Manage the mess
Tidy workshop, tidy mind

DIY Music
Adding long-range radio to a project

DESIGN YOUR FIRST

PCB

Make your circuits permanent

Turn your ideas into reality

Add custom artwork

+

Learn sewing
Make your own cable bag

Vacuum tubes
Learn to use retro components

BUILD HAT
**Control LEGO®
with Raspberry Pi**

LEDS PICO **BUBBLE WRAP** PLA
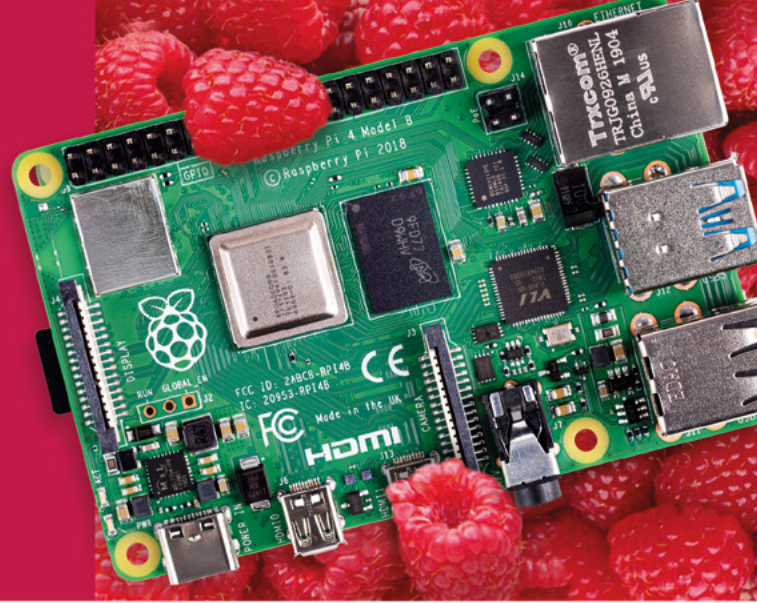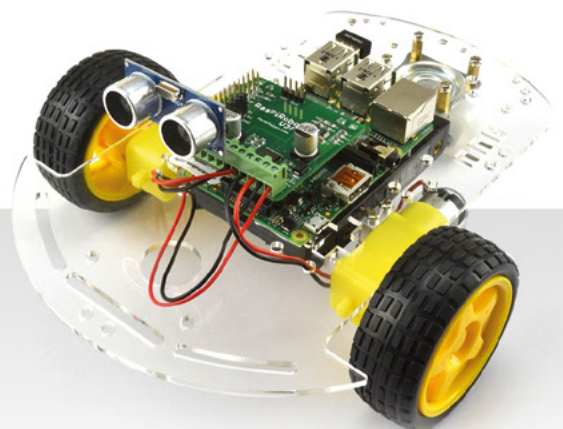
Welcome to HackSpace magazine

Printed Circuit Boards (PCBs) are almost ubiquitous in the modern world. They are cheap to make and easy to assemble. For a long time, though, they've been seen as the preserve of professional manufacturers and only the more dedicated hobbyists. However, in the last few years, they've become more accessible than ever before. There's a wealth of beginner-friendly design tools, and getting PCBs made is both simple and cheap.

> There's a wealth of beginner–friendly design tools, **and getting PCBs made is both simple and cheap**

In this issue, we're looking at how to bring your builds up to the next level, with custom-made PCBs that look great and hold your components securely. It's much easier than you might think.

If all this talk of PCBs feels a bit modern for you, take a look at page 70 where we're revisiting that retro-tacular technology from yesteryear: vacuum tubes. They may not be cheap or fast, but they look cooler than any of this modern sand-based electronics, and sometimes that's more important.

BEN EVERARD

Editor ben.everard@raspberrypi.com

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

get in touch at
hsmag.cc/hello

GET IN TOUCH

 hackspace@
raspberrypi.com

 hackspacemag

 hackspacemag

ONLINE

 hsmag.cc

Available on the
App Store

GET IT ON
Google Play

Contents

112

Tutorial

Sew a bag

74 Make a bag for storing all your maker bits and bobs

Cover Feature

DESIGN YOUR FIRST PCB

Make your circuits permanent
Turn your ideas into reality
Add custom artwork

34

94

30

06

Interview
Wesley Treat

56 Making 1950s roadside America, one sign at a time
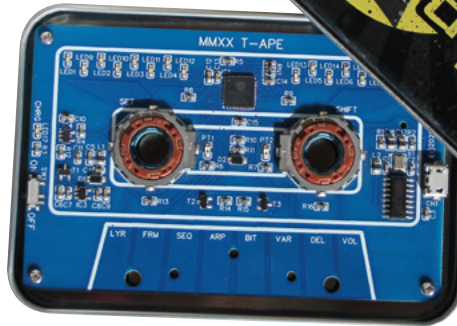
Build HAT
Smart Lego
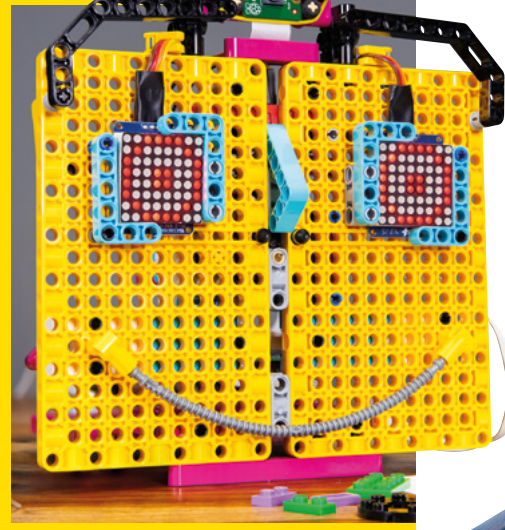
26 Unlock the robotics potential of your LEGO® kits

108

70

Eco-Arduino

By **Jiva Materials**   ✈ **jivamaterials.com**

P CBs, as you'll know from our cover feature this issue, are currently constructed from epoxy resin and fibreglass. To extract the metals used in their construction, they need to be shredded and incinerated, which is an inefficient, time-consuming, and energy-consuming process. Electronic waste is a growing problem, and with the current squeeze on semiconductor prices, it makes sense to have a PCB substrate that makes it easier to reuse components without chucking a load of waste in a landfill.

Enter Soluboard®, from Jiva Materials. This PCB substrate is based on natural materials, which delaminates when it's soaked in warm water, allowing the natural parts to be composted and the metals to be recycled.

We like the thinking behind this product, and we also like the woody aesthetic. Here's how it looks when incorporated into an Arduino Uno. ◻

Right ↗
The makers estimate that, compared with traditional materials, Soluboard® PCBs would save 10.5 kg of carbon per square metre of PCB
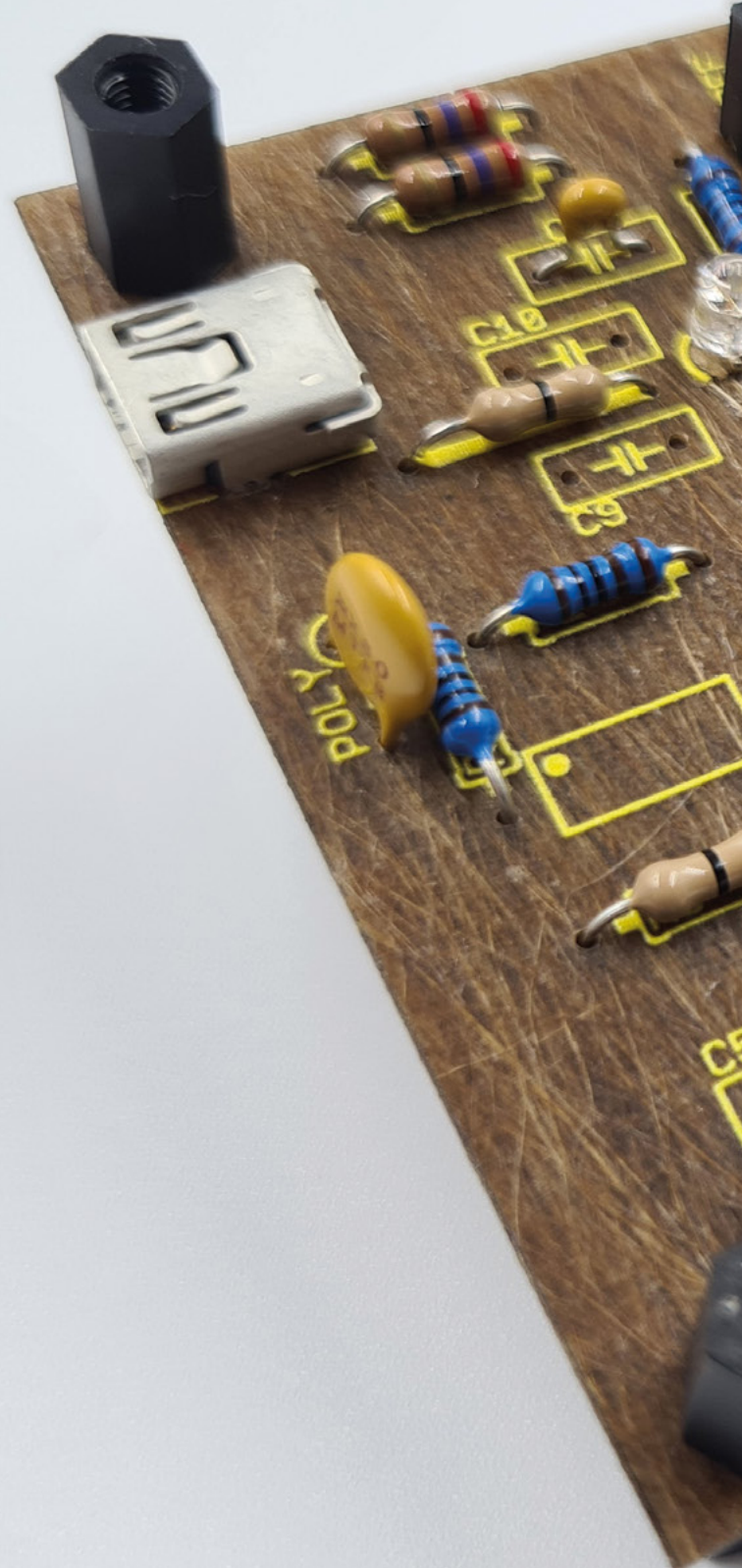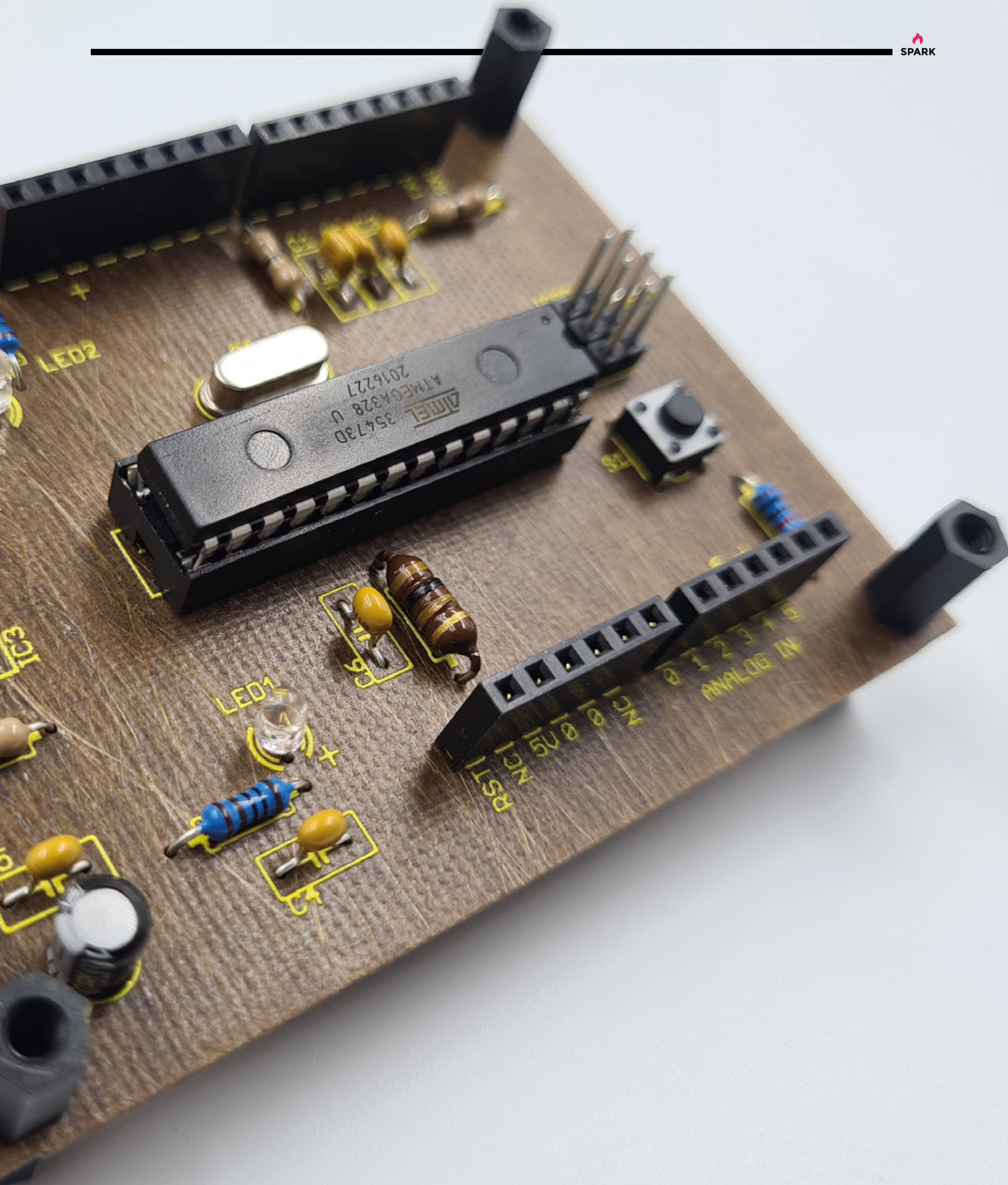
MMXX T-APE

By **mayojaster**   🚀 **hsmag.cc/T-APE**

Forget what you think you know about synthesizers: this, from mayojaster, is something (as far as we're aware) completely new. Instead of using samples, or square waves generated by something like a 555 chip, it uses what the maker calls a 'bytebeat', which is software that generates a fractal that the synthesizer can use to generate sounds.

We can't vouch for the maths behind the sounds, but we can tell you that we like the way it looks, and we like the way it sounds. If you want one, buy it from Tindie, unless you're in a recently post-Brexit country, where you're best off ordering from mayojaster's own website so they can calculate the VAT properly. ◻

Right ⟶
The MMXX T-APE
features ten
capacitive keys,
two encoders, two
light sensors (so you
can use it as a kind
of theremin), and a
headphones output

MMXX T-APE

LED12 C12 C13 R5
D4   IC1   RN1 LED13 LED14 LED
       C14   LED5 LED6 LED
R8   R9
                          SHIFT
   C15
PT1 R10 P.T2   R1 C16 C17 R2
     R11
D2   L2 C1 X1 C2
R6   R7

T2   T3   R16   CN1
   R14 R15   C3

SEQ   ARP   BIT   VAR   DEL   VOL
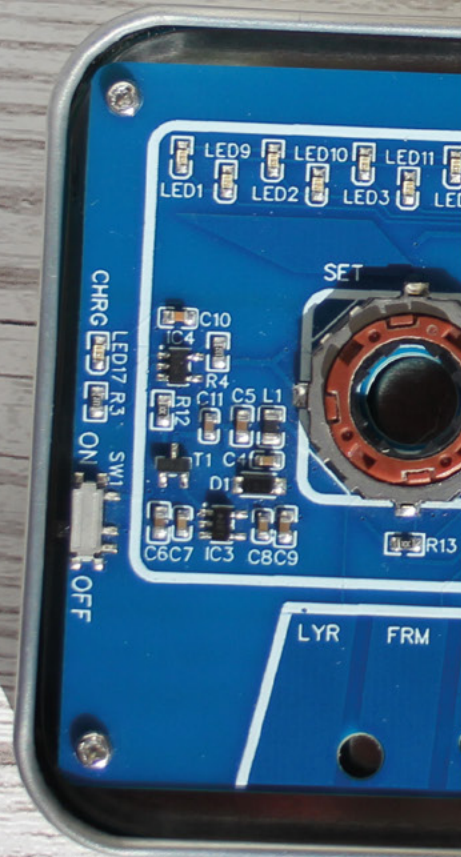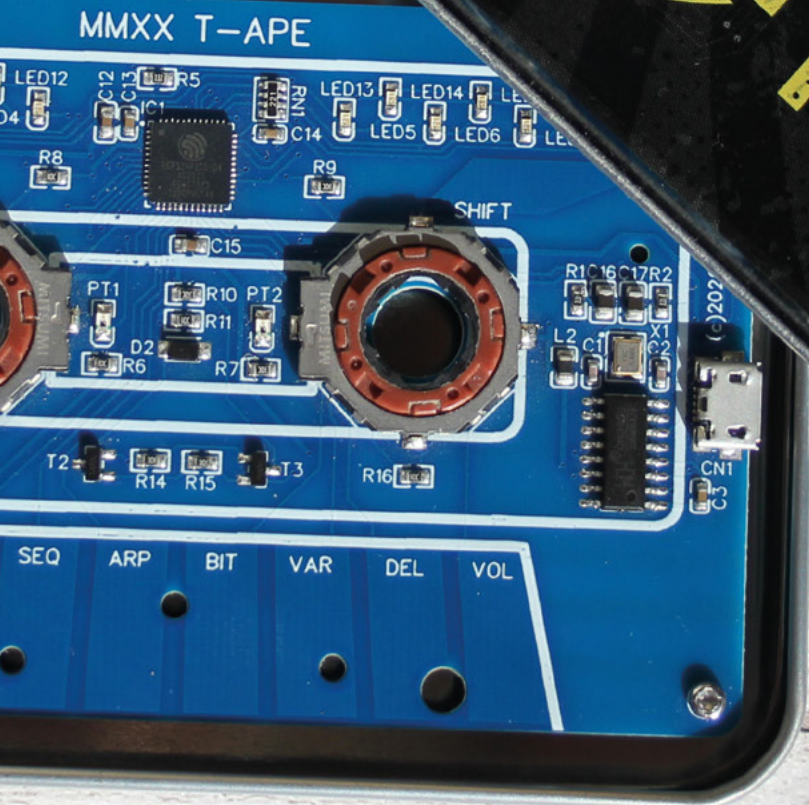
Raspberry Pi Fight Stick

By **Stuart Brand**   **hsmag.cc/FightStick**

This glorious piece of work is a self-contained retro games emulator, lovingly crafted out of polished, shaped, and drilled aluminium. The buttons are custom-made, the joystick is custom-made, and it can be used either as an all-in-one gaming system or as a USB joystick. Naturally, there's a Raspberry Pi at the heart of this device. ▢

Right ⤢
Stuart's also made a sumptuous Atari console, which looks just as good as this does
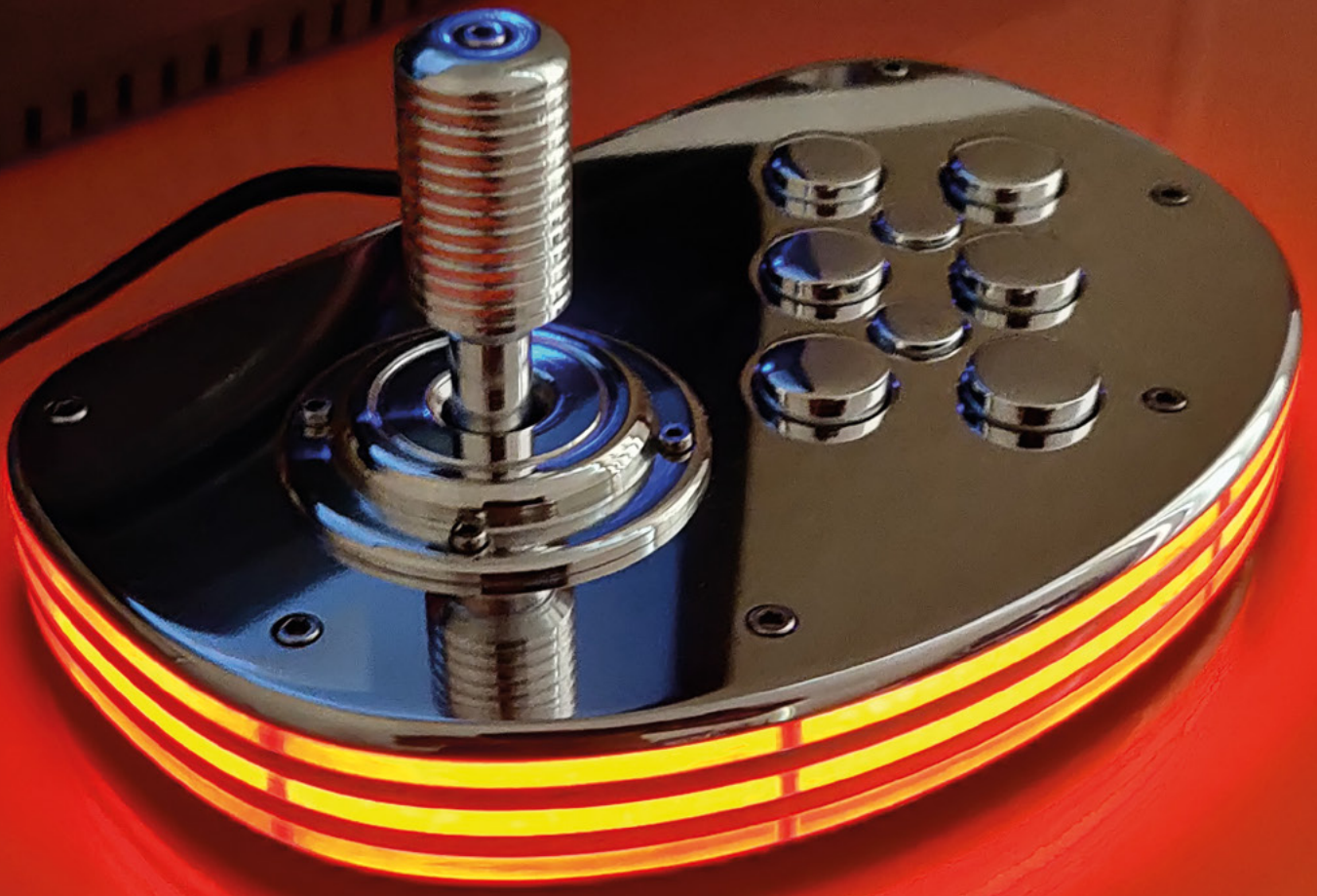
Digital picture frame

By **Democracity**

hsmag.cc/DigitalPictureFrame

We're big fans of upcycling old hardware here at HackSpace Towers, but there is some tech that's just too far gone to rescue. One approach to this is to rip out the guts, but keep the vintage-looking shell of the old kit. That's what Instructables user Democracity did with this Sony Micro TV.

In place of the old CRT that the TV originally used, there's a TFT LCD screen, behind a bezel made out of black acrylic and the original glass of the TV. Images are served up by a Raspberry Pi 4, and the whole thing is held together with a lot of hot glue. ◻

Left ◩
Democracity
wanted to keep
the old buttons
and knobs of the
1960s TV

Scrap-metal horse

By **Rickdadick450**   **hsmag.cc/MetalHorse**

Since Magdelenian people drew on the cave walls at Lascaux, we've felt an artistic connection to horses. Now Reddit user Rickdadick450 has added to the long tradition of horse-inspired art with this scrap-metal sculpture. According to the maker, this took 20 hours of work, and is made entirely of scrap. It's the attention to detail that gets us about this work; the way that the hard angles of the metal come together to suggest the natural shape of the animal's face. Brilliant stuff. ◻

Right ↗
If you like this, you'll love the Kelpies, the giant metal horse heads in Falkirk, Scotland
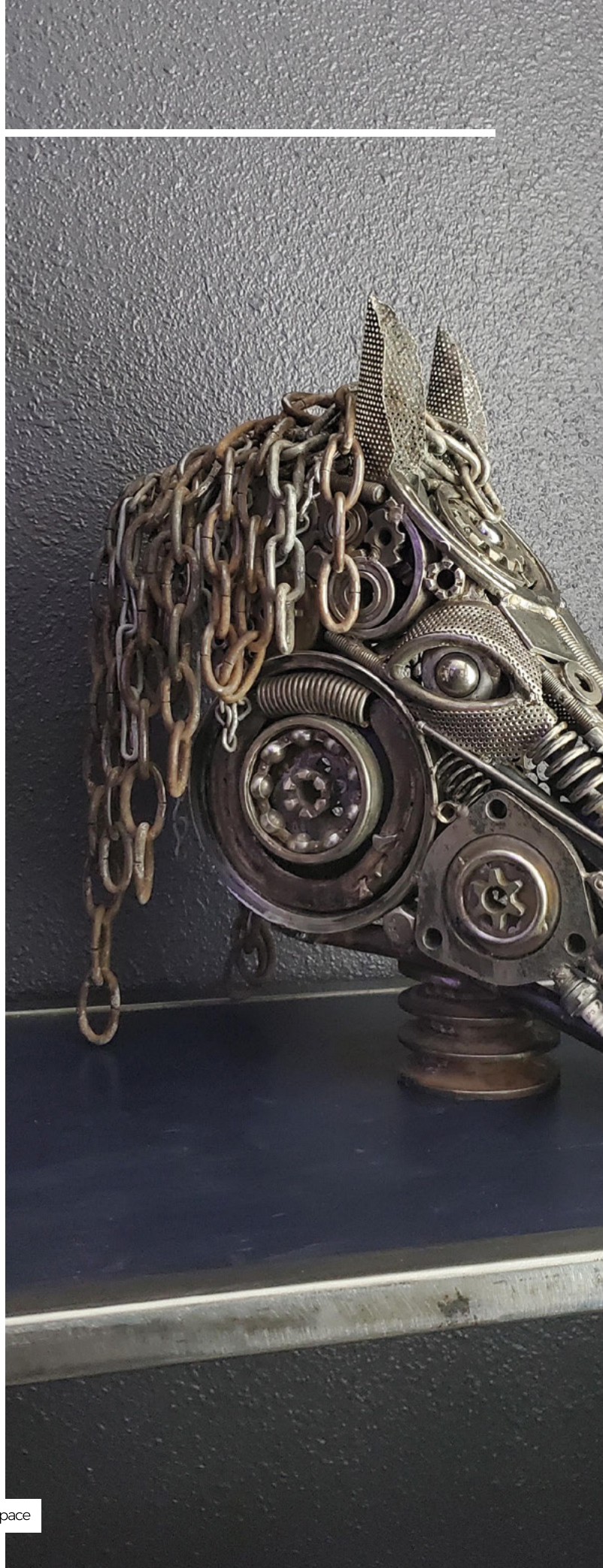
Heat shrink tube cutter

By **Mr Innovative**   **hsmag.cc/TubeCutter**

f you're soldering wires together in an enclosure that's going to live outside, or as part of a machine that moves, you'll probably want to protect the bare wires from the elements (and from each other) with something non-conductive that will wrap around the exposed wire. That something is usually heat shrink tubing. This machine, made by YouTuber Mr Innovative, has been built to cut heat shrink tubing to consistent, pre-programmed lengths.

The machine feeds tubing into two lengths of metal pipe, while a blade passes up and down in a narrow gap between the pipes; it looks like a guillotine, but for plastic rather than decadent aristocracy. Mr Innovative used an Arduino Nano, two stepper motors to drive the moving parts, and a 2.8-inch touchscreen, held together with a custom PCB. Yes, you could use a craft knife instead, but why not automate when it looks as cool as this? ▯

Right ⏎
If we had one of these, we'd watch it all day long. It really is mesmerising
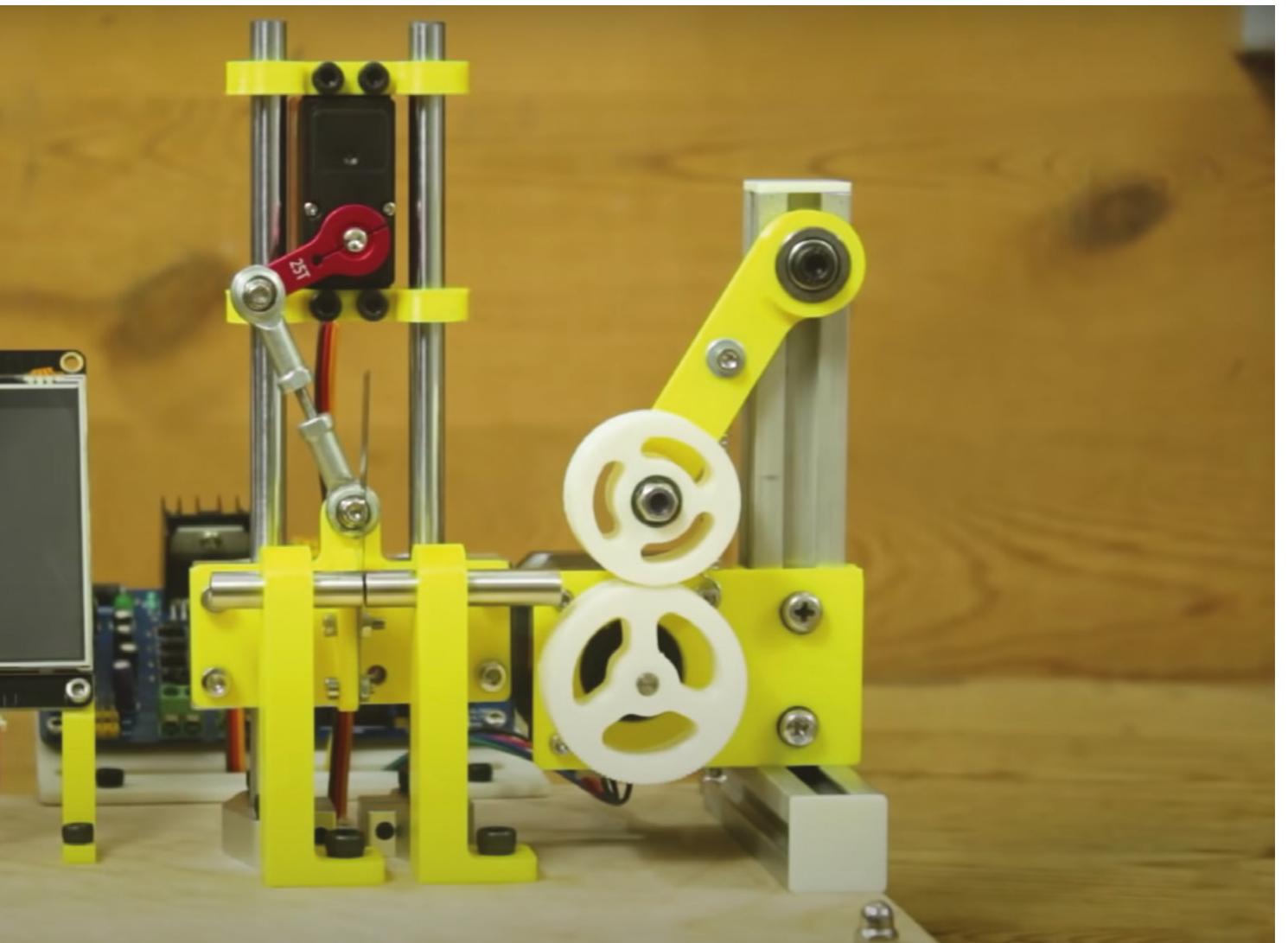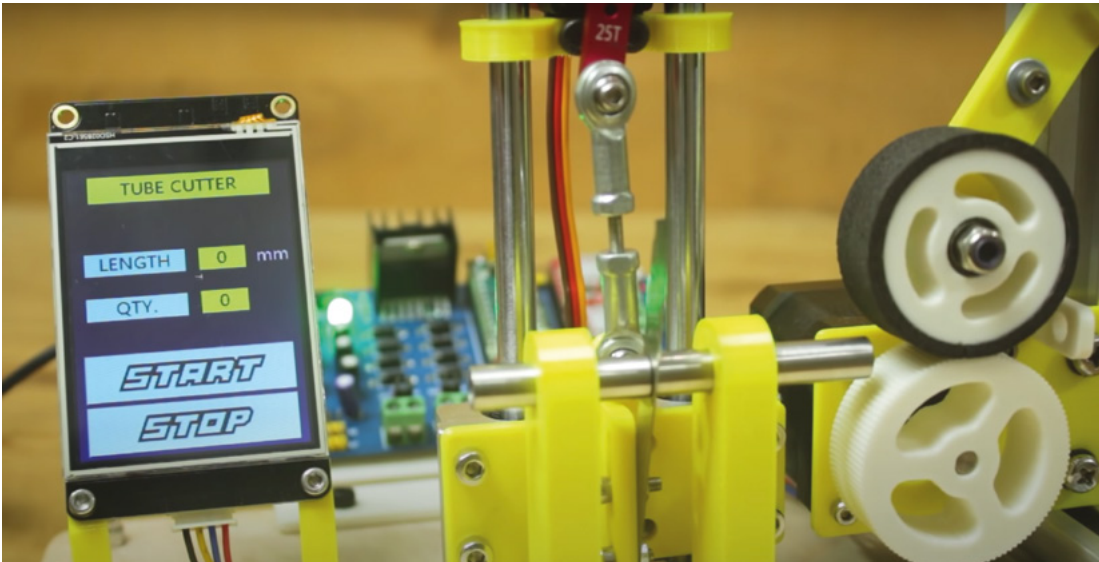
Objet 3d'art

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

We're breaking new ground here at HackSpace towers, for this is the first time we've featured an animation on these pages. Floaty Boaty, shown here, is a 240-frame (very) short film showing the classic 3D-printed Benchy in its element, sailing the seven seas. Well, we say seven seas – in fact there are 240 individual seascapes, designed by visual effects professional Mark Theriault.

