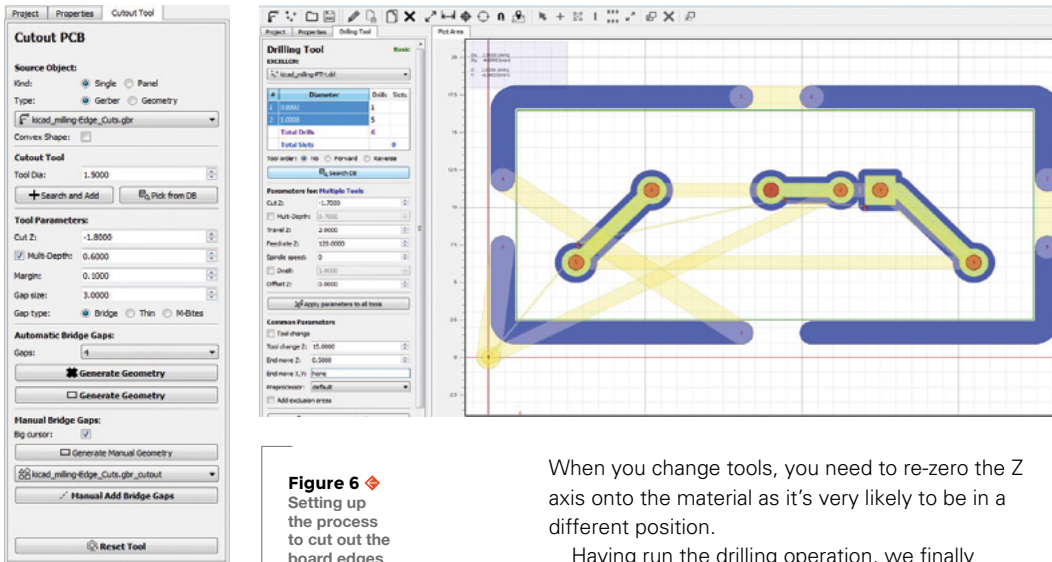
DRILLING AND MILLING

We left everything else at the default settings and then clicked the Generate Geometry button, and the geometry red lines appeared in the plot area preview. In the Project tab, select the new geometry object and move to the Properties tab. We left everything at its defaults, apart from we reduced the feedrate XY to 120 mm/min and the Z feedrate to 60 mm/min. These are nice and slow for an initial experiment, but with experience, you'll likely increase them a little. We then clicked the Generate CNCJob Object and saved the G-code file in the same way as we did the earlier file for the B.Cu layer.

All that remains are the drilling operations. Select the Excellon file in the file tree, then move to the Properties tab and click the Drilling Tool button. We again have slightly cheated in using the default tool as it essentially creates a vertical Z axis movement and, as such, the hole size in reality is governed by the diameter of the cutting bit you use. In the Drilling Tool tab, we made sure our Cut Z was set to -1.7 and we adjusted our feedrate down to 120 mm/min (**Figure 7**). We left our Travel Z value at 2 mm as, with a slow Z feedrate, lifting the tool high adds a lot of time to the drilling operation. With that set, we clicked the Generate CNCJob Object button and then again highlighted it in the file tree, moved to the Properties tab, and saved the G-code.

We are now ready to mill our PCB! We've covered all the aspects of this in previous parts of the series,

Figure 6 ◈
Setting up
the process
to cut out the
board edges

Figure 7 ◈
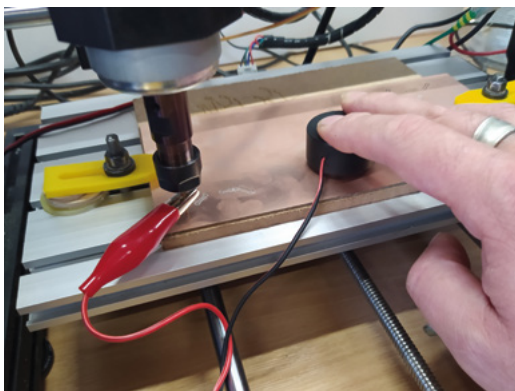Setting up the
drilling operations

so refer back to them if needed. We ran our isolation routing file first, then the drilling, and finally the edge cut operation. It's important, especially with the isolation routing operation, to set the tool Z axis zero point accurately: as the depth of cut is only tiny, any small error will make a large difference. Using the Z axis probe we set up last month is very handy, and as the PCB material is conductive, you can place the probe block upside down on the material and then zero the tool directly onto the material (**Figure 8**).

Having completed the isolation routing, we then jogged the machine back to the zero point using the 'Safe position' button in Candle that we set up earlier in the series with a custom command. We raised the Z axis a little to give us room to swap the tools. We used a 0.75 mm drill/mill tool to create our holes through our pads; this is fine for the components we were going to fit to this project, but some through-hole components may of course need larger.

Figure 8 ◈
Using the Z probe upside down to zero
the tool directly onto the PCB board

When you change tools, you need to re-zero the Z axis onto the material as it's very likely to be in a different position.

Having run the drilling operation, we finally swapped the tooling for the 1.5 mm PCB cutter, re-zeroed, and cut the board edges. This worked perfectly, but we realised that having four 3 mm tabs was overkill for this tiny PCB and we could certainly have reduced it to two tabs. A little work with a rotary tool and our board was free. It can be a good idea to check over the board with a magnifier to make sure that copper has been removed where it needs to have been. Also, you can use a multimeter with a continuity buzzer to check everything is routed successfully. Finally, we soldered up our completed test board just to prove it all works OK! ▫

EDITING **THE CODE**

Before we get excited and run our G-codes on the machine, we need to make some edits to the G-code files. As the files stand, they have not got a spindle speed set and they also have some codes for tool changes where the G-codes will pause and lift the Z axis by 15 mm, which may cause your CNC 3018 Pro to crash. Open each G-code file in turn and delete the code you can see in the image and add an 'S1000' to set the spindle to full speed after the 'M03'. Also, scroll to the very last line of each of the G-code files and remove the G0 Z15 lines before re-saving the edited files.

```
G01 F120.00

M5
G00 Z15.0000
G00 X0.0000 Y0.0000
T1
M6
(MSG, Change to Tool Dia = 0.1000)
M0
G00 Z15.0000

M03
G01 F120.00
G00 X5.7912 Y7.3093
G01 F60.00
G01 Z-0.1120
G01 F120.00
```

QUICK TIP

As the cutting forces in PCB milling are quite low, often people will stick their material to the work bed using double-sided tape. This means you may not need to add work-holding tabs.

Isomorphic keyboard: hex keys

Use Raspberry Pi Pico to make an isomorphic music keyboard with hexagons

MAKER

Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies, Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

You'll Need

- Stripboard **magpi.cc/ stripboard**
- 102 tack switches 4-pin 6×6×5mm **magpi.cc/ tackswitch**
- Hex Keys CAD diagrams **magpi.cc/ pibakery**
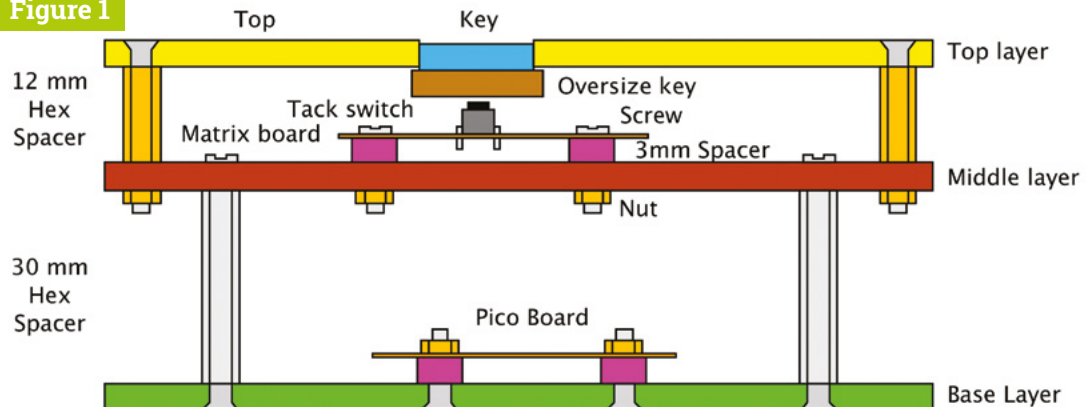- Access to laser cutter

▶ **Figure 1** Our double box concept

One alternative layout to a conventional piano keyboard is an isomorphic keyboard. Here, the notes are laid out so that adjacent keys differ by only a semitone. It is said that such a keyboard is easier to learn to play in any key. This month, we show you how to make a USB interfaced MIDI keyboard with an isomorphic layout.

01 Isomorphic keyboards

While the principle of adjacent keys differing only by a semitone is simple enough, there is considerable variety in exactly how they are laid out. The arrangement of keys can be on a square grid, or an offset grid; the keys themselves can be square, round, or, our favourite, hexagonal. There are a number of different layouts that detail exactly what keys are where; these have names like Gerhard, Park, Maupin, Wicki-Hayden, Harmonic, Janko, C-system, and B-system. We are going to use the Harmonic type that was used by the now defunct C-Thru Music company.
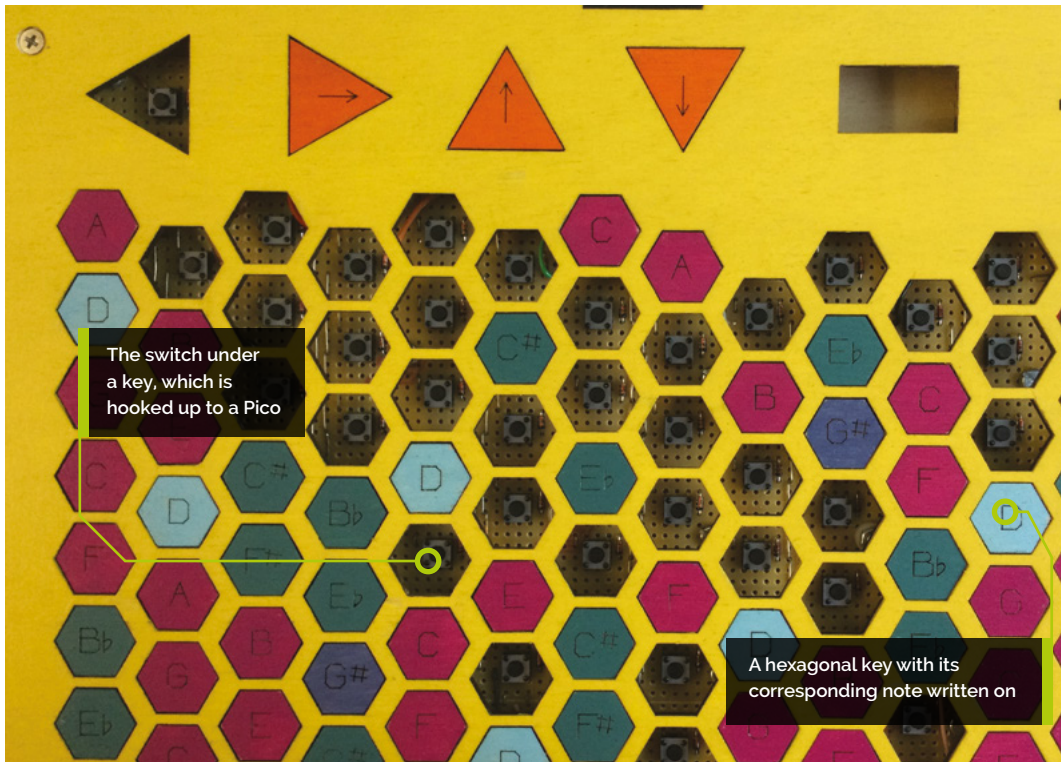
02 The enclosure

We came up with the concept of a two-layer box – this is shown in **Figure 1**. The top surface is where the keys are. They are cut out and press down on the tack switches underneath. To stop them falling out when the box is turned upside down, each key has an oversize version of itself glued to it. The tack switches are mounted on a large piece of stripboard, supported by 3mm spacers screwed to the middle level. Underneath this is a Raspberry Pi Pico board, supported on five 30mm pillars between the base and middle level.

03 Scan the Matrix

A scanning switch matrix is a way of reading a lot of switches from a few inputs/outputs. **Figure 2** shows the schematic of a small 3×4 matrix. There are many variations on this, but we have chosen to use a column drive with a row input. A walking zero signal is applied to the columns, and pressing a key connects a column to an input. As you know what column was low when an input was low, you

Figure 1

The switch under a key, which is hooked up to a Pico

A hexagonal key with its corresponding note written on

know what key was pressed. We will be making a 15 column by 7 row matrix for this project.

04 The laser files

We have started using the software LightBurn to produce our laser cutter artwork, and all the files can be found on our GitHub page. If you use another system, then you can get a full version of this software on a 30-day free trial to use or convert the files. Alternatively, we have also supplied SVG files for each sheet you have to cut. The top panel also includes vector engraving on the keys – this has to be done first at a lower setting so as not to burn all the way through.

05 Enclosure construction

Figure 3 shows the construction of the two-layer box. Note how the side with only a 1.5 mm tab sticking out goes to the middle layer of the box, and the 3 mm tabs on the other side go to the top and the base. We assembled and glued the sides into two rings, and then we reinforced each corner with a small length of half round dowel fixed with superglue. Then we plugged up the bottom end with Blu Tack (tack putty) and dribbled some Gorilla clear glue down the joint for added rigidity. See **Figure 4**.

06 Painting

We have come up with a new way to paint plywood boxes that doesn't show brush marks, by using a half-inch foam brush and normal emulsion paint. This allows us to use paint from the wide range of match-pot colours, and gives a nice flat finish. To keep things clean from finger marks, we add three coats of a water-based clear matt varnish. We used yellow for the top and base, and matt black for the sides and visible edges of the box.