Mark's made the source files for this work available at the link below. As well as the 240 slightly different wave scenes, there's a frame to keep the whole thing in one place, a modified version of the Benchy boat, and a peg to hold the Benchy in place on the waves. Mark warns that the process takes hours, so you'll also have to think about lighting if you try this at home. ▫

🡵 **hsmag.cc/FloatyBoaty**

PART
01

A beginner's guide to lock picking and locksport

If you have an enquiring mind, the kind that can focus on a puzzle for a long time, you'll definitely find some aspect of lock picking or locksport that you will enjoy!

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!

Lock picking and other locksport is a popular pastime in many hackspaces and at many maker events. It's a rich vein of interest for a curious mind, and there are lots of different aspects to the scene. In this article, we'll take a look over a small collection of locksport-type activities, and explore some of the basic skills to open common locks. In the second part, we'll look at more DIY aspects of locksport – making tools, stripping and reassembling locks, and much more.

The commonest place to start with lock picking is pin tumbler-style locks. Pin tumbler locks come in a range of types, including the common Yale door-lock in the UK, or Schlage in the US. Many padlocks are commonly pin tumblers, as are European-style door-locks. In **Figure 1**, you can see a cross-section of a tumbler lock. Inside the body of the lock is a barrel that turns when unlocked. There are a number of pin chambers inside the lock – in the case of our diagram, there are five. Therefore, it would be referred to as a five-pin lock. The pins sit in holes that are drilled into the housing and into the barrel. There are two separate pins in each hole – the lower pins are the 'key pins' and the upper pins are the 'driver pins'. Above the pins in each channel are springs that push the pins downwards. These are represented by the purple areas in **Figure 1**.
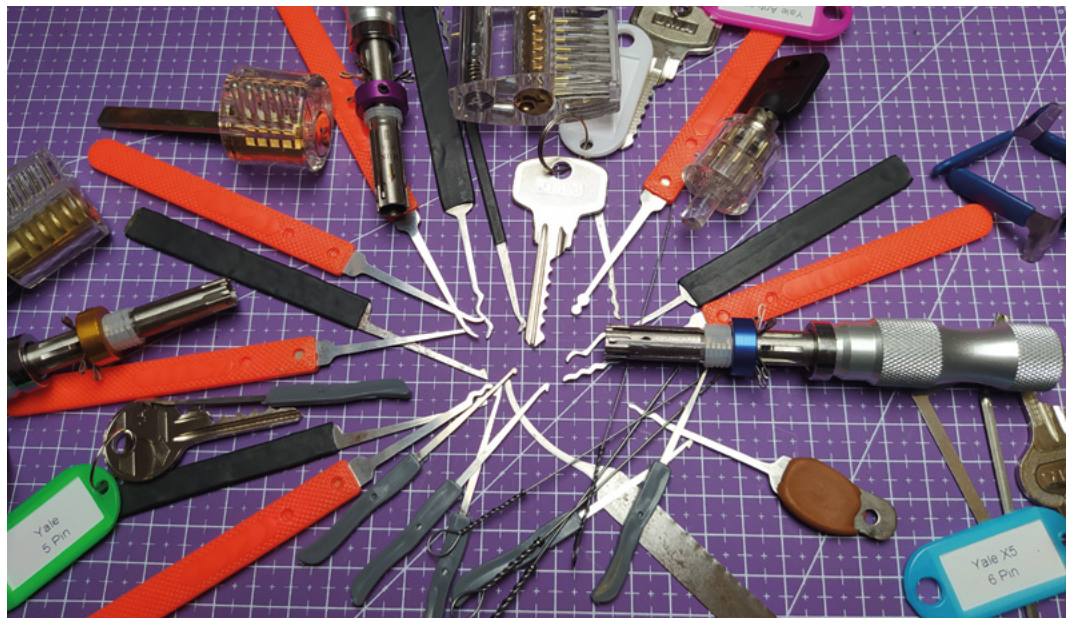
Right ◈
A collection of locksport practice locks, lock picks, and other tools – some commercially bought, and some DIY

When the key is not in the lock and it is in the locked position, the key pins and driver pins drop into the barrel and the driver pins stop the barrel from being able to turn. The key pins are specific lengths and a corresponding key, when inserted, will lift the key pins so that the top of them will align with the top of the barrel. This means that the driver pins are lifted into a position just clear of the barrel. This point, at the interface of the lock body and the barrel, is called the shear line. When the pins are aligned at the shear line, the barrel then can be turned and the mechanism is unlocked.
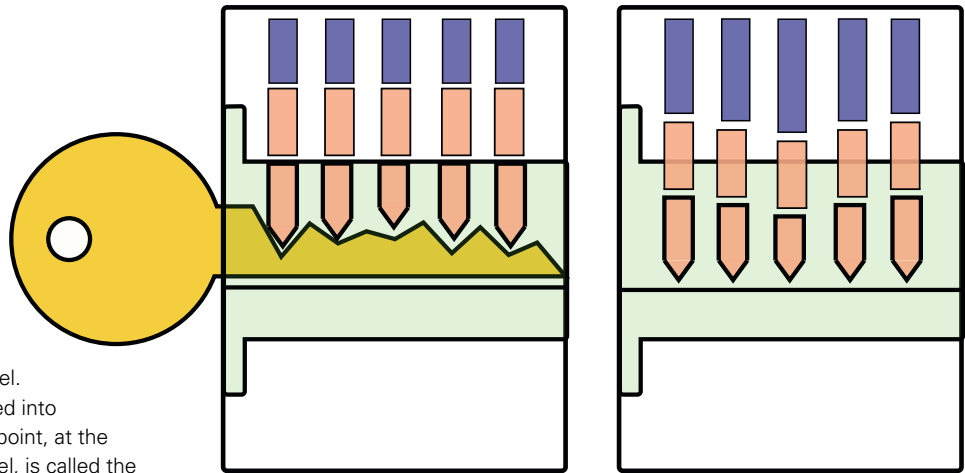
PICKING HOLES

There are a few conceptual approaches to picking a lock, with the most common approaches being 'single-pin picking' or 'raking' a lock. Single-pin picking is a challenging approach to learn, and requires a fair amount of practice to develop the sensitivity to be able to feel the different internal parts of the lock and the small movements you need to make. Single-pin picking, however, is seen as perhaps the most elegant and, when performed well, efficient way to open a lock.

As we will explore, both single-pin picking and raking of tumbler locks rely, in part, on tools that create a rotational force in the lock, similar to the rotational force you apply when using a key. These tools are commonly referred to as tension tools, torque tools, or tension wrenches, and these names
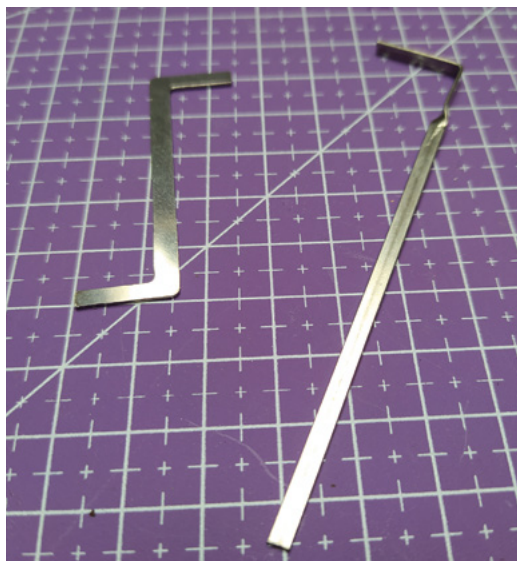
are interchangeable. Two common types of tension tool can be seen in **Figure 2**. On the left we can see a flat-style tool, and on the right a bent flat bar tool. Both these types of tool are great to start using on standard pin tumbler locks, but there are many more types of tension tools out there for you to research and experiment with. It's true to say that of all types of tools in locksport, people are constantly reacting to

> **There are many more types of tension tools out there for you to research and experiment with**

new lock designs, or coming up with new tool designs, so there is lots to research.

The tension tool is used in combination with a lock pick. Again there are no hard and fast rules about lock pick selection, and I'd urge you to read around and look at what other people used and experiment. For single-pin picking, the most commonly used tool is the small half-diamond pick (**Figure 3**), closely followed by the small hook pick. You can see, however, in **Figure 4** that a huge range of lock pick designs exist – but to begin with, you probably won't need many.

Single-pin-picking a pin tumbler lock relies on the fact that no lock can be dimensionally perfect and there will be a slight tolerance in manufacturing that we can exploit. Essentially the exploit is based around the idea that if you apply a rotational force onto the barrel, as if trying to turn a key without a key fitted, then one of the pins will be the first to bind. To try this, take a torsion wrench tool and insert it into the keyway of a practice lock. If you apply a small amount of force, the lock will turn ever so slightly and then stop, or 'bind'. As an exaggerated example in **Figure 5**, you can see a top view of the lock with the third →

Figure 1 ◆
A diagram showing the basics of a common pin tumbler lock, and the locked and unlocked states

Figure 2 ◪
Two of the most common styles of tension wrench

YOU'LL NEED

- ◆ **A practice lock**
- ◆ **A tension wrench**
- ◆ **A half-diamond lock pick**
- ◆ **A Bogota rake lock pick**

QUICK TIP

In the second part of this look at lock picking, we will explore creating DIY tools and lock picks.
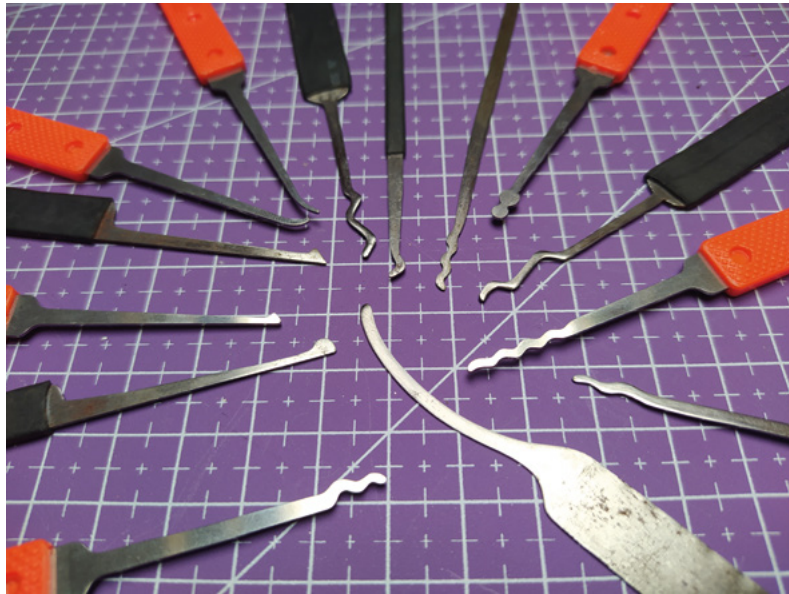
Figure 4 ◆
There are hundreds of lock pick designs, each with slightly different applications, uses, and target locks. However, you don't need many to begin!

Figure 3 ◩
Two common picks for single-pin-picking tumbler locks are the half-diamond pick and the hook pick

QUICK TIP

Keep a note of the pin setting order as you discover it, then it's easier to pick the lock next time.

pin of the pin channels off the centre line. This would be the first pin to bind if the lock was being rotated clockwise. Although that is an exaggerated example, it is true that the tolerances in manufacturing, however fine, will always mean that one pin channel binds before others. Next, let's take a diamond pick or a hook pick and gently insert it all the way back to the back of the lock keyway. If you look at keys for tumbler locks, you will see that there are different grooves cut into the body of the key and these correspond with the shape of the keyway in the lock. This primarily means that the only keys that can enter the keyway

> The tolerances in manufacturing, however fine, **will always mean that one pin channel binds before others**

are limited, one part of the security of a pin tumbler lock. But, for us, the shape of the keyway as we advance may limit or change the type of tools that we can realistically use and fit into the lock.

Having inserted our diamond pick all the way to the back of the lock, bring the pick back towards you and try to feel the rearmost pin. We don't know which pin is binding at the moment, so the test is to try and lift each pin with the tip of the pick. The one pin that is bound will feel stiffer than the others, which should all feel similar and quite loose when you lift them. When you find the pin that is binding, it will feel stiffer to lift. Gently lift the bound pin and, at the moment the pin aligns with the shear line, you should feel that the lock turns ever so slightly more and you may hear a slight click as the tension in the pin releases. You have set your first pin – congratulations! If you release the tension, the lock will reset and the pin will drop back down, so keep the tension on the lock as you continue. To continue to single-pin-pick the lock, you repeat the process of lifting each remaining unset pin to try and identify the new binding pin and then set that pin by lifting it. Continue with this and, when you reach your last pin, the lock will turn under the tension from your tension tool.

Per **Figure 6**, most people begin practising lock picking with the lock in-hand. Curling one hand around the lock, and using your thumb or a finger to apply some tension on the wrench, is a common approach that works well. One benefit, and simultaneously a failing, of this approach is that sometimes you may feel a pin set and the cylinder slightly rotate through the hand holding the lock rather than through the pin. This can be useful,
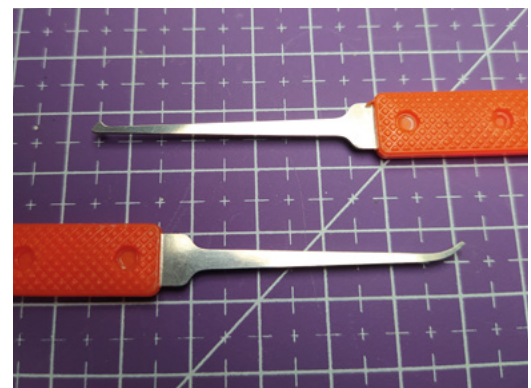
STAYING LEGAL

There are lots of rules you should check out about locksport. Different regions have different laws about owning and carrying lock-picking equipment and tools if you aren't a registered locksmith. Check out these laws in your region. The other two golden rules that you should consider sacrosanct in locksport are: first, never pick a lock you don't own, and secondly, never pick a lock you rely on. The first rule is pretty self-explanatory, but the second one needs a little more thought. Picking locks causes some wear on locks and there is a chance that something may go wrong. A broken and stuck lock pick may permanently damage a lock and, if it is a lock you depend on, then you are in trouble. It's always best to have a set of practice locks to work on and tinker with.
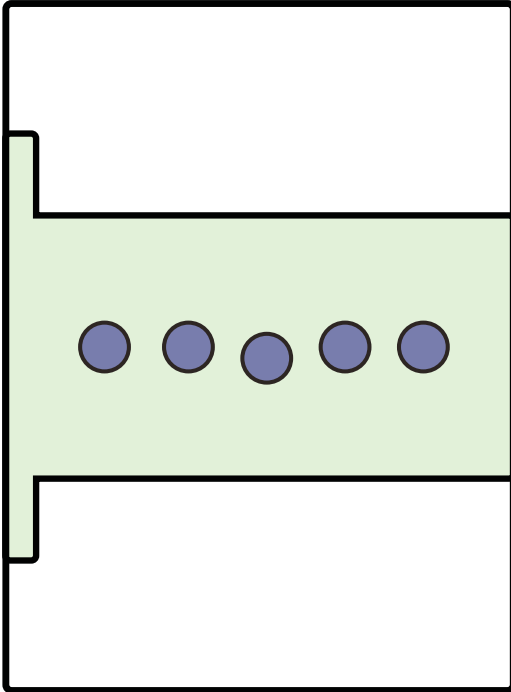
but doesn't emulate picking a lock in a door. A quick fix is to use a small desk vice to hold the lock whilst you work on it. In the next part of the series, however, we will make a small practice board to mount our lock into.

Raking is another manual approach to defeating a pin tumbler lock. It's similar in some ways to single-pin picking, but is perhaps easier to perform as it requires less feel. There are a variety of rake tools and, in the next issue, we will explore making tools for both single-pin picking and for raking. To use a rake tool, (**Figure 7**), you again use a tension tool or tension wrench to apply a small amount of rotational pressure to the lock. Insert your rake tool fully into the lock. As you apply slightly more pressure, you move the rake tool in and out and up and down in the keyway to create rapid movements of the key and driver pins. As you are holding some rotational pressure on the lock core, again as with single-pin picking, pins should set and remain in position as they reach the shear line. Once all the pins have been raked to the shear line, the lock should open and turn using the tension tool.

Pin tumbler locks are common as door-locks, such as the lock pictured in **Figure 6**, but the same pin tumbler technology is also in many Euro cylinder-style locks and also in many padlocks (**Figure 8**). Whilst padlocks are unlikely to be re-pinnable, they can often provide a nice challenge and are often lower-cost

than their door-lock counterparts. It's pretty easy to find see-through plastic locks online that are sold at quite reasonable prices – these are very helpful when getting started.

SETTING YOU UP

'False setting' – If you have a lot of tension applied on your tension tool, it's possible to push a pin past the shear line and for the key pin to bind the lock. You might feel that that pin is set, but it actually isn't. The only way to cure a false set on a tumbler lock is to release the tension, which releases all the pins back to their locked position, regardless of whether they are set correctly or false set. However, if you have →

DO NOT DESTROY

Lock picking and 'locksport' tend to be catch-all terms that cover all non-destructive approaches to opening a lock. Destructive approaches are where a lock is wholly or partially destroyed as a process to open the door or system. Destructive approaches can vary from drilling and cutting to destroying with plasma cutters or even thermite! Needless to say, locksport is primarily looking at the non-destructive methods of successfully bypassing or opening a lock. Within non-destructive methods there are numerous approaches, a main distinction being 'manual' or 'automatic' approaches. Manual describes, when applied to common pin tumbler locks, the use of a lock pick and a tensioning device, which are moved inside the lock by hand. Automatic approaches include the use of items like 'picking guns' that automate movements using mechanical systems. Whilst interest in both manual and automatic techniques is prevalent in locksport communities, we are going to stick with manual techniques as these are the most common starting points for beginners.

Figure 6 ◈
Practising lock-picking a pin tumbler lock in-hand makes this a very portable hobby!

Figure 5 ◩
An exaggerated top view showing how manufacturing tolerances mean that pin chambers are not perfectly aligned
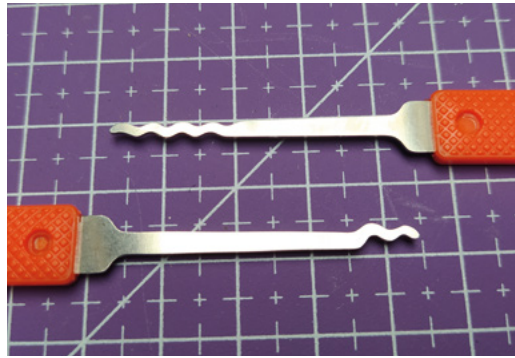
worked out a small part of the sequence of which order the pins set, your previous efforts aren't all in vain!

The budget-type, starter lock-picking locks usually all feature standard driver pins which are all the same length and are a simple small metal cylinder. More advanced security driver pins exist that have different geometries which can increase the challenge of picking. Spool or mushroom pins are designed to false-set easily, as they have thinner sections which means, as they are lifted, they can pretend to unbind the lock and allow the core to turn slightly before binding on the thinner part of the security pin. This can trick you into believing you have correctly set that pin to shear line. There are numerous techniques you can research to overcome these pins, and you can either collect these pin types from locks you have stripped or re-pinned, or you may be able to buy small amounts of security pins directly. In the second part of this series, we will look at how to re-pin a lock, which allows you to change the pin order in a practice lock, and also allows you to fit security pins into a lock to practice with.
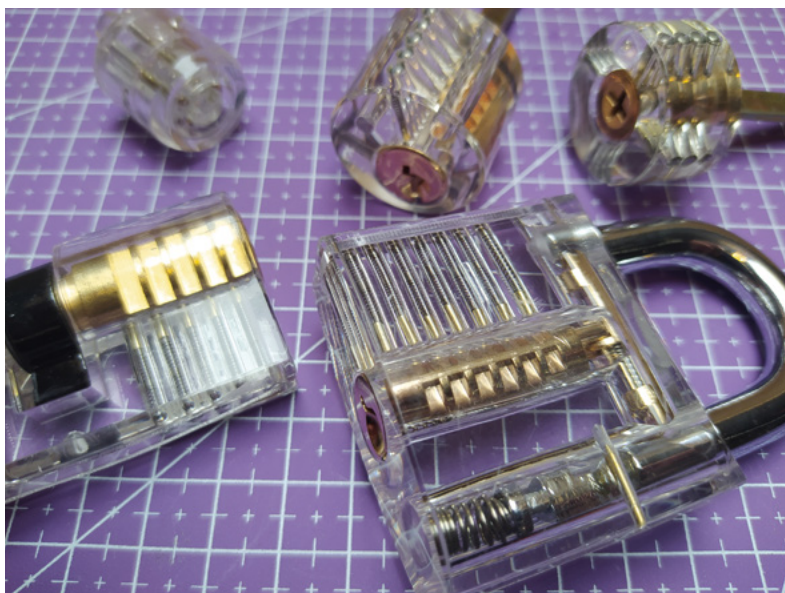
RIGHT COMBINATION

Another accessible area to begin locksport is the manipulation of combination locks. Before you go and buy a safe to practise on, it's probably a good idea to start a little smaller. Dial and ring combination locks are widely sold as padlocks for lockers, or for garden storage containers, etc. At the budget end of these locks, many of them have been explored for manipulation-type vulnerabilities. A couple of classic locks to explore, which have had large amounts of exploration in the locksport communities, are the 38 mm Master Dial Combination lock (model number 1533), and the Master Lock 7640 (**Figure 9**). Lock manipulation on these padlock-style combination locks have common themes but a variety of approaches. A common theme is that you apply pressure on the lock's shackle, pulling away from the lock as if trying to force it open. Applying pressure and then rotating the dials allows the locks to give clues as to their combinations, by perhaps having stiff-to-move sections, or making a different noise or a slight movement when certain parts of the mechanisms align. Researchers have often worked out and published how these clues can be processed to ultimately come up with the correct combination to open the lock. In the case of the Master Lock 7640, you can (with the lock unlocked) reset the combination to whatever you choose. Applying pressure to the shackle on the 7640, one approach is to simply work from the furthest away from the shackle dial and feel when the shackle gives/moves slightly, indicating the correct number has been found, then moving on to the next closest dial. We found that with our 7640 we could, with some practice, successfully open the lock from an unknown combination set by someone else in less than ten minutes. If you haven't got someone to set the combination, you can test yourself by setting the combination under a towel and mixing the dials. The 38 mm Master 1533 is a classic lock for manipulation and if you search online, there are numerous manipulation techniques and approaches. Most use feedback from manipulation to find the first number directly, and then there are some simple maths approaches to work out the third number directly, and then to curate a reduced list of possibilities. Approaches vary, but the website **hsmag.cc/8tries** shows a technique and has tools that mean you can open this lock by only ever trying a maximum of eight combinations.

Another area of interest is 'bump keys' or 'bumping locks' (**Figure 10**). It's an approach that brings us close to some of the automatic techniques for opening pin tumbler locks, but still using manual tools. A bump key

QUICK TIP

If you are struggling to feel separate pins, spend some time just probing the front pin in a lock with a half-diamond pick as practice.

is a key made from an official key blank, just like the type a locksmith would cut a replacement or spare key from for your lock. Therefore, it fits into the keyway of the target lock. A bump key is cut in such a fashion that, for each pin, the corresponding valley in the key is cut to the maximum depth possible for that lock type. The bump key is inserted into the lock and then removed one bump or one pin's worth. You then apply some tension to the key using your fingers and then, using a small block of wood, a tiny hammer, or commonly the back of a screwdriver, you give the bump key a sharp tap. Inside the lock, the angled parts of the key valleys transmit the energy of the tap into the pins, thus causing the key pins to hammer up the driver pins. This is often described using pool as an analogy, where you strike the cue ball into an object ball and the cue ball stops dead and the object ball moves away. As the driver pins are shocked away from the key pins, this can create a gap at the shear line and the rotational tension you are applying to the bump key causes the lock to open. A common addition is to place an O-ring over the shoulder of the bump key to allow the key to bounce back into its partially retracted position, allowing you to reset and bump again quickly and repeatedly as needed.

We hope you have enjoyed this overview of some of the basic manual approaches to locksport. Next month we will look at some easy ways to make lock picks, tension wrenches, and other tools from common items, to allow you to build up your own personal locksport tool-kit. □

TOOOL

There are heaps of tutorials and guidance online and many places to discuss locksport. A great forum full of all different locksport information is an online forum, **lockpicking101.com**, which has sections covering a huge range of different aspects of locksport, from creating DIY tools through to exploration of high-security lock design, and much more. The other great resource for getting into locksport is local groups or, indeed, lock-picking events at Maker Faires or at your local hackspace. One international organisation, TOOOL (The Open Organisation Of Lockpickers), has chapters all over the world, and many chapters have their own websites with their events listed and a host of other locksport resources.

Figure 9
Lock manipulation is an interesting area of locksport. Simple combination padlocks often have documented vulnerabilities that allow them to be opened with manipulation

Figure 10
A collection of commercially bought bump keys that fit a range of Yale-style keyways

Build HAT

Control LEGO® from your Raspberry Pi

LEGO® **has introduced generations of children to making.** The little blocks slot together in almost endless possibilities and, if you add in mechanical components from Technic sets, you can create some complex builds.

Through the years, LEGO® has built a few different programmable electronics systems for adding silicon brains to your builds. MINDSTORMS® and SPIKE Prime Sets have given users an interface for controlling their plastic bricks with code and, through this, have introduced countless children to physical computing.

Of course, LEGO® is not the only organisation helping children learn physical computing, so perhaps it's inevitable that LEGO® and Raspberry Pi have worked together to create the Build HAT. This add-on to Raspberry Pi computers lets you control the latest generation of LEGO® motors and sensors from Python code running on your Raspberry Pi. Here, of course, you have access to the full range of modules and hardware that you can usually connect to your Raspberry Pi. With extended headers, you can connect additional HATs, or other components, to the GPIO pins as long as they don't use the UART connection (14 and 15).

To make it easy to mount your Raspberry Pi, LEGO® has created the Maker Plate. This lets you attach your Raspberry Pi securely to your build.

MORE POWER

While you can power the Build HAT directly from Raspberry Pi, the motors run at 7.5 V, so you'll need an additional power connection to the barrel jack on the Build HAT if you want to run motors. Raspberry Pi has also launched an official power supply for this.

There are four connections on the Build HAT that can talk to sensors and motors from the SPIKE Prime kits, or the most recent generation of LEGO® Technic. The available options include motors with integrated sensors for precise control, distance

sensors, force sensors, colour sensors, LED matrices, and more.

The Build HAT and power supply are available now from all the usual Raspberry Pi suppliers, for $25 and $15 respectively.

While the Build HAT was designed for education, there's plenty for makers to get their teeth into. Want to live-stream video from your LEGO® car? Add a Raspberry Pi camera. Want to trigger your LEGO® build from data on the internet? If you can get the data onto your Raspberry Pi, you can send it to your LEGO® build. Want a platform to quickly prototype an idea for a project? There aren't many quicker ways of building than plugging LEGO® bricks together. ◻

Right ◈
Add cameras and other Raspberry Pi-compatible sensors to LEGO®
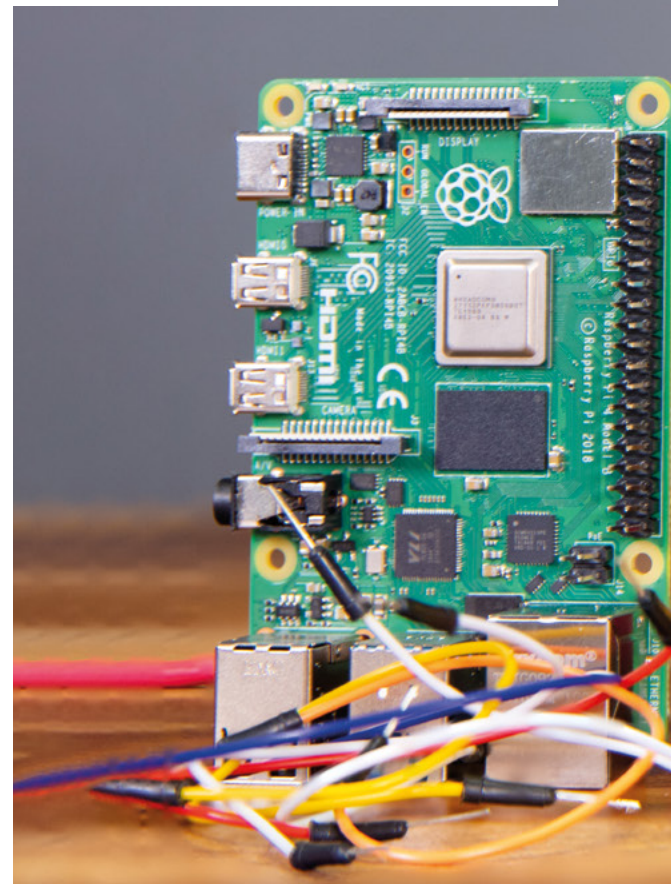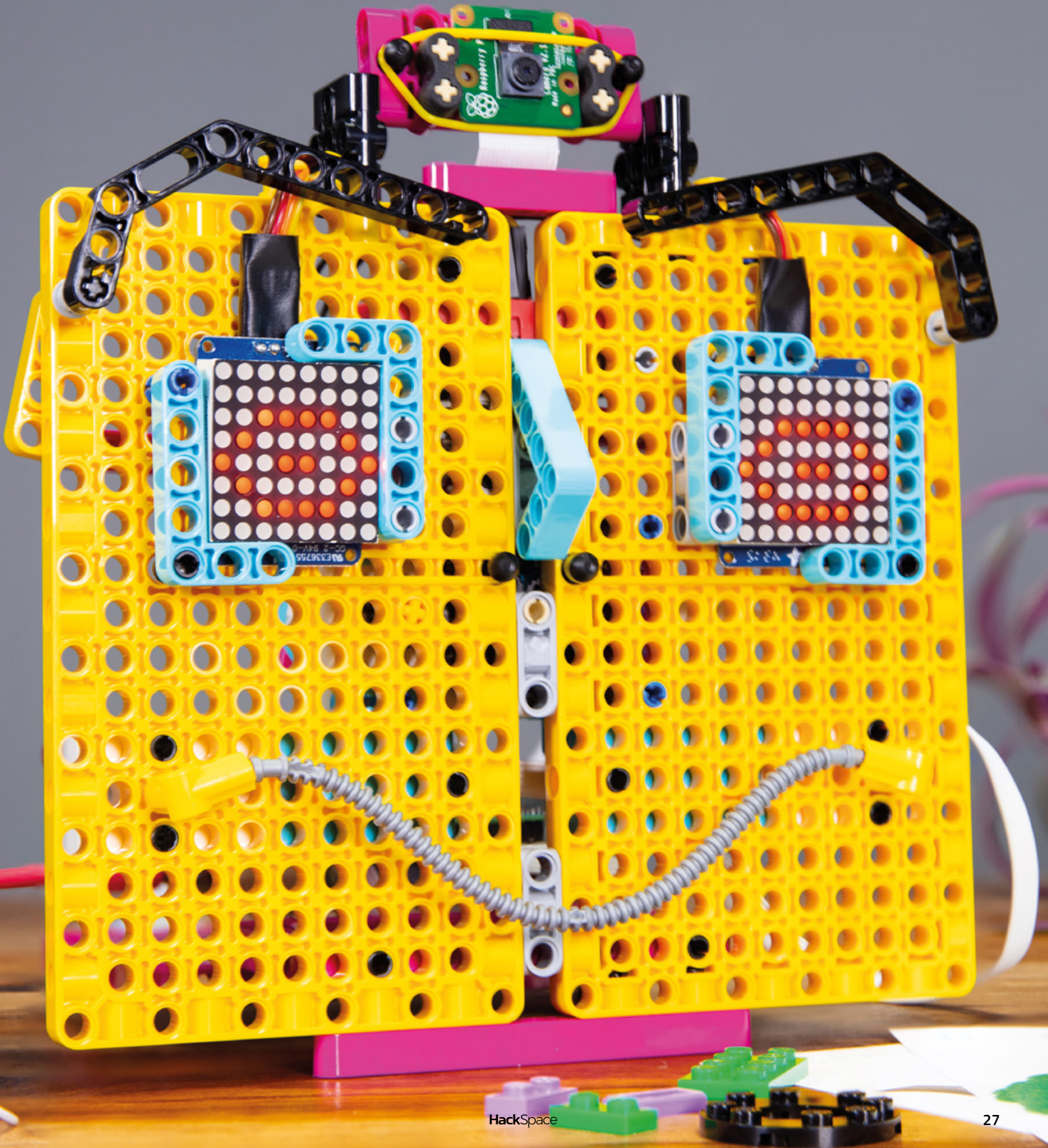
Letters

3D PRINTING

It seems that you're starting to take 3D printing seriously. First there's your columnist talking about using 3D printing to renovate her house. I like this; for anyone lucky enough to be able to buy a house, it's the biggest expense of your life. If 3D printing is mature enough that she trusts it with her house, that says a lot more than printing Marvel or Dungeons & Dragons figures, as pretty as they are.

And now in this issue there's '50 top 3D prints'. I can see myself using a few of these! Sometimes it's a pain to sit on the ring road for an hour each way to get to B&Q for a simple tool. But if I can print a tool in a few hours, then I can set the printer away and fill the time with something else useful. I get it now. I'm a convert.

Ian Wood
Southampton

Ben says: From the outside, 3D printing looks like a cult. But it's not: it's just a tool. And just like when you get a hammer, everything's a nail, when you get a 3D printer, everything can be solved with PLA. Need a woodworking jig? Print one. Need a picture hook? Print one. A switch? Print one (never mind that you can get switches for pennies, with overnight delivery – we've still printed them). Once you've solved a couple of problems with a 3D printer, you'll start seeing other things that you can do with it. You go from solving problems to creating opportunities. And if that starts with printing a D&D figure, so be it.
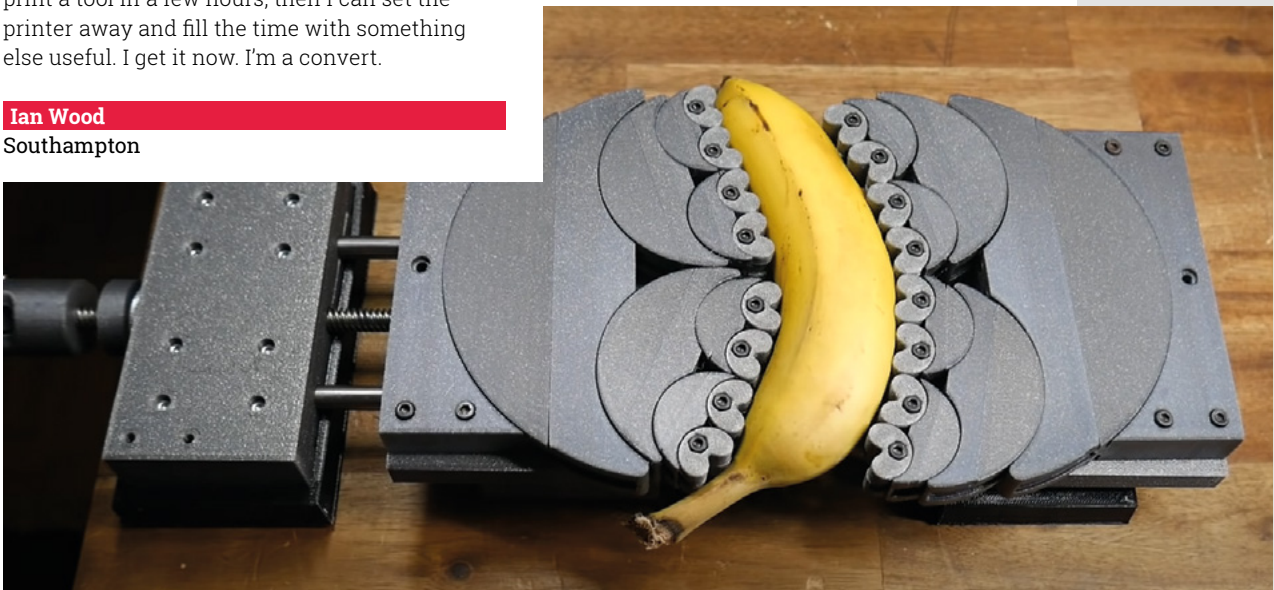
PCB PLEASE

Your series on KiCad has been incredibly useful. And even more so now that you've delved into PCB design with it. I've been mucking about with circuits on perfboard as a step up from having everything on breadboards (seriously – I know they're meant for experimentation, but I've got projects that reached breadboard stage years ago and are still working fine, thank you very much). But perfboard feels like a bodge (yes, even more of a bodge than breadboard).

So thanks for getting me started with the KiCad extension of FreeCAD. I'm still a way off getting things made, but it feels like I'm getting there.

Brian
Stoke-on-Trent

Ben says: Have we got a treat for you! Turn to page 34 for the complete guide to designing, assembling, and troubleshooting your first PCB. It's been a bit of a journey getting things set up, sent to get manufactured, then getting the PCBs back and finding out that I'd made an elementary mistake and have ended up with fancy coasters… but, as the old saying goes, it's good to learn from your mistakes, and it's even better to learn from someone else's.

OSKITONE

"Technology isn't the answer to everything" – I didn't expect to see that line from a man who makes and sells technology [Oskitone's Tommy Marshall, who we spoke to for issue 47]. But it's true!

"I don't like computers. I don't like them at all" – the man has a window into my soul! I feel seen, as the kids would say!

In all seriousness, technology is not the answer to everything. We need to go for walks in the trees, get out into nature, talk to people… we need food, shelter, and to tinker with analogue through-hole DIY electronic synthesizers. I know what I'm asking for this Christmas.

Stephen
Croydon

Ben says: I too liked what Tommy was saying in that interview. I love technology, both for the things we can do with it, and the fact that there's always something new to learn. But at the same time, I get frustrated with computers. Spending a weekend building an analogue synth, then a Sunday afternoon playing tunes on it, is embracing technology, but it's a different sort of pastime to playing with a computer. That's why we like basic electronics: you can build something that it's possible for one person to understand. When you know that you've soldered every component of your new toy, you get a wonderful sense of understanding, of being at one with the instrument as you compose your latest masterpiece. I also know what I'm asking for this Christmas!

CROWDFUNDING
NOW

Thumby

Is this the smallest gaming device?

From $19 | hsmag.cc/thumby | **Delivery:** February 22

When it comes to technology, how small is too small? Does such a thing exist, or is smaller always better? TinyCircuits specialize in making itty-bitty electronics. Previously, they've made development boards, arcade machines, and even other handheld games machines. However, Thumby, at just 29.5 × 18 × 8.5 mm is their smallest device yet. In that diminutive space, it packs a 72 × 40 pixel monochrome screen, a four-way D-switch, and two buttons. It comes pre-loaded with five retro-style games, and can be programmed in either MicroPython or Arduino C++.

Powered by RP2040 – the same microcontroller as Raspberry Pi Pico – Thumby has plenty of processing power that you can harness for your gaming pleasure. Along with two processing cores running at 125MHz, there's 2MB of flash for your games.

Obviously, with a device this small, poking tiny buttons becomes a problem. We've not been able to get our hands on one, so we can't say how easy it is to play. We suspect that the small size will force developers to think about the controls a bit more. If your gameplay requires unforgiving and exact button

> **Thumby is undoubtedly a novelty device,** but sometimes constraints build great art

presses, that might be a problem. The other side of this is that it's really easy to carry around with you. It's small enough to fit on a key ring (though again, we can't say if it's tough enough to last long on the rough-and-tumble environment of a key chain).

We're particularly interested in the link cable. This lets you use a micro USB port to connect two Thumbys together for multiplayer gaming. How well you can fit the graphics for two-player games on a 72 × 40 pixel screen is a challenge for you.

Thumby is undoubtedly a novelty device, but sometimes constraints build great art. The diminutive size makes it look fun, and that alone is a great motivator to get people up and programming their own games. ◻

BUYER BEWARE ⚠️

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

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG **34**

DESIGN YOUR FIRST

PCB

Give your electronics projects
their final form

PG **50**

HOW I MADE:
AN AUTOMOTIVE HEAD UNIT

Convert analogue signals from
your car to digital, graphical
loveliness with a Pico

PG **56**

INTERVIEW:
WESLEY TREAT

If you've ever wanted a tiny grill
that's perfect for cooking two
sausages, he's the man for you

PG **64**

IMPROVISER'S TOOLBOX

What masterpieces can you
make with all that spare
bubble wrap?

DESIGN YOUR FIRST
PCB

Creating permanent circuits needn't be complicated

CB design may seem like a hugely technical endeavour that's the remit of elite engineers, but the truth is, it's not. If you can build electronics on a breadboard, you can design a PCB. There's no magical difference. Instead of wires, you just have traces of copper. That's the only difference.

Turning PCB designs into real boards used to involve either harsh chemicals or harsh prices. While the home-spun route still requires some careful use of powerful chemicals, it's now easier and cheaper than ever before to get your PCBs manufactured. Small boards are often only a few pounds, including delivery, and only take a few clicks to order.

While you can use protoboard to make your projects permanent, there's a whole heap of advantages to building custom PCBs.

- You can get the PCB any shape you like, so you can fit it into any chassis or enclosure
- It's easier to make your circuit small
- It's much easier to use modern surface-mount components
- They can look great
- They can be structural in your makes
- It's easier to make your design rugged

There's no one thing on this list that is the main reason for everyone. This author particularly enjoys PCB design because it can help make great-looking builds, but for you, the reason might be different.

Fortunately, whatever the reason, it's the same easy process for designing a PCB that can be manufactured. We're going to create a PCB that's mostly aesthetic, but you can build whatever you like. Now, let's take a look at the software we'll use. →

KEY CONCEPTS

Build up from the basics

I n this tutorial, we're going to use EasyEDA. It's good for entry-level PCB designs. It's fairly easy to use and you can get started without having to install any software. Point your web browser to **easyeda.com/editor**. You don't have to sign in to an account, but if you do, you can save your projects in the cloud.

There are two basic steps you need to do to create a PCB: design the schematic and then convert this into a PCB design. EasyEDA does both of these. Technically, you don't have to create the schematic first – you can dive

There are two basic steps you need to do to create a PCB

straight into the PCB, but there's a high chance something will go awry, so we'd strongly recommend starting with a schematic. You've probably seen schematic diagrams before. They show how electronic components connect together in your circuit.

We're going to create a name badge. This is a nice easy test for your first PCB. It contains your name in the middle – this is in exposed metal, so reflects the light from the LEDs. There are also two touchpads that you can use to control the LED animations.

We've used entirely through-hole components, except for Pico, which we've surface-mounted.

However, you can use whatever you like. Switching to surface-mount LEDs and resistors would let you make the PCB a little (or a lot) smaller if you wish. The advantage of designing your own PCB is you can use exactly the parts you want to use.

We've only used a few components, so it should be easy to understand:

LEDs A triangle, bar, and two zigzag arrows show these light-emitting components in your schematic. EasyEDA colours them in for an easy reference.

Resistors There are two ways of representing resistors in schematics: you can have zigzag lines, or a thin rectangle. The former is more common in the USA, the latter in Europe. They are both completely equivalent, and you can use either in EasyEDA by selecting the EU or US option in the resistor drop-down.

Headers These are any connections, whether they're pins, solderable holes in the PCB, or some custom connectors.

Touchpad There's no specific symbol for a touchpad, so we've used a header. There's nothing special about the touchpad, it's just an area of exposed metal that you can touch.
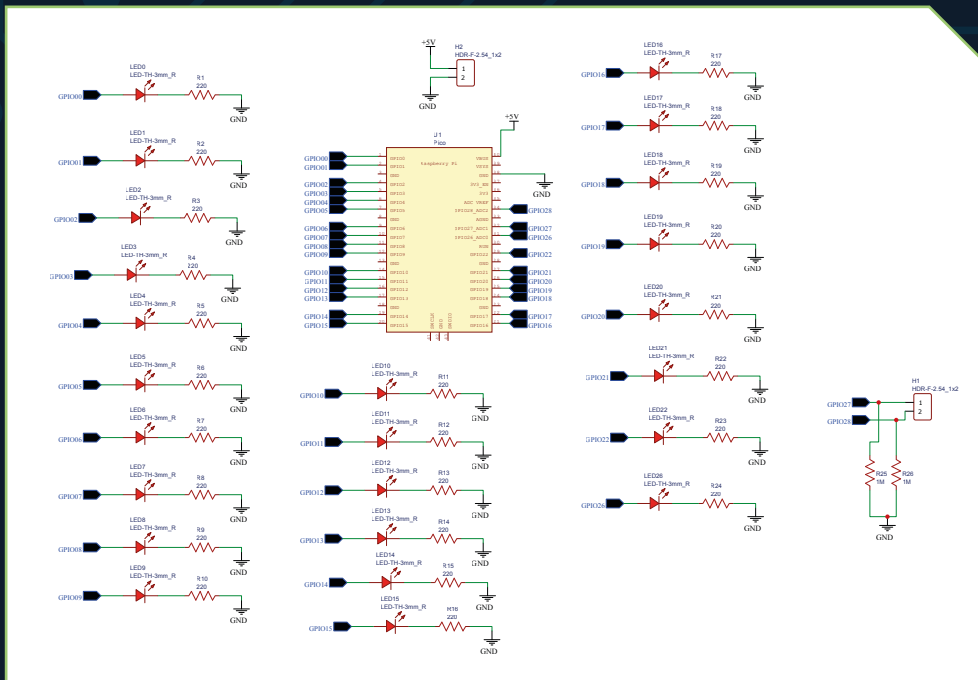
Pico Larger modules are represented by rectangles with the relevant information written in them.

There are a couple of things about this that you may be unfamiliar with if you've only used schematics to build circuits before.

Firstly, each component is linked to a physical design of its mounting on the PCB (known as a footprint). That means that when we add an LED, we don't just add any LED, we add an (in our case) 3 mm through-hole red LED. The colour isn't critical – it's just what will show up in the 3D render – but the size is. You might decide you want 5 mm, or 0805 SMD LEDs. The choice is yours, but you make the choice when you're adding components to the schematic. You can alter it afterwards by highlighting the component in the schematic, and changing the footprint in the right-hand pane.

The second thing you might not have come across is nets. Net, short for network, is a connection between two or more components. If you built your circuit entirely out of wires, the majority of connections would be a simple wire between two components. In this case, each wire is a net. However, in some cases, you might have a single connection that spans multiple components – for example, you might want to connect lots of LEDs to ground. In this case, the ground net might be made up of multiple wires. In each case, though, it's a continuous connection that doesn't go through any components.

In a schematic, components are linked by wires, but this can very quickly become very messy, with wires going all over the place. An alternative is to use net labels, which are labels used instead of wires. In **Figure 1**, you'll see that we've connected each GPIO to a net named after the GPIO pin. We then use this net to power an LED. EasyEDA knows that the two are connected and acts as though a wire is linking the two. You can link more than two things to the same net. For example, we've linked lots of things to the ground net.

When you don't explicitly label a net, EasyEDA will generate a name for it based on the names of the components that it links. It's important to know that every connection has a name. →

Figure 1
The design for our PCB – there are lots of LEDs, a power connector, and two touchpads

Below
You can view this project at hsmag.cc/PCBBadge
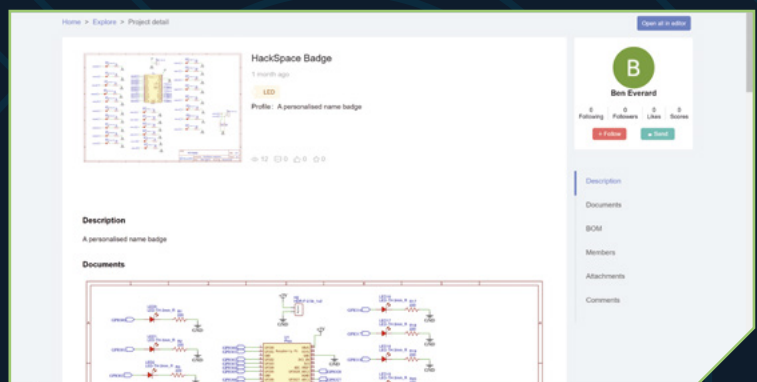
ADDING
COMPONENTS

Building the first schematic

asyEDA Designer is an online tool at easyeda.com/editor. You don't have to create an account to use this, but if you do, you can keep your projects online and easily access them from any computer.

Go to File > New > Project to create a new project.

First, let's add the LEDs. LEDs come in all sorts of shapes and sizes, so it's important to make sure you've got the right footprint attached to the symbol. In the EELib tab, scroll down until you find the LED symbol. There's a drop-down underneath it that lets you select the right footprint. Each one comes in Red, Green, and Blue flavours, but this doesn't affect our circuit in any way. We used LED-TH-3 mm – TH stands for through-hole, and we used 3 mm LEDs. You can use whatever size you like. Different LEDs will give different final effects. For example, you could use larger surface-mount LEDs to create a line effect, or smaller surface-mount LEDs to create more of
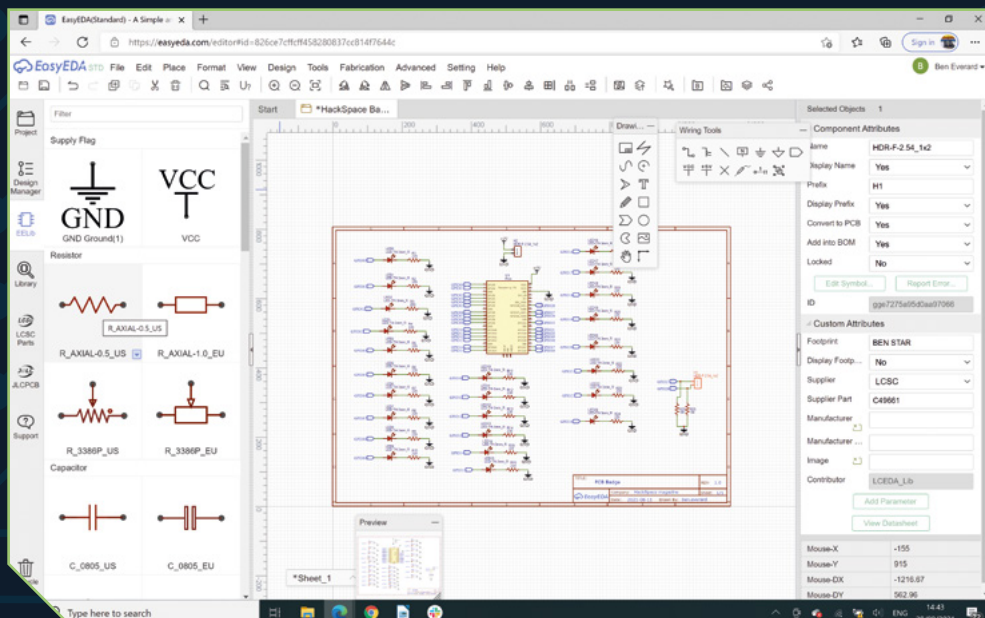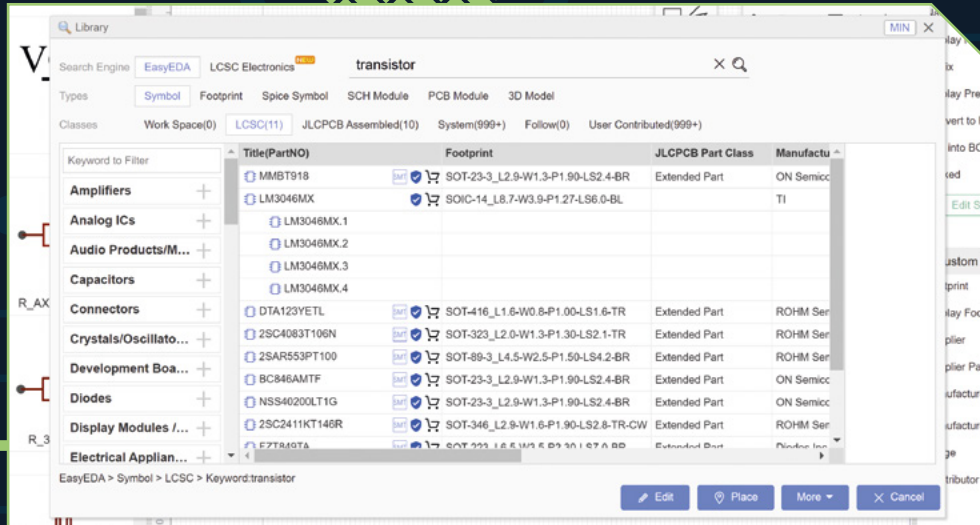
a sparkle. If you go up to 5 mm LEDs, then you might end up with a wide PCB – which could be useful if you've got a lot of text to fit on there.

Once you've dragged your LED on there, we also need a resistor. Again, these come in different footprints, so make sure you get the one that's right for you. The through-hole options are given in proportions of an inch. We've used AXIAL-0.3 which fits smaller resistors and keeps things compact, but you can go for 0.5 for larger resistors or a bit more space. By default, they'll be 1 K resistors. Once they're on the schematic, you can highlight them and change that to 220 ohms.

Next, we'll add the headers and connect them up. We'll use HDR-M-2.54 and you can use the drop-down to select 2 × 2. There are two sets of 2 × 2 headers; one is for power, the other will be used as a touchpad.

So far, so easy. LEDs and resistors are both common components in more-or-less standard packages. What happens if you want a component that isn't in the standard library?

Well, there are two options: EasyEDA lets users share footprints, and you can search to see if someone else has created a footprint that you can use. Alternatively, you can create your own. Let's look at both of these options now.

User-contributed footprints are a bit of a risky option because there's no guarantee that the person designing it had any idea what they were doing. They may not even have used it themselves. If you use one, it's entirely possible that you'll find the PCB doesn't work

when it arrives. However, they can be a very quick way to get a footprint, and there's a huge selection to choose from. If you do decide to use a user-contributed footprint, then you should spend some time checking it against your parts datasheet to make sure that you're happy with it before committing it to your design.

Since we're creating a name badge, we want the mechanics of it – the Raspberry Pi Pico that drives the interface – entirely hidden away. There

User-contributed footprints are a bit of a risky option

are lots of Raspberry Pi Pico footprints available, but most of them have holes – either for pin headers, to access the test points, or for the USB components. We want a Pico footprint that's entirely one-sided, so we created one.

Click the Library button, and make sure EasyEDA and Symbol are selected, then enter Pico One Sided into the search field. User-Contributed should now be an option, so click on that. The one you want is Raspberry Pi Pico Castellated One Sided by user Ben Everard (make sure it's got castellated in the name – this author created another one while writing this article that doesn't work properly, but there doesn't seem to be an option to stop sharing it). Double-click on this line and you should be able to add it to the schematic. →
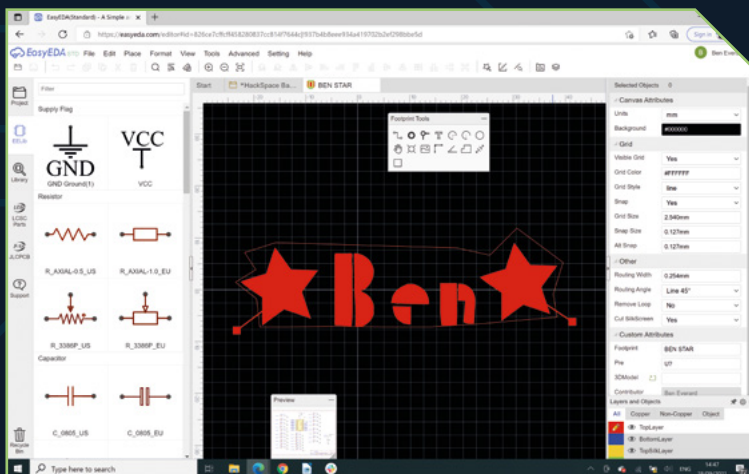
ADDING

ART

Creating our own footprints

That's all our standard components added. The one thing left is the decoration on the front. This is partly functional as we can use the exposed metal on the front as touchpads. However, this isn't a one-size-fits-all solution as obviously we have names of different lengths. This author goes by a three-letter name, so to fill the space, he added a couple of stars. You might have a longer or shorter name, so we'll leave it up to you. The important thing is that this exposed metal can be either a touchpad or simply decorative. In our circuit, we've used the two stars as touchpads, but you can use letters if you'd rather. You can also make more than two touchpads, but you'll have fewer pins spare for LEDs. The choice is yours.
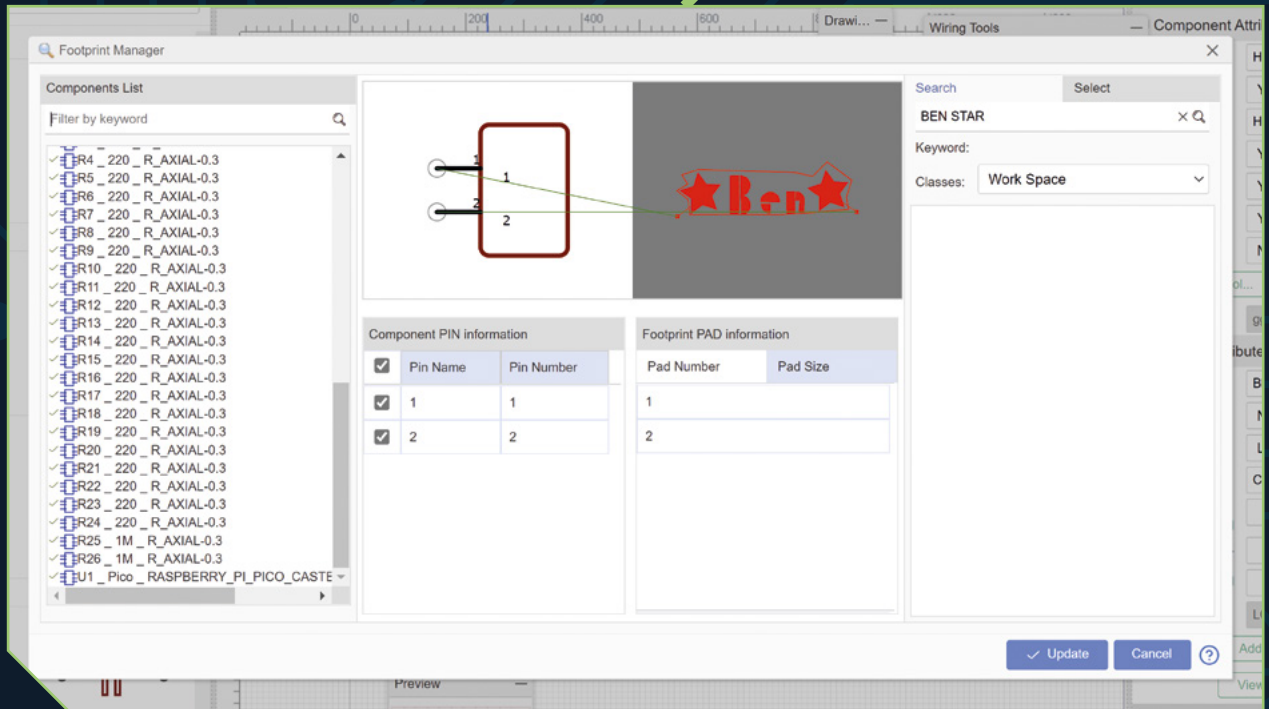
The exposed metal graphics on the front area are one footprint that we created in a graphics editor. EasyEDA can import image files that you can import from other graphics programs. We used Inkscape to create ours. We started a new file, added the text with the font and effects that we wanted, then we selected this text and used Path > Object to path to convert it into a format EasyEDA can use. While EasyEDA can import SVG files, we found that it corrupted these and it worked much better to import a PNG.