07 Painting the keys

Figure 5 shows the colours we used to paint the keys. They keys can be painted any colours you like, but keep the groups consistent for the
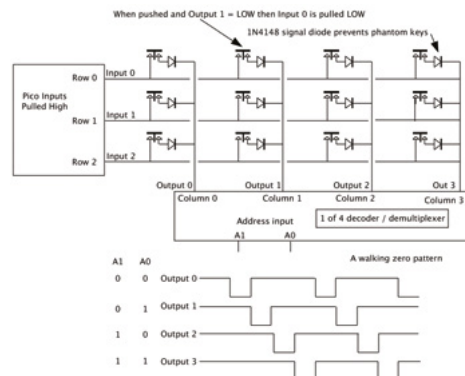
Figure 2

◀ **Figure 2** A 3×4 switch matrix

Figure 3

> **Once painted, you can glue the keys to the oversized key blanks**

best playing experience. Again, three coats of matt varnish are applied to each key. Once painted, you can glue the keys to the oversized key blanks. We used a small drop of superglue in the middle of each and pressed down for ten seconds. Try to get them centralised; if not, this can be corrected with a disc sander. The top and bottom flat of each key blank should be about 1mm from the edge of the actual key.
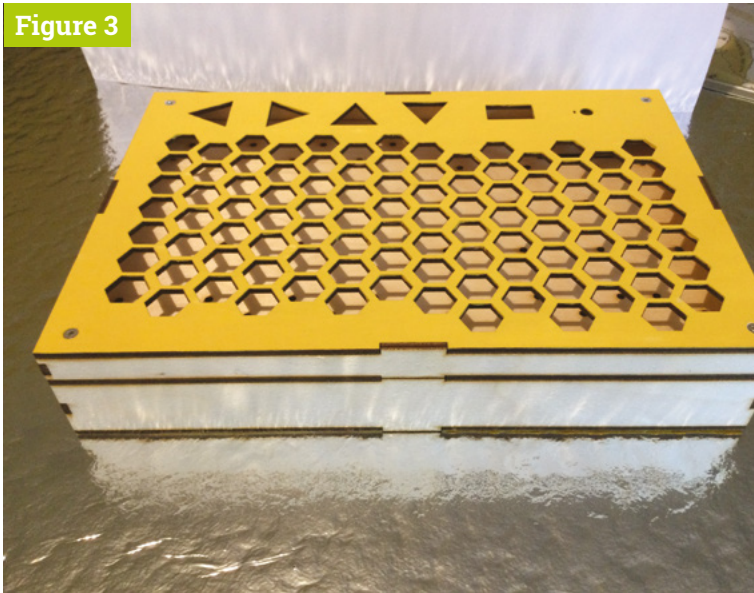
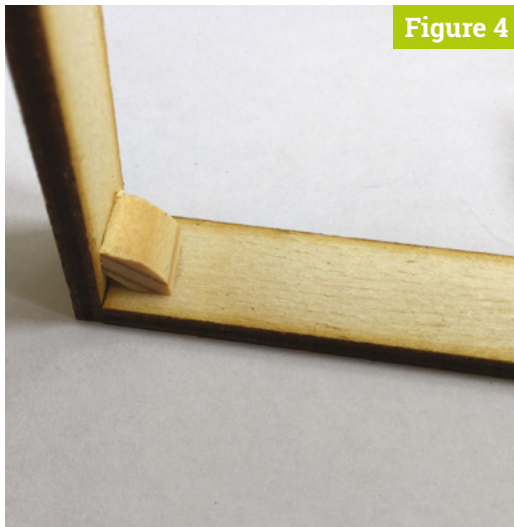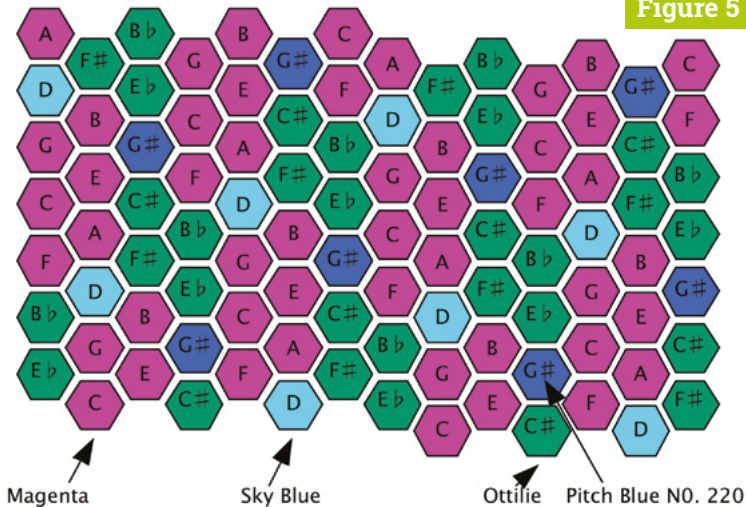▲ **Figure 3** The assembled double box

Figure 4

▶ **Figure 4** Reinforcing the corners of the two rings formed from the sides

▼ **Figure 5** The colours we used for the keys – magenta and sky blue are the white keys on a piano

Figure 5

Magenta · Sky Blue · Ottilie · Pitch Blue N0. 220

08 The stripboard

We need a piece of stripboard 101 holes long by 58 strips high. Unfortunately, we couldn't find one, so we had to make one out of three pieces of board. Fortunately, this could be done by buying just one 119 by 455mm sheet of stripboard. The maximum height of this board is 46 strips, so we cut out a 101 hole by 46 strip high main piece, a 71 hole by 12 strip, and a 30 hole by 12 strip piece. We also got a 58 hole by 8 strip piece for the triangular control button board from the same piece of stripboard. See our GitHub page for more notes about joining these together.

09 The Matrix

Figure 6 shows the first five columns. There are two additional diagrams showing the rest of the wiring on our GitHub page, but the basic pattern is the same. The even and odd numbered columns both have identical patterns of key placement and track breaks round them. We fitted the switch first and then, before soldering it, cut the tracks between the switch pins and surrounding the tracks. To make sure you have picked up the correct place to make the break, insert a wire through the place indicated, then turn the board over and make the break where you see the wire.

10 The Matrix 2

Make sure the tack switches are flat to the board when you solder them. They have a habit of pushing themselves out once the board is turned over for soldering. To make sure they are OK, solder

one of the four pins and check. If it is not flat then hold down the corner of the key with your finger nail and reapply the soldering iron and push it into place. Do the same for the opposite corner. Once two corners are in place, simply solder up the other two. Then fit the 1N4148 diode next to each switch; get it the right way round.

11 The Matrix 3

The long tracks of tinned copper wire with a black dot on them denote a joint where one wire enters the hole and another leaves it. Here, push all the wires in place before soldering them. We found this easier if these holes were enlarged slightly with a 1.2 mm drill. When fitting the keys, turn the top over and insert from the back; we numbered the columns and rows on the board and wrote the column and row number, along with its name, on the back of each key. We then placed a blank board over the top and inverted it. Finally, we attached each key to the top with masking tape so they would not drop out when we fitted the top – see **Figure 7**.

12 Warp one, Mr Sulu

One problem we encountered was that the top board had an inwards warp. This meant that when the top retaining screws were tightened, the top was pressing some switches. To counter this,
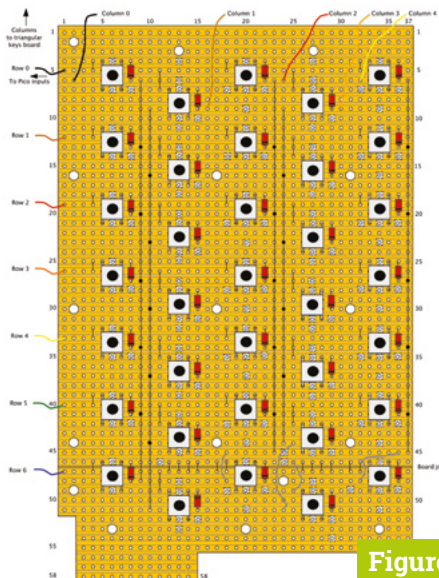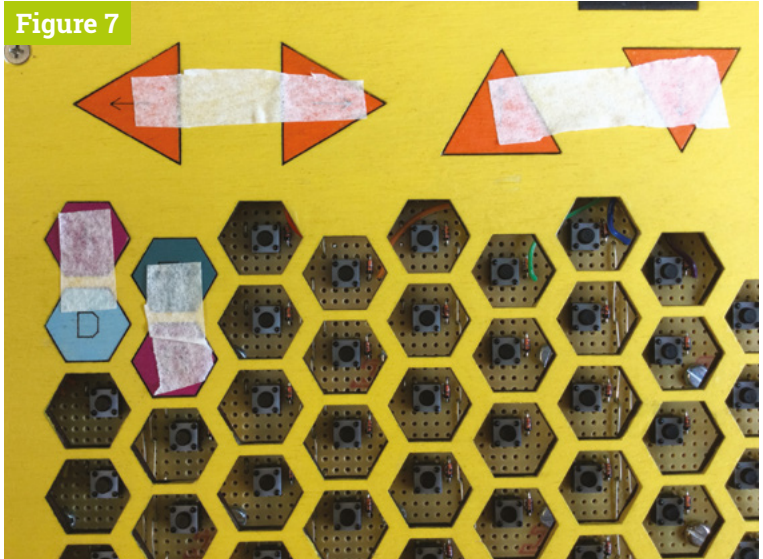

Figure 6

Figure 7

one or two securing holes were turned round so that a screw pushed up on the board, correcting the warping. We used a 20 cm pan-head M3 screw with a 3 mm spacing and three 0.5 mm solder tags on the underside of the board – see **Figure 8**. The solder tags allowed for fine adjustment, making the switch tops flush with the board. We had to cut away at the oversize key mount on surrounding keys to ensure the push-up screw was not fouled.

13 Part 2

Next issue, we'll look at the construction of Raspberry Pi Pico, control switches, OLED display, and rotary encoder, as well as how to wire it all together. We will also look at the software we need to test what we have built, and the final software bring it all to life. ▥

Figure 8

▲ **Figure 7** Use masking tape to stop the keys falling out when fitting the top to the switches

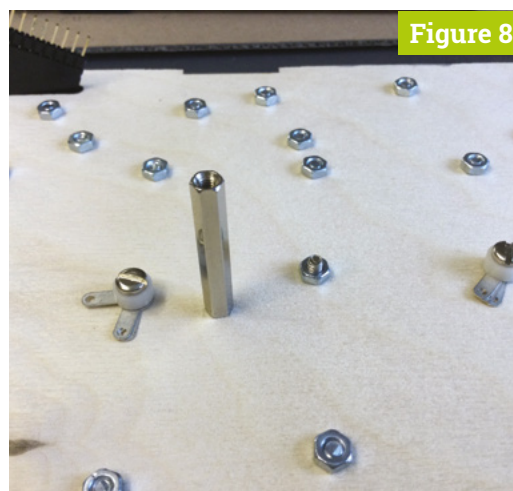◀ **Figure 6** Physical wiring of a third of the switch matrix

Top Tip 👍

Deburr the stripboard holes

When drilling through the stripboard, you'll notice that there will be bits of copper that could make contact with the screw. Deburr the holes by lightly applying a larger drill bit by hand to remove the copper at the very edge of the hole.

◀ **Figure 8** Two push-up support screws and the central base support

FreeCAD: Working with circuits

An open-source workflow for electromechanical creations

Jo Hinchliffe

🐦 @concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!
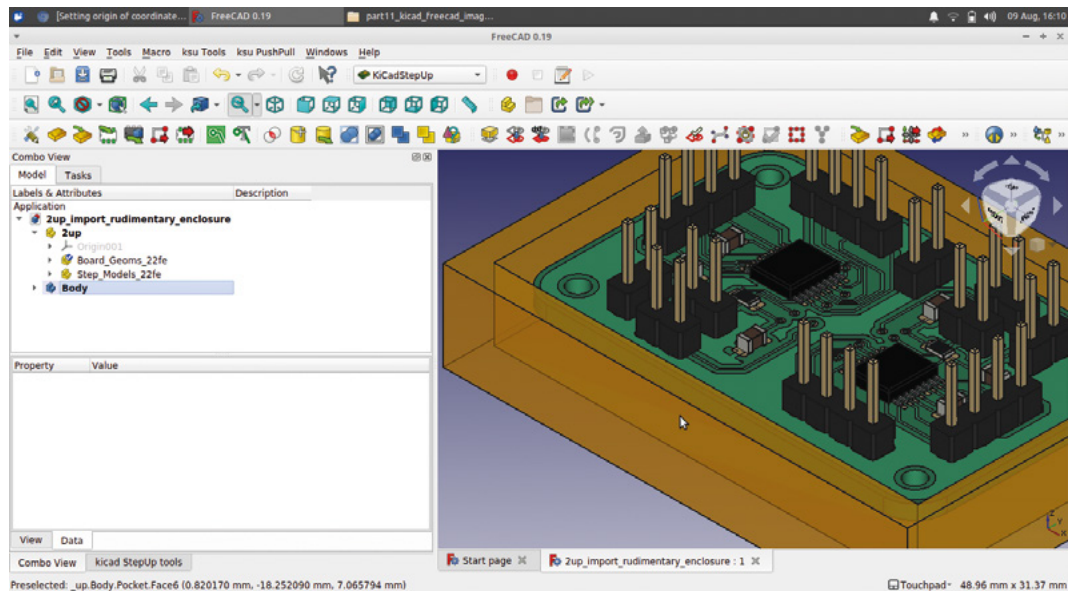
Figure 1 ⬈
A PCB file imported from KiCad with missing models added with FreeCAD, and a rudimentary enclosure design added

YOU'LL NEED

◆ **A computer and a copy of the 0.19 stable version of FreeCAD and a copy of the current stable version of KiCad**

**K**iCad is an excellent open-source PCB design package that can render populated 3D images of circuit board designs. In combination with FreeCAD, we can create custom component models for KiCad or import KiCad PCBs whole, allowing us to design enclosures or other surrounding systems. Let's check out some FreeCAD tools that simplify these processes.

We've looked at KiCad before in issues 17 and 18, where we went through the entire process of creating a simple PCB. One of the concepts in KiCad is that a schematic component is assigned a physical footprint from a list; that footprint consists of numerous items: copper pads, silkscreen drawings, and more, which all appear in your PCB design. Commonly, a 3D model is assigned to the footprint so that you can view your components and your PCB in 3D. Occasionally a 3D model

might be missing or might not fully represent your component, or indeed, you might be working with a custom footprint and component and need to create a 3D model for your part. FreeCAD has us covered for this – there's an add-on workbench that contains specific tools to make this process much easier.

Start FreeCAD and then install the KiCadStepUp Workbench. To do this, go to Tools > Addon manager and find the KiCadStepUp Workbench in the list, then click Install. Once downloaded and installed, you will be prompted to restart FreeCAD. Once restarted, you should see the workbench in the workbench list. Start a new project in FreeCAD and move to the KiCadStepUp Workbench, where we'll first take care of a little housekeeping.