In EasyEDA, go to File > New > Footprint, then File > Import > Image. Select the image file you want. There are sliders to control the corners and simplification; however, we haven't found these to be particularly useful at all. There's a certain amount of simplification that seems unavoidable.

You'll also need to select the size at this stage. You can obviously make it as big or little as you like. For reference, our final board is 76 mm wide.

When you click Insert Image Into PCB, EasyEDA will put the image on the top layer of the board. This is where we want it, but it won't expose the metal. To do that, we also need to add our shape to the top solder mask layer.

Slightly confusingly, the top layer of copper is inclusive. Where you put a track or an image – basically anywhere that shows red – is where copper goes. The solder mask layer is exclusive. That means that where you put a shape is

Below ◈
The Footprint designer lets you customise almost any element of how your component fits into the PCB

where the solder mask (protective layer that goes over the top of the PCB that's traditionally green) doesn't go. Therefore, you want the same image twice placed directly on top of itself to add the copper to that area and remove the solder mask.

Copy and paste the image on top of itself, then highlight the name and right-click. Select Attributes and change the layer to TopSolderMask.

We also want to stop traces running through the name as that will impinge on the effect we're going for. To do this, select the 'Solid region' tool and draw a solid area tightly around the graphics. Right-click to finish drawing. However, we don't want a solid region, we want an empty region, so press **ESC** to exit the tool, then highlight the region and change the drop-down in the right-hand pane to No Solid.

We're almost there now. We've added our graphics and connected everything up. We just need to add pads that our traces can connect to. Click on the Pad tool (the circle in the tool box) and add the ones you want. Press **ESC** to leave the tool. We need to set the properties of our pads, so click in them to bring up the properties in the right-hand pane. Change layer to Top Layer, and Solder Mask Expansion to 0.

The final act of creating our footprint is to draw a trace connecting the appropriate pieces of artwork to the pad. Select the Trace tool, and draw a connection. Right-click to finish the trace.

That's our footprint finished. This is a slightly unusual footprint in that, most of the time, you'll want to draw a footprint for an existing component, not some customised artwork.

Where you put a track or an image is where copper goes

However, once you're familiar with the tools, the task is fairly similar.

Use File > Save to save the file and give it a name. We now need to assign this to one of our footprints. Back in the schematic, select the header that you want to be a touchpad and click in the Footprint attribute in the right-hand pane. This will bring up a window that lets you search for your new footprint and add it.

You should now have a complete schematic with components. Let's now look at building the PCB. →
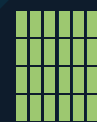
PCB DESIGN

Turning the schematic into reality

We promised you **PCB design, and so far, we haven't actually done any PCB designing – we've just created all the bits that we need to use to design the PCB.** Let's fix that and design our PCB. We can convert the schematic into a PCB with Design > Convert Schematic to PCB. There's a pop-up telling you about unfinished nets. This is expected because we haven't connected every possible thing on Pico. However, it's not a bad idea to double-check this. If you click Yes, you'll get a list of nets that aren't connected – this should just be the unused pins. If there's anything unexpected in there, take a closer look to make sure everything is connected as expected.

Once you're happy with the nets, use Convert Schematic To PCB again and, this time, click on No. EasyEDA will then present you with a PCB with all the components. There's a box asking about the size of the PCB, but you can just ignore this for now as you can adjust it later to fit around your components.

You'll see light-blue lines connecting your components. This is known as the 'rat's nest' and represents connections that you need to make. They're not actual connections on the PCB. As you wire up the PCB, these lines will disappear.

You can drag and drop your components where you want them. It's up to you how you lay it out, but if you want to follow in our footprints, you'll first need to place Pico on the rear of the PCB. Select it and right-click. In attributes, select Bottom Layer.

The main task in this PCB design is getting everything laid out how you want it. If you like everything neat and symmetrical, then you can use the grid. Press **ESC** to leave the current tool and then click on the black background. In the right-hand pane, enable the snap, and set the size you want. If you prefer a more organic, free-form look, then you can disable it.

Our design is, from an electrical standpoint, quite simple, so you do not need to worry too much about being able to connect everything – it will almost certainly be possible. However,
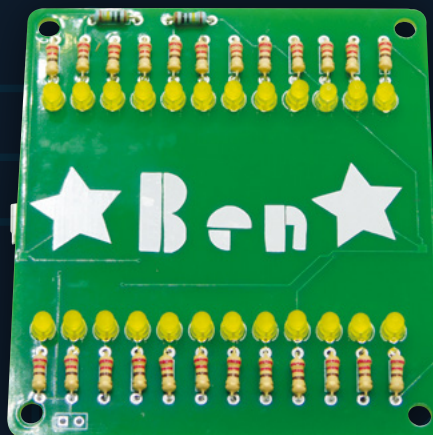
Right ⬇
Our finished PCB manufactured and soldered up
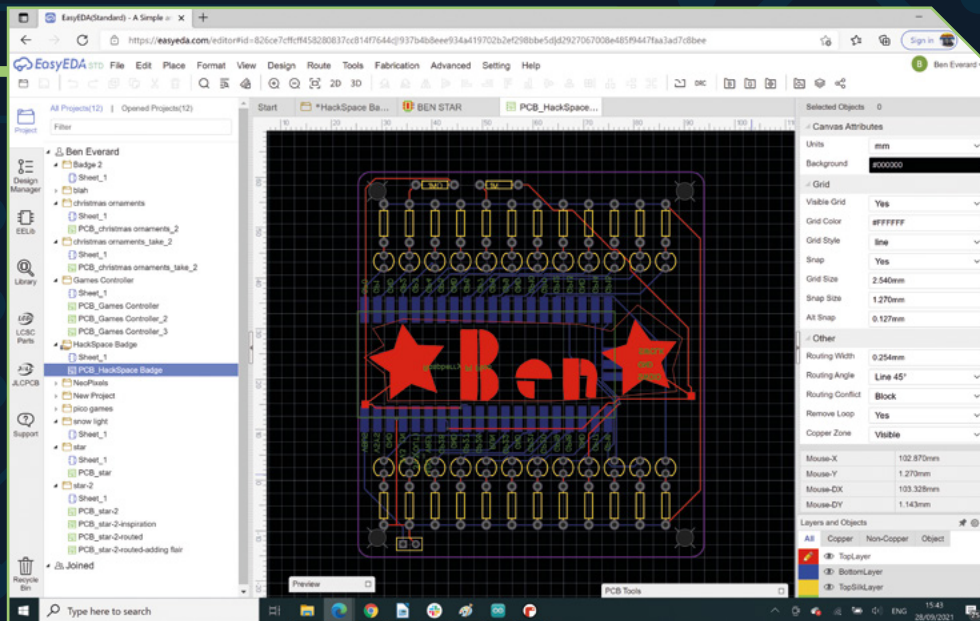
it is easier if you place the LEDs near their respective pins, and also the resistors near their respective LEDs.

Once you've got everything placed where you want it, it's time to route. You can either do the routing manually, or use the autorouter to do the hard work.

Autorouters have a bad reputation, and they're far from perfect, but on a PCB like this, where there's quite a lot of connections but enough space that everything should find its way through, they can save a lot of time. Let's start with that. You can always go back and do it manually if you don't get a result you're happy with.

EasyEDA has cloud and local options for autorouters. We've had some difficulty using the cloud router recently, so would recommend installing it locally. There are instructions on how to do this at **hsmag.cc/Autorouter**.

Whichever option you choose, it still works in the same way through the web app. Go to Route > Autoroute > Autoroute. The default options should be fine. Press Run to kick it off and then wait while it finishes.

If you're lucky, it will route all of the components and that's it. You might find that there are one or two nets left unrouted, and you have to do these manually.

Routing is one of those things that can be complicated or simple. Generally, the difference is the speed of the signal you're routing. If you're trying to shove a lot of data through a small area of PCB very quickly, then you can have problems. However, if you're trying to switch a few LEDs on and off, then really you just need to worry

EasyEDA has cloud and local options for autorouters

about getting a connection. As long as two pads are connected, that's probably all you need to worry about.

Hopefully, most of your tracks are now connected, but if you have some nets unrouted, let's look at how to do it by hand. Any unrouted nets should still have their rat's nest lines showing you where they need to go. Tracks go on one side of the PCB, but you can send them →

SURFACE FINISH

One of the options you'll be given when ordering your PCBs is the surface finish. There are typically three options: HASL, HASL (lead-free), and ENIG. HASL stands for Hot Air Solder Levelling, and it means that the PCBs are dipped in a vat of molten solder and then blown with hot air to remove any excess. ENIG is Electroless Nickel Immersion Gold. In both cases, the surface finish protects the copper from oxidisation. ENIG is flatter and may be more suitable for components with very small pads, but for our purposes, either can be used. ENIG is typically much more expensive, but does look awesome. HASL is shinier and cheaper. Again, the choice is yours.
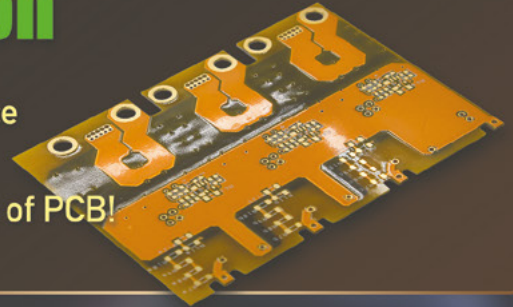
Build a Makerspace for Young People

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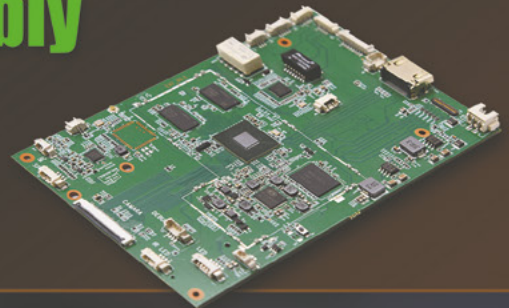
Sign up today: rpf.io/makerspace

Raspberry Pi

#RPiLearn
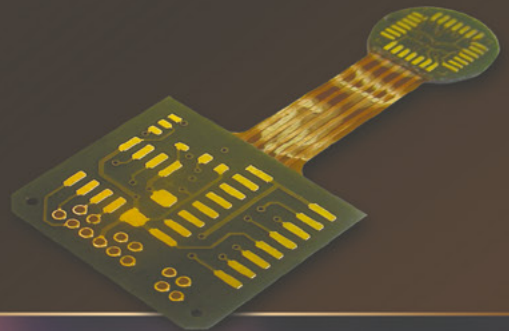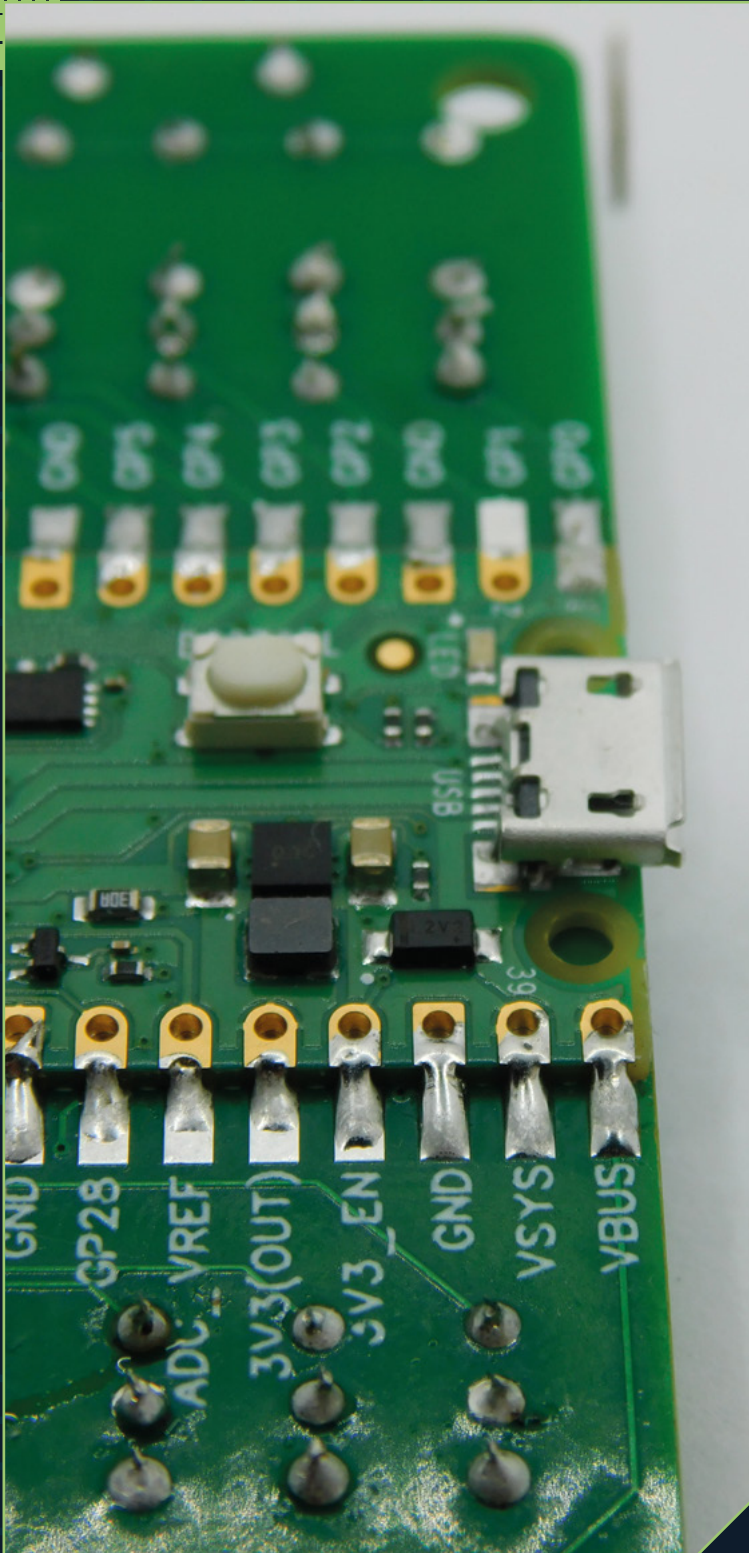
Left ◈
You can either power the PCB through the USB port or through the power supply headers

between layers using vias. These are little holes that go through the PCB and are metal-plated so you can attach traces to them from either end. Using these, you need to find a path around the other objects and traces on your PCB. As the autorouter hasn't managed to route these traces, it's likely that you'll need at least one via to jump over or under any obstacle.

Adding a trace isn't any more complicated than using the Track tool and drawing the path you want to take. EasyEDA won't let you run a trace too close to anything it shouldn't connect to, so it should be quite straightforward.

When adding a via, you first have to place the via, then add it to the appropriate net before you can connect a trace to it. Use the Via tool to place a via, then press **ESC** to leave the tool, and then click on the via to highlight it. In the right-hand pane, there's a Net option, and you can type in there what net you want to connect it to. If you're unsure, click on one of the pins that you're connecting, and you can copy-paste the Net.

You may also want to remove the labels from the silkscreen layer. This makes the finish a bit tidier. Just select the label (and only the label) and press **DELETE**.

That's all there is to creating your first PCB. The final thing is just to check that it's all correct, and then get it manufactured. There are a number of design rules that your PCB has to follow – things

PROGRAMMING

Obviously, Pico will need programming to take advantage of the PCB we've created. You can program it however you like, but we've used CircuitPython. This has the touchio module that can take advantage of the touchpads we've created.

You'll first need to download and flash the CircuitPython firmware from here: **hsmag.cc/CircuitPythPico**. If you've not used CircuitPython before, there's a good Getting Started guide at **hsmag.cc/CircuitPIntro**.

Our code to flick between two flashing patterns is at: **hsmag.cc/PCBBadgeCode**.

Press the left-hand star to change patterns, and press the right-hand star to change the speed.

OTHER PCB DESIGN TOOLS

EasyEDA isn't the only option for designing your PCBs. Here are a few of the other popular choices:

- **KiCad** is an open-source tool that we've looked at before in HackSpace magazine. It's perhaps more fully featured than EasyEDA, and is used by professionals as well as hobbyists. If you want to try it out, one important difference is that you don't assign footprints to your schematic components when you design the schematic. Instead, there's a step between designing the schematic and designing the PCB where you annotate the schematic.

- **EAGLE** is widely used to design PCBs; however, there are limitations with the free version, including only being able to design PCBs up to 80 cm². Also, the free licence is on a year-by-year basis and may not be available in the future.

- **Altium** is used more by professional electronics engineers than hobbyists. At an eye-watering £265 per month, it's out of reach of all but the most dedicated hobbyists. It's not a tool we're familiar with at HackSpace towers, but we're told it does perform excellently.

- **Fritzing** – This software was popular for breadboard layout, but does include the ability to take designs from breadboard through to schematic and PCB. If you're already familiar with the software and use it, then it could be useful for simple designs, but it's a little unloved now.

Below ◈
The castellated holes on Pico make it easy to solder onto a board. Solder one pad first to make sure it all lines up, then once you're happy with the placement, solder everything else up

> **There are a range of PCB manufacturing options available**

like making sure tracks are the right distance apart and nothing's overlapping. EasyEDA should have ensured these were all followed as you did the design. However, if you move components around after they've been routed, things can go a little awry. It's a good idea to check them before committing to an order. Go to Design > Check DRC and it will ensure that everything looks good (or give you a list of things to fix). You can view and change the current design rules by going to Design > Design Rule. The standard EasyEDA ones should be fine for most purposes, but, in some cases, you might want something a little different (such as if you're milling your own PCB and have different tolerances than a PCB factory).

To get your PCB made, you need the Gerber files from EasyEDA. Go to to Fabrication > PCB Fabrication Files (Gerber). You'll get a pop-up asking about DRC checking, but if you've already done this, there's no need to do it again. Finally, select Generate Gerber to download your file. This will actually be a zip file containing separate files for the PCB, solder mask, silkscreen, drilling, etc. It's almost, but not completely, standard between different PCB manufacturers.

You should just be able to upload this zip file on a web interface, view a rendering of your board, and then order it. The boards in this article were made by PCBWay (**pcbway.com**).

We can't wait to see what you create. Let us know at **hackspace@raspberrypi.com** or HackSpaceMag on social media ▢
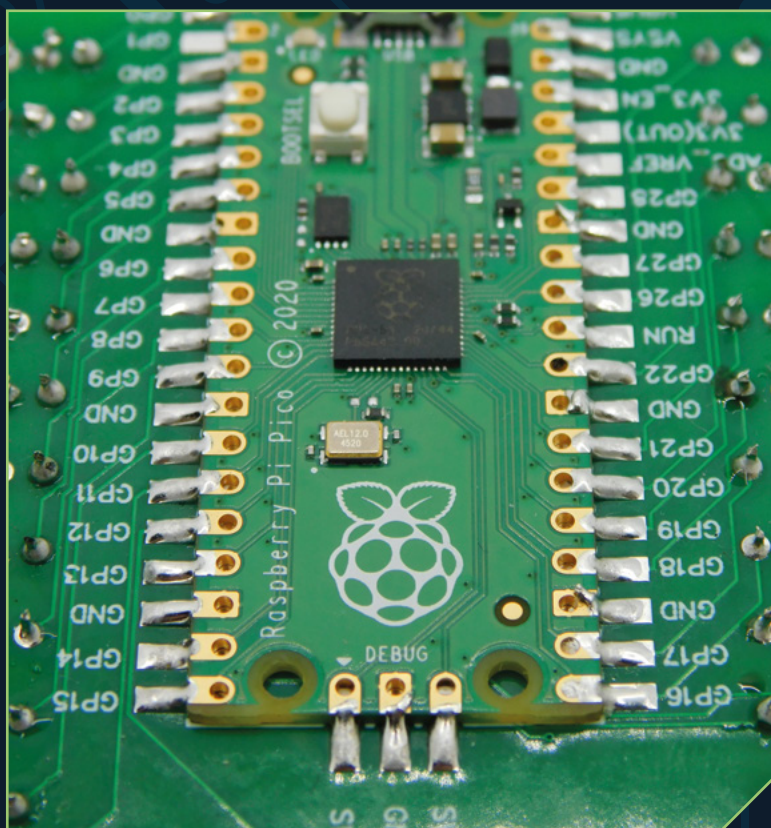
SUBSCRIBE TODAY
FOR JUST £10
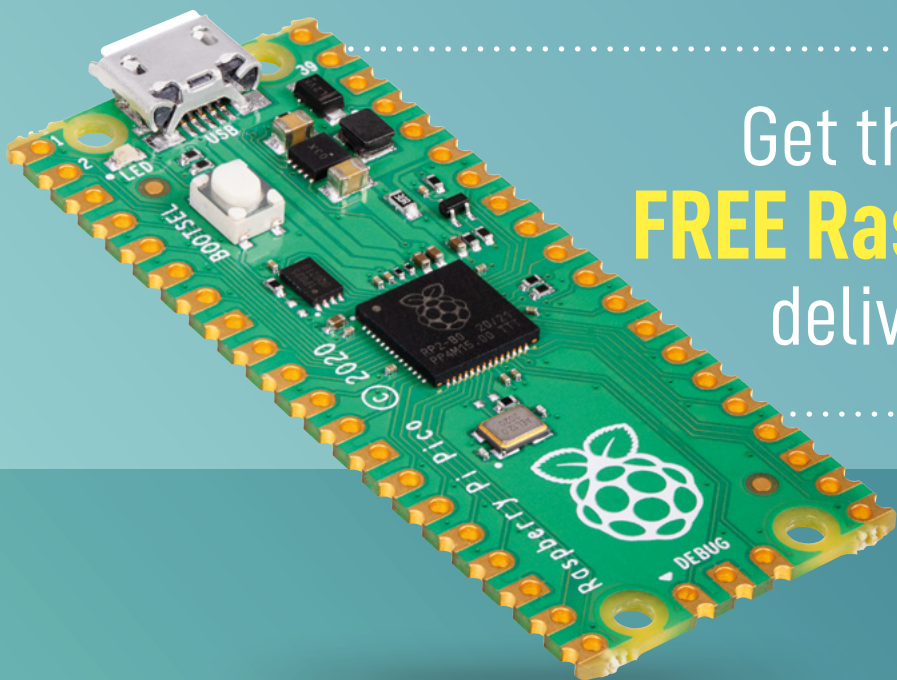
Get three issues plus a
FREE Raspberry Pi Pico
delivered to your door

hsmag.cc/FreePico

UK offer only. Not in the UK?
Save money and get your
issue delivered straight to your
door at **hsmag.cc/subscribe**.
See page 66 for details.

Subscription will continue quarterly unless cancelled

<h1>How I Made</h1>

AN AUTOMOTIVE DISPLAY UNIT

Customising my car the geeky way

By **Brian McKeon**

My Sony car stereo was failing, sometimes refusing to start, so I thought about an improvement: a 7" display head unit. I presently have three conventional gauges above the Sony and wondered if I could put some software gauges into a more modern head unit.

Modern cars feature an OBD-II communications bus which allows aftermarket head units to connect to that bus and get signals such as engine RPM, temperature, etc. However, my 1990 Toyota MR2 is pre-OBD-II, so my head unit would need to take some voltages as gauge inputs.

Android head units on eBay cost around A$130, including postage. They have built-in radio, audio power amp,

Bluetooth, GPS, WiFi, the ability to link with phones, and lots of Android apps. An equivalent Raspberry Pi setup would cost A$238: A$94 for a Raspberry Pi 3; A$90 for a 7" touchscreen; A$29 for a GPS module plus antenna, and A$25 for a 4-channel power amp. The Android unit option was cheaper and mechanically more robust than attempting to assemble multiple modules with the Raspberry Pi option – the packaging is important in a car where vibration can be an issue.

I would need to write an Android app to display gauges, and an electronic interface to fetch the analogue input voltages. The options for the analogue interface would be via Bluetooth, WiFi, or USB. I had a Raspberry Pi Pico that has a USB connection and can read a few analogue voltage inputs – as it's powered over USB, there's no need

Figure 1 ◩
Driving data displed
on your dashboard
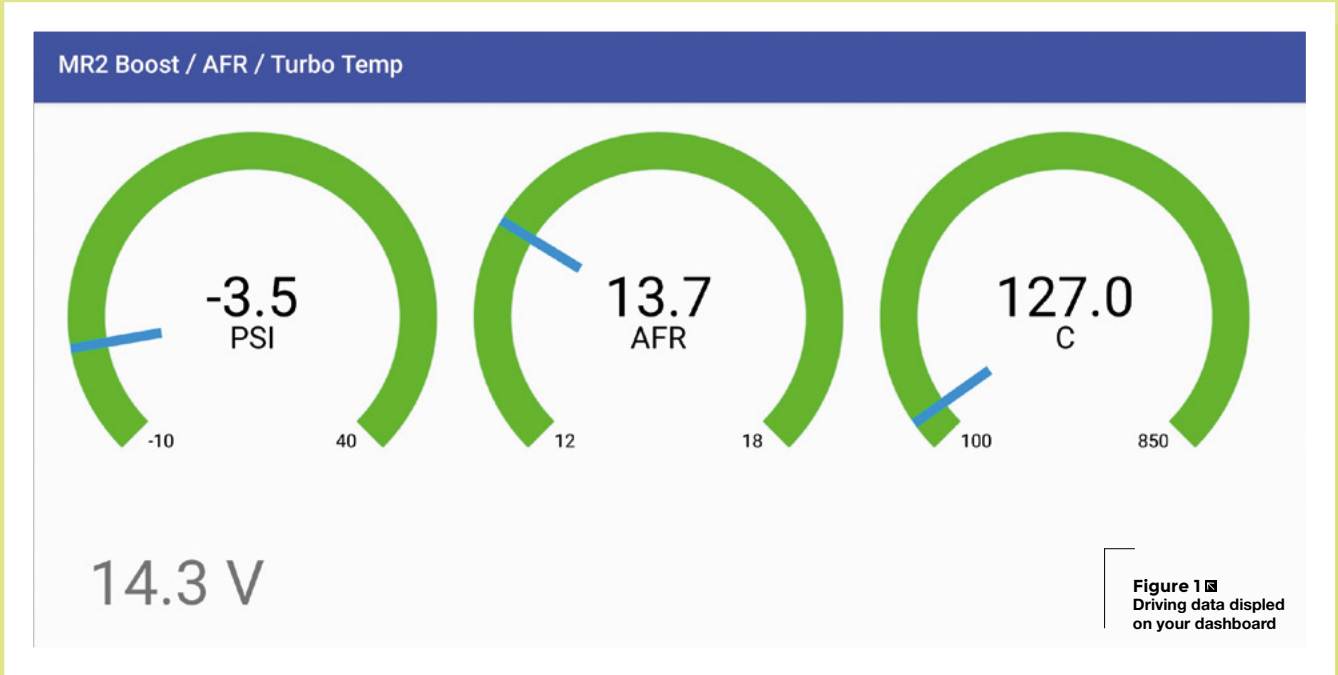
for further power supplies. A Pico retails for around A$8; however, mine came attached to the cover of a HackSpace magazine bought from the newsagent.

My background is in embedded C programming, and I found the Android model has a significant learning curve. However, it is very interesting once you understand how it works – it is also very powerful once you get the hang of it.

The software development environment (SDE), Android Studio, is a free download. If you do want to play with the app, however, it's a monster package. At one point I deleted and reinstalled Android Studio together with a few prototype applications – the Windows Recycle Bin revealed 150,000 files totalling 15GB had been deleted!

The Android Studio's display can be seen in **Figure 1**. The gauge on the left is the turbocharger boost; the gauge in the middle is the air/fuel ratio (AFR), and the gauge on the right is the turbo temperature value that I calculate from the boost signal. I use this to decide when to shut down the engine for turbo cool-down. If boost indicates the engine/turbo is working hard, the temperature will begin to rise, and it

A Pico retails for around A$8; however, mine came attached to the cover of a HackSpace magazine bought from the newsagent

will decrease if the boost is low. I used a thermal mass and a thermal resistance to ambient, then fitted this to a maximum temperature of 850°C (a cherry-red exhaust) after an extended period of heavy-duty operation, a cool-down time of 30 seconds after light duty, and five minutes after heavy duty. These are typical times that people use for turbo cool-down in these situations.

The cool-down target's set to a turbo temperature of 220°C, which seems to be a typical maximum temperature for synthetic oil. We need the cool-down time so that oil and coolant remain circulating until the turbocharger bearings have cooled and won't cook (burn) the oil if it isn't circulating. I wait until this gauge shows less than 220°C before shutting the engine off. →

Figure 2 ◆
A simple voltage divider protects Pico's input
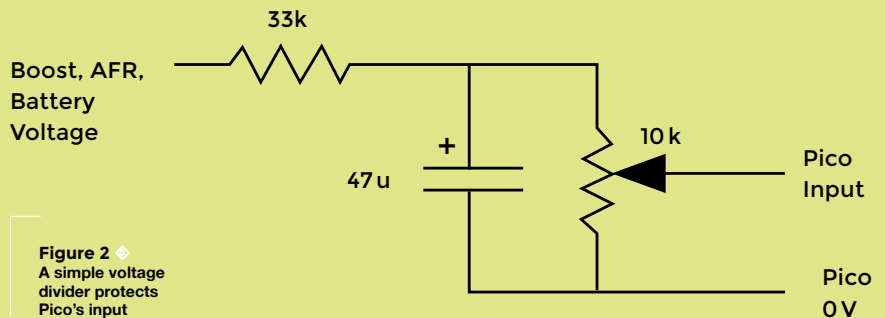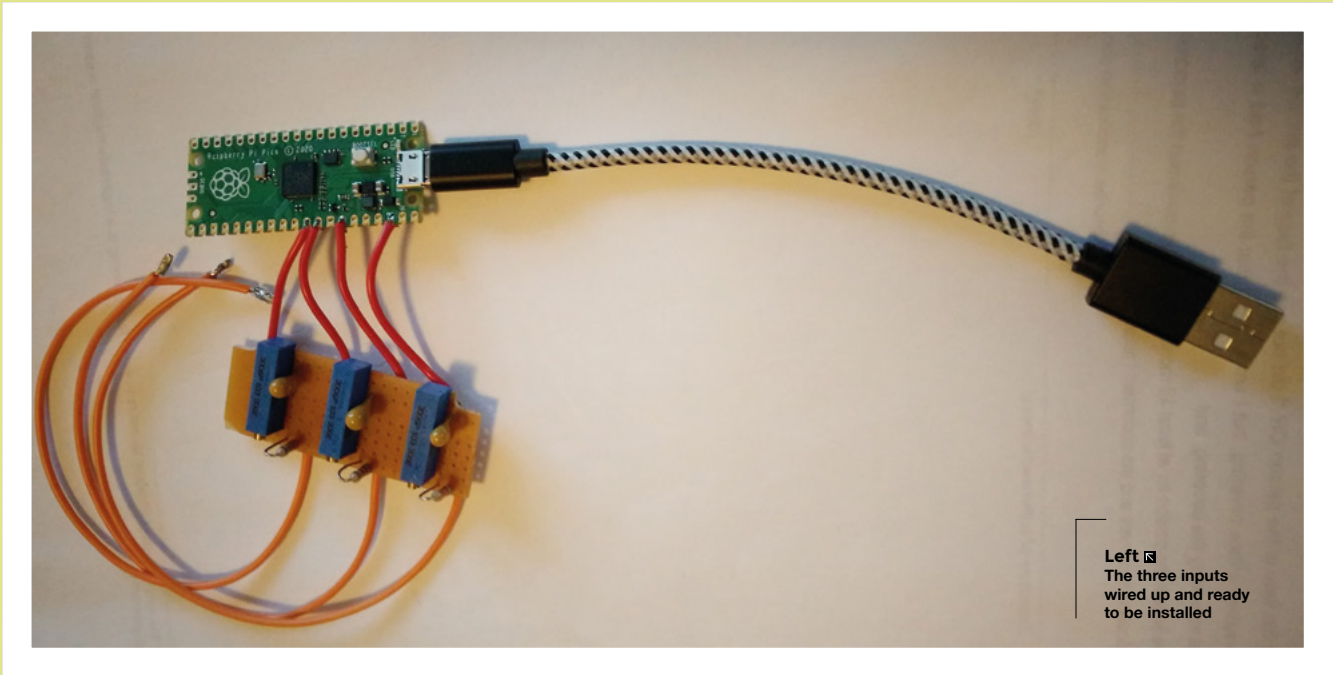
The value shown underneath the gauges, '14.3 V', is the battery voltage.

The Pico has an internal 3.3 V supply, so input voltages need to be kept in a 0–3.3 V range. I added resistor dividers on the three inputs to convert a 0–17 V range into a 0–3.3 V range. The trimpot lets me calibrate the input.

Figure 2 shows one of the three input circuits. All three Pico voltage measurement inputs have the same 0–17 V range, and calculations of voltage to battery voltage, boost psi, and AFR signals are in the Android app.

The 47 u tantalum capacitor, plus parallel 8k or so, would result in a 400 msec filtering time constant for any automotive electrical noise, and this wouldn't impact the rate that these signals change.

My boost sensor has a 12 V and 0 V connection and an output signal that is 4.14 V at +40 psi and 0.37 V at -11 psi. I measured some voltages while I connected the sensor to a water-filled plastic tube in the shape of a 'U' (**hsmag.cc/UTube**). I raised and lowered one side of the U to get pressure changes. I could then work out an equation to convert the incoming Pico voltage to a boost pressure.

My AFR measuring device is an 'Innovate Motorsports 3918 MTX-L PLUS', which I have mounted near the engine. It gives 0 V for 7.35 AFR and 5 V for 22.4 AFR. It needs a good 12 V connection as it supplies the heater in the O_2 sensor, so I've wired this 12 V supply from the engine bay.

Starting with the Pico, I loaded MicroPython: **hsmag.cc/ProgRPI**. Then I entered the following Python app onto the Pico using Thonny on my PC (**thonny.org**). The Python code is:

```
import machine
import utime
import math

pinBoost = machine.ADC(26)
pinBatteryVolts = machine.ADC(27)
pinAFR = machine.ADC(28)

while True:
    print( "S,"+str(pinBoost.read_
```

Figure 3 ◈
I can display whatever data I want on the screen
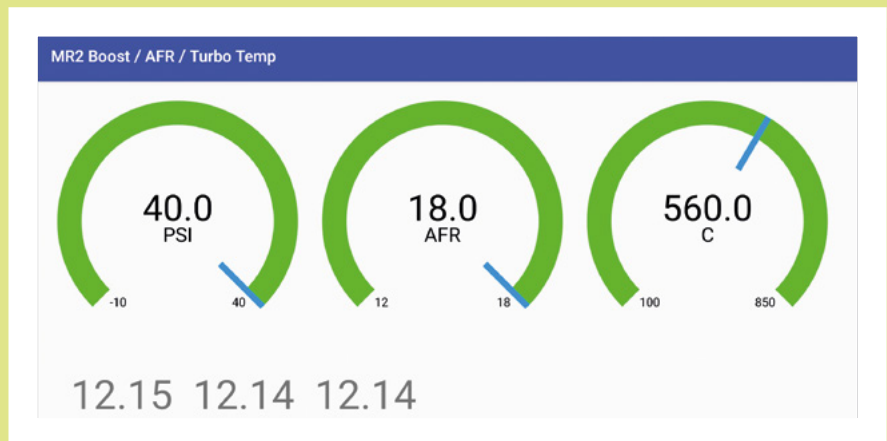
```
u16())+”,”+str(pinBatteryVolts.read_
u16())+”,”+str(pinAFR.read_u16())+”,”)
    utime.sleep_ms(500)
```

The program displays lines like 'S,10786,12979,13347' a couple of times per second. The numbers will move around depending on what voltage is at the three inputs.

When you save this Pico program, there are options to save to your PC or to the Pico. You should save this to the Pico so that it will run automatically when the Pico is restarted. You can then unplug the Pico.

I found a gauges software library and also a USB serial I/O library. Many thanks to

Figure 4
I had complete freedom to choose the dials I wanted

Anas Altair for the great SpeedView library, **hsmag.cc/SpeedView**, and Felipe Herranz for the USB library, **hsmag.cc/USBSerial**.

Although the head unit has USB ports, these wouldn't connect with Android Studio to allow app download. I could download the app on a Samsung tablet, so I did most development there, and then tested the finished versions on the head unit. I had to put the app binary on a USB flash drive, plug this into the head unit USB, and then install it from the USB flash drive.

My head unit was Android 9.1. I noticed that there are similar units that come with Android 10, though I haven't tried that. Android 10 includes specific changes to do with Android Automotive OS extensions – I don't know if these might interact. Even

with the Samsung tablet being Android version 9, the Samsung did behave differently to the head unit, and I had to make changes to get it to work on both. So I am wary about different Android versions.

To load the Android app on the head unit, you need to enable the Debug mode: **hsmag.cc/DevOptionAndroid**. Then, put the file **app-develop-debug.apk** onto a USB flash drive and plug this into the head unit. Go to the file manager and the USB drive and then click on the **app-develop-debug.apk** file. Android should then install the app. If the Pico is then plugged into the head unit, the app should start and display the gauges.

Once the app is running correctly, disable the developer option as per the previous link. Developer mode is only needed to →

225.9Km/h

24.4 Km/h

46.7 Km/h

allow loading of the app as it is not being delivered through Google Play. You shouldn't leave Developer mode enabled as it could allow apps to be loaded from places other than Google Play.

Before installation of the head unit, I powered it up on the bench and adjusted the calibration trimpots. The boost voltage input is normally in the 0–5 V range. If a boost voltage of more than 10 V is connected, the app will display three voltages in the lower line where it normally only displays battery voltage. If you have the unit connected to a battery, you can connect the three input signals to the battery positive – the app will then show three values. I measured the battery voltage and adjusted the three trimpots to get the three displayed voltages close to the voltmeter reading. The display looked like this (**Figure 3**) when I had adjusted the trimpots for a 12.15 V measured battery voltage.

I found that the head unit drew 0.5 A when it was on, and a hefty 0.3 A when it was off. 0.3 A is too high to be left connected to a battery unless you drive your car daily. My car battery is in the 72 Ah

you turn the ignition off. The Android file system itself is designed around the Linux EXT4 format, which has journaling – this prevents the file system from corruption if power is lost unexpectedly. Even though

These power connections mean that Android has to boot whenever the car is turned on, which takes around 40 seconds

capacity range – 0.3 A would flatten the battery in a week. Therefore, I've wired the power to the head unit so that all power is from the accessory circuit of the MR2. These connections on the head unit were +12 V and ACC. The head unit has an ILLUM connection for night use – this should be wired to a parking light signal. I found that ILLUM gives back-light on the controls to the left of the screen, and also dims the display a bit.

These power connections mean that Android has to boot whenever the car is turned on, which takes around 40 seconds. These connections also mean that Android can have the power shut off unexpectedly when

it won't be corrupted, this doesn't mean that all file updates have been completed. You might have selected a song on the music player, turned the ignition off, and Android may not have had time to record your selection for when you restart the head unit. You would need to be quick to see this sort of problem, though the impact is pretty minor. If you do anything like change settings, it would be a good idea to turn the power off at the head unit power switch before turning the ignition switch off. This would let Android know to clean up before it is turned off.

The head unit installed in my 1990 Toyota MR2. The AFR is not connected yet. I mounted the head unit using two strips of black-painted aluminium angle attached to the side of the head unit and the front of the dash. The Pico and resistor network are in a small plastic box attached to the back of the head unit.

If you want to try other gauge styles, here are some examples from the SpeedView library (**Figure 4**).

Have fun! □
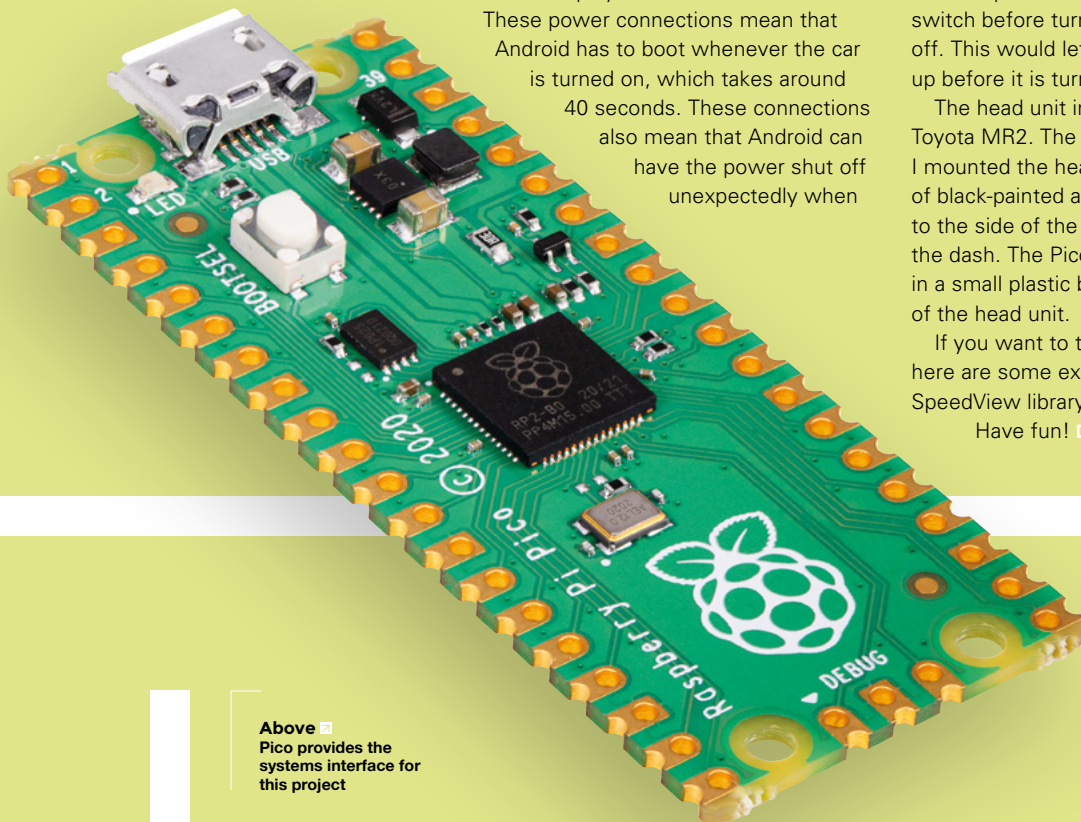
Above
Pico provides the systems interface for this project

HackSpace magazine meets...

Wesley Treat

Drawing inspiration from the rusty, neon space age

A ccidental YouTuber, Etsy seller, sculptor, photographer, and scholar of mid-century roadside art, Wesley Treat is a man of many talents. But he'll be the first to tell you that he's not an expert in any one thing in particular: instead, like a lot of self-taught makers, his number one skill is teaching himself new things.

We caught him just as he was heading out the door to the Catskill Mountains Maker Camp in New York State, to ask him about learning, the space age, and why imagination making things for yourself is the best kind of making. →

Wesley Treat My background is actually in theatre; that's what my degree is in. But I've always been a photographer. Since I was a kid, I've had a camera in my hand. In the late 1990s when the web just started to become a thing, I started doing web design, just to learn new skills. And I started writing about roadside attractions, offbeat stuff, photographing those and putting them up on a website.

That got the attention of a couple of guys in New Jersey who publish a magazine called Weird NJ. And it's all about that kind of weird stuff. They asked me if I wanted to write on a series of books they were producing; they were doing one on each state, and I was running a website on weird stuff in Texas.

I picked up a couple of sheets of aluminium and learned how to put it together with rivets. I couldn't weld back then; this was like ten years ago, and welding was something that was not even in the realm of possibility. I still don't know how to weld aluminium; it's a lot more difficult than welding steel.

When I found out about pop rivets, it just completely opened up. I could drill holes, and I could go to a store and buy a pop rivet gun, and I could finally make one of those signs. And I made one – a big red arrow. That was my first sign and I sold it on eBay. I sold it for nothing, a ridiculously low price.

I came up with a couple more designs and started selling stuff on Etsy. And then I found that a lot of interior designers will hit Etsy for ideas. And so they started contacting me – people doing restaurants

I don't remember when I made the decision to make my first video about making things. It was a video on etching circuit boards at home. I had developed a circuit board specifically for the signage I was making, just to animate bulbs and things and learn how to program on an Arduino, taking that chip and putting it into its own circuit board. I learned just enough to make it work through forums and taking parts and other people's circuits and that sort of thing. And it still surprises me that all the ones that I've made on those circuit boards still work amazingly, →

> " It's become an obsession: a lot of them that I've shot over the years have **disappeared already. So I'm trying to preserve them as I can** "

So they contacted me, and so I got the *Weird Texas*, *Weird Oklahoma*, and *Weird Arizona* books.

At that time, I was driving through a lot of small towns and I was noticing all this old signage. You know, the motel signs with the big arrows and neon. It's become an obsession: a lot of them that I've shot over the years have disappeared already. So I'm trying to preserve them as I can.

At some point I wanted to start working with my hands again because, even as a kid, I was always kind of making little things. And I decided I wanted to make one of these signs. I know a couple of guys who collect them because they're in the industry, and they know when stuff's coming down and they have the money to buy them up. They've got storage areas full of these old signs, but a guy like me... I couldn't get hold of one, so I figured I'd make one.

and that sort of thing. It's slowly become a thing, and I have a good client base now, a small number of clients who keep me busy.

HS **And then you started putting things on YouTube.**

WT I've thought about this many times; I can't remember how it started, how I got into it. I had long had an interest in video editing. Back when I was in college, you know that's when non-linear editing just started to become a thing. At that time I spent like $1000 on a video capture card, which was a lot of money at the time. I was still using tape, and it was frustrating, because the technology was just horrible. I never could get video and audio to sync up when editing that sort of thing. I just got too frustrated with it. But then, you know, YouTube came along and I started to think about it again.

even though I only kind of knew what I was doing.

I've always liked the idea of trying to tell the story visually. The challenge is making it understandable, so viewers can understand the process without me explaining it explicitly.

I talk more [in videos] now because I've learned that I've got to put myself in there. But I still mostly do it without speaking.

Anyway, Jimmy DiResta tweeted about that first video, which was cool. And I think at the time I got around 100 views from that. That's not a lot, but it was a thrill that people were watching this video of mine. And it's still one of my most popular videos. And then I did another one and another one, and it has slowly become a thing.

HS **It's rare that we know at the time how significant things are.**

WT I didn't know this was gonna turn into an actual deal. There was no light bulb moment when I saw some particular video

HS **What is it about mid-century stuff that you like so much? Is it just that there's a lot of it kicking about in Texas and Oklahoma?**

WT I guess it's hard to explain why something visual appeals to you. The signage, especially that Googie style, is linked with all the mid-century aesthetic. Signs today are all just backlit, acrylic, and vinyl. They're strictly functional. They're utilitarian.

But back then they had character, they had personality, they got your attention. People would open up a shop, and it was a

point of pride to get a sign like that out in front of your shop. People were proud of those signs, and they would, you know, they would design something unique to stand out from the other 100 motels along the same road. And they had to have character.

The signage, architecture, the houses, the furniture inside them, it all just kind of comes together.

HS **The optimism of the early space age. You make a lot of things with stars on them.**

WT I made a video that explains the history of that. The spiky balls are called Sputniks, after the first satellite. When the USSR launched Sputnik, we didn't know what it looked like, exactly.

But one of the first US satellites, Vanguard 1, it had that spiky look, and so the name Sputnik got attached to that design. And it was the space age, you know; people were excited about it. And so that design motif started to pop up

> There was no light bulb moment when I saw some particular video by some **particular maker that inspired me. Life is accidental. And random**

by some particular maker that inspired me. Life is accidental. And random.

everywhere, even on signs for motels. And I love them. I don't know why exactly. Maybe because they serve no function other than to look cool.

And I've always been kind of a space nerd, too. So that space aspect, tied together with the Googie style, really does it for me. And then they've always got the neon and the bulbs on them. And I love that too.

Even what's going on now, with SpaceX and stuff, it doesn't compare with the enthusiasm people had at the time. I grew up with the space shuttle; we were already in space. But at that time, in the 1950s, just launching an object into orbit was mind-blowing. I mean, it was science fiction finally coming to life for them. So I

think it really crept into a lot of the culture at the time. If you wanted to seem like part of the future, everyone was incorporating that aesthetic.

HS **Speaking of the Sputnik video, you followed that up with a video on failure. Not just mentioning it in passing, but taking the time to discuss failure. That's unusual.**

WT I do fail quite often, and I often include it, because it's funny just to see me screw up.

But I get a lot of comments from people saying that they appreciate that I do that, that I include the mistakes and the screw-ups and, you know, just me grabbing a piece of metal like an idiot when it's hot, forgetting that I just had a torch on it. It's funny, but you know, let people laugh at me. I'm not an expert. I'm not an expert at anything. People have said that they appreciate me showing my mistakes, →

because it makes it more accessible for them.

Some people present as authorities on, woodworking, for example. And they show you, 'this is how you do this... this is a perfectly cut tenon, this is a perfectly cut dovetail'. And if you, the non-expert, try to repeat that, it's never gonna happen. That's a guy who's done that many, many times.

So people get frustrated when they attempt to repeat something they see in a video. But when they see somebody else struggling, I think it's actually encouraging. That guy, you know, I see the stuff that he makes, and they see him screwing up along the way. And they're like, 'Oh, you know what? I can screw up and I can still get to that same destination'.

I screwed up loads with the project I'm working on at the moment. It's like a briefcase for my Little Wiener grill. That fits the grill and the tools, and the condiments, all in one case. I've got all these cutouts that exactly fit everything. I don't show it, but I screwed that up four or five times and had to chuck several of those just because of dumb mistakes.

I don't include any of those in this one, because it doesn't really help the narrative at all.

And none of them are humorous at all – it's just things like I had trouble with one router bit that kept slipping in its collet – there's nothing funny or informative about that, so I left it out.

HS You look like you're having fun while you're making things. Are you enjoying it as much as you look like you are?

WT I do get frustrated sometimes, when things aren't going right, but I try to keep in mind that that's part of learning, and as long as I'm learning I'm having fun. I don't really have a speciality: I collect skills. I just get bored easily. So I like to try. I like to try new stuff, you know, just to see if I can make it happen.

I've heard a lot of people talk about how they hit this part in a project where they start wondering if they're really enjoying it. Once you get over that, then it's fun again...

and when it finally turns out the way you want it, you get that high. But yeah, I enjoy what I do, as long as I'm doing something new, learning something new, or improving on some new skill.

I've just started 3D printing. Creality just sent me a printer to try out. The speed bump is that I hadn't learned Fusion yet. So the modelling is where I'm having the issue. But now I'm learning, and I printed a couple of custom things... I see what people mean when they say it's a hole you're gonna fall into.

It was one of those things, I didn't really have much interest in getting one. Because

> If I've made something because it amuses me, and then it amuses someone else enough that they want to own one... that's a fantastic feeling

I couldn't think of what exactly I would do with it. But now that I've got it, I'm thinking of all kinds of stuff to modify.

Like, there's this mount in my car, for my phone, but it doesn't, like, stick out far enough, so I've been modelling this attachment to extend it and then hang my cable on it.

HS How do you find your clients?

WT I do a lot of sign work and stuff like that. I do a lot of stuff for bars and restaurants – that's still my main source of income, and I show some of that stuff. As long as it's a little bit different or interesting enough, I'll make a video out of it.

I'm getting to the point where I have the privilege of picking and choosing some of those commissions now. If I've got a sponsor for something, then I can just come up with something bizarre. I've got a huge list of ideas that I can choose from – just things that I think will amuse me, but which might not pay the bills.

**HS The Wiener Grill, by the way, is absolutely superb. I laughed when I saw it.**
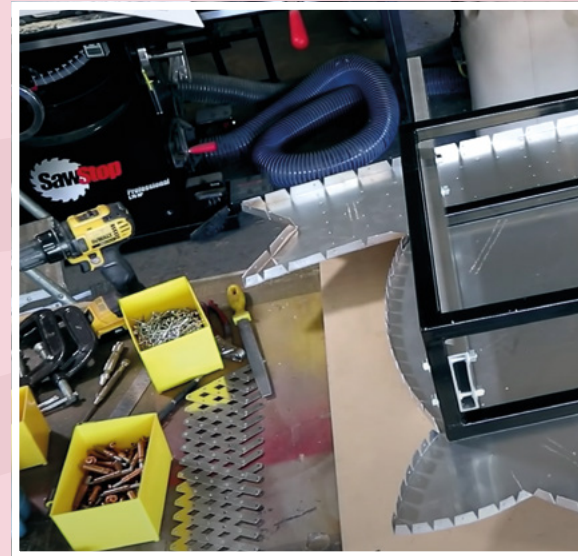
Just the idea of a barbecue that's only big enough for one sausage sandwich.

WT It was inspired by something Laura Kampf did – this miniature grill that was perfect for a couple of sausages and some vegetables. I just wanted to see if I could do it even smaller. When I did that, people just wanted to buy them. That was my first product development. I didn't know if they were going to sell, because they're kind of expensive to make because I'm not doing thousands of them. Especially right now, like, metal is expensive, but it's gotten a pretty good reaction.

I learned that it's fun: I enjoy when somebody places an order for something I've made. Like, packing it up and sticking it in the box and putting that label on it – I'm kind of proud of that. Sending that thing out, knowing that somebody actually wants to buy something that came out of my head.

It's different when a business needs a sign and they want me to do it. If I've made something because it amuses me, and then it amuses someone else enough that they want to own one... that's a fantastic feeling. ◻

ROCKET FUEL

BUBBLE WRAP

IMPROVISER'S TOOLBOX
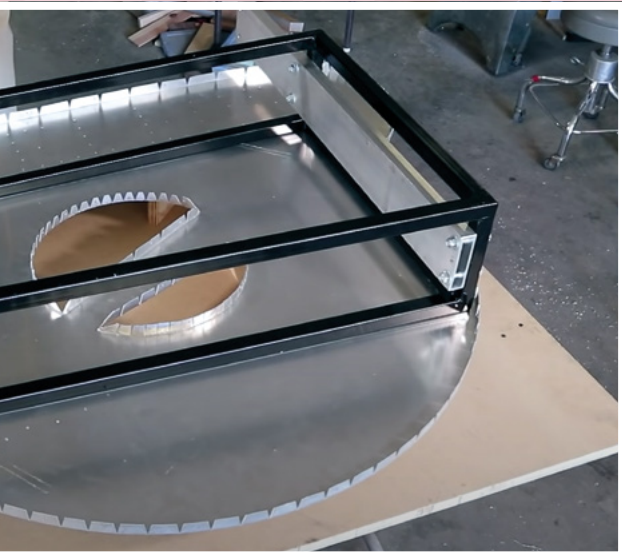
The plastic protective sheets can be used for far more than instant stress relief. Rosie Hattersley finds it's not just about the sneaky pop

Rosie Hattersley

🐦 @RosieHattersley

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

Bubble wrap usually finds its way into our lives as a result of a satisfying splurge on something delicate enough to need protection: trinkets, vases, items with sharp edges that might do damage tend to come swaddled in the inflatable wrapping. Receipt of such items invariably leads to an immediate urge to pop the packaging in a bid to produce the loudest, rudest-sounding parp, but if you can resist its tactile entreaty, bubble wrap can be used for all sorts of secondary tasks. Perhaps unsurprisingly, most such reuses capitalise on its cushioning effect: line your drawers or tool-box with bubble wrap to stop tools clattering and damaging the tools, or wrap a handle with the stuff to prevent sores from gripping it tight. Artistic options can also be found.

Bubble wrap was originally intended to be used as a form of textured wall covering, which sounds quite zany until you factor in that it is a product of the late 1950s. This means its invention is broadly contemporaneous with other curvy creations such as bubble cars, bubble houses, and bubble dresses (but a full 20 years after the introduction of the Aero chocolate bar). The putative, tactile wallpaper was first constructed from two shower curtains, but the concept didn't really take off. Undaunted, its creators, engineers Alfred Fielding and Marc Chavannes, set up the Sealed Air Corporation and pondered its commercial possibilities, with its more recognisable use as a packaging material emerging several years later. Nearly $10 billion of bubble wrap is now sold annually.

"Bubble wrap was originally intended to be used as a form of textured wall covering"

BUBBLE WRAP INSULATION

Because bubble wrap uses trapped air, it has thermal qualities that lend themselves to both blocking heat or preventing its escape. This makes it ideal for keeping cold things cool and hot things warm, so bubble wrap is a good substitute for other methods of chilling or heating. Take a box of salad out of the fridge, wrap it in bubble wrap, and pop it into your backpack and you'll find your lettuce, cucumber, and accoutrements are still cool an hour or two later.

It also makes bubble wrap ideal as insulation taped into place on the inside of windows to stop your greenhouse or shed from getting really cold (website **BuildItSolar.com** suggests simply spraying water onto the window pane to be covered, and then smoothing the bubble wrap into place), or to pop over plants to keep the frost at bay. It's just as useful on too-warm days to filter the sunshine without blocking out all the light, or to stop delicate leaves from drying out and wilting. →

Project Maker
BUILD IT SOLAR/ GARDENERS' WORLD

Project Link
hsmag.cc/BWInsulation

Left ⬙
Keep your plants toasty warm this winter

BUBBLE WRAP KITE

Project Maker
San Diego Kite Club

Project Link
hsmag.cc/BWKite

—

The near weightlessness of bubble wrap is another quality upcyclers can easily exploit. Since it's also fairly easy to cut into shapes and fasten with sticky tape, its use as a material for building kites seems obvious. There's a straightforward how-to video here: **hsmag.cc/BWKiteMake**.

California blogger Nancy Kissack photographed characterful undersea-themed examples that the San Diego Kite Club made from the bobbly stuff, complete with trailing tentacles and eyes that seem to float above the kite's surface and stare down at onlookers. The transparent 3D material belies the fact it's just paint.

Right ⬦
Pretty bubbles in the air, they climb so high, almost touch the sky

BUBBLE WRAP WALL DECORATIONS

I t's perhaps no surprise to find bubble wrap used in a range of children's art projects, from eye-catching motif printing (see hsmag.cc/BWPrinting) to sculptures made from coloured bubble wrap: hsmag.cc/BWSculpture.

The 'relaxing art form' creations of Kobus Hattingh take things a little further, with life-size animals sculpted from the plastic packaging material. Kobus says he starts his projects by using double-sided sticky tape to attach pieces of bubble wrap to the wall he wants to decorate, gradually building up a 3D design that he then paints or covers with inexpensive gift wrap. Some of his designs seem more reminiscent of a theatrical set than an everyday interior.

Project Maker
Kobus Hattingh

Project Link
hsmag.cc/BWDecor

Left ⬦
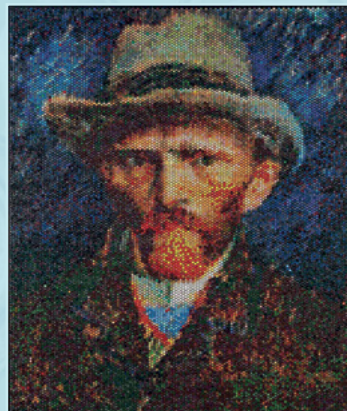**Add some texture to your rooms**

BUBBLE WRAP ARTWORK

B ubble wrap artist Bradley Hart recreates iconic paintings and photographs by injecting individual bubbles with acrylic paint. Together, his 'pixelated photorealistic pictures' combine to form an 'album of memories'. Bradley uses a computer algorithm to provide a visual overview of his highly detailed bubble wrap artworks' progress, and keep the colour work as close to the instantly recognisable original image as possible.

The laborious painting technique, which mimics the pointillism effect of impressionism or the oversized dots of pop art hero Roy Lichtenstein, is painstaking and time-consuming, in sharp contrast to the disposable plastic derived from crude oil, of which we produce an overabundance. Bradley says his aim is to highlight "the irony of applying such delicate physical artistry to a mass-produced material, and the indestructible nature of plastic versus the fragility of bubble wrap". Intriguingly, the idea came about when Bradley read about bubble wrap's original purpose as a wall covering, and decided to recreate it. □

Project Maker
Bradley Hart

Project Link
hsmag.cc/BradleyHart

Left ⬦
Packaging doesn't have to be boring

HackSpace
TECHNOLOGY IN YOUR HANDS
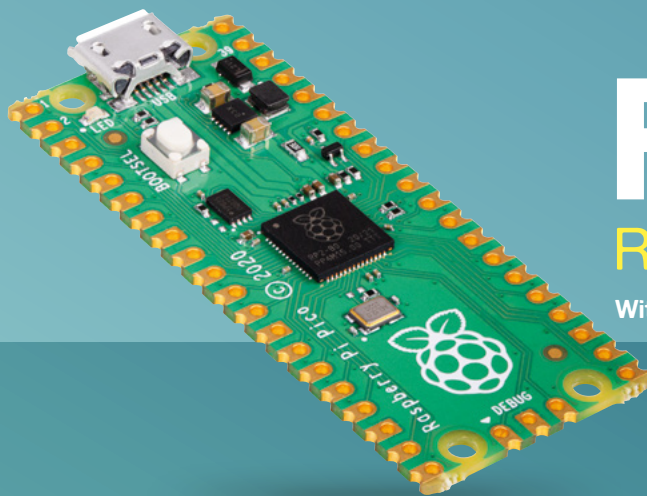
FORGE

HACK | MAKE | BUILD | CREATE

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

Get to grips with glowing electronics

Mike Bedford introduces the predecessor to the transistor – the valve or tube – and shows you how to experiment with one in a simple circuit

Mike Bedford

Despite loving all things digital, Mike admits to being a bit of a luddite, vinyl records and all. So diving into vacuum tubes proved a perfect distraction.

In December 1947, John Bardeen and Walter Brattain at Bell Laboratories invented the transistor and, in so doing, pretty much consigned its predecessor to the history books. That predecessor was the thermionic valve as it's called here in the UK, the electron tube as our American cousins would call it, or just valve or tube for short. Yet the triode valve, as invented by Lee de Forest in 1906, which he actually called the 'Audion', was a game-changer. Transitioning the world from a reliance on steam-engines, gears, and levers to one empowered by electronics is an achievement that shouldn't be forgotten,

hence our showcasing of this vintage technology. We're going to investigate the working of valves, explaining what goes on inside their glass enclosures. We're then going to show you how to experiment with a valve-based circuit, and how to migrate it from the breadboard to a cool-looking retro curiosity that wouldn't look out of place on a bookshelf next to any other piece of historical technological paraphernalia. We'll also see that valves aren't quite as obsolete as you might have imagined.