Once in the workbench, click Edit > Preferences and scroll down the icons on the left-hand side to find the preferences for the KiCadStepUp Workbench. Click that icon, and at the top of the
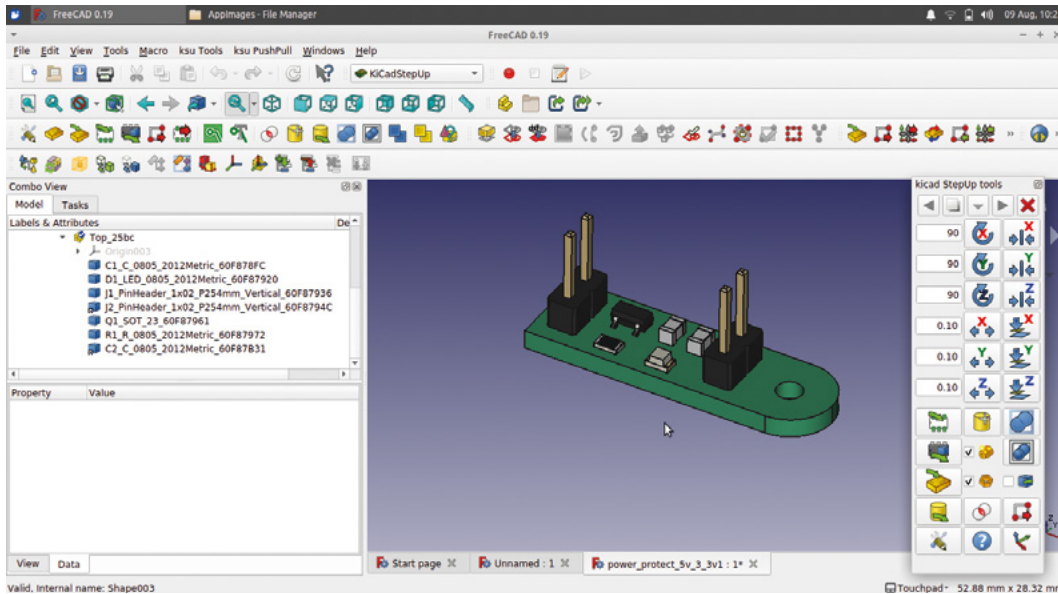
page, there is an area to set up the paths to where KiCad stores its 3D models. To find these, open KiCad and click Preferences > Configure Paths, this should open a dialog box that shows the locations and paths set within KiCad. Copy the paths to the relevant 3D model file locations and put them in the boxes in the FreeCAD KiCadStepUp preferences dialog.

The first thing you might want to do in FreeCAD is to move over to the KiCadStepUp Workbench and do a quick test to import a 3D model of a KiCad PCB. In FreeCAD, open a new project and then move over to the KiCadStepUp Workbench and click the yellow icon with a green arrow which reads 'Load KiCad PCB Board and Parts' as you hover over it. Navigate to one of your KiCad PCB board files (the file in the KiCad project that ends .pcb) and open it.

You should notice a couple of things (**Figure 2**). The first thing is that when you use a tool icon on KiCadStepUp, it automatically launches a floating toolbar with a collection of the most commonly used KiCadStepUp tools on it. You can move this out of the way, or indeed close it, but the toolbar will persist when you move to other workbenches, which, later in this tutorial, we will find useful!

Secondly, you should see your PCB import as a 3D model. You will receive a couple of messages about the import – just close these, and you should be left with your PCB in the preview window. The imported PCB doesn't have the traces visible or the silkscreen items. You can import the silkscreen using a KiCad DXF export and importing into FreeCAD, but most of the time, we probably won't need that level of detail in FreeCAD. You can create the copper traces and pours – we will look at this later in the tutorial – but doing so adds complexity

to the model that you may not need and might make it difficult or slow to move the PCB model. You will also see that the different components on the PCB are listed in the file tree view as separate parts. This means that the components can be treated in a similar way to any FreeCAD part. You can toggle individual parts' visibility, for example, by highlighting the part in the file tree and pressing the **SPACE** bar.

Sometimes when working in KiCad, you'll design a board using a component or components that don't have a 3D model for you to use in the 3D viewer. →

GETTING PERMISSION

It's highly likely that if you are creating 3D models for KiCad components, you will want to save them into the 3D packages folder so that you can quickly find them and use them in KiCad. We run KiCad on Linux, and by default, when installed, the **library** and **modules** folders are only read and writeable as the root user – the general user won't be able to make changes. On Linux, we can use a terminal to switch to be the root user and use the classic `chmod` command to make the relevant KiCad folders read and writeable for our regular user login. Press **CTRL**, **ALT**, and **T** to open a terminal, then switch to root user using the command `sudo su` and put in your password when prompted. Once logged in as root, input the following command using the file path to the folders you want to make read and writeable. We've opted to make all the folders inside our KiCad **modules** folder read and writeable, and as such, our command and file path read: `chmod -R a+rw usr/share/kicad/modules`.

QUICK TIP

We've used the 'tooltips' to describe tool icons in every part of this FreeCAD series. If you are struggling to find the tool icon, hover over a few and read the tooltips!
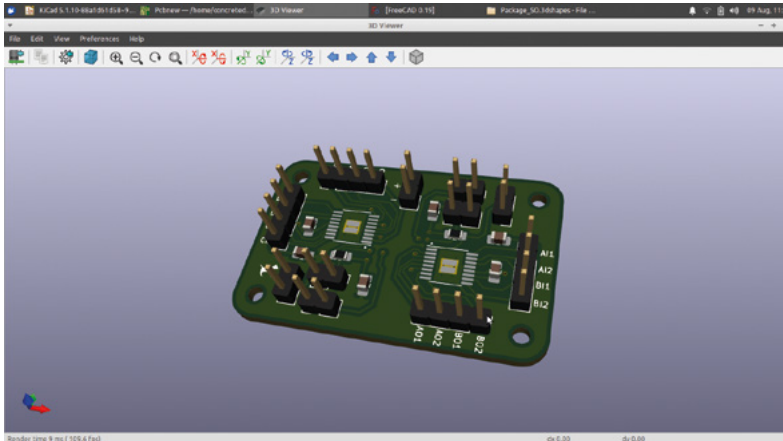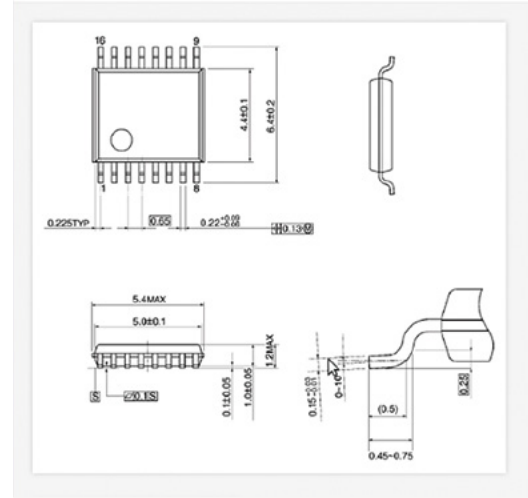
Figure 3 ◆
A PCB we have designed in KiCad that uses an IC package that doesn't have a 3D model in the standard libraries

Figure 4 ◆
Most electronics component packages have details of their dimensions within their datasheets, which can be found online

Or indeed, you might be creating a custom footprint and custom components for use in KiCad and they won't have a standard 3D model or models. We say 'models' because often in KiCad, an object or part in the 3D viewer will have multiple models lined up to create the 3D view. In **Figure 3**, you can see a PCB we are working on that uses a 4.4 mm wide TSSOP16 package IC which is missing in KiCad's 3D viewer. The footprint is there, and we can see the copper pads, vias, and silkscreen, but the actual placed component isn't there. The footprint that is there is actually a 3D model of the small part of PCB board it populates and all the geometry of the footprint. If we create a model of the component that would be placed on top, we can add this to the list of items included in the KiCad 3D viewer to complete our 3D board view.

CENTRE LINES

When we found the dimensions for our IC package, we noticed that the dimensions for the pin spacings were drawn and labelled relative to the centre line of the pin. As we had this information, it made sense to use it, and as such, we added an extra point constrained to the centre of the short vertical line of the pin. This meant that we could then use this point on each pin rectangle to positionally constrain the pin without converting any dimensions off the original drawing. To add a point in the Sketcher Workbench, you simply use the 'Create a point in the sketch' tool.
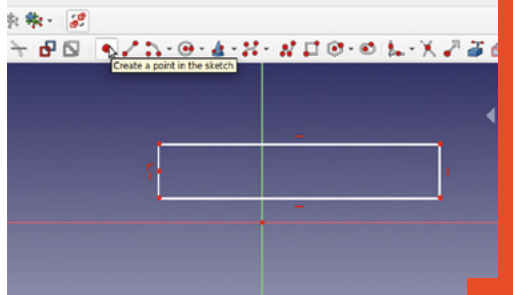
QUICK TIP

We covered all of the FreeCAD skills used to create our simple IC model way back in the first few parts of this series starting back in issue 37!

The first step of this is to identify some accurate dimensions for the part. Most electronics components will have these details in their datasheet, which are available online. We found a diagram for the correct TSSOP16 package that matched our component and used this to create our simple 3D model.

Armed with our dimensions, we fired up FreeCAD and began to model our part. We went for a simple approach that dimensionally represents the component but is only representative in design and lacks detail and refinement. We used the Part Workbench and began with a rectangle sketch that we cloned 16 times and aligned to create the pins of the IC component. Using the simple clone tools on the Sketcher Workbench is definitely useful for creating multiple similar parts such as these pins. Our IC is a little over 4 mm wide, and as such, each pin rectangle was 0.65 mm wide. It's worth noting that at this scale, FreeCAD will automatically slip between mm, µm, and nm when you are applying constraints, but if you replace the unit letters, you can, of course, input them as fractions of a mm.

Having laid out our 16 rectangles to represent the pins, we extruded the sketch on the Part Workbench up to a height of 0.5 mm. Right-clicking on the extrusion in the file tree, we clicked Appearance and set the colour to a silver/grey (**Figure 5**).

We then added a second sketch via the Part Workbench and drew a simple rectangle that would represent the body of the IC component. As we had positionally constrained our pin sketches around the origin point of the XY datum plane when we constrained the rectangle, similarly it meant it was in the correct position. We then extruded the
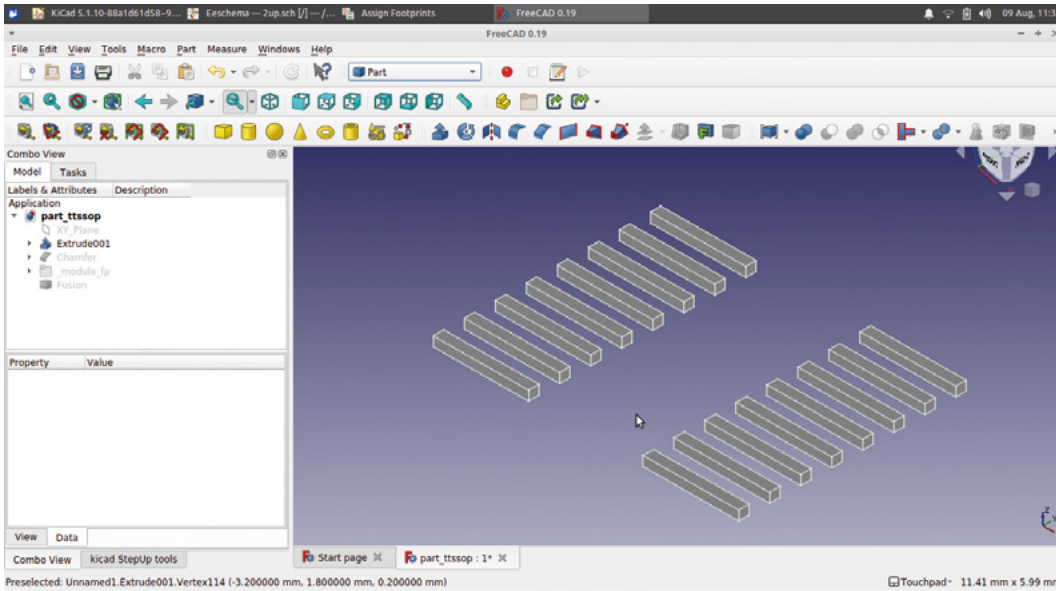
rectangle to the height of the IC package taken from the datasheet. We added a tiny chamfer to the upper outer edge of the extrusion as a token nod to modelling prettily! Finally, we changed the appearance of the extruded rectangle to black – you can see this in **Figure 7**. Once we were happy with the look of our package, we selected the pins and the body of the IC model and used the Union operation on the Part Workbench to create a single fusion object. This is important as KiCadStepUp can only export a single object as the 3D model to be used in KiCad.

Having created our simple model, the KiCadStepUp Workbench has some great tools allowing us to check the fit and position of the model relative to the KiCad footprint model. On the KiCadStepUp Workbench, click the 'Load KiCad PCB Footprint' tool and navigate to the footprint file for the component you are modelling.