VALVE BASICS

If you've struggled to get your head around the workings of transistors, I have some good news. The operation of valves seems so much more intuitive than the migration of electrons and holes across the junction between n-type and p-type semiconductor materials. To set the scene, though, we're going to take a step back from de Forest's triode to investigate its predecessor, the diode. Diodes are now semiconductor devices, but we're thinking of the diode valve, which first saw the light of day in 1904 courtesy of John Ambrose Fleming. Although Fleming's creation was subtly different, most commercial diode valves had a cathode, an anode, and a heater, all of which were housed in a glass enclosure from which the air had been removed. The cathode is a metal cylinder coated with alkaline earth metal oxides. Inside the cathode, but not touching it, is a wire filament, similar to the one found in old-style light bulbs. When a current is applied to the heater, it glows orange, heating the cathode and, in so doing, causing the earth metal oxides to emit electrons which normally remain close to the cathode. The diode also has another electrode, the anode, which is another metal cylinder surrounding the

Above
Building an ornamental valve circuit on a stained and varnished wooden plinth seems just right

cathode. Now, if a voltage is applied to the anode, and that voltage is positive with respect to the cathode, the electrons emitted from the cathode are drawn to it, so a current flows. On the other hand, if the voltage on the anode is negative with respect to the cathode, the electrons are repelled, so no current flows. Like the familiar semiconductor diode, therefore, it allows a current to flow in one direction but not the other.

Where it gets really interesting is when a spiral coil of wire is wound in the space between the cathode and the anode. This coil is called a grid – specifically a control grid – and is de Forest's critical contribution. If a negative voltage is applied to the grid, it repels the electrons flowing from the cathode to the anode, and
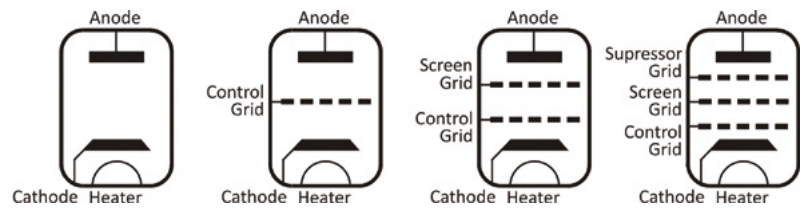
the greater the voltage, the more it reduces the current. Importantly, though, small differences to the grid voltage can control a high-voltage, high-current signal passing between the cathode and the anode. The triode was, therefore, the first-ever amplifying device.

From what we've seen so far, you'll probably have realised that valves are categorised as something-odes, where the 'something' dictates the number of electrodes. We've encountered diodes and triodes, which begs the question of whether there are also tetrodes, pentodes, or more-odes. There are, although most do the same job as triodes but do it better.

The snag with triodes is that there is a stray capacitance between the control grid and the anode, and this can cause problems. In particular, that capacitance places a practical limit on the frequency →
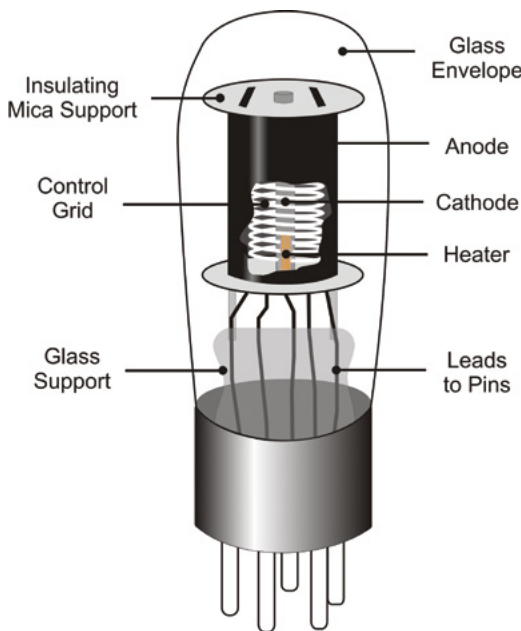
VALVE **VOLTAGES**

Valves needed two supply voltages, one for the heater and one for everything else. 6.3 V was usually used for the heaters, although some dual valves, like the 12AT7, allowed the two heaters to be connected in series and used with a 12.6 V supply. The main supply voltage was higher, often much higher, to allow high-power operation with the relatively small anode current that common valves were able to handle. A couple of hundred volts was typical for radio receivers; kilovolts were sometimes used in transmitters.
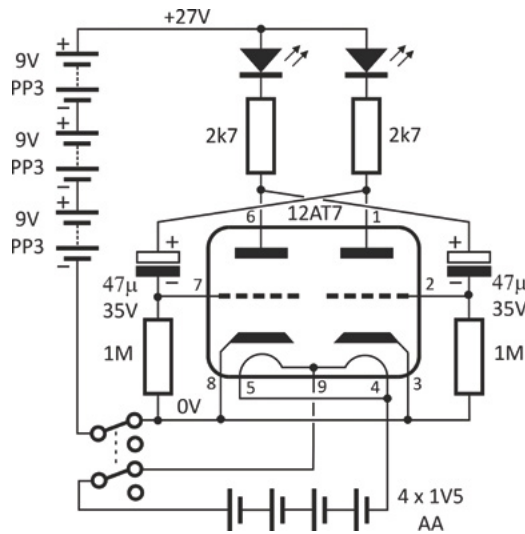
at which the valve can operate. The solution was to add a second grid, which was also a spiral-wound coil of wire, but this time between the control grid and the anode. This second grid is called the screen grid, and is normally connected to a positive voltage, somewhat less than the anode voltage. This is a tetrode.

Unfortunately, the tetrode had a different drawback. That positively charged screen grid further accelerates the electrons en route to the anode to such a velocity that electrons are emitted from the anode, and this affects performance. The solution was to add a third grid, constructed like the other two but located between the screen grid and the anode. This is the suppressor grid, and the resulting valve is a pentode.

There are also hexodes, which fulfil a somewhat different function, but that's a story for another day.

HANDS ON

A hands-on exercise is often one of the best ways of learning, so we thought we'd suggest a simple experiment using just a single valve or, more accurately, a single component that has a pair of valves

in the same glass enclosure. The circuit is called an astable multivibrator, which is a square wave oscillator. We're going to use it to flash two LEDs alternately. That means it's easy to see that it's working and, if you decide to take it beyond a mock-up on a breadboard, you'll end up with an interesting-looking contraption that evidently does something.

We bought the components from Cricklewood Electronics, although the components aren't difficult to obtain, and you'll probably have some of them already. Here's what you'll need. The valve is a 12AT7, which has the alternate part number ECC81, and is a double triode. Valves aren't available from all electronic component suppliers, and a 12AT7 will cost £12–£15. You'll need a socket for the valve – these are referred to as bases, and the type needed for the 12AT7 is a B9A base. The capacitors can be either electrolytics or tantalums, and the voltage rating shown is, of course, a minimum. Because this is an experimental project, unless you already have a good stock, we suggest you buy several pairs of capacitors, from the one we show down to 1μF. We used standard red LEDs, but if you want to build the multivibrator as a curiosity to keep, we suggest you also buy a pair of panel-mounting neon lights around 10 mm in diameter – we'll explain why later. You'll also need three 9V PP3 batteries together with three of the associated clip/leads, four 1.5V AA batteries, a 2×2 AA battery box, and if that box has a PP3-type connector instead of a flying lead, you'll need a fourth PP3-type battery lead. Finally, if you're going to go beyond the breadboard design, you'll need a double pole on/off or changeover switch.

Our circuit is shown above and we'll provide an inkling of how it works later. For now, though, you'll just be using it as a guide to wiring up the circuit on a breadboard. Wiring it on a breadboard will be intuitive except for making connections to the valve, because the valve can't plug into a breadboard. Instead, solder solid conductor wires onto each pin of the base, and plug the other ends of each of those wires into the breadboard. Our base had the pins numbered but, if yours doesn't, the pins are numbered clockwise, starting with the pin immediately clockwise from the gap, while viewing the base from the side with the pins. Oh, and when you first apply power, don't expect it to work immediately. Valves only work when their heaters have reached the correct temperature, and that can take quite a few seconds. Since you'll have to solder, even with a breadboard, you might choose a different method of prototyping, like the one at **hsmag.cc/AstableMultivibrator**.

Experienced valve design engineers are thin on the ground, and this author doesn't count in their number. So the circuit design is one he found and modified

slightly. A search will reveal two or three variants, so we suggest you use our circuit just as a starting point. Some circuits, such as the one in the previous YouTube video and the one at **hsmag.cc/VacuumTubeAstable**, have the grid resistors connected to the positive supply instead of 0 V, although in the first video, the schematic isn't shown, so it needs some detective work. We tried both and barely noticed any difference in performance. Also, some circuits use a higher voltage, more typical of valve circuits. Bizarrely, with the capacitor values shown in other circuits, and what theoretically should be needed, the LEDs flashed several times per second, so we had to increase the values of the capacitors substantially. We never did figure out why – perhaps there was a quirk with our valve, unlikely as that might be since it was new, or perhaps it was concerned with the unusually low voltage we used. The circuit in the first video also used a low voltage, but the component values can't be read. Needless to say, there's plenty of scope for experimentation here, and if you want the fuller story, take a look at **hsmag.cc/Multivibrator**. In your experiments, the only thing to take care with is the resistors in series with the LEDs – if the value is too small, you'll blow up the LEDs.

So, how does the circuit work? Put simply, oscillation occurs thanks to the cross-coupled capacitors between the anode of each triode and the grid of the other. This causes each triode to turn off as the other triode turns on. The reason this doesn't happen so quickly that you wouldn't see the LEDs alternating, is that capacitors take time to charge up, the time depending on the value of the capacitor and of the resistance through which it charges and discharges.

BEYOND THE BREADBOARD

Having experimented on a breadboard, if you decide to turn your final valve circuit into a keepsake, the photos will give you some inspiration, but a few words of guidance might be useful. Valve circuits were often built onto a metal chassis with the valves mounted on top and most of the other components wired to the valve bases in the hollow area below. We chose a similar approach but, in the interests of aesthetics, we used a wooden plinth. The horizontal surface was made from veneered plywood, which shouldn't be too thick, and the sides were made from offcuts of moulded wooden door architrave, reduced in height to only as much as necessary to accommodate the batteries, mitred at the corners, glued and pinned. The whole thing was
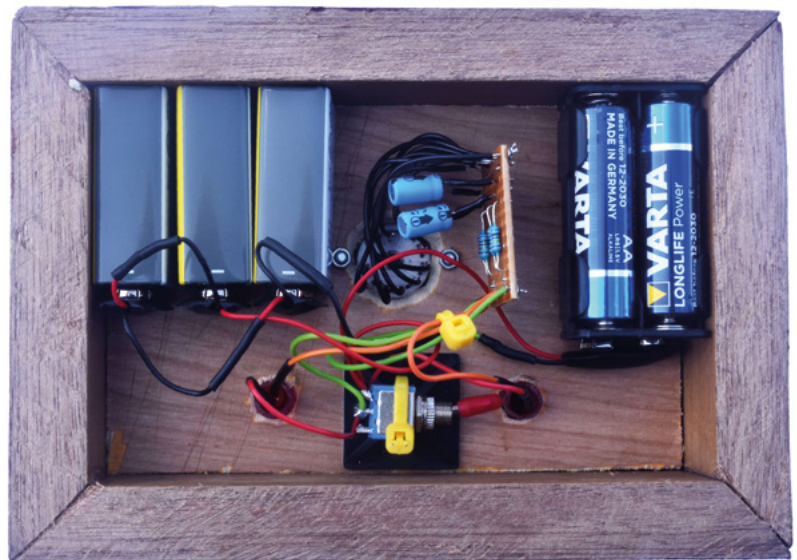
painted with a wood-coloured varnish. It's easiest to make all the solder joints to the base before you fit it to the plinth, especially if the plywood is thick.

Finally, a comment on the LEDs, which we thought weren't visually in keeping with a valve-based circuit. The ideal indicators would have been panel-mounting neon lights, but these require a much higher voltage than 27 V. So we bought a couple of suitable-looking neon lights, but used them only to disguise the LEDs, sawing off the bottoms, removing the neon lights, and inserting the LEDs in their place. ☐
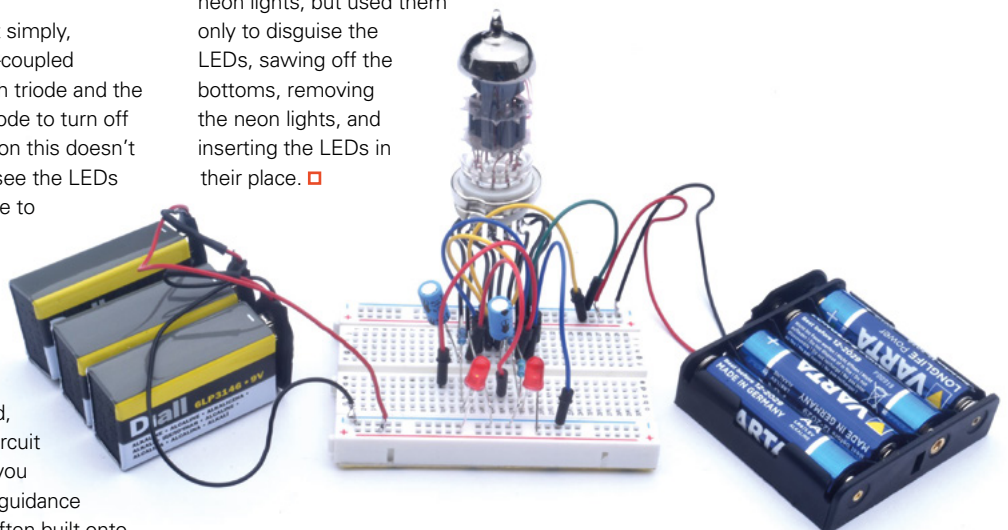
VALVES **TODAY**

Valve amplifiers are still used by some rock guitarists and a faction of the audiophile community because of their fabled warm sound. In other applications, valves aren't used for their pleasing sound but for much more prosaic reasons. A very special type of valve, called a travelling wave amplifier, or TWA for short, remains the primary vehicle for high-power, extremely high-frequency space applications.

Make a fabric drawstring bag

Tidy up that makerspace by stowing away your cables, wires, switches, and more in some handy homemade bags

Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor who also enjoys dabbling in the fabric and fibre arts fields. She owns more yarn than she will ever use in her lifetime.

QUICK TIP

Check the tension of the stitches by sewing on a piece of scrap fabric before you get underway, and adjust the machine as necessary.

Right
One completed drawstring bag to hold your wires, various components, or whatever you want. Adapt the pattern to meet your needs

t's probably fair to say that most makers and crafters have a lot of 'stuff' – cables, wires, PCBs, nails, brushes, threads, balls of yarn… the list is a varied and very long one. It's probably also fair to say that one of the great challenges we face as makers is storage, so why not fashion your own drawstring bags to hide away and organise some of those bits and pieces?

In this tutorial we are going to make a medium-sized, lined drawstring bag in which you can store your paraphernalia, and the beauty of the make is that you can adjust the size to your own requirements; make it bigger, smaller, wider – the choice is yours. It's also worth mentioning that, apart from a very basic knowledge of how to use a sewing machine (i.e. how to thread it up and sew the most basic stitch), you don't need to know a huge amount about sewing, as this is an easy make. Also, the sewing machine that you use does not need to be all-singing, all-dancing (as some can be eye-wateringly expensive) – it just needs to sew a straight line.

STEP 1
PREPARE YOUR FABRIC

First, you'll need a paper template to use to cut out your fabric. Using tracing paper, draw with a pencil and ruler a 45 cm × 35 cm rectangle. Then draw a second 22 cm × 20 cm rectangle (this will be the pocket). Cut them out and pin the larger template to your main fabric. Cut two of these pieces, and then repeat with your lining fabric. (I used a calico for the lining as it's durable and inexpensive.) Take the smaller template and pin it to a contrasting piece of fabric, and cut out just one of these. You should have four large pieces and one small. Now give them a quick press with an iron.

Below �)
Five pieces of fabric ready for sewing – take your time cutting out, as preparing your fabric properly will often lead to a better finish
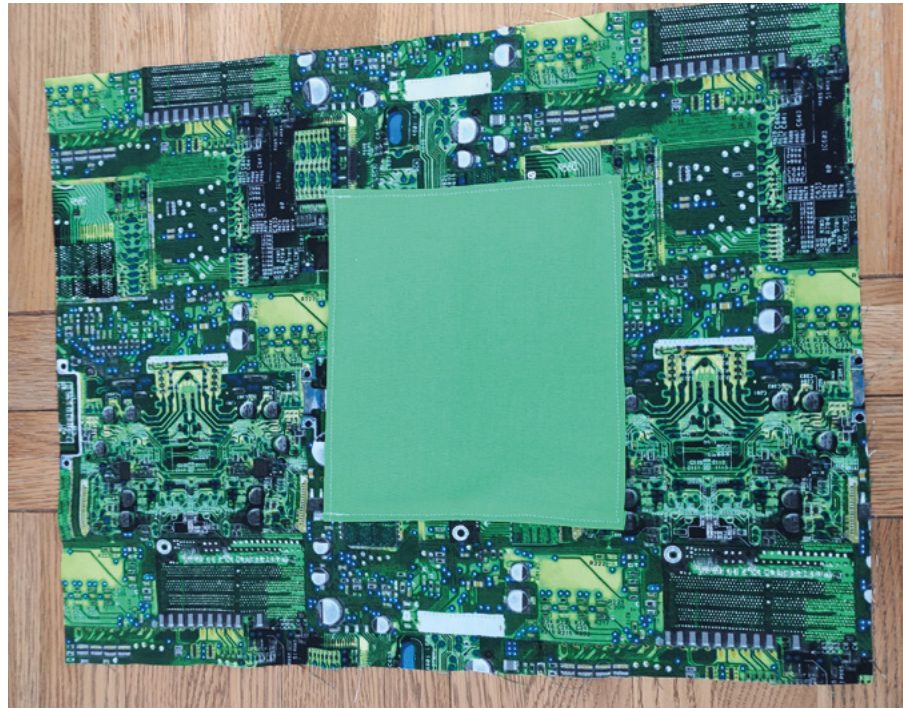
Above ↗
The pocket sewn onto the front piece of the bag – white thread is always a good option colour-wise, and is a sewing staple

STEP 2
PRACTICAL POCKETS

Optional, but useful, we now need to attach the smaller piece of fabric to the 'front' of our bag. Take the smaller piece and fold one of the long edges over by 0.5 cm, press with the iron, and then fold again, so you have no raw edges. Press, pin and, having ensured that the sewing machine is threaded up and ready to go, sew along that seam using the most basic stitch on any machine – the straight stitch. Take it slowly and remove pins before you reach them.

That's the top seam done. Now do the same with the other three sides, but do not sew those yet. Take one piece of outer fabric and place face-up on a table. Take the pocket and position it on the main fabric, ensuring it's central. Pin in place, using the pins you've used to hold the three seams down, but leave the top of the pocket open. Now, sew those three sides to the main fabric. As you sew along the first side and approach the corner, lift the machine's foot, swivel the fabric round and continue to the next corner until all three sides are attached. Voilà – one functional pocket!

STEP 3
SEWING THE MAIN BAG

Take the two outer pieces and put them right sides together, ensuring that the piece with the →

YOU'LL NEED

◆ **½ metre of fabric** (cotton weight) **for the outside**

◆ **½ metre of lining fabric** (also cotton weight)

◆ **Small piece of contrasting cotton fabric for pocket** (optional)

◆ **Approx. 2 metres of ribbon/cord/ tape for the drawstring**

◆ **Tracing paper**

◆ **Pencil**

◆ **Ruler**

◆ **Iron and ironing board**

◆ **Scissors –** for paper and dressmaking

◆ **Pins/clips**

◆ **Safety pin**

◆ **Thread**

◆ **Tailor's chalk**

◆ **Seam ripper**

◆ **Access to a sewing machine**

Right ◆
Ensure you leave a 12cm gap in the side of the lining, and trim off any stray threads as you go along

Above ◪
Right sides out, it's starting to resemble a bag. The iron is your best friend when it comes to a crisp outline

pocket is not upside-down. Sew around the left-hand, right-hand, and bottom sides, leaving the top open. I suggest using a 0.5 cm seam allowance. Then, do the same with the lining fabric pieces, but also leave a 12 cm gap on one side so that we can turn the bag to the right side later. It's always a good idea to press with the iron as you go, as it keeps things looking professional, so, after sewing, press the outer and inner pieces and press open the seams.

STEP 4
ATTACHING IT ALL TOGETHER
Turn the outer fabric right sides out and poke the corners out with a pencil so they are neat. Trim stray threads as you go. Then, place this piece inside the lining, right sides together, so the side seams and top edge line up. Then, pin or clip these pieces together and sew around the entire top edge.

Next, reach in through the gap you left in the lining side seam and pull everything out to the right side. It's now starting to look like a bag! We also need to sew up that gap in the lining; so, with the seam allowance turned inward, sew a line of stitching to close that

QUICK TIP

Use a lighter, plain-coloured fabric for the lining, as it makes it easy to see what's inside.

gap. Remember, this will be unseen on the inside of the bag, but keep it as neat as you can.

Next, push the lining inside the bag and you'll need to press it again with the iron at this point to flatten it out, especially the top edge. Then sew a topstitch around the upper edge, around 0.5 cm from the top, and when done, turn the bag inside out so you are looking at the lining.

### STEP 5
THE LAST BIT OF SEWING
Grab your ruler and tailor's chalk and measure 2 cm from the top of the bag. Draw a line all the way around (i.e. on both sides). Then measure another 2.5 cm below that, again drawing a line on both sides. This will give us a 'channel' for the ribbon or cord to go through. Note: If you were to use a thicker ribbon or tape as your drawstring, you could make this channel deeper so that whatever you are using fits properly – just measure the width and decide how deep you need it.

Next, turn the bag back to the right side and sew around those two lines that you have drawn. We don't want to sew the front of the bag to the back here, we just want two lines of stitching over the drawn lines. Again, take your time, place the lower side under the presser foot and work your way

SCISSOR HAPPY

It's worth mentioning here that scissors are not all made alike; generally, scissors are intended for a specific purpose. Dressmaking scissors, for cutting fabric, should NEVER be used to cut paper, as incorrect usage will blunt them and make them next to useless when it comes to cutting fabric. In this tutorial, make sure that you cut your paper template with general-purpose paper-cutting scissors, and cut your fabric with scissors designed for that job. There are lots on the market (**hsmag.cc/FabricScissors**) if you don't already own a pair.

Also, we've lined our bag so that we have a neat, professional finish, with no fraying ends inside the bag. If you wanted to skip the lining, you could trim the inside fabric edges with pinking shears (**hsmag.cc/PinkingShears**) which have a serrated blade and cut a zig-zag edge in fabric to prevent fraying.
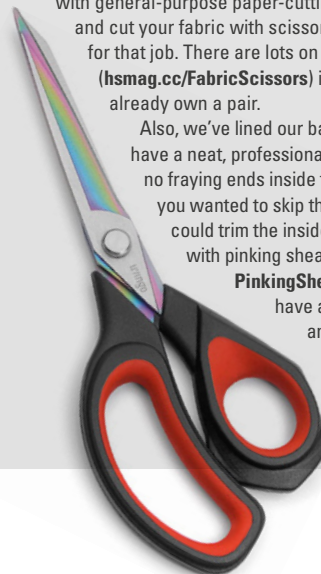
around. Ensure that on the seams on either side, you do a few backstitches, as this will help us when we come to insert the drawstring.

Now, take a seam ripper or small pair of pointed scissors and, on the outside of the bag, carefully remove the stitching between the top and bottom lines of channel stitching at the side seams. (This is why we backstitched at the seams – a little reinforcement!)
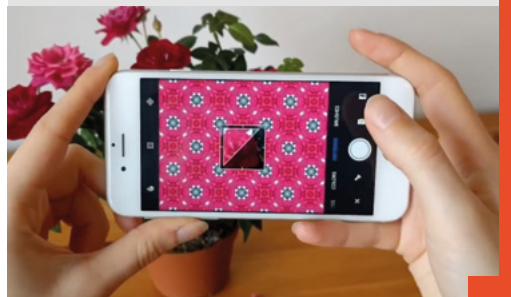
STEP 6
THE DRAWSTRING!

You'll need two pieces of ribbon or cord, each around three times the width of your bag for your drawstring. I used a fabric tape. Take your safety pin and attach it to one end of the drawstring. Then, feed it through the channel all the way around the bag, pull it out, remove the safety pin, and gently knot the two ends together. Do the same again with the other piece of ribbon, or whatever you are using, but insert the drawstring on the other side of the bag, thread all the way through, remove the pin, and gently knot. And there you have it – a usable piece of storage, easy to open and close, and hopefully customised to your own bespoke requirements.

FABRIC FAVOURITES

The range of fabrics available to buy both online and in fabric stores is huge, and the breadth of designs is also vast. I bought my cool circuit board-inspired 100% cotton from here: **hsmag.cc/CircuitBoardFabric**, but there are a myriad of different designs to choose from. Or, consider designing your own fabric, something that we've previously covered in HackSpace magazine – **hsmag.cc/DesignFabric**.

In this tutorial, we are using a simple cotton fabric, but if you wanted something thicker or more sturdy to house slightly heavier items, you could try heavyweight fabrics such as denim or twill which have a thicker weave. If you wanted to make a basic bag and just have press studs at the top instead of a drawstring, you could even use a leather/leatherette or PVC, and you wouldn't need a lining.

MIX IT UP

Now that you've made your first bag, why not make more for fellow makers? Or these bags could replace wrapping paper at Christmas gift-giving and be recycled each year. As previously mentioned, you can make your paper template bigger, smaller, narrower – there really are no rules and no sewing police, so do your own thing! You could also add a press stud to keep the top of the pocket secure, add another pocket on the inside, use stamps and ink-pads to stamp your own design on the outer pieces, or use a batting/wadding material between the outer fabric and lining to give some rigidity and cushioning to your bag. Or, why not make your own fabric drawstring to the right width in a contrasting fabric? Make these handy bags to fit your needs, and start to organise some of those lonesome odds and ends that need a home. ☐

Left ◆
This is where we create the drawstring channels – adapt them to the right depth to fit whatever drawstring you are using

QUICK TIP

Take your time when sewing, and don't feel the need to rush. This will result in a neater finish.

Left ☑
Carefully remove the stitching between the top and bottom of the channel at the seams, ready for your drawstring to be threaded through

Below ◆
Use your imagination and make them out of any fabric you like or have in stash

Isomorphic keyboard: hex keys

Use the Pico board 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

Last month we showed you how to make a switch matrix and enclosure for an isomorphic layout keyboard. This month we will finish the construction, and look at the Pico board, 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 to bring it all to life.

01 The control keys

We added four triangular control keys to change various non-playing parameters, and built them on a separate piece of stripboard from the main matrix. They are, however, wired into the main matrix as an extra column, number 14. **Figure 1** shows the physical layout diagram of this. Note that the top left-hand corner of the board has been marked with a small white mark, as we found it helped us orientate the board while adding the components. During play, these are used to switch the MIDI channel and shift the octave of the keyboard.

02 The OLED display

We used a 128 by 64 OLED I2C display to provide visual feedback on what notes were being sent. You can use any colour, but we used the type with a strip at the top in one colour, and the rest in another. The specific one we used had yellow LEDs for the top 16 rows, with the rest of the display blue. Like our other Pico projects, we fitted a right-angled header on the display, pointing inwards to give a low profile mounting, and milled out half the thickness of the box to accommodate the display's bezel and connector strip.

03 Rotary encoder

We added our normal KY-040 rotary encoder to interact with the display and control the Note On velocity during play. Due to the library we used to read it, the clock and data signals needed swapping round in order to get the rotation correct – that is clockwise rotation producing an increasing count. **Figure 2** shows how we did this by swapping over the two connectors in the ribbon cable. Note the green and yellow wires cross so we can keep the interface connectors the same as we normally use on our other projects.

04 The Pico board

The whole keyboard is driven by the Pico controller board, and the schematic is shown in **Figure 3** (overleaf). The OLED and rotary encoder are directly connected to the Pico, as are the row inputs of the matrix. These are connected to a 4-,

▼ **Figure 1** Triangular control keys – physical layout

Figure 1

Row 0   Row 1   Row 2   Row 3   Column 14

37
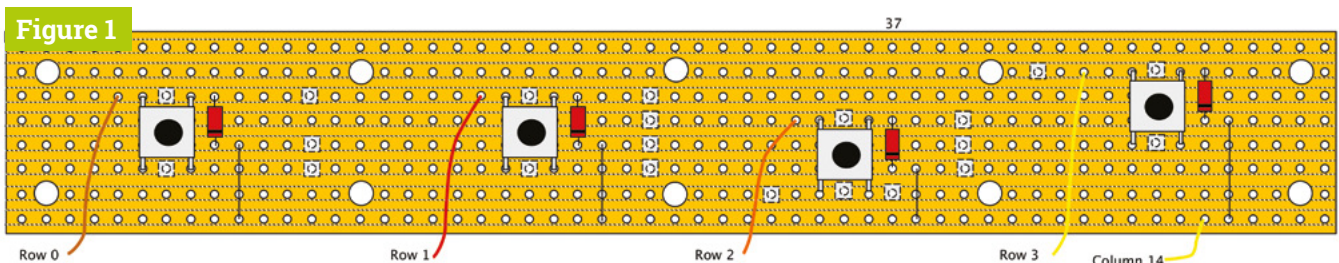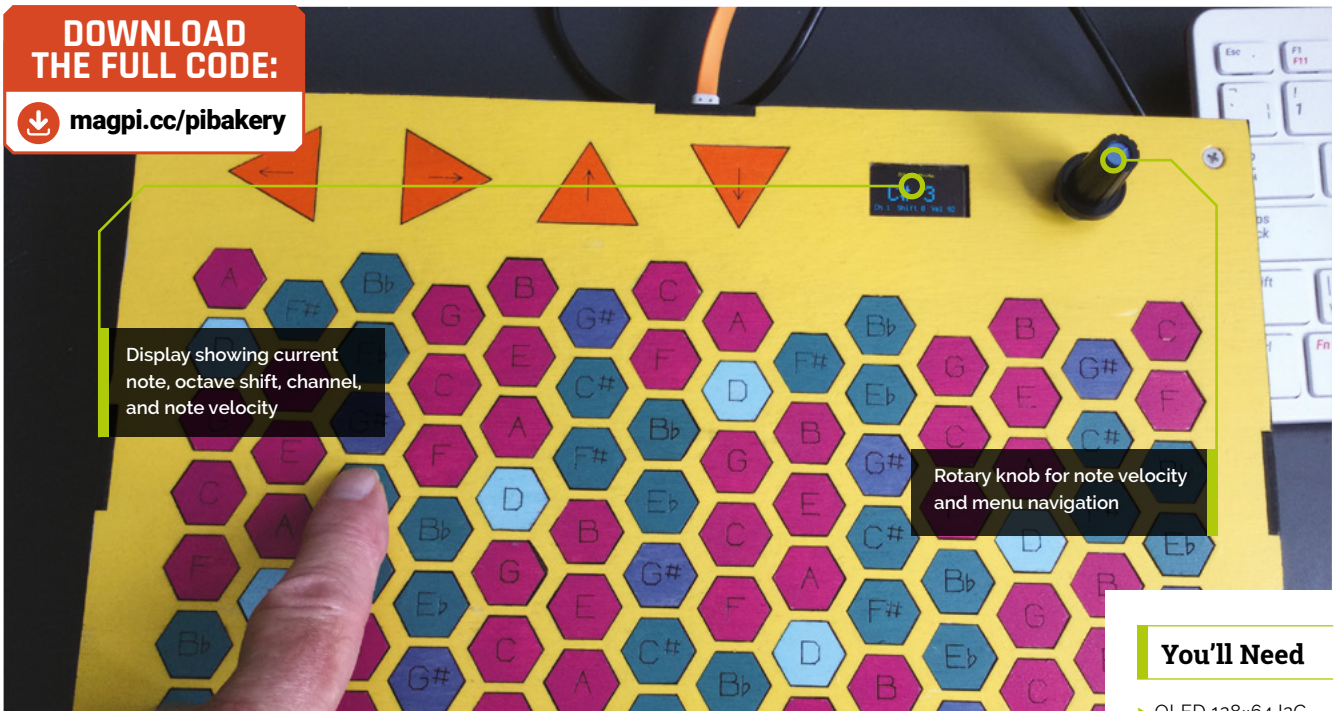
Display showing current note, octave shift, channel, and note velocity

Rotary knob for note velocity and menu navigation

5-, and 8-pin header respectively. The matrix column signals, on the other hand, are generated by two 74HS138 decoder chips, because otherwise we would not have enough Pico pins. Each chip produces a logic zero on the output determined by three binary bits input. By using an extra two pins, we can enable one decoder or the other.