Select the footprint and it will be imported into the preview window of FreeCAD and appear as a folder of items in the file tree view (**Figure 6**). When we imported the footprint, it aligned pretty well with our model of the IC, but we wanted to move it very slightly. If you have created your model away from the origin point, you can right-click and select Transform in the usual way to move the part around

in the preview window to get it roughly positioned. Then in the KiCadStepUp toolbar, there are some clickable buttons for the X, Y, and Z axes, and you can set tiny increments in their dialogs. We tweaked our model in increments of +- 0.01mm to get the pins of our package well aligned on the pads of the imported footprint (**Figure 7** overleaf). You can also see that the KiCadStepUp toolbar is available for use when open and undocked, even though we are on the Part Workbench. This is a really useful feature. →

> We used the Part Workbench and began with **a rectangle sketch that we cloned 16 times**

Figure 5
Our very simplified approach to creating the IC pins – the model will be representative and will be accurate in the area that the component sits in, but isn't finely modelled

Figure 6
Loading the KiCad footprint file into FreeCAD allows us to position and check our 3D-modelled component
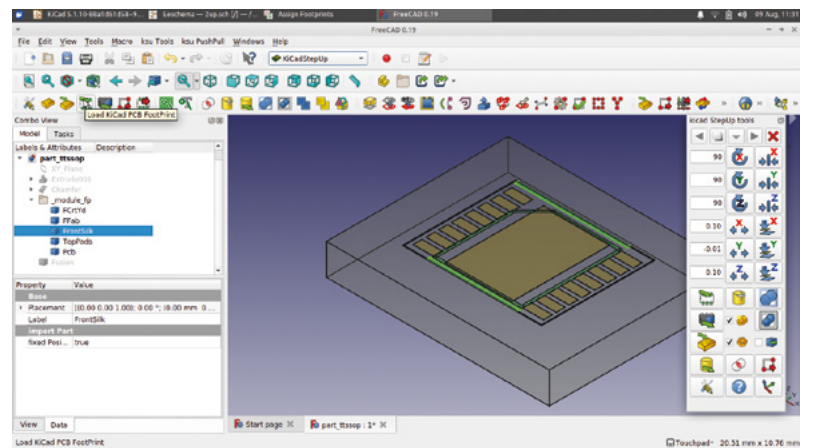
Once you are happy with the position of your component, it's time to export the model to the correct file path to be then used in KiCad. Again, with the model selected as our single fusion object in the file tree view, we clicked the 'Export 3D model to KiCad' tool icon. When you click this icon, some dialog boxes will appear: the first two are 'mesh deviation' and 'crease angles', which we left with their default values. The next two dialogs ask if you want to change the colours of the parts of your model from their current values to suggestions that match common options used in KiCad. We left both our colours on the 'as is' option to use the colours set in FreeCAD, but feel free to experiment. Once you have OK'd the dialogs, two files will be created: a WRL file and a STEP file. On our machine using the stable Linux app image of FreeCAD 0.19, these two files were written to the folder that our app image is stored in. Find the two created files on your machine and note that they are titled with whatever the highlighted part was called in the file tree, in our case, 'fusion'. We renamed both the files to a more useful and descriptive name, **TSSOP16_4.4mm** – we then copied and pasted them into the correct KiCad folder. In our instance of KiCad, the 3D packages are located as **usr/share/KiCad/modules/packages3d**. Within that folder, we then navigated to the folder that would make sense to store the model we had created; for our model, this was the **Package_SO.3Dshapes** – we pasted our two files here. Having saved everything, we can now close

> *KiCad allows you to connect multiple models to each instance of a footprint, so we can add our component here*
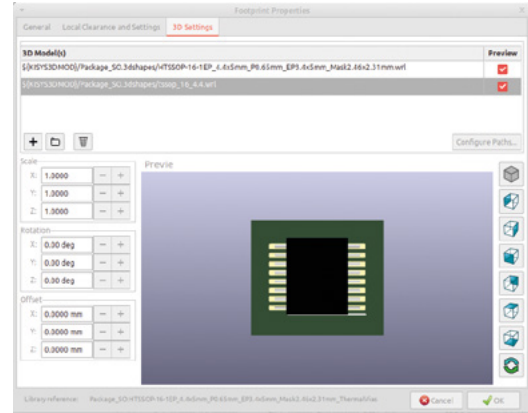
FreeCAD and launch KiCad once again. Opening up our PCB project, we now need to assign the additional component model to our footprint. There are a few different places in KiCad that we can create this association, but as this is a complete PCB project, we did it from within PCBNew.

Opening the project in PCBNew, we selected the component footprint and pressed the **E** key to edit it. In the resulting 'Footprint Properties' window, there is a tab called '3D Settings'. At the top of this tab there is a window with a list of 3D models currently associated with this footprint. In this case, there is just one model, and the model is the 3D render of the footprint without the placed component on top. Brilliantly, KiCad allows you to connect multiple models to each instance of a footprint, so we can add our component here. First, click the '+' icon to create a new empty item in the list. Next, click the folder icon, which is next to the + icon, to set a path to the component model file. Navigate your way to find the WRL version of the files we exported from FreeCAD and moved to the correct KiCad module subfolder. Once selected, you should see a preview of the model. Click OK and check that the new listing is ticked in the checkbox (**Figure 8**). Close the Footprint Properties window and then select View > 3D Viewer. If everything is well, you should now see your PCB rendered with your nice new 3D component model. In our project, we repeated the above steps twice as we had two of the same IC component on our board. You can see the 3D view in KiCad is now complete with all its components (**Figure 9**).

Figure 8
Adding our new 3D model to be rendered in position on our PCB project in KiCad

Figure 7
With the fusion part selected in the file tree, you can use the translation tools from KiCadStepUp Workbench to nudge the part into an exact position
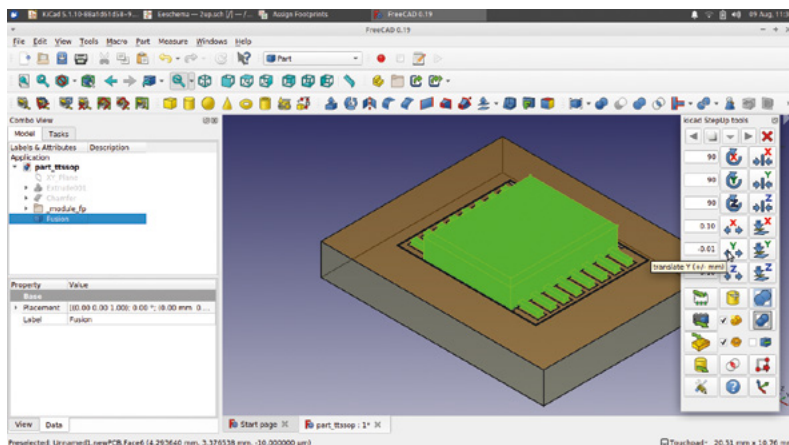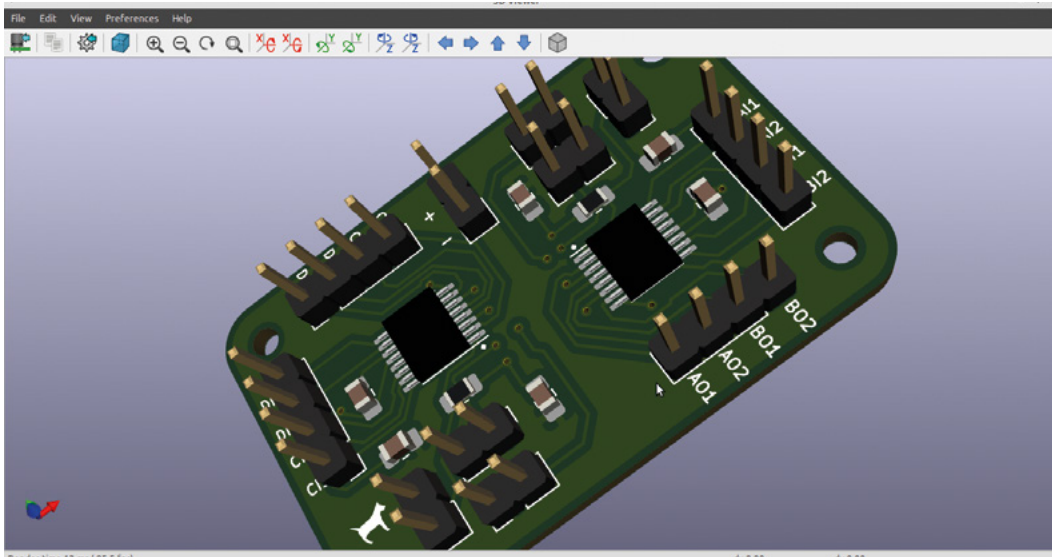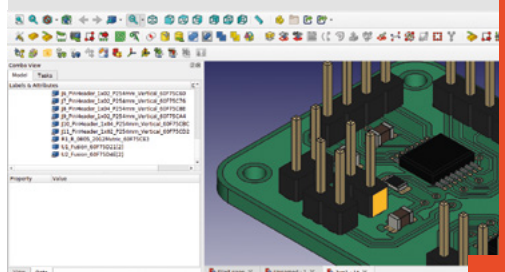
While creating models for missing 3D components for KiCad is a great achievement, we have only really scratched the surface of what the KiCadStepUp Workbench can be used for. Of course, the most obvious use is to be able to create CAD models of projects around the PCB model. All the usual FreeCAD tools are still at our disposal, and as a super-quick example, we made a simple tray enclosure using a body in the Part Design Workbench (**Figure 1**). You can use all the usual tools and tricks; for example, you can toggle between section and full view in the Sketcher Workbench with the PCB model behaving like any other object. We created a pocket in our padded enclosure idea and moved the PCB model into a rough position inside the pocket. As an example of how useful this tool can be, we highlighted both the enclosure body and the PCB model in the file tree view and then clicked the 'Check Interferences and Collisions' tool icon on the KiCadStepUp Workbench. It took a short while to process, but you should see that the Report View window opens and a list is slowly generated of any components that collide with any other components in the complete design. If you have parts of your PCB touching or overlapping your enclosure or design, they will be picked up in this operation; however, it will also list the touching components on the PCB, for example, where a pin passes through a hole. This is useful in itself, but you might also notice that it creates collision parts in the file tree – these are small 3D parts created only where collisions and interferences occur. It's not immediately apparent

why this might be useful, but if you consider that you could use these collision parts in combination with boolean operations to cut away parts of an enclosure where objects are overlapping, you can see that these models have merit, and, of course, if they aren't of use, you can simply delete them! ⬚

FINER DETAILS

Often a simple model of a PCB is going to be most useful, enabling us to design around it and see how it fits within our design. As such, KiCadStepUp imports the simplest version of the PCB file that it can, simply the board edges and the component models, so we end up with a model that has the dimensional geometry of our PCB. However, we can also use KiCadStepUp to add more detail to our PCB model. One of the simplest ways is to add the traces and copper pour layers into our 3D model. To do this, select the imported PCB part in the file tree and then click the 'ksu tools Add Tracks' tool icon. When you click the item, reselect the KiCad PCB file for the project and then click Open. This may take some time and also will increase the complexity of the model, but you should see your traces and copper pour appear on your PCB model.

Surface-mount soldering

No holes, no problem!

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.

When most people learn to solder, they learn to work with through-hole components. These are the sort with legs that poke through holes in PCBs or breadboards. However, peek at any modern electronics and you'll see that very few components are through-hole. Almost all are surface-mount. This means that they don't poke through a PCB, but are soldered to pads on one side of it. These sorts of components are far easier to work with in factory assembly lines, but can seem scary for hobbyist makers.

However, surface-mount soldering shouldn't be much more difficult than through-hole soldering. We got a whole bunch of surface-mount practice kits (these are available cheaply from most direct-from-China websites) to put the different ways of working with surface-mount devices to the test.

METHOD 1: SOLDERING IRON

You don't need specialist kit to solder surface-mount components. In fact, the only thing you need that you may not have for through-hole work is a pair of tweezers. A good pair of curved tweezers are essential for placing components.

These don't need to be really fancy, but it's best not to cheap-out too much as really cheap tweezers are prone to bent prongs, which can send small parts flicking across the workshop, never to be seen again. We use the SparkFun curved tweezers (**sparkfun.com/products/10602**).

You may want to consider the tip shape of your soldering iron as well. It's perfectly possible to solder surface-mount with the larger, cone-shaped tips that tend to come with soldering irons. Some people prefer fine cone tips; others prefer chisel tips that help get into the pads. It's just a matter of personal preference.

Soldering surface-mount with a soldering iron depends a lot on the package. There are, broadly speaking, two types of surface-mount component: those with exposed pads that you can get to with a soldering iron, and those without. Some ICs only have

Right ◈
You can check your soldering with the test points at the bottom, and apply power to the main circuit for blinking lights (Pro tip: use less solder than this)
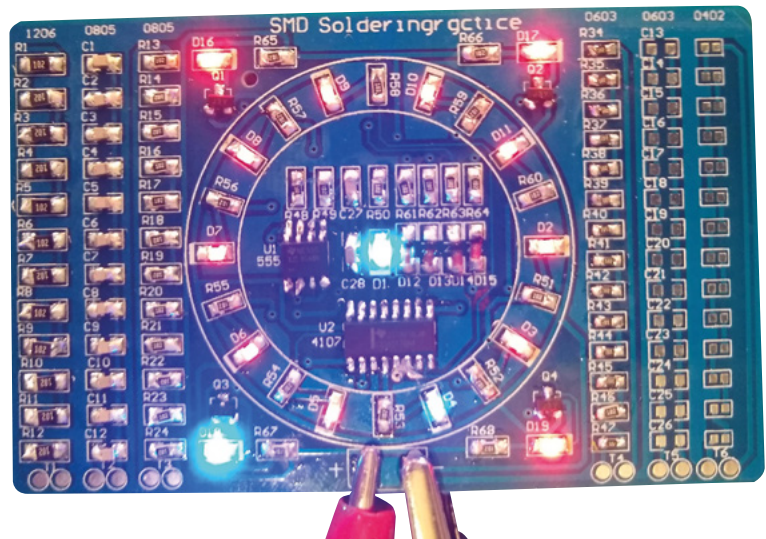
MAGNIFICATION

If you want to use small components, you're almost certainly going to need some form of magnification. There are lots of options, depending on just how small you want to go. The simplest option is probably a pair of magnifying glasses. If you want to go really small, you might want to consider a microscope. A stereoscopic microscope is best as it retains the depth-of-field perception you get in real life.

However, depending on your eyesight, plenty of surface-mount work is possible without magnification.

pads on the underside of the package – these are a whole different beast when it comes to soldering with a soldering iron, so we won't cover them here. Instead, we'll look at components for which you can get a soldering iron in contact with the pad. This covers most passive components (such as resistors, capacitors, LEDs, etc.), as well as transistors and simpler chips.

There is precisely one secret that makes surface-mount soldering with a soldering iron easy: tin one of the pads of a component before you place it.

Surface-mount components are placed on top of two or more pads that you solder them to. Using a soldering iron, simply melt a small blob of solder onto one of the pads. Then pick up the component with your tweezers and place it onto the (now cooled and solid) solder blob. While holding the component in place, reheat the blob so that it melts, and hold the component as the solder cools again. Once the solder is cool, the component will be held firmly in place. Make sure it's lined up correctly – if it's not, just reheat the solder and try again. Once it's in place, you

SIZES

Resistors, LEDs, capacitors, and other two-pad components come in different sizes. The size is usually given as a four-digit number, such as 0805. The first two digits are the width, and the second two are the length. However, they can be in either metric or imperial units.