05 The Pico board construction

Figure 4 shows the physical layout of the Pico board. Note that here we are not only using the normal Pico pins, but test pad 6 to bring out the bootloader switch to the back panel. We knew in advance that there would be a lot of dismantling to be done if we ever needed to press it, and we were very glad when we found it necessary during development. **Figure 5** shows the point on the

back of the Pico board to solder this wire. Note the column output connection order was chosen to simplify the wiring of this board.

06 Wiring the matrix

We used ribbon cable with crimped headers to do all the interconnections. As we only had 21 cm lengths, we needed to use two lengths to make them long enough to fit for assembly. They were joined together with a strip of long pin headers and, as shown in **Figure 6**, we needed to use two back shells to make up the required number of column pins. These connectors have to live inside the lower processor enclosure after assembly, so some of the wires had to be extended by a few centimetres in order to reach where they were going on the matrix board.

You'll Need

- OLED 128×64 I2C display SSD 1306 yellow/blue **magpi.cc/piboled**

- KY-040 rotary switch **magpi.cc/pibky040**

- Momentary push switch **magpi.cc/pibmomentary**

▼ **Figure 2** Swapping over the clock and data in the encoder cable
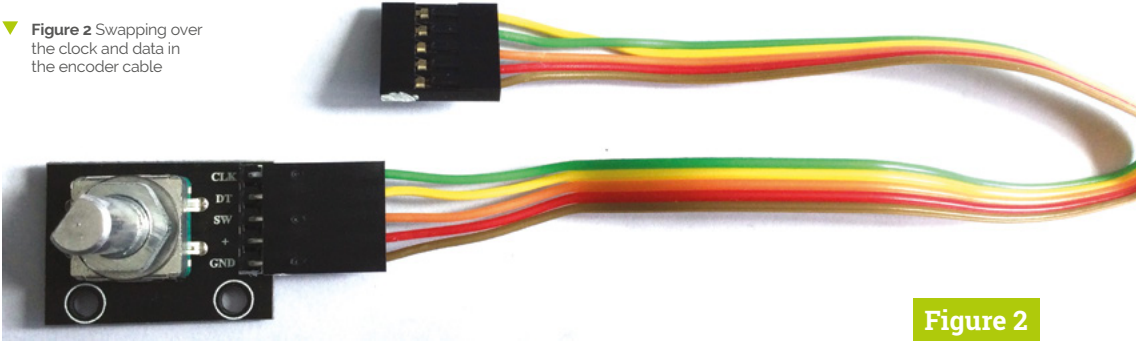
Figure 2
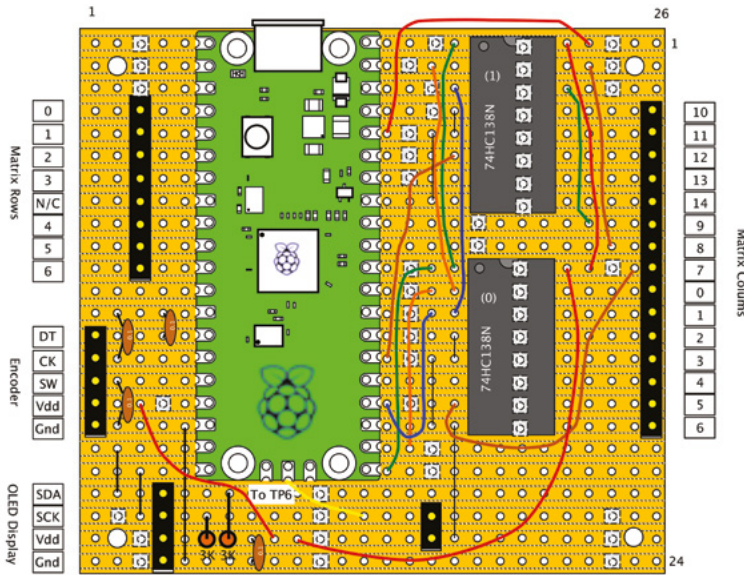
Figure 4

Break track on every row
on this column

Boot Mode
push button

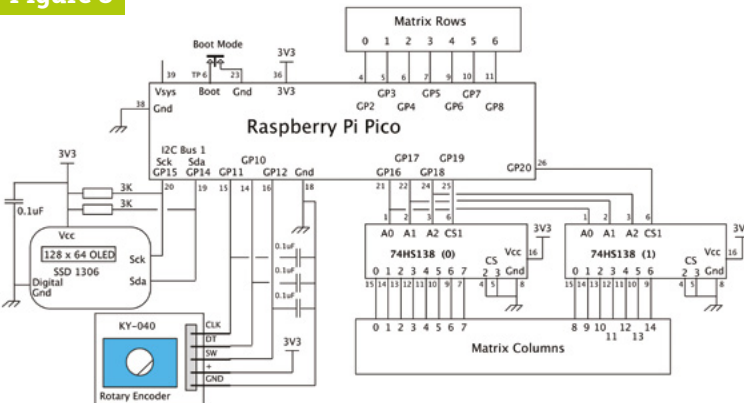▲ **Figure 4** Physical
layout of the
Pico controller

Top Tip 👍

Fitting the lid

If you find a key
will not click when
pushed down, you
probably have
a wire that has
moved over a key.
Remove the top
and rearrange
the wires.

▼ **Figure 3** Schematic of
the Pico controller

Figure 3

07 Test all the parts

Before engaging in the final assembly, make
sure all the sections work. **Figure 7** shows the
bottom section with the Pico processor and leads.
Connect the display and encoder to the ribbon
cables, but don't mount them on the top yet.
Also connect up the column and row connectors
to the matrix. Then hop over to our GitHub page
and run the examples to test the encoder, display,
and matrix. With the matrix, check that every key
gives a key number. If any key shows repeating key
numbers, it is likely that breaks have not been fully
made in the wiring around those keys.

08 Mounting the top

With the OLED display fixed into place and
the rotary encoder bolted on, you can now attach
the top. Put the top side ring into place, and make
sure all the ribbon cables for the matrix are tucked
inside the lower box (**Figure 8**). We found it best
to fit three columns at once, turn them over while
pressing a board over the top, and then tape them
up. We did try turning the whole lot of keys over
twice, but a lot fell out. Do this with the top on a
thick book so you can get it flat with the encoder
shaft not fouling it.

09 Final assembly

Now attach the ribbon cables to the display
and the encoder, and place the top on the side
rings. Just put one of the countersunk screws in the
top and just engage the thread, but don't screw it
up yet. Make sure no wires are sticking out of the
sides; push them back if they are. Make sure the
side ring is clicked into the middle tabs, and then
tighten up the top screw. Insert the other screws
and tighten them up carefully, checking no wires
poke out and the tabs remain engaged. Test that all
the buttons click when pressed.

❝ Before engaging in the final assembly, make sure all the sections work ❞

Figure 5
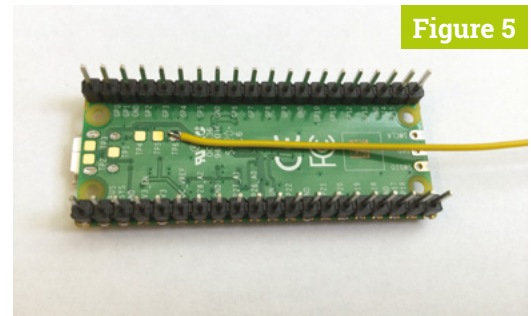
▲ **Figure 5** Accessing the bootloader signal

10 The software

We decided to use CircuitPython from
Adafruit, having used MicroPython, and C in our
previous two projects. It had the big advantage
of having a USB MIDI capability, but does not
have any interrupt handling capability. This was
nearly a showstopper until we found an example
of using a rotary encoder using the GPIO pins. We
found the OLED driver complex to understand,
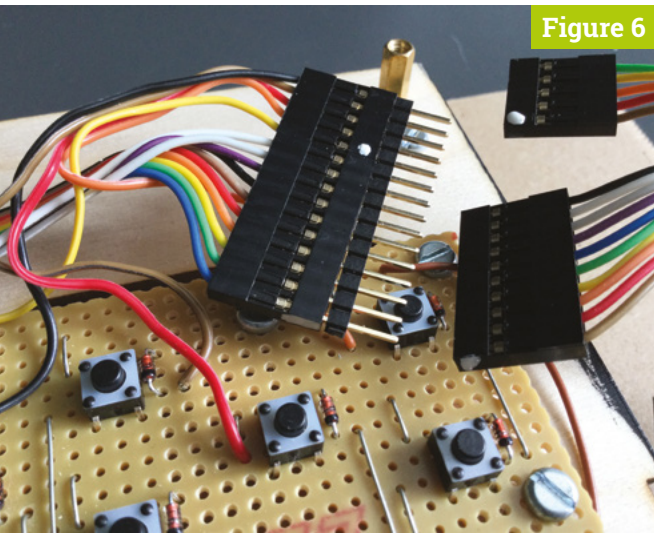especially for the simple display we wanted to

Figure 6

Figure 7

produce, and the examples we found weren't all that comprehensive. But, in the end, we prevailed. See the implementation notes on our GitHub page for details of how to set up your Pico.

11 Play mode

In the play mode, the note name and octave number are displayed when you hold down a key; when more than one is held down, the last key is shown. The buffer can cope with up to ten keys being held down at once. The left/right triangle controls change the channel, and the up/down ones change the octave. The rotary encoder selects the note-on velocity. When playing Channel 10, the percussion names are displayed. To get into the instrument select mode, you push the switch incorporated into the rotary encoder.
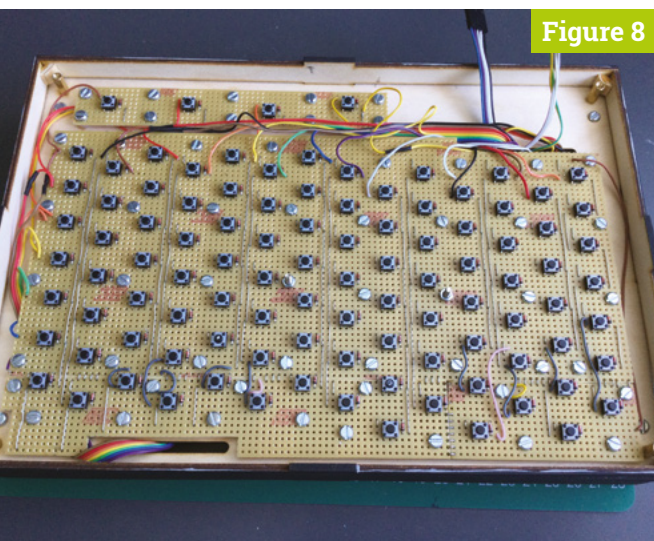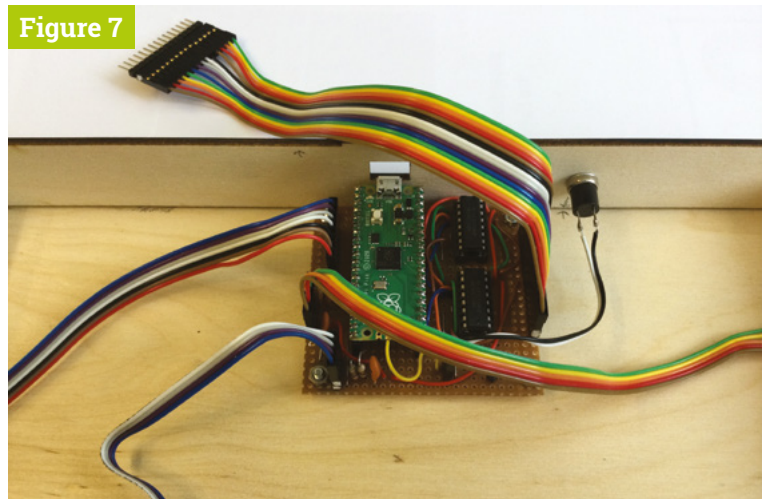
12 Instrument Select mode

Once in the Select mode, a list of instrument groups is shown. Turn the encoder to highlight the required group, then push the encoder knob again, and you are taken to the first instrument in that group, in a list of all 128 instruments that you can scroll through. Push the encoder to send a MIDI message to change the instrument, and the left/right controls to change the channel. The up control takes you first to the instruments and then back into play mode. Note that, once in the Select mode, the down button acts the same as pushing the encoder button.

13 Chords you can play

Each note appears twice on the keyboard, left and right side. Given any starting note, the note above it is a perfect fifth up, to the left is a minor third, and to the right is a major third. So you can play a major or minor triad, as shown in **Figure 9**. Chords shapes are the same for any chords in any key. This is reputed to be much easier to learn than a conventional piano keyboard, and once you know how to play one chord, those in other keys are the same shape, but just transposed. Ⓜ

Figure 6 Connecting the matrix column signals to the Pico on the bottom layer

Figure 7 The Pico board ready for connection to the peripheral devices

Top Tip 👍

Other chords

For more chord shapes, see **magpi.cc/chordshapes**.

Figure 8 The top layer waiting for the keys to be placed

Figure 9 Note intervals and simple chords with the Hex-Keys layout

Figure 8

Figure 9: Minor Chord — Key. 1 Semitone (Low–High): Minor 3rd, 5th, Major 3rd, Any Key (7 Semitones, High–Low). Major Chord — Key.
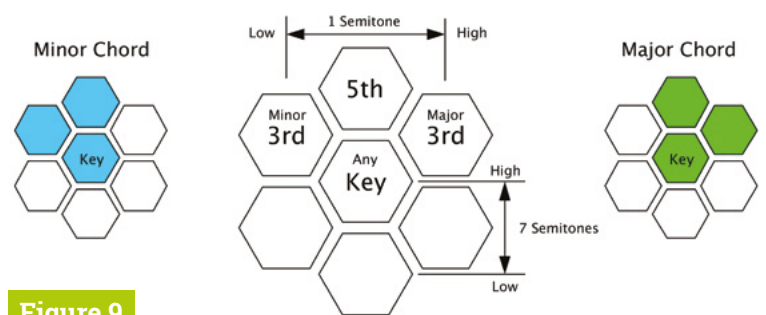
Figure 9

Getting Meshy in FreeCAD

Get your designs ready for 3D printing,
CNC milling, and more

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!

S TL files have been around since
the 1990s, and in simple terms,
they contain a mesh of triangular
facets that describe the shape of the
design. STL files are a common mesh
format to use in 3D printing where they
are imported into a 'slicer' piece of software that
generates the G-code for the 3D printer to follow.
As such, STL has become a standard file type for
sharing 3D print designs on sites such as Thingiverse
and GrabCAD and more. Whilst heavily associated
with 3D printing, STL can often be used for other
processes such as CNC milling or routing.

Most of the time, for simple parts, creating an STL
file in FreeCAD is simple and somewhat automated.
As an example, let's quickly create any object in the
Part or Part Design workbench. In **Figure 2**, you can

see we have created a cube object and a sphere
object in the Part workbench and cut part of the
sphere out of the cube.

With the cut object selected in the file tree, click
File > Export and you should see that FreeCAD
suggests you export the object as an STL file.
Give the file a name and export it, and it's ready
to be sliced for 3D printing. In the background,
FreeCAD has made a lot of sensible choices about
the resolution of the mesh of our exported part.
Put simply, FreeCAD makes a choice about the
number of triangle facets our exported STL mesh
object contains.

Sometimes we might want to have more control
over the mesh that's created for our part, and we
might want to change the resolution to get the best
results for our project. This is one area where the

Figure 1 ◇
A simple example of
a mesh with an issue
we have discovered
using an automated
analysis tool
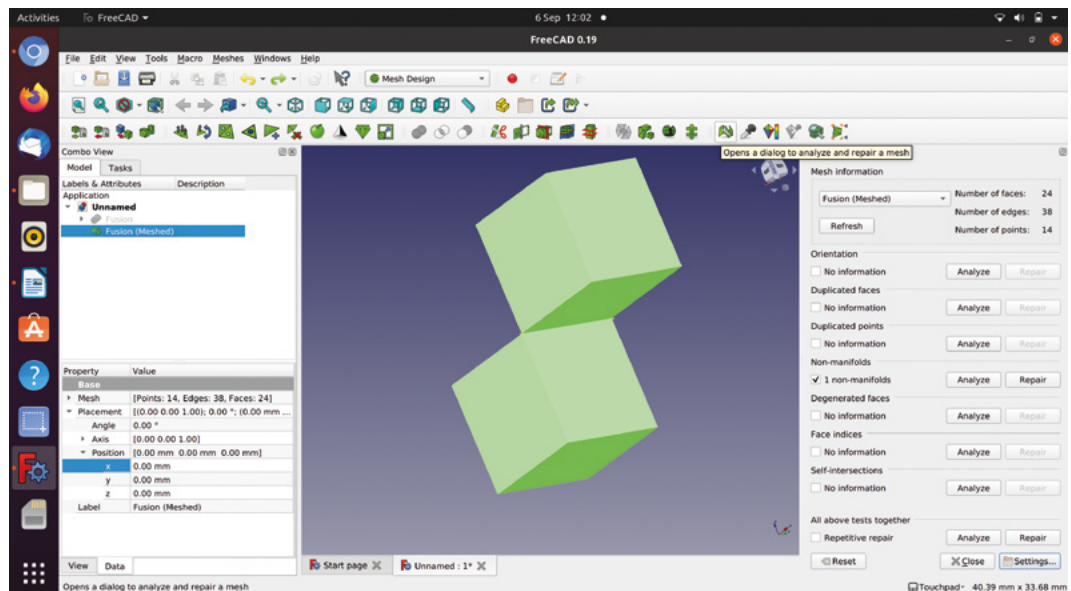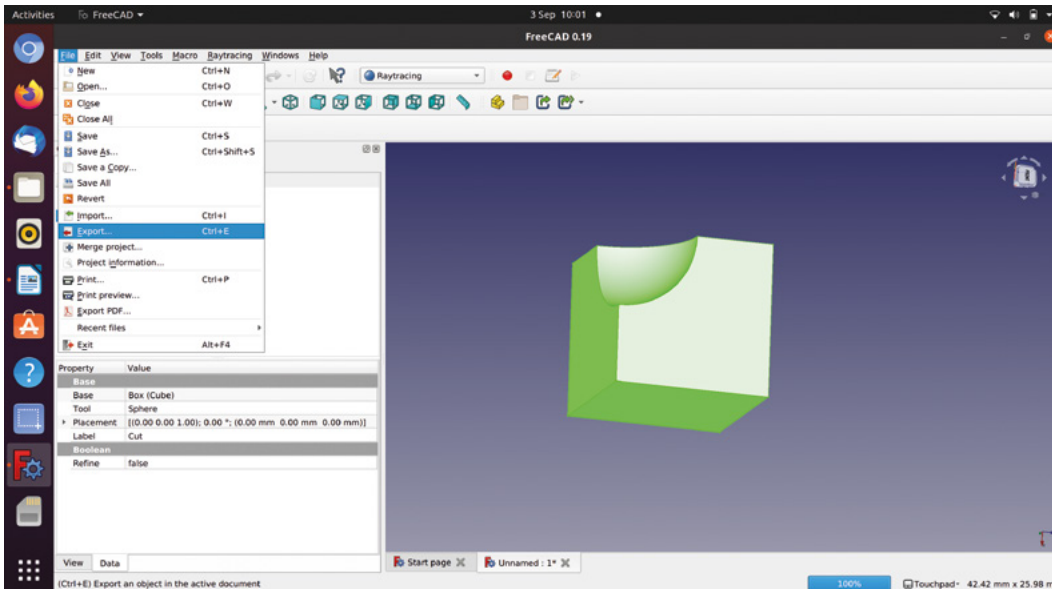
more specialised tools on the Mesh workbench can help. As an example, we have a nose cone part for a rocket designed – let's move to the Mesh workbench using the drop-down menu to create some different resolution meshes to compare.

On the Mesh workbench, highlight the nose cone part and then click the tool that reads 'tessellate shape' on the tooltip as you hover over it. This tool is also accessible by clicking Meshes > Create mesh from shape. You should see a dialog box in the combo view area, and you should see two changeable values on the Standard tab. The first value, 'Surface deviation',

should be 0.10mm, and the second value, 'Angular deviation', should be 30 degrees. If they aren't, set them to these values and then click OK (**Figure 3**).

RESOLVING RESOLUTIONS
You should now have two objects in the file tree: your original nose cone part and the new mesh object. Similar to other items in the file tree, you can toggle the visibility of the mesh and the part by highlighting the object and pressing the **SPACE** bar. Select the original nose cone part and set it as invisible so that we just see the mesh object in the preview window.

You can see in **Figure 4** that the created mesh consists of lots of triangles closely approximating the part, but we can adjust the values to increase the
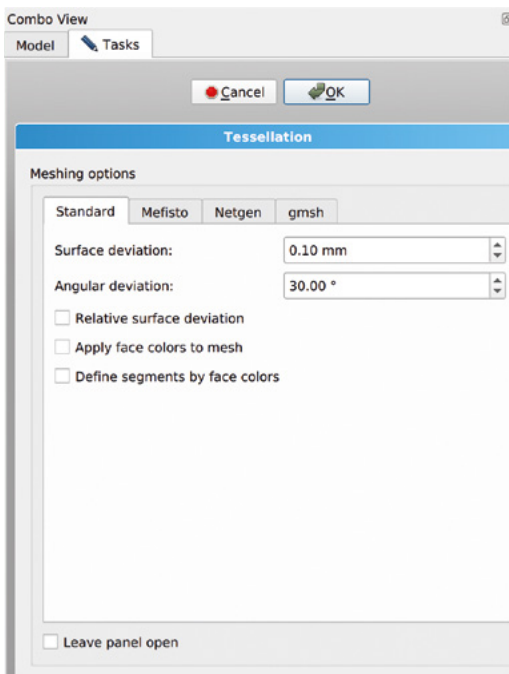
> FreeCAD has made a lot of sensible choices about **the resolution of the mesh of our exported part**

resolution of the mesh if we delete our first mesh, make the nose cone part visible, and highlight it so we can create a new mesh. For this mesh, let's set the surface deviation to 0.03mm and the angular deviation to 10 degrees. You should see, similar to **Figure 5**, that the resulting mesh is a higher resolution and more precisely matches the original part. Of course, this also increases the complexity of the mesh and, if taken too far, may well begin to slow down your graphics card performance.

To export a mesh from the Mesh workbench, you can highlight the mesh in the file tree, right-click, and select 'Export mesh'. There are numerous options of STL types, but we tend to export as 'Binary STL' and, →
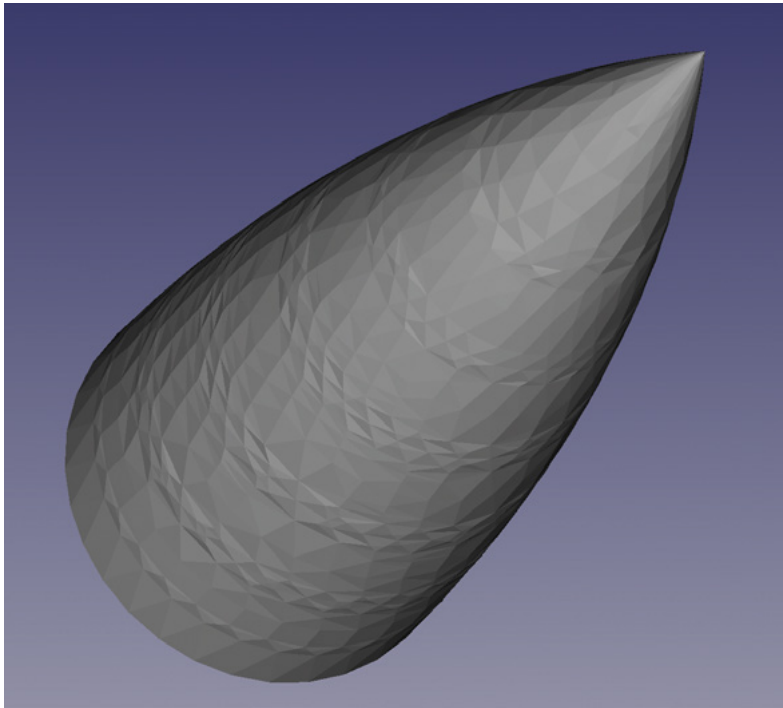
Figure 4 ◆
Our first attempt at
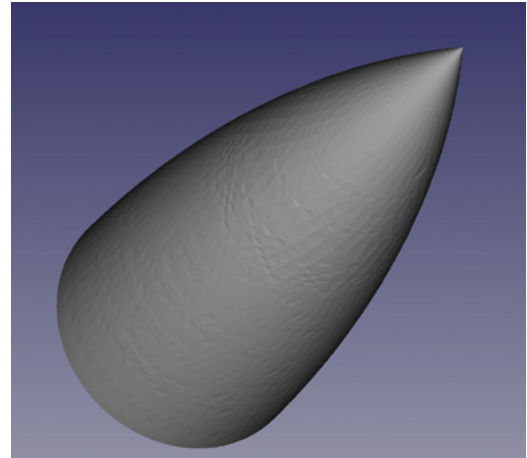creating a mesh of the
nose cone part

Figure 5 ◆
Our second mesh resulting in a
higher resolution mesh, which more
closely matches our nose cone part

these seem to work well with our slicing software for 3D printing. One thing to note is that the Export mesh function seems to want us to add the '.stl' to the file name in the name input box.

Of course, creating a mesh file from an object we made in FreeCAD is just one small aspect of the work of the Mesh workbench. One area it is particularly useful in is for editing and amending mesh files. This scenario often arises when we've found an object online that we wish to 3D-print but will require some reworking. Often, if you are downloading an object from a website such as Thingiverse or GrabCAD, you may only have the option to download a mesh file/STL, or indeed you might have the project file but it is made in software that you don't own.

As an example, we have an STL file of a hands-free door puller. Let's pretend that, in use, we decided it would be better if the hook faced the other way round. We could, of course, with the knowledge within this FreeCAD series of articles, draw this part from scratch in a matter of minutes, but let's use the Mesh workbench to make some changes. Click the 'Imports a mesh from a file' tool icon and navigate to the STL file and select to import it (**Figure 6**).

The Mesh workbench has some tools that are similar in approach to the Part workbench, where

> The Mesh workbench has some tools that are **similar in approach to the Part workbench**

we create simple solid objects and combine them using Boolean functions to stick objects together, or subtract one shape out of another. If we want to turn around the hook part of our mesh file, we will create a copy of the hooked end and rotate it using these tools.

REGULAR SOLIDS
First of all, click the 'Build a regular solid' tool icon or select the same tool by clicking Meshes > Regular solid. You should now see a small window with a drop-down to select various shapes, cubes, cylinders, etc., and some entry boxes to add dimensions. We will create a cube 35 mm long, 50 mm wide, and 10 mm high. Similar again to working with solids on the Part workbench, if you highlight the cube object in the file tree, right-click and select Transform, we can then move the cube into place over the hooked end of the door puller (**Figure 7**).

Next, select the two objects in the file tree and click the 'Intersection' tool icon. You will now see an additional item in the file tree labelled 'Intersection'. If we toggle the cube item to be invisible and then select the intersection item, we can then right-click and use Transform to move the intersection object, which is a copy of the hook section, away from the

QUICK TIP

The Intersection, Union, and Difference Boolean operation tool icons look like green versions of their counterparts on the Part workbench.

rest of the door puller object. Next, make the cube visible once more, select the door puller object and the cube and then perform a 'Difference' Boolean operation on it. You should now have a version of the original door puller with no end and a copy of the hooked end (**Figure 8**).

All that remains is to move and rotate everything into the new position, and then we can use the 'Union' Boolean operation to create a new single mesh object. We can then highlight the new door puller union object and click the 'Export a mesh to file' tool to create our new STL file ready for 3D printing.

Continuing to use our door puller tool project as an example, let's explore how we create a Part object from a mesh. Moving to the Part workbench, we can highlight the mesh – in our case called 'Union' in the file tree view – then click Part > Create shape from mesh. You should now see that a part has been made of our door puller object (**Figure 9**). Let's make our mesh object invisible in the file tree to explore our part. You should see that the part object creates its faces and edges based on the triangles in the mesh, which means that it is an accurate copy, but also means it can be a little challenging to work with. You can, of course, select faces and map sketches to faces, which is pretty straightforward on larger flat sections of a part but gets challenging if you consider the curved part of our puller. It can also be useful to convert meshes to parts if you want to recreate aspects of a mesh design. Converting a mesh to a part means that you can use the 'Link to an external geometry' tool in the Sketcher workbench to pull in reference points to create sketches over parts converted from meshes.

Sometimes we might want to work with extremely complex mesh files. If you've ever downloaded a complex model, such as a model of a person or object that's full of complex organic curves, you might notice that your graphics card starts to slow down →

Figure 6 ◈
Importing an STL file to the Mesh workbench for editing
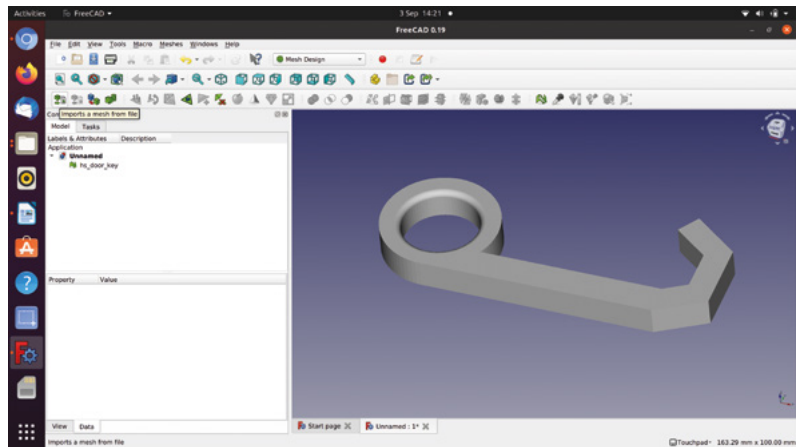
OPENSCAD

FreeCAD is a community-developed, open-source piece of software that uses other open-source pieces of code and modules within it. Another excellent open-source piece of CAD software is OpenSCAD, which uses a code-based approach to 3D modelling. Files and models created with OpenSCAD can be imported directly to FreeCAD. FreeCAD has an OpenSCAD workbench to enable people to use the best of both software approaches.

In the Mesh workbench some of the functionality, including the Boolean operations we are using in this article, require OpenSCAD to be installed and a path to the OpenSCAD executable to be set up in FreeCAD. Head over to **openscad.org** and download and install the latest version for your operating system. Once installed, back in FreeCAD, move to the OpenSCAD workbench and then click Edit > Preferences and scroll down to select the OpenSCAD icon. In the Preferences dialog, the first item is an input box to set the file path to the OpenSCAD executable – your system should have detected the correct path, but it's worth opening the file location and checking it is all set correctly. Once checked, click the Apply button and close the preferences.
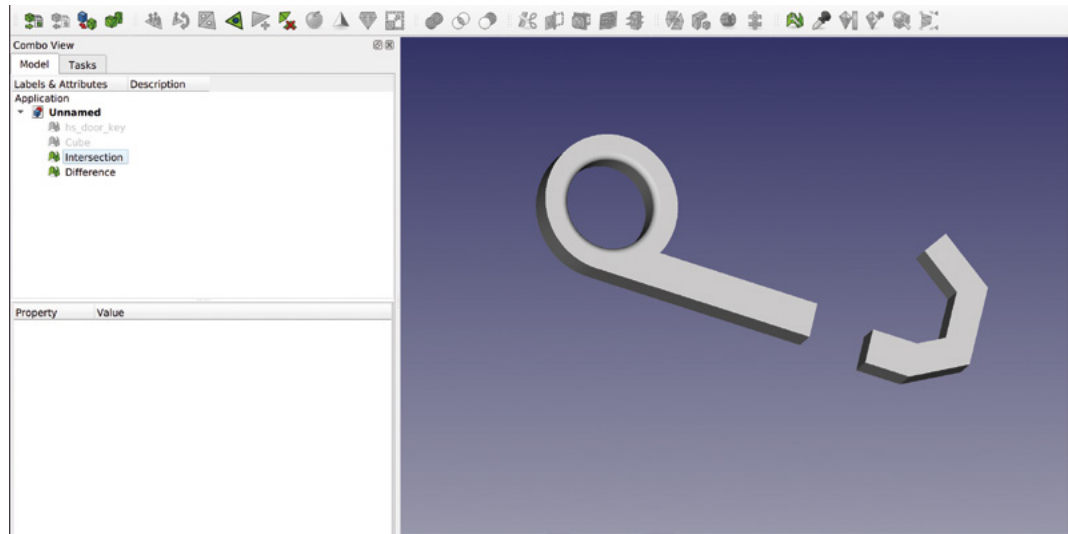

Figure 7 ◈
Creating a mesh solid cube and positioning it over the part we intend to rotate
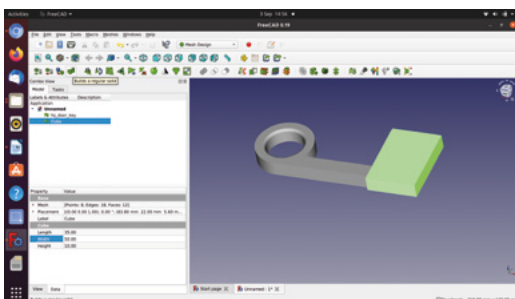
QUICK TIP

You might want to install the brilliant Rocket workbench using the Add-on Manager if you want to play with the nose cone design.
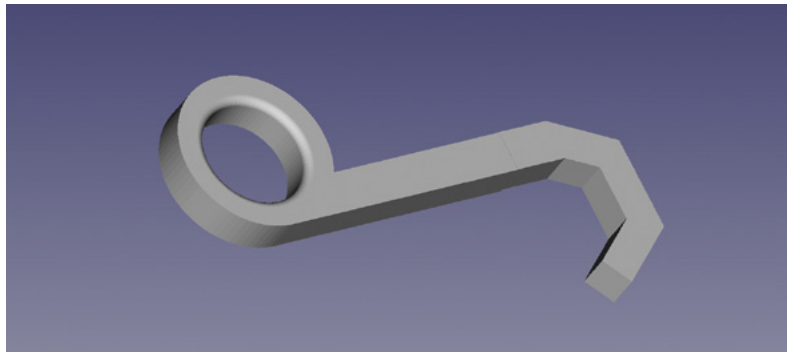
> **Mesh decimation must be used in moderation, as taken too far, it's easy to completely destroy the object**

Figure 8 ⏎
New parts created and ready to be put back together in our improved configuration

Figure 9 ◆
With the two parts of the puller rearranged, we combine them back into a single mesh using the Union Boolean operation ready for export

when you manipulate the object. The FreeCAD Mesh workbench has a handy tool to play with optimising the mesh resolution that may be able to reduce the data, or put simply, the number of triangles in a given mesh while still retaining enough detail. This tool is called the 'Decimation' tool. Mesh decimation must be used in moderation, as taken too far, it's easy to completely destroy the object, which can also be a fun experiment. In the preview window, let's zoom in so that we have a close view of a part of the internal chamfer on the large hole in our door puller object. Notice that when zoomed in, you can see the facets that make up the mesh surface. As this is a small file, we've made this mesh a pretty high resolution, but let's explore the effects of decimation. First, select the door puller object, in our case called 'Union' in the file tree. Notice that in the dialog box, once selected, we get some details of the number of points, edges, and faces our mesh contains. For us, these are 1606

points, 4818 edges, and 3212 faces. Next, click the 'Decimates a mesh' tool, which appears as a green diamond-shaped tool icon. In the Decimating dialog box that appears, set the reduction level to 50 percent and leave the absolute number box unchecked. Click OK. In the preview window, if you are still zoomed in, you should notice the resolution of our puller part change, it should look cruder and the triangles should be more obvious (**Figure 10**).

If we check the details of our mesh object in the dialog box, we can see it has reduced the number of points, edges, and faces by 50 percent to 803 points, 2409 edges, and 1606 faces. This carries through to affect the file size of an exported mesh, and indeed, exporting this decimated version creates a file size percent smaller than the none decimated mesh. If you zoom out on our 50 percent decimated mesh version of the door puller, you'll see that it makes very little real-world difference to the resolution of the object, and the size and shape of the object is unchanged. You can experiment with the decimation value – it can be quite amusing to see what happens to a shape when you approach the near 100 percent decimation values.

Sometimes we might work with mesh files, or indeed create mesh files, that have issues and problems. The Mesh workbench has a tool that enables us to quickly and repeatedly analyse meshes for problems. With our mesh object highlighted, click the 'Opens a dialog to analyse and repair a mesh' tool. This should open a large dialog box on the right-hand side of the preview window. There is a list of potential problem items that FreeCAD can analyse a mesh for. You can analyse a mesh for each item on the list individually, or you can click 'Analyse' on the 'All above tests together' option at the bottom of the list. Close that dialog box for now, and let's quickly make a new simple object with a problem!

In a new project, move to the Part workbench and create two identical default cube solid objects. These default to 10 mm cubes, using Transform, move the second cube object up 10 mm in the Z axis and across 10 mm in the X axis. Finally, highlight both cubes in the file tree view and perform a Union operation to make the two cubes – one part called 'Fusion'. Moving to the Mesh workbench, with our fusion object highlighted, click the 'Tessalate shape' tool icon to create a mesh object. You can see the resulting mesh imported in **Figure 1**. Make the original solid part fusion object invisible in the file tree, highlight the mesh object, and then reopen the mesh analysis dialog as before. Double-check that your mesh object is the object listed in the Mesh Information section (especially if you have multiple projects open), then click the Analyse button on the 'All above tests together' option. The tests will run quickly, and you should see just one error reported.

You should see that in the list of analysis areas in the dialog, the 'Non-Manifolds' section has a check box with one non-manifold issue found. A simple way to think of this common error to understand when something is non-manifold is to consider if it is possible to be manufactured. In this instance, our two cubes are only connected at one edge and they do not overlap at all, which means, in essence, that the connected edge is infinitesimally small. Of course, it's impossible to create a connection between these objects, and as such, they are described as non-manifold. Another way to create a non-manifold example is to create a tube with no ends where the surface of the tube has no thickness – this can be done using lofting tools or sweeping operations in FreeCAD, for example.

FIXING THE ISSUE

Having analysed our cubes example and found the non-manifold edge issue, we may well be tempted to click the Repair button next to the offending issue in the dialog box. If you try this, FreeCAD will automatically apply a repair, but be prepared for this to fail as it will only work occasionally on simple problems, such as a small gap on a surface that needs closing. It's always worth a try, however, and you can always undo any odd changes the repair functions apply. For the sake of completeness, let's manually create a solution. If we move back to our part object and delete the fusion object so that we have our two separate cubes, move the upper cube back over and into the lower cube creating a couple of millimetres where they overlap. Then create a new mesh and analyse it – you should see that you have cured the manifold issue.

Finally, while we've looked at some useful stuff on the Mesh workbench, we have only really scratched the surface of what's available. As ever, there is lots more information, and tutorials, to be found online on both the FreeCAD community forums and in the FreeCAD wiki. ◻
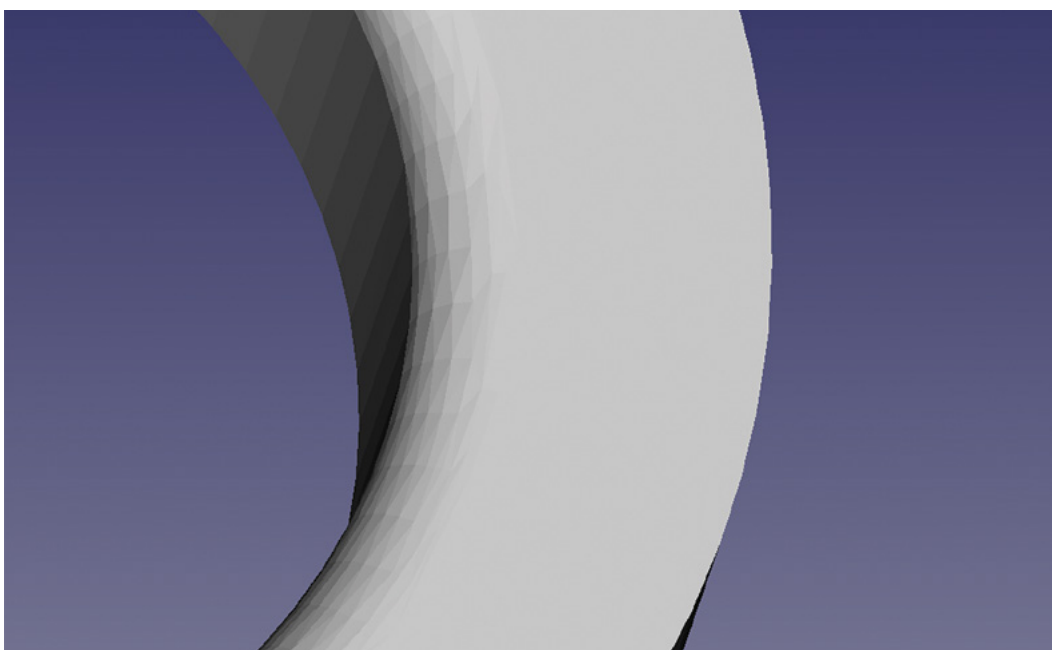
Figure 10 ◈
Using the 'Decimate a mesh' tool to reduce the resolution of a model can make a complex model easier to work with

CODE
THE
CLASSICS
VOLUME 1

Brimble

Crookes

Gillett

Malone

Tracey

Upton²

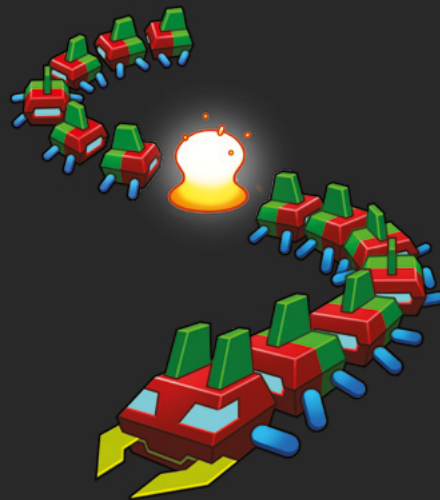# Dealing with a messy workshop

Keeping your spaces clear and well organised is just as important as choosing the right tools and consumables for your next project

Dr Andrew Lewis

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

Space is one of those things that makers never seem to have enough of. If you have a table to work on, you'll wish you had another table to keep your tools and notes in order. If you have a whole workshop on multiple levels, you'll probably wish you had an extra bit of room tacked on at the side for storage. 'If I just had a bit more space', the mantra goes, 'then it wouldn't be such a mess in here'. This isn't true, of course. No amount of extra space is going to make you more productive and tidy unless you manage it properly.

THE FOUR HORSEMEN OF THE SHOP APOCALYPSE

There are four main consumers of space in the workshop: tools, consumables, inattentiveness, and stalled jobs. The easiest of these to address is

inattentiveness. Neglecting small acts can lead to big problems. Not putting a tool back in its proper place, letting rubbish pile up in a corner, and putting things aside into piles to 'sort out later' are prime examples of small acts that can turn a workshop into an unusable space. If you lack the strength or willpower to tidy up at the end of each day (which is strongly recommended as a practice, as even 15 minutes of tidying can make the world of difference), then at least allocate a regular period of time in your calendar exclusively for tidying. Treat this time like an important meeting you must attend. If you can't make it, you have to reschedule it as soon as possible. The good news is that tidying up is an ideal task to break up into manageable chunks. You don't have to clean everything in one go, and a few weeks of regular effort will make a huge difference to your space.

QUICK TIP

You might find you won't lose out much by buying and selling tools second hand to free up space. You might even make a profit.

RISE OF THE MACHINES

Machine tools can take over a workshop very quickly, especially if you subscribe to some of the American YouTube makers who all seem to have a workshop the size of a barn (or several barns) and sponsorship deals to match. If you have a workshop with limited space, then you need to be brutal when it comes to deciding what tools you keep in your workshop. Ask yourself what you will lose if you get rid of the machine. Does the machine do something unique that you can't do another way? Could you sell the machine now and buy another one if you need it in the future? As a general rule, if you haven't used a large machine tool for twelve months, then you should consider getting rid of it.

As an example, imagine that you have a small workshop with a tabletop lathe and milling machine. If you find that you are using the milling machine very infrequently, you might be able to sell it and add a Z axis to the lathe instead. You'd lose the ability to mill larger pieces of metal, but much of the other functionality would be preserved by the modified lathe. That might be enough for the type of making that you do.

If you think that selling and buying large machines to suit different projects seems extravagant, consider the fact that your entire workshop is actually a tool for storing and facilitating your practice of making things. Once you start to think of the space as a tool, then you see how the value of any machine tool can be weighed against the value of the space itself. It's just a matter of deciding where the balance lies – is the functionality of the space worth more than the cost of the tool? You can follow the same pattern of thinking for smaller tools, although it's easier to be forgiving if the space they occupy is less imposing. In some cases, modern hand-held tools like portable bandsaws and belt sanders can mostly replace the functionality of larger floor-mounted machines. →

TUTORIAL

Consumables and material storage is always a difficult thing to manage. Some consumables are critical to a process of making, and not having that consumable at hand will prevent you from continuing your project. The amount and type of consumables that you keep in stock at any time will be decided by the sort of projects that you do, but don't fall into the trap of thinking that you have to bulk-order and store everything. Having reliable suppliers for your consumables gives you the option to buy items to arrive just in time for your project. It also gives you no excuse not to reorder stock parts as you use them up. If you see that you're starting to run out of a consumable, order more of them straight away. Don't leave it until you've completely run out and need to wait for a delivery before you carry on.

Salvaged materials are the most difficult thing to deal with in a small workshop. As a general rule, don't pick up salvage unless you have a specific plan for it. It's tempting to grab every free or cheap piece of material that looks like it might be useful, and store it or strip it down for parts. If you do really want to strip something down for parts, do it immediately, and then dispose of the bits that you don't want. Salvage feels like the right thing to do, but it's only eco-friendly if you actually reuse the parts, otherwise you're just storing recyclable materials. It can also take a lot more time to reuse a piece of scrap than it does to start with a fresh piece of material, so be aware that you'll probably have to allow extra time when you're scheduling jobs to clean off salvage and make it presentable. With offcuts of rigid materials, it makes sense to cut them to a standard size before you store them. There's some wastage, but regular pieces are much more convenient to work with than jagged offcuts.

MAXIMISE YOUR SPACE

When you store things, you need to decide where they are going to go, and keep some sort of record of what's in a storage space so that you can find things when you need them. Labelling boxes and trolleys is an obvious strategy that not only informs

CONTAINERS

Sometimes you will need to clear your bench at short notice to work on another project. Maybe an urgent repair has come into your workshop, or a deadline has been pulled forward. Having some large containers or trays that you can use to shift the contents of your desk into is a useful tip. Meat roasting tins are excellent for heavy items, while lighter items can go into plastic litter trays. Clear the desk into the tray, label and date the side, and move it out of the way to work on the emergency. When you're done, you can switch back quickly and all of the same parts will be there waiting for you. This also works if a project stalls because of a parts shortage, so you can switch quickly to the next project while you are waiting for the parts to be delivered.

you of where an item should be when you need it, it also informs you where it is supposed to go when you've finished using it. Labelling doesn't need to be exhaustive. It's not a complete inventory of every component in storage – it's a general guide to the sort of things stored there.

Even more generally, divide your item storage into three categories, and place items accordingly: immediate, shallow, and deep. Immediate storage is the toolbox that lives on your workbench, things that you use on a daily basis, and things that you want within arm's reach. Shallow storage is usually cupboards or drawers, where you'll keep things like rolls of LED lights, bottle cutters, or a sewing machine that you use several times a year, but not every day. Deep storage is for the things you need to keep, but won't be using for several months. This is the workshop equivalent of the warehouse that Indiana Jones put the Ark of the Covenant into. Things that go here might be critical spare parts for a machine tool that you probably won't need to use, but can't afford not to have. This storage can be less accessible than your regular storage, and might even be off-site in a shed or an attic. You don't need to get in here often, so it's not too important if it takes a while to get to it.

You can make more of your available space by making it reconfigurable. Rolling trolleys and cabinets on wheels can be repositioned to maximise space, or bring certain tools and consumables closer to you when you need them. When they're not in use, you can slide them out of the way. If you can keep your cabinets all of the same size (in particular the same height), then they'll be even more flexible, and could be used to set up a temporary workshop in a different location. □

Above ◈
Labelling the edge of a trolley makes it easy to see what's inside, even when it's underneath a work surface. Keep the labels general, or you'll end up trapped in the practice of adding new labels every time you buy fresh consumables

Left ◩
Doors are not dead space. You can store smaller items on the outside of cupboard doors using galley-style shelving. Nail varnish shelves and spice racks are particularly useful for storing paints and small tubes of glue. They're cheap, lightweight, and easy to fit

QUICK TIP

Nothing fills up the materials storage space faster than good intentions.

Make a Pico musical cheese box

Create syncopated rhythms and computer-generated music using a Pico

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**.

Make a portable MIDI controller that you can use to create complex rhythms and musical sequences and use as a tiny keyboard. In a cheese box. The article 'Make Music with Pico', in HackSpace magazine issue 46, showed how easy it is to use a Raspberry Pi Pico device to send MIDI notes to a synthesizer. We are going to build on that to create a MIDI instrument, the 'Pico musical cheese box', as shown in **Figure 1**. The musical cheese box uses a ring of twelve pixels to display notes, and twelve buttons for user input. You can use the cheese box with a synthesizer program that connects to MIDI, and there is also a specially written Pure Data patch for it.

INSIDE THE BOX

Figure 2 shows the contents of the box. There are quite a lot of wires, but the task of assembly wasn't too onerous. The box contains a twelve NeoPixel ring and twelve buttons which are connected to a Raspberry Pi Pico. Each button has its own dedicated input pin on the Pico.

Figure 3 shows the circuit for the box. The button signals are pulled low when a button is pressed. To simplify the wiring, the ground signal is connected in a ring around the switches. The pixel ring display only needs a single data connection, along with power and ground connections.

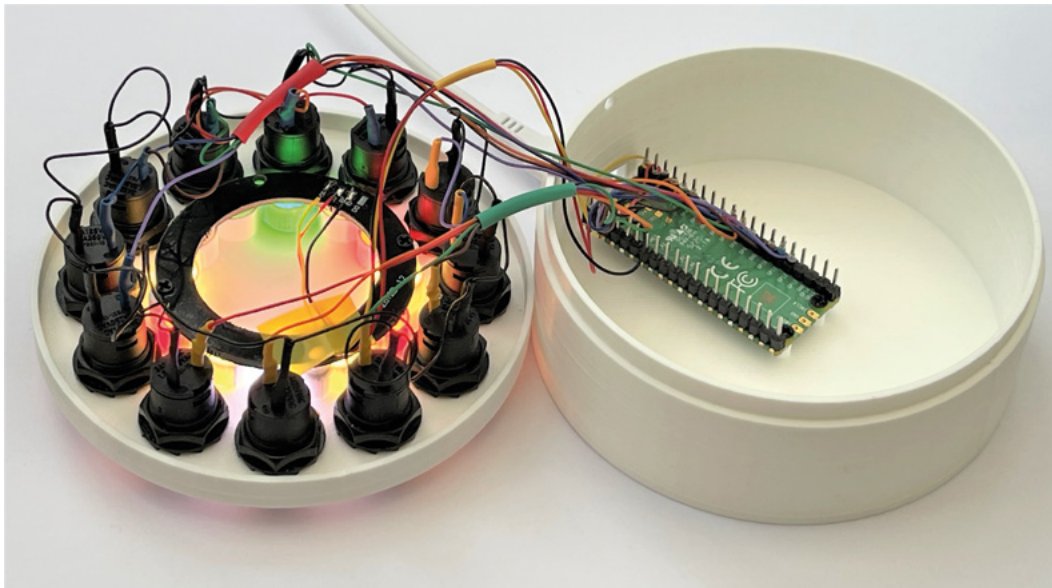The prototype used wire wrap wire for the connections. One end was wrapped round pins on

YOU'LL NEED

- **A Raspberry Pi Pico**

- **A 12-pixel WS2812 pixel ring** (search for 'WS2812 ring 12')

- **12 × coloured push-buttons** Ideally eight white buttons for the notes and a red, green, blue, and yellow button for the controls. The buttons should be non-locking, push to make

- **A micro USB cable to link the Pico to the host**

- **Connecting wire** The author used coloured wire wrap (search for '30 AWG wire wrap') which needed a wire wrap tool (search for 'wire wrap tool'). He also used a fair amount of heat-shrink tubing

- **A Raspberry Pi or machine running Windows, macOS, or Linux running Pure Data to make the sounds**

- **A box** There is a 3D-printable design, or you can just drill some holes for the lights and buttons in a cheese box and stick the pixel ring and Pico inside it

- **Screws** You'll need some screws size M2 4 mm in length to fix the pixel ring and Pico to the case (search for 'laptop screws')
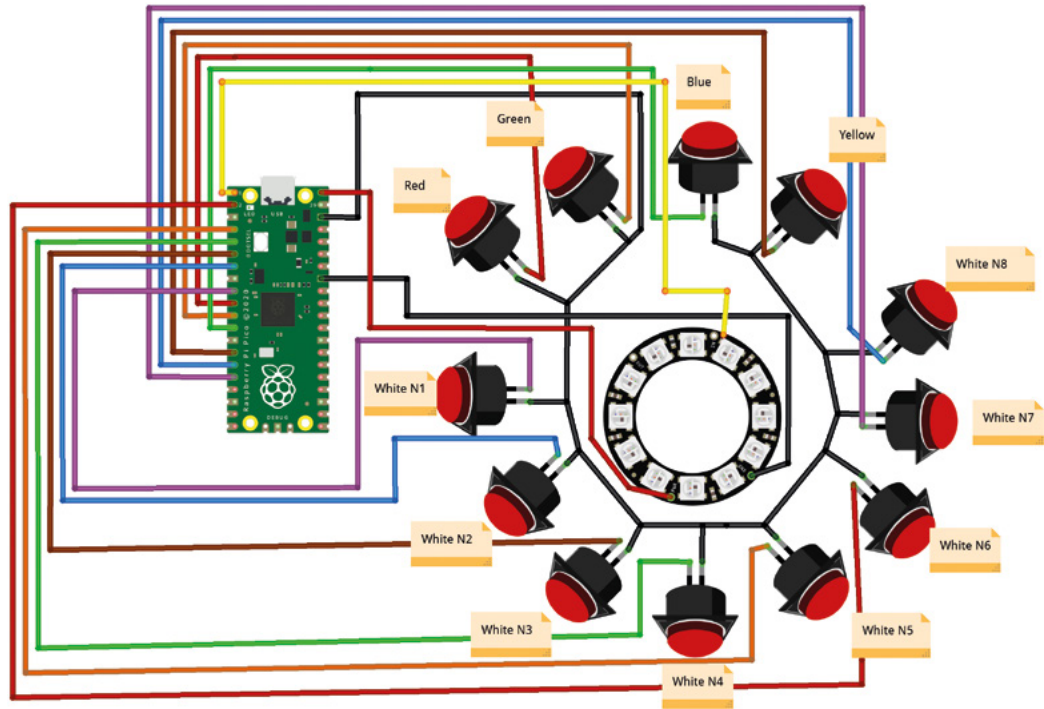
the Pico device, the other end was hooked inside the hole in the pin on the switch, and then soldered in place as shown in **Figure 4**. All the switch connections were encased in heat-shrink tubing.

HARDWARE MAPPING
The buttons are used by the CircuitPython program running inside the Pico that controls the device. The program contains `Note` and `Select` classes to represent the two types of button on the device. Note and Select objects are created with hardware pins and pixel number settings when the box starts.

```
notes = (
    Note(pin=board.GP6,pixel=11,num=0),
    Note(pin=board.GP5,pixel=10,num=1),
    Note(pin=board.GP4,pixel=9,num=2),
    Note(pin=board.GP3,pixel=8,num=3),
    Note(pin=board.GP2,pixel=7,num=4),
    Note(pin=board.GP1,pixel=6,num=5),
    Note(pin=board.GP12,pixel=5,num=6),
    Note(pin=board.GP11,pixel=4,num=7)
    )

selects = (
    Select(pin=board.GP7,pixel=0,col=Col.RED),
    Select(pin=board.GP8,pixel=1,col=Col.GREEN),
    Select(pin=board.GP9,pixel=2,col=Col.BLUE),
    Select(pin=board.GP10,pixel=3,col=Col.YELLOW)
        )
```

The code above is the Python that creates the collections of Note and Select objects that provide the user interface for the box. The note values are assigned a pin, a pixel number, and a note number

> **You can use the cheese box with a synthesizer program that connects to MIDI**

which will be used with they are played. The select values are assigned a pin, a pixel number, and the colour of the button which will be used to select the appropriate voice. This colour is then mapped to an instance of the appropriate **Voice** class. The mappings opposite were used for the first cheese box. You can change them to match the hardware that you build. Look for the statements in the program (they are near the bottom), and change the GP and pixel values to match your hardware.

BOX SOFTWARE
You can download the software for the box from the GitHub site for this project: **hsmag.cc/CheeseBox**. The software is written in CircuitPython, which can be installed on your Raspberry Pi Pico using the image from here: **circuitpython.org/downloads**. The software uses the Adafruit adafruit_bus_device and adafruit_midi libraries, which need to be copied into the **lib** folder on the CircuitPython drive of your Pico. You can find these here: **circuitpython.org/libraries**.

You can use Thonny (**thonny.org**) or Mu (**hsmag.cc/MuEditor**) on your computer to open the program and download it into your Pico. →

Figure 3 ◈
This diagram took almost as long to draw as it took to build the hardware

Figure 4 ◈
Wire wrap cable is single core, which makes it very easy to work with. It is also very flexible, so it is less likely to strain the connections at each end

" Different elements in the application are
managed by software objects called classes **"**

A full description of how the program works would take more space than we have for this article, but we can take a look at how it is structured. **Figure 5** shows a simplified class diagram for the software. Different elements in the application are managed by software objects called classes. The `Button` classes (Note and Select) manage the information for the buttons. The `Voice` classes (Keyboard and Rhythm) manage the information for the voice tracks that are played. The `Setting` class holds setting information for a collection of setting values. The `Step` class is based on the `Setting` class and holds information for a step in a track. The `MidiBox` class contains lists of buttons and voices, and repeatedly updates them to keep the box running and provide the illusion of several things happening at the same time.

MAKING TRACKS

The software manages four performance tracks: one keyboard track and three rhythm tracks. A track is selected by pressing one of the coloured buttons. When a track is selected, the pixel next to its selection button is lit in the colour of the button. The red keyboard track lets you play notes. When this track is selected (indicated by a red pixel next to the red button), the box starts a MIDI note playing when
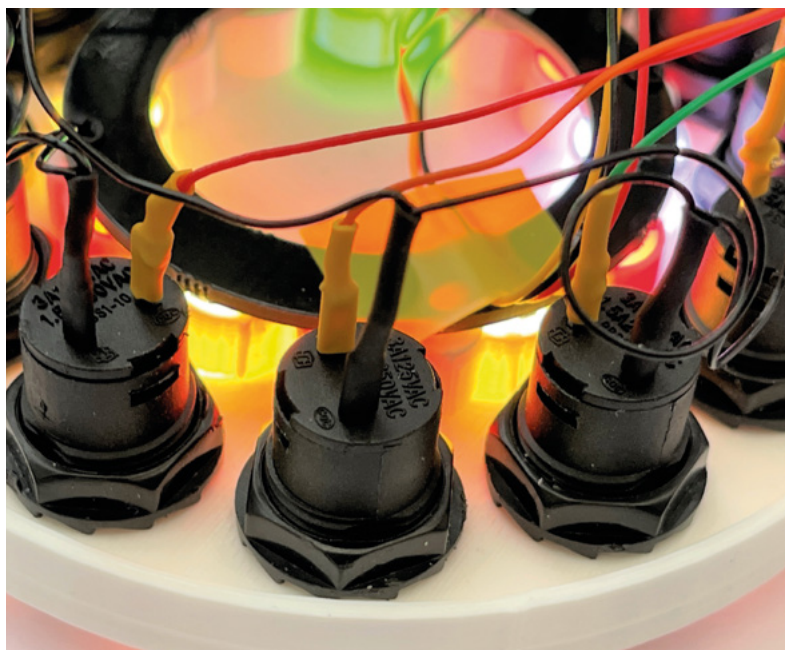