In this case, 0805 is 0.08 inches wide and 0.05 inches long. It's equivalent to a metric component size 2012 (2 × 1.25 mm). Unfortunately, it doesn't seem to be common practice to state which units you're using. In most cases, the standard sizes don't overlap (there's no metric size 0805, for example). The two that you need to be particularly careful with are 0402 (which can mean 0.04 × 0.02 inches or 0.4 × 0.2 mm), and 01005 or 1005. Despite being a UK-based magazine, in this article we're using imperial units because they are more common in hobbyist usage.
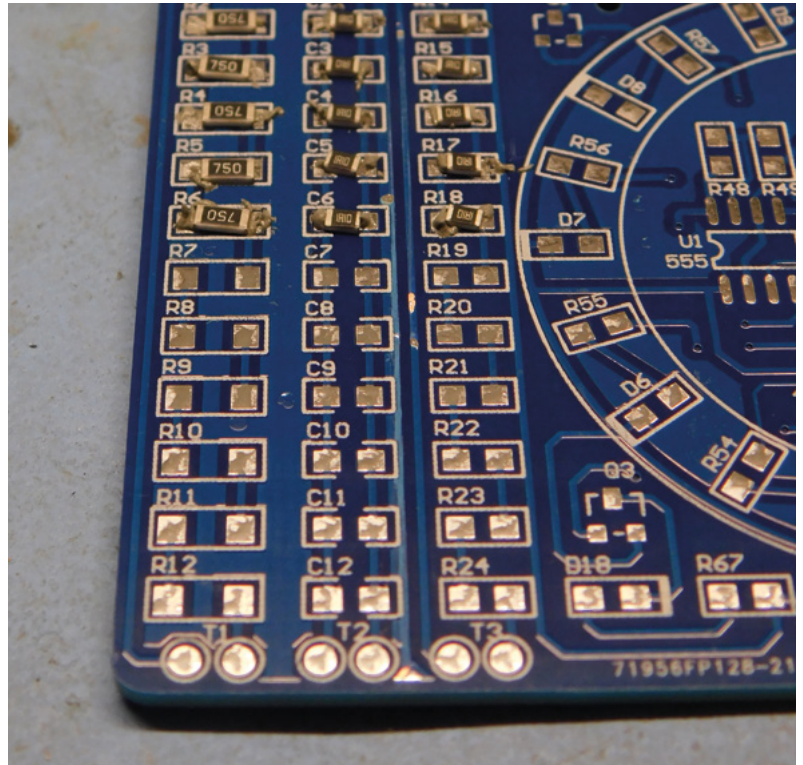
can solder the rest of the pads. If you're soldering an IC or component with a lot of pads, it can be useful to solder an opposite pad next, as this will help make sure it's properly lined up.

When soldering the other pads, it's usually best to just heat the pad. You don't really need to worry about trying to solder onto the component itself – the molten solder will flow there naturally.

How hard you find soldering surface-mount components like this depends on your eyesight and how steady your hand is. Without magnification, this author can do 1206 and 0805 components without any significant challenges. 0603 are a little difficult, and anything smaller is a bit hit and miss. However, with practice and using magnification, many people manage to solder much smaller components.

HOT PLATE

Probably the first step up from doing surface-mount with a soldering iron is doing it on a hot plate. This, as it sounds, is a plate that you put the PCB on that heats up. They come in many forms. It used to be common to build your own or repurpose an electric skillet, but now they're available cost-effectively, it's becoming more common to buy them.

The idea is simple: you put some solder paste on the pads, and put the components into this goo that's a mixture of little balls of solder suspended in flux. Then heat the whole lot up until the flux boils off and the solder melts. Once it all cools down again, you →

Above ⬈
You don't have to be too precise – surface tension will pull the components into place

Surface-mount soldering

—

stencils are available for our surface-mount practice kits, so we have to do it by hand.

You can get solder paste dispensers that deposit a small drop of paste on the pad, but we don't have one of these in the HackSpace mag workshop, so we had to do it the old-fashioned way.

Solder paste is quite stiff and hard to work with, so we mixed in a little flux paste to make it easier to work with. Then, we used some hook-up wire as a tool to dollop a tiny bit of this paste onto the pad.

The key is to add as little as possible. The right amount looks impossibly small if you're used to through-hole soldering, but remember that the solder serves a slightly different purpose in the two methods. With through-hole, there's usually a small gap between the wire and the hole that the solder needs to fill. With surface-mount work, no gap needs to be filled – you just need the tiniest amount to stick the component to the board.

When heating components up, there is a risk of thermal shock if you heat them too quickly. It's common practice to heat the whole board slowly up to about 150°C and let it sit there for a little while. At this temperature, you shouldn't damage any of the components, and you can let everything get up to a nice, even temperature. After a minute or two at this temperature, you can heat it up to your final temperature (typically, this is about 25°C above the temperature that the solder melts at, so something like 245°C for lead-free solder). All solder is a bit different. You may get recommended profiles from your solder manufacturer, or you may have to experiment a bit to find what works.

Once the solder is melted, you just want to cool it down. It's usually best to leave the board on the hot plate as it cools, so it doesn't cool too fast.

should have a nicely soldered-up circuit.

Sounds simple, and actually, it is, but it can take a little practice to get right. The first challenge is getting the solder paste on the pads. The 'proper' way of doing this is using a stencil which is a sheet (usually, but not always of stainless steel) with the pads cut out. Place this over the circuit and you can place the solder paste in exactly the right place. We'll be investigating this more in part two. However, no

> **"**
> **Solder paste is quite stiff and hard to work with, so we** mixed in a little flux paste to make it easier to work with
> **"**

REFLOW OVEN

Hot plates give you a lot of control over the temperature of the board, but they're susceptible to outside influences. A cool breeze or a hot day can change the way the board heats and cools. The other big disadvantage of a hot plate is that it can only do single-sided boards. If you have problems with any of these, then you may need a reflow oven.

The basic idea is exactly the same as a hot plate: heat up the board until the solder melts. However, in
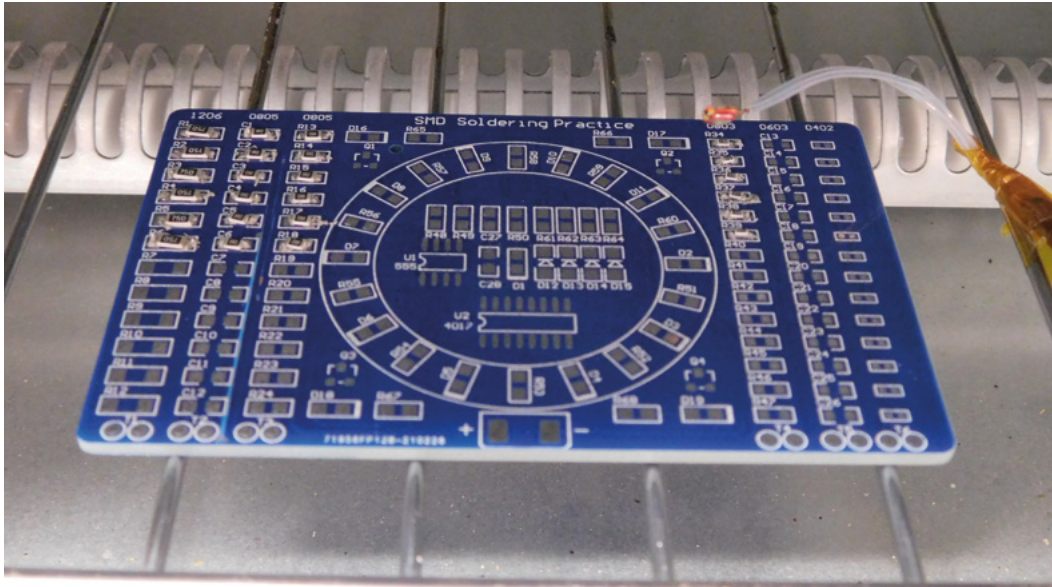
an oven, the air around the board is heated up in a more controlled environment.

There are commercial ovens available, but they are expensive – many people choose to make their own out of a toaster oven.

Again, you need to paste the PCB and heat it through a reflow profile that stops it heating or cooling too fast.

The disadvantage of a reflow oven is that you can't see it as the soldering progresses. On a hot plate, you can give components a little poke if necessary to get them into the right position. In an oven, this isn't possible.

If you need to do a double-sided board, there are a few techniques you can try. One is to use a high-temperature solder for one side and a low-temperature solder for the other side. You can also try just doing both and hoping the surface tension of the solder holds it in place (this is often fine if you're using small components). In professional settings, a dot of glue is sometimes used to stop the components falling off the bottom.

That's the basics of surface-mount soldering! It's not as scary as it first seems, especially if you start with the larger components. In the second part of this series, we'll look at stencils and hot air rework. ☐

LEAD-FREE SOLDER PASTE

In the HackSpace mag workshop, we use lead-free solder almost exclusively. While we know many of you still use leaded solder, we strongly recommend you use lead-free for solder paste. The reason is that it gets everywhere, and you're far more likely to end up with lead in your system than with spooled solder.

Of course, just because it's lead-free doesn't mean that it's healthy. It's still fairly toxic; just less toxic than leaded solder. Wash your hands thoroughly after working with it.

Flux is also an unpleasant chemical. Wash any exposed skin thoroughly after using it, and avoid inhaling fumes (ideally this should be done with a fume extractor, but at the very least, make sure you don't put your head over the circuit board when it's heating). This is easier to achieve with surface-mount soldering than through-hole.

Above ◩
Preparing a PCB to go in the reflow oven

Left ◆
Just a little dab of solder paste is enough. Try to use as little as possible
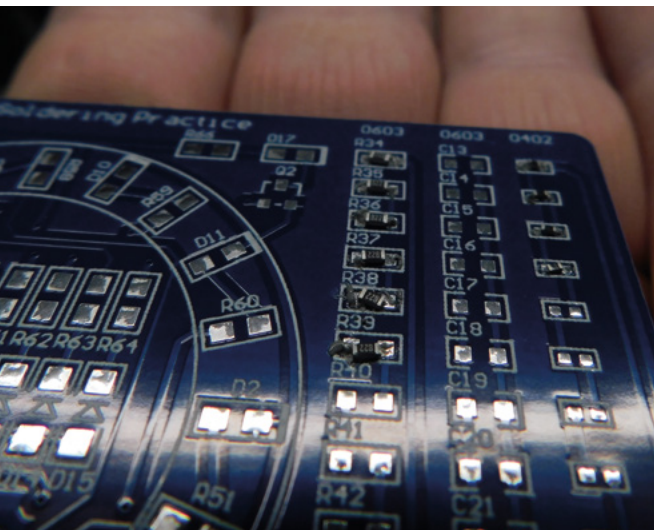
Make a drum machine with Pure Data

Explore Pure Data and use it to create
an electronic drummer for your band

Rob Miles

🐦 @robmiles

Rob Miles has been
playing with hardware
and software since
almost before there was
hardware and software.
You can find out more
about his so-called life at
robmiles.com.

Every great band needs a good solid
drummer. But if none of you can
actually play the drums, you may have a
problem. So why not make an electronic
version powered by a Raspberry Pi
or your computer? In this article, we
are going to find out how to do just that, using
something you may not have seen before, the Pure
Data visual programming language. Our drum will
have two sounds: a kick drum and a tom-tom.

WHAT IS PURE DATA?

There are lots of ways of telling a computer what to
do. Programs written in Python, JavaScript, or C++
are like recipes that specify a sequence of steps for
the computer to follow. Pure Data mimics making
sound and graphics by connecting physical devices
together. A Pure Data program looks more like an
electronic circuit of connected components. The
connections between the components are called
'patch cords', and a Pure Data program is called a
'patch' or a 'canvas'.

Pure Data was created by Miller Puckette, a
Distinguished Professor of Music at University of
California, San Diego. The program that provides the
Pure Data environment and run time system is freely
available for all major platforms.

MAKING A NEW PATCH

Select File > New or use the shortcut key **CTRL+N**
to make a new patch. This opens a window, as
shown in **Figure 2**. You use Edit mode to create and
modify your patch. When you've finished editing, use
the Edit menu item to exit Edit mode, or the **CTRL+E**
shortcut key to toggle edit status. When you are not
in Edit mode, the cursor is a pointing arrow.

START WITH A BANG

The drum will be triggered with a mouse click. Later
we will discover how to get a beat automatically.
A Pure Data patch contains different kinds of
items. One of these is a 'bang'. When clicked with
the mouse, a bang generates a trigger signal on
its output. The user will click the bang to make
the drumbeat.

To add a bang to your empty patch, select Put to
open the list of items, as shown in **Figure 3**. Select

THE **ORGANELLE**

The Organelle is a Pure Data-powered musical instrument. It contains a Raspberry Pi
compute module and provides a portable environment for creating and playing patches.
Find out more at **hsmag.cc/organelle**.

Figure 1 ◈
**An Organelle instrument with its distinctive wooden
keyboard. The owner has replaced the standard black
knobs with coloured ones to prove that he is a proper artist**
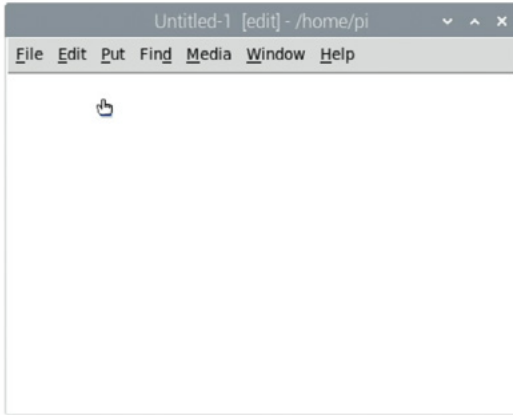
Figure 2
This is an empty patch in Edit mode. Note that the cursor is a pointing hand and the word 'edit' is displayed in the title of the window

Bang and a bang object will be added to your patch. A patch can contain multiple bang items; we could add ones for different drum sounds.

Figure 4 shows how a bang is represented in a Pure Data patch. The thick bars at the left-hand side of the top and bottom show the inputs and outputs. The input is at the top, and the output is at the bottom. A bang will pass through triggers it receives on its inputs and send a trigger to its output when clicked. We are going to use the bang to trigger the production of a signal that will be processed to make the drum sound.

RAMPING UP
The starting point for the kick drum sound will be a 'ramp' signal that goes from 0 to 1. The ramp will last 200 milliseconds (a fifth of a second), which will be the length of our drum sound.

Figure 5, overleaf, shows the ramp signal will change over time. The signal is linear, which means that at the halfway point (100 milliseconds), the output level will be 0.5. This would make a very boring drum sound, so we will use some maths on the values in the ramp to make them more interesting.

SENDING A MESSAGE
We are going to use a Pure Data `line~` object to produce the ramp signal. The ~ (tilde) on the end of the name of the `line~` object indicates that it creates an audio signal. We need to tell the `line~` about the ramp we want – and to do that we need to create some messages. The first message tells the `line~` the value we want at the start of the ramp (0),
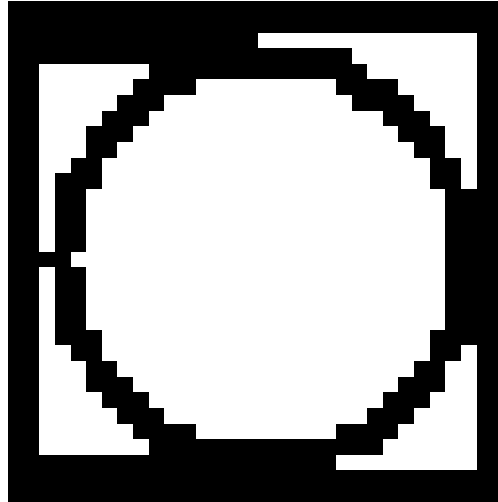
and the second sets the value at the end of the ramp (1) and gives the length of the ramp in milliseconds (200).

We will send these messages by using a `message` object that sends messages when triggered. A message contains a list of objects separated by spaces, and messages are separated by commas, so the `message` object we want to create looks like **Figure 6**, overleaf.

This figure shows the `message` object that we will use. Note that `message` objects have a particular shape that makes them easy to spot on a patch. The first message is '0' (set the ramp level to 0), and the second message is '1 200' (ramp up to 1 over 200 milliseconds).

If we trigger the `message` input it will send the messages from its output. Add the `message` to →
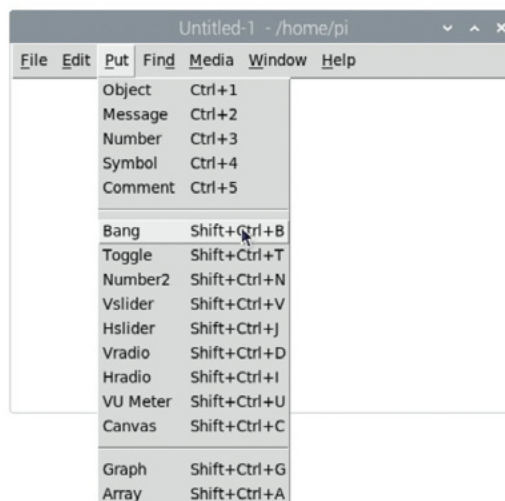
Figure 4
The user clicks inside the circle to trigger the bang. A bang will not send triggers if the patch is in Edit mode

Figure 3
The Put menu item displays items that can be placed in a patch. Note that each has a short cut to speed up editing
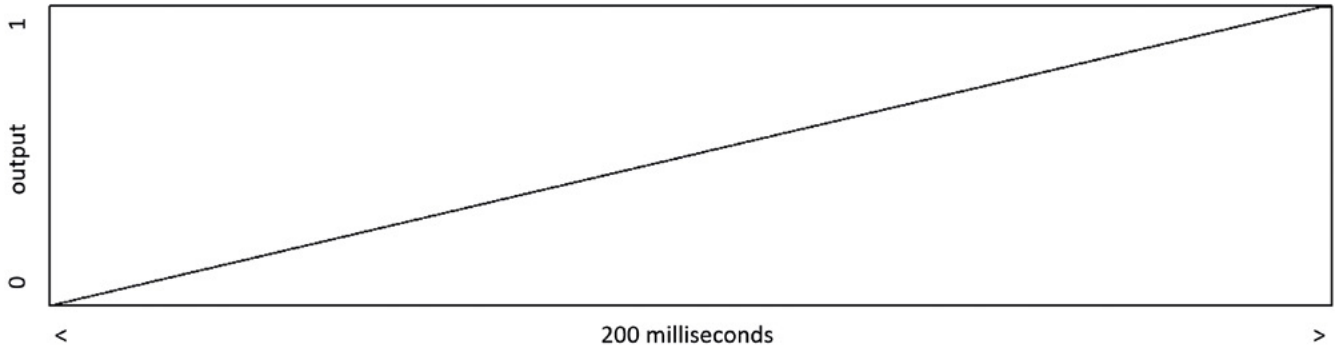
Figure 5 ◆
This shows how the
ramp signal looks on
a graph. The Y axis
shows the output
level, and time goes
along the X axis

Right ◆
There's lots of help
on the website
puredata.info

Figure 6 ◩
The message object
accepts a trigger
on the input at the
top and then sends
all the messages
it contains out of
the output

the diagram by using the Put menu to add the
`message` object, drag it into position under the bang,
and then click inside the `message` on the patch to
enter the messages.

MAKING A LINE
Next, we add our `line~` object to make the line.
There is no Put option to create a `line~`. We add an
'empty' object and then tell Pure Data that the new
object is a `line~`. Start by adding an object using Put
and then type `line~` into the empty object to tell Pure
Data what kind of object it is.

   Figure 7 shows the patch so far. Signals are
going to flow down the patch, starting with an
output from the bang. To make connections, hover
the mouse over a connection point until it turns
into a ring. Then, left-click and drag the connection
endpoint over to the destination connection, where
the cursor will turn back into a ring. Release the
mouse button to complete the connection. You can
delete a connection by hovering the mouse over
the connection line until it changes to an 'X'. Then
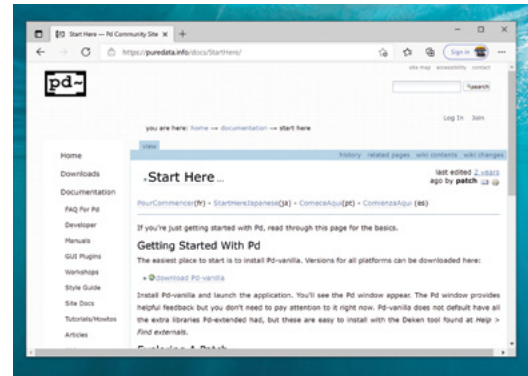
click to select the line (selected items turn blue), and
press the **DELETE** key on your keyboard.

SOUND SHAPING WITH MATHS
Now that we have our ramp, it is time to turn it
into something more interesting. We are going to
do this by processing the values in the ramp data.
Computers treat sound as sequences of numbers.
Each number (or sample) represents the volume
level at a certain point in time. Each second of sound
produced by Pure Data is made up of 44,100 such
samples, in the range -1 to +1. When each of these
numbers, in turn, is converted into a voltage level, a
sound signal is produced.

   The ramp produced by the `line~` object is 200
milliseconds (a fifth of a second) in length, which
means that it contains 8,820 values. The first sample
value in the ramp is 0, and the last value in the ramp

THE LEGENDARY
ROLAND TR-808

Electronic drum machines have been adding beats
to music for many years. One of the most influential
devices is the Roland TR-808, which was launched in
1980. Roland cut the cost of the TR-808 by designing
analogue circuits to make the drum sounds rather than
storing audio drum samples in expensive computer
memory. They even used 'broken' transistors to produce
the noisy signals that they wanted. Only around 12,000
were made, but the TR-808 acquired a cult following and
it is said to have been used on more hit records than
any other drum machine. What we are doing with our
Pure Data drum machine is building a digital process
that recreates the behaviour of analogue components,
which is a nice example of technology going full circle.
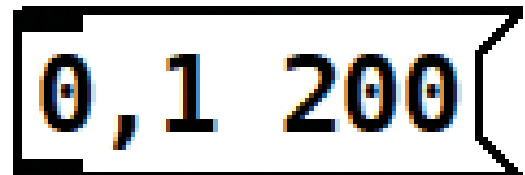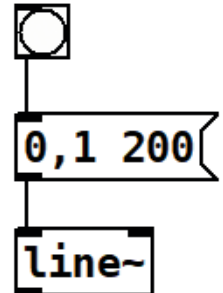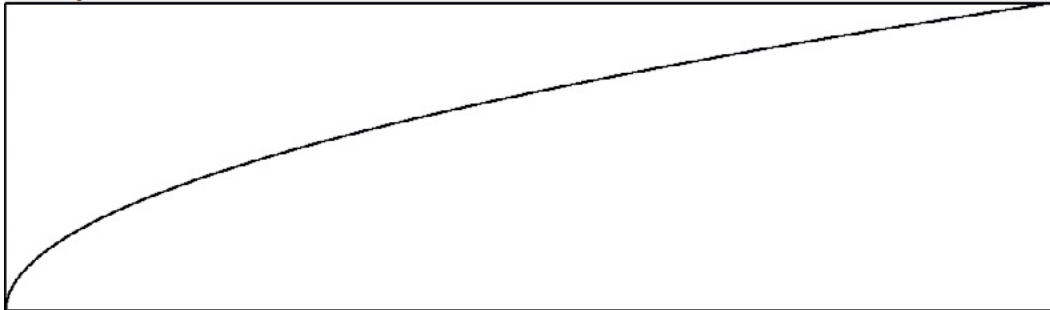
is 1. What we want to do is convert the data values in the ramp into a sound signal that can be played to make a kick drum sound. We can do this by performing some mathematical transformations on the values in the ramp.

At the moment, the ramp is the straight line; you can see this in **Figure 5**. We could use this to produce a sound that contains the same frequency all the way through, but that would just sound like a beep. To make the sound more interesting, we are going to change the shape of the ramp by taking the square root of each value in the ramp.

Figure 8 shows the effect of the square root operation. The ramp still starts at 0 and ends at 1, but now it has quite a different shape. Next, we are going to increase the number of waveforms we are going to produce from the ramp by multiplying each value in the ramp by 20, and finally, we will convert the ramp into a series of sine waves that can be played on the output to make the drum sound.

The ramp will be sent to a **dac~** object to make the sound output. The audio theory behind how all this works is too lengthy to put here, but the sequence of patch operations looks like **Figure 9**, overleaf.

If you add the objects shown in **Figure 9** to your patch and connect them together, you should hear a healthy thump when you click on the bang at the top (remember to leave Edit mode first by pressing **CTRL+E**). If you don't hear anything, make sure that Pure Data is connected to the correct audio output. By default, Pure Data will send audio over the HDMI connection to the monitor. If you add a sound card or USB audio device, you can use that instead (but you will have to select it using the File > Preferences > Audio dialogue). Make sure that the sound devices you want to select are plugged in before you start Pure Data.

> It turns out to be very easy to convert the kick drum **sound into an electronic tom-tom sound**

MAKE A TOM-TOM SOUND

It turns out to be very easy to convert the kick drum sound into an electronic tom-tom sound. We increase the frequency of the sound. At the moment, the kick drum sound contains 20 waveforms that are played over a 200-millisecond time period (a fifth of a second). You can just about count the waves in **Figure 10**. The 20 waveforms are produced by the ***~ 20** object in the middle of the patch. Note that this multiplication doesn't make the patch →
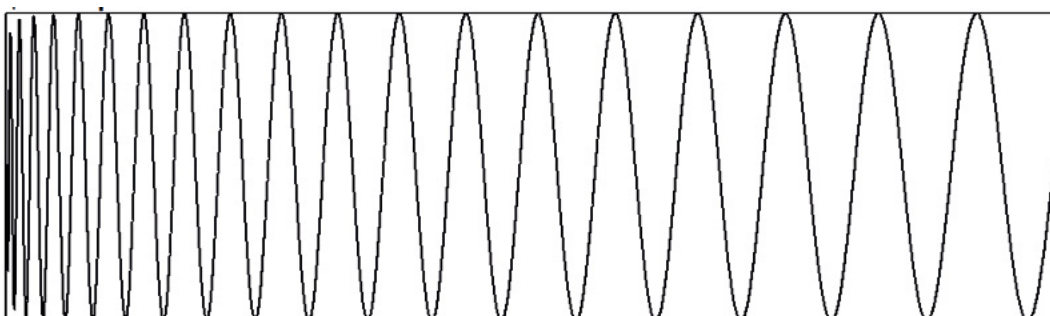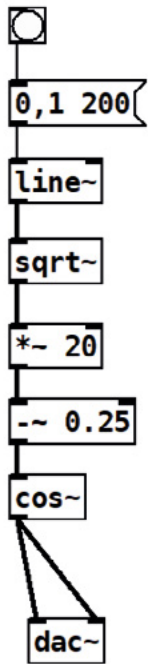
Figure 9 ◆
Pure Data draws the audio signals (the ones after the line~ object) as thicker. The objects that process audio have a tilde (~) after their name. The dac~ object accepts two inputs: one for left and another for right

longer; it increases the size of each of the values in the sample. When these values are passed through the `cos~` operation, they will cycle through the cosine values in a circle 20 times, leading to 20 waveforms. If we change the value to `*~ 50`, the sample will contain 50 waveforms, which means the drum now sounds more like a tom-tom. You can create a tom-tom in the same patch by using the mouse to select all the objects in the original patch and then cutting and pasting them into a new area in the patch.

AUTOMATIC DRUMMER

We can make the drums automatic by using `metro` objects in our patch. A `metro` (short for metronome) fires trigger events at regular intervals. By setting different intervals for each drum, we can get a good beat going.

The patch in **Figure 11** puts out a good strong beat. It uses two metronomes: one for each drum. The value in the `metro` object gives the time in milliseconds between each output. The tom-tom is played every 300 milliseconds, and the kick drum is played every 600 milliseconds. You can experiment with the timings of the two beats to get interesting, syncopated rhythms. The `metro` objects are enabled by a `toggle` object at the top of the patch. This is rather like a bang, but it is used to enable or disable an object. In the case of the auto drummer, the toggle turns the metronomes on and off.

SOUND ENGINEERING

You can have a lot of fun playing with sound patches and making different sounds. You could try making the frequency even higher. You could reverse the direction of the ramp so that the pitch starts low and goes high. You could add more mathematical processing. These patches only scratch the surface of what you can do. You can also use Pure Data to create audio waveforms for use in sound synthesis – look out for future articles about this. Pure Data has support for MIDI (Musical Instrument Digital Interface) signals, which means that patches can interact with keyboard and controllers and send commands to other musical instruments. You could use this to allow the beats of our drum to be controlled remotely over MIDI, perhaps from a Raspberry Pi Pico device. Check out the article *Make music with Pico* in HackSpace 46 on how to get a Pico to send MIDI messages. ▢
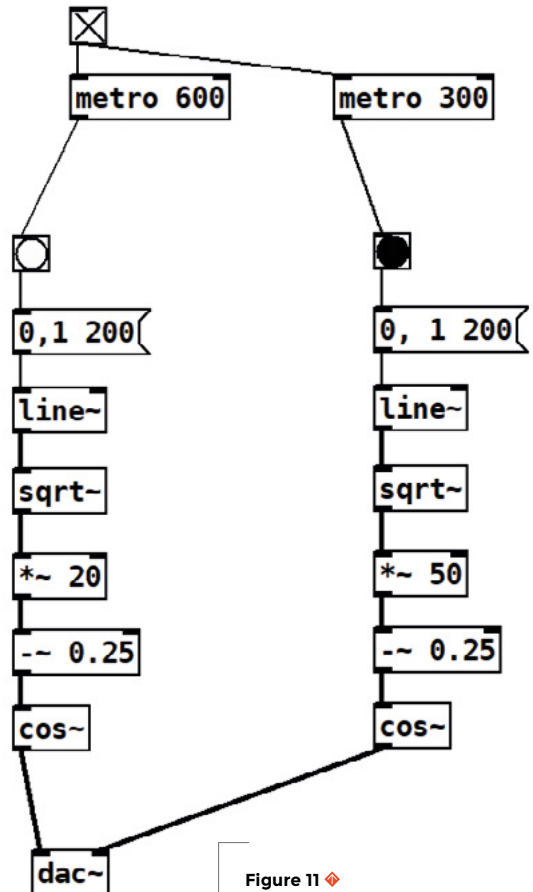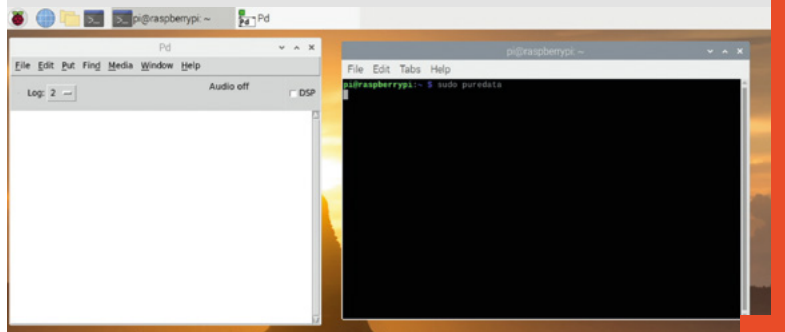

Figure 11 ◆
The drummer sends the kick drum to the left channel and the tom-tom to the right for a truly stereoscopic experience

GETTING **PURE DATA**

If you are using a Raspberry Pi, you can download and install Pure Data from the command console:

```
sudo apt-get install puredata
```

This installs the Pure Data application and creates a shortcut for it in the 'Sound and Video' menu on the desktop. However, the author has discovered that Pure Data works best on a Raspberry Pi if you run it in Supervisor mode. You can do that from a terminal window:

```
sudo puredata
```

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG
110

LASER
ENGRAVER

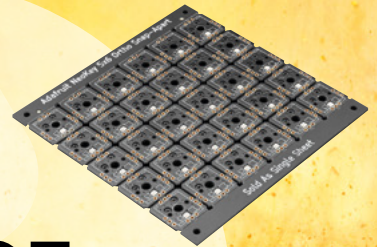Burn images into
almost everything

PG
112

WATCHY

An e-ink watch you can
make at home

PG 104

BEST OF
BREED

Build your own macro keypad

Alternative and additional keyboards for your computer

ONLY THE BEST

When even an extended keyboard just isn't enough

By **Marc de Vinck** | @devinck

I f you know a programmer or gamer, and you ask about their setup, you most likely will hear about the type of keyboard they use. And I don't mean wireless, corded, standard, or extended. I mean the details, all the way down to the type of switches used, and profiles of the keycaps. Do they like light pressure, high pressure, silent, or super-clicky? It's a popular topic amongst many of us. I admittedly am typing away on an Apple keyboard right now (Gasp! I know). But when I switch over to my PC, which I do regularly for some VFX work, or product modelling, I do admit I like my Cherry MX Brown switches, and the feel of a definitive press of the button. And maybe a little RGB lights for fun too.

In this Best of Breed, I'll be looking at kits that use mechanical switches as their input. OK, there is one exception, but we'll get to that later. You would typically think of these as computer accessories, but the love of tactile switches has taken over a lot of the microcontroller market too. Maybe it's the realisation of how nice the switches feel, the ability

to customise them, the ease of a snap fit, or maybe it's a supply chain issue with many other parts, putting many kits on temporary hold that has brought these products to light? I really don't know, but they seem to be everywhere, and I like it!

Adafruit MacroPad **vs** Qwiic Keyboard Explorer

ADAFRUIT ◈ **$49.95** | **adafruit.com**   **QWIIC** ◈ **$49.95** | **sparkfun.com**

If you like mechanical switches and customisation, then the Adafruit MacroPad is something to look at picking up for your desk. I just got my hands on one, as they seem to be in high demand, and I have to say I'm totally impressed. And I see a lot of kits!

The MacroPad features a 3×4 keyboard controller and an RP2040 from the Raspberry Pi Foundation. Why is that special? Adafruit says it has 64 times more RAM than the Apollo Guidance Computer and they even added 8MB of flash memory. I bet NASA would have loved to be able to have done that so easily! What it really means is lots of power to run code quickly. You can press a key and have a full list of macros displayed, then click another button to execute the macro, fast! Or maybe have the whole keypad change colours according to their function? I have so many plans for this kit; I may need two!

You can pick up the MacroPad PCB on its own, but this one is the full kit. It includes the keys, keycaps, and an enclosure. Once you build the kit, you have access to the twelve keycaps, but also a rotary encoder, an RGB NeoPixel LED, and my favourite addition, a 128×64 SH1106 Monochrome OLED display. That's the part that makes this kit a must-have accessory for any computer.

Left ◈
The spinner is an underrated addition to your keyboard

Below ⬉
What would you add to the prototyping area?

The Qwiic Keyboard Explorer from SparkFun is the firm's entry into the mechanical keyboard DIY marketplace. The kit features 14 mechanical keys, all connected to SparkFun's Qwiic Pro Micro dev board. And like most of the other kits in this roundup, you can set macros for each key and have them execute quickly. But unlike some of the other kits, this one also features a prototyping area, which I think would be handy for more advanced users. And you can easily expand the capabilities of this keyboard using SparkFun's no-solder Qwiic connector that can be found on the Pro Micro board.

The Qwiic Keyboard Explorer kit comes with everything you need to build the complete keyboard, including Cherry MX blue switches and the translucent keycaps. Check out SparkFun's website for more info, including Arduino example code that will get you up and running quickly. →

VERDICT

Adafruit MacroPad

This is the best kit for adding quality keys to your setup.

10/10

Qwiic Keyboard Explorer

A complete kit with a nice prototyping area.

8/10

Keybow 2040

PIMORONI ◈ **£49.50** | **pimoroni.com**

've featured the Keybow from Pimoroni in an earlier Best of Breed, but this is the new flavour of the kit, and I thought it warranted a second look. The original Keybow was connected to a Raspberry Pi Zero, which is great accessory, but the new evolution of the Keybow 2040 features the relatively new microcontroller from Raspberry Pi, the RP2040, integrated right into the board.

You'll have to choose what version you'd like: clicky (Gold), linear (Silver) or tactile (Copper) Kailh Speed switches, as the kit comes with everything you need to get up and running, including the switches and keycaps. And since they are the standard type of keyboard switches, you can simply plug and play, no soldering required. It even ships with the RP2040 preprogrammed and ready to use. Building your own keyboard, featuring 16 keys, couldn't be easier.

Below ◈
Possibly the most beautiful macro keypad

VERDICT

Keybow 2040

Mechanical keys and LEDs – what's not to like?

9 /10

CHERRY MX MECHANICAL
4-KEY TESTER

ADAFRUIT ◈ **$6.95** | **adafruit.com**

Nobody said picking out the type of Cherry MX mechanical key switch for your next project was easy. For the uninitiated, Cherry MX mechanical switches come in various types, and they all have a very different feel and sound. With this simple kit, you can easily try them out. It might be a little overkill for just a hobbyist, but if you are making a full-size keyboard or building projects for others, this can be a real time- and money-saver.
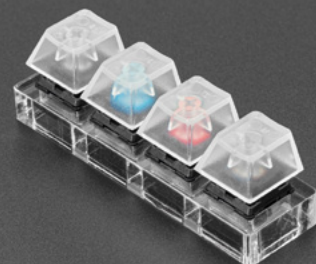
NeoKey 5 × 6

ADAFRUIT ◆ $29.95 | adafruit.com

You know you want to add some high-quality, clickety-clack buttons, or silent ones, to your next project or computer. But you don't know exactly how many, or what configuration. And that's where the NeoKey 5 × 6 from Adafruit jumps right in. Sold as a single 5×6 PCB, you can easily snap in your MX-compatible switches and have a beautiful 30-key keyboard. But the genius of this PCB is the ability to snap each key apart. You can have one single key, a strip of six, a 2×2 grid, or really any combo of 30 keys. It's perfect for your custom configuration. And each one has an RGB LED too! It's a great solution for the DIYer who needs something custom.
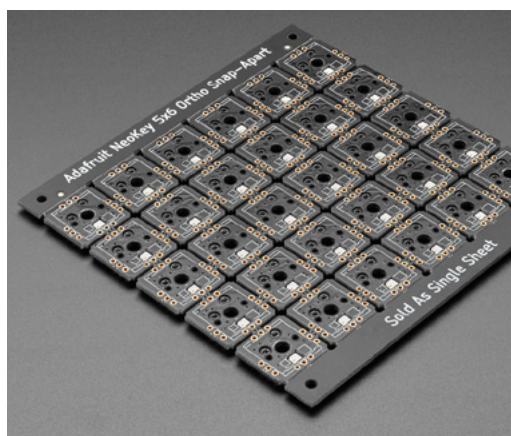
Left ◆
DIY custom keyboard layout

VERDICT

NeoKey 5 × 6

Use as is, or break it apart!

9 /10

Adafruit NeoKey Trinkey

ADAFRUIT ◆ $6.95 | adafruit.com

When you just need one button, then you need the Adafruit NeoKey Trinkey. I can see a lot of use cases, including making it mute your audio, run a macro, or simply relocate your Escape key. The board comes populated with an RGB NeoPixel and a SAMD2 that can easily run CircuitPython or Arduino code, so it's capable of doing a lot! It's specifically designed to allow you to add a Cherry-MX compatible mechanical switch and keycap. Or you could just use the on-board capacitive sense pad. Just slide the board into an unused USB port on your computer and it will power up and can be programmed. →
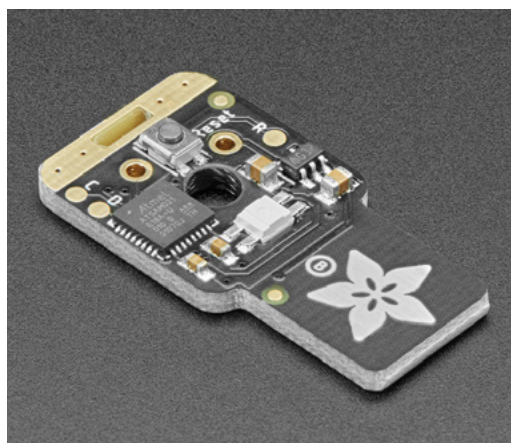
Left ◆
Get a large enough USB hub and you can build a keyboard

VERDICT

Adafruit NeoKey Trinkey

When you just need one more key!

10 /10

Math & Automation Touch Keyboard for micro:bit

**DFROBOT** ◈ **$29** | dfrobot.com

Y es, I know this doesn't have any clickety buttons. And yes, I can hear the classic cartoon record scratch sound in my head as I start to write this review. But I included this kit simply because, when I was researching all the tactile switch kits out there, I came across this cool little extended numerical keypad board. And yes, I know, it doesn't have switched keys! But that's another reason I included it in this roundup. I really like the capacitive touch and layout of the PCB. And each pad has an LED too!

The Math & Automation Touch Keyboard for micro:bit is an easy-to-use shield that includes 16 programmable touch keys, each with an RGB LED. The board also features a vibration motor, a buzzer, two motor ports, two servo ports, and nine expansion ports. That's a lot more than just a simple numeric keypad! Head over to the product wiki for detailed documentation and examples. OK, now back to mechanical switches! Sorry for the tangent. □
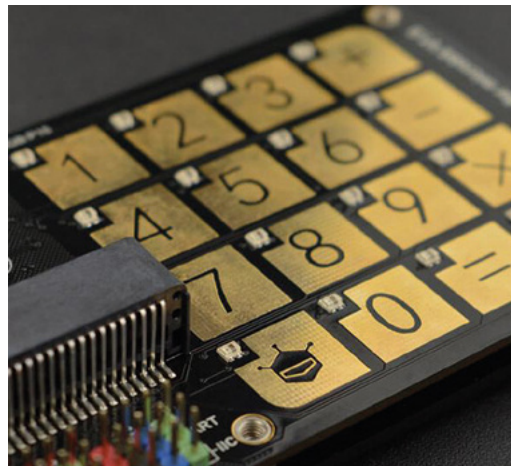
Left ◈
The gold touchpad looks divine
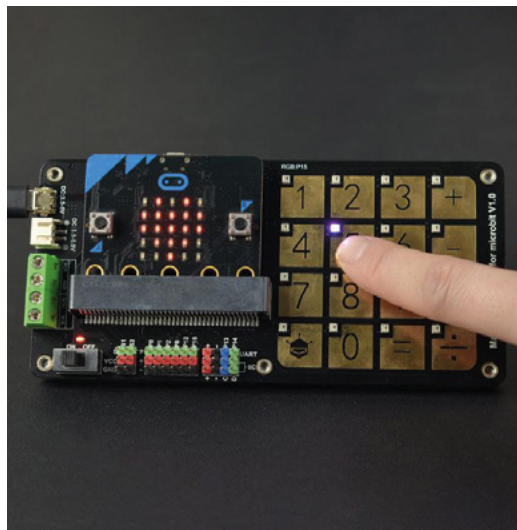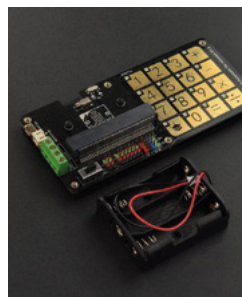
VERDICT

Math & Automation Touch Keyboard for micro:bit

A nice way to add a numerical input to your micro:bit.

8/10

KITTY PAW
SILICONE KEYCAP MOULDS

ADAFRUIT ◈ **$3.95** | adafruit.com

OK, so you are the type of person who thinks simple QWERTY is just so passé. You want custom! Well then, the Kitty Paw Silicone Keycap Moulds might be the perfect project for you. Start with this mould and pick up some UV-cured resin to make your own custom Kitty Paw Keycaps. Too much work? Well, don't worry, you can get one premade too!
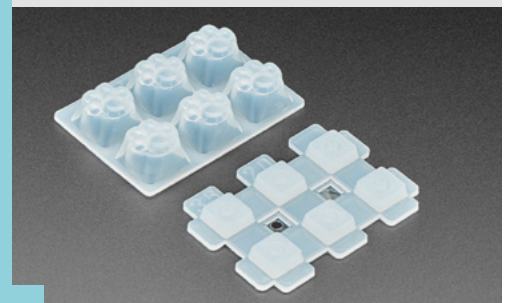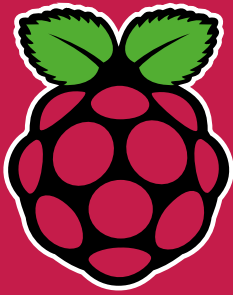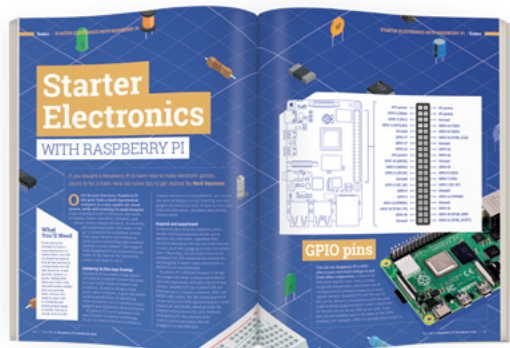
4050 laser engraver

Create designs with perfectly controlled burning

By **Jo Hinchliffe**   🐦 @concreted0g

As part of the CNC 3018-PRO series of tutorials, we played a little with a laser module that could be attached to the machine for engraving, and we were keen to experiment with something larger. We picked up a cheap kit for a 40 × 50 cm working area CNC laser engraver for £135. This item is available online and is sold under a few different names – searching '500 MW desktop 4050 laser engraver' returns a few similar results. We ordered what we believed would be branded 'ChronosMaker', but actually, on arrival, the controller board was labelled 'Vigotec', but apart from that, it was identical to what was pictured.

It arrived well-packed, with a decent set of build instructions. Assembling the frame and running gear was straightforward and took a couple of hours using the cheap tools that came in the box. The assembled frame is pretty rigid and, as there are no real cutting forces with a laser diode, it's fit for purpose – don't expect this kit to be rigid enough to convert into a CNC router, though. The only problem we encountered was a small missing part: a spacer that created a standoff for one of the V pulley wheels that held the laser carriage assembly to the X axis. We rectified this by filing a nut down to carefully match the height of the missing part, taking a measurement from one of the others. Not the biggest problem – we were probably just unlucky. It's worth reading the instructions fully before assembly, as there are some T-slot nuts that need inserting into slots several steps before they are utilised, because the slot ends become inaccessible in subsequent steps. Of course, we found that out the hard way and had to partially disassemble.

The axes are driven by stepper motors and lengths of GT2 belts and pulleys. We were slightly sceptical about the belt attachment, which is a simple bolt through a captive nut, with the bolt end clamping down on the belt. However, we've found that this has worked very well and has given no problems in operation. We were pleased to see that the Y axis is driven via two stepper motors rather than a single stepper with some kind of linking drive-shaft, as other designs have. The kit was supplied with all the cabling
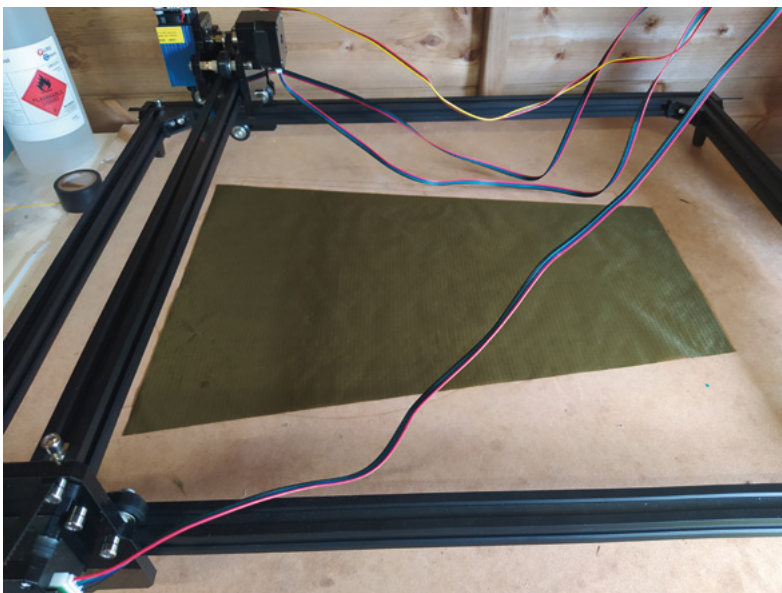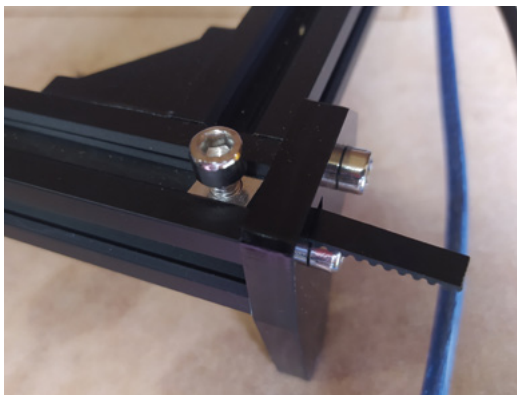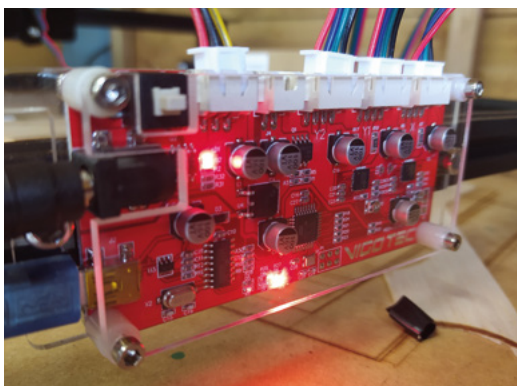
Above
We were sceptical at first about the bolts holding the belt tension, but it has worked perfectly

required pre-terminated, so once it's mechanically assembled, it's simple to plug everything together. Our machine came without software but suggested LaserGRBL, which is a free and open-source Windows only control environment and G-code sender. Once downloaded and installed, this software worked perfectly with the machine and allows for vector images to be imported to cut and a range of image files to be imported and converted for engraving.

The laser diode supplied with the machine is a blue laser with a wavelength described as 450 nm and a 500 mW power rating – this is at the lower end of power output for modern laser diodes, but still is capable of cutting and burning a variety of materials. As such, you need to be very careful to not damage yourself or others when in use. The kit came with some green laser PPE goggles, but there are differing opinions on how well these would actually protect your eyes. It's worth considering that for between £30 and £50, you can buy laser goggles that are designed to offer protection across a range (190 nm to 490 nm

Below
The controller is simple to connect up, and works well with LaserGRBL software out of the box

in our case) which cover this frequency well and have a documented and proven level of protection. Also consider if other people may suddenly arrive during laser operation, or if family pets may wander into the environment and take precautions to reduce this risk.

As with most laser diode engravers, the work rate to engrave is slow – we can only imagine how many days it would take to engrave a design that would fill the work area. That said, with some tinkering in LaserGRBL, it is possible to get great results when engraving. We have engraved onto wood, card, and paper. We have also cut through a variety of materials, including paper, thin balsa wood (2 mm sheets with three or four passes), card, and fine rip-stop nylon.

One issue you will need to solve with this machine is ventilation and fume management. If you run this machine in an enclosed space, you will very quickly fill it with fumes. The laser diode has a small fan mounted above it, but this is mainly to cool the laser enclosure during operation. As a stopgap solution, we used a desk fan pointed across the machine to push fumes towards the open double doors of our work shed. However, if we continue to use the machine a lot, we will probably mount an extraction tube that travels with the laser module that vents outside.

For the price, we think this is an excellent machine if you have a use case for it. It's straightforward to build and operate, and is incredibly accurate. ▫

Above
It's worth considering upgrading from the green plastic goggles that come with the kit to something specified and rated for the wavelength of the laser

Left
Engraving onto card works very well, with good results

DIRECT FROM SHENZHEN

REVIEW

Watchy

Temporal e-ink art to adorn your wrist

SQFMI ◈ $59 | sqfmi.com

By **Ben Everard**   🐦 @ben_everard

Below ⇘
After six days without charging, Watchy is still going strong

The last couple of years has seen a plethora of hackable smartwatches designed to be programmed through open frameworks, such as Arduino.

For what is quite a bit of technology, there aren't many parts to Watchy. It comes as a PCB, an e-ink screen (which plugs in via a ribbon cable), a watch-case, and a battery. Assembly is surprisingly easy: pop the ribbon cable on, plug in the battery, snap on the case, and insert the watch-strap. If you opted for the aluminium case, there are an additional four screws that hold everything in place. There's also a bit of adhesive that you can use to hold the screen in place, and a screen protector that can also be added. It took us longer to find an appropriately sized Allen key in our disorganised workshop than it did to assemble the watch, (we later realised that there was an Allen key included which we hadn't spotted).

Add some power (via the micro USB cable) and you've got a watch.

We tested it with the aluminium case, which is quite industrial-looking. If this isn't to your tastes, there's a selection of 3D-printable options on the Watchy website (**hsmag.cc/watchcases**).

This isn't a small watch, but it's certainly one of the smallest hackable watches we've come across. We found it unobtrusive and comfortable to wear. While the aluminium case is tough, it isn't waterproof and doesn't offer any protection to the screen other than the stick-on screen guard.

HACKING

Watchy is unashamedly a watch for programmers, and some parts of configuration can only be done via code. For example, the default seven-segment watch-face includes the current weather forecast.
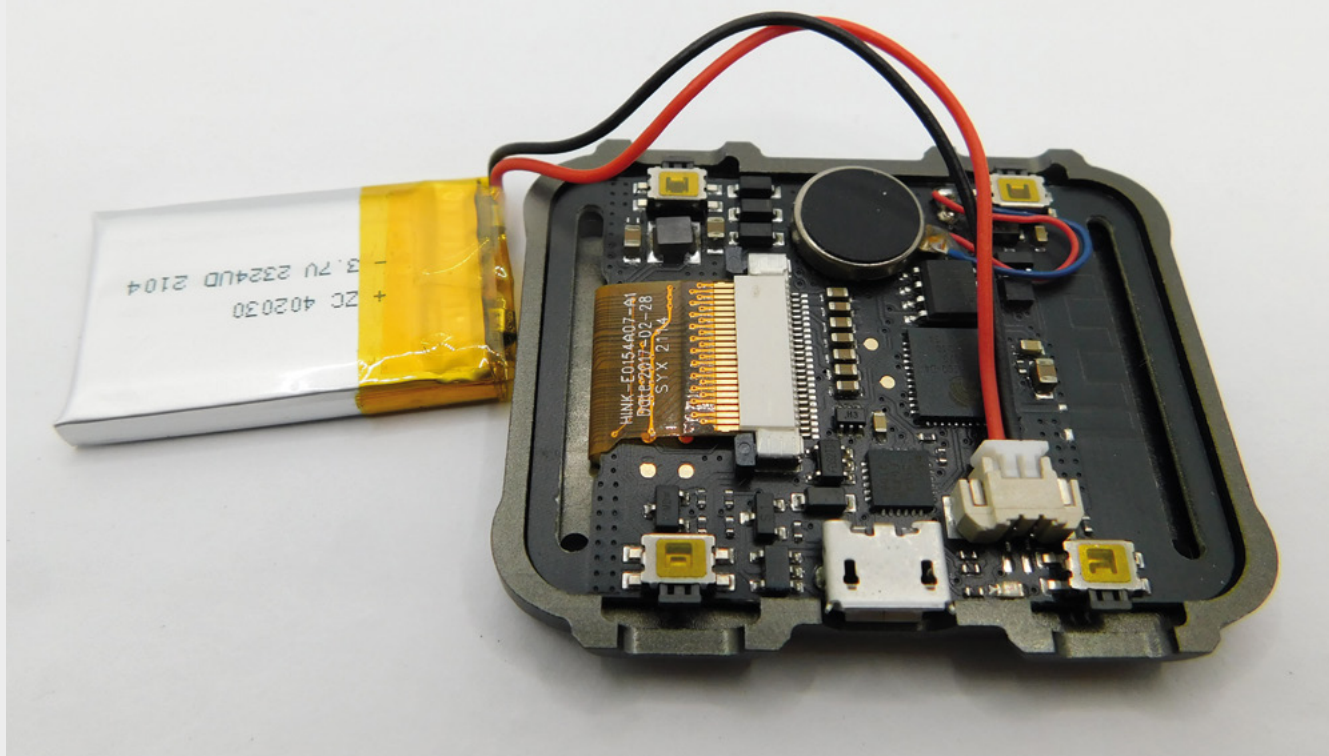
However, this is hard-coded to New York unless you make a change to the configuration file, recompile, and upload. We'll look more closely at how to build your own watch-face in a future issue.

The other stand-out feature of Watchy is the e-ink screen. This is easily readable in daylight and always on (unlike watches with TFT-based screens that typically turn off to save power). There are 200×200 pixels, so plenty of space to display information, but they can only be black or white (not greyscale or colour), so there's no anti-aliasing, and text looks a little more blocky than it would on an equivalent-sized TFT. The other downside of this display is that it's not readable in a dark room. That's simply the payoff for low-power displays. It's personal preference, but it does feel nicer to have a display that turns on and.

It's this low power that is probably our favourite feature of Watchy. Exactly how long Watchy will run depends significantly on the WiFi. Running it as just a timekeeper, ours lasted around a week. With it also fetching data over WiFi every hour, this dropped to a few days.

> **The stand-out feature of Watchy is the e-ink screen. This is easily readable in daylight and always on**

There are loosely two schools of thought about smartwatches. Some people fundamentally want a watch that perhaps has an extra feature or two. Other people fundamentally want a wearable computer that they can bend to any purpose. Watchy will appeal to the former more than the latter. The always-on e-ink screen is a good canvas for expressing yourself in time-based artwork, and there's already a range of watch-faces you can play with. There are enough smarts to augment this a little bit with, for example, the weather and step counting. However, if you want to do more complex online or interactive things with the watch, you'll start to run into the limitations of the form factor and display technology.

There are a couple of things that we'd like to see on the watch: there's no charge indicator, there's a missing connection that makes it hard to maintain low-power Bluetooth connections, and we'd be very interested in an e-ink screen with more colours (some are available in similar form factors). However, these are minor issues. Watchy is a great platform for building your own expressive timepiece. □

The Oskitone POLY555 is a polyphonic, analogue, square wave synth based on the 555 timer chip – 20 of them, to be precise. Each timer chip is at the heart of a sound-generating circuit linked to one of the synth's 20 musical keys. It's an approach that is simultaneously simplicity itself and massive, massive overkill. Most importantly, it sounds absolutely epic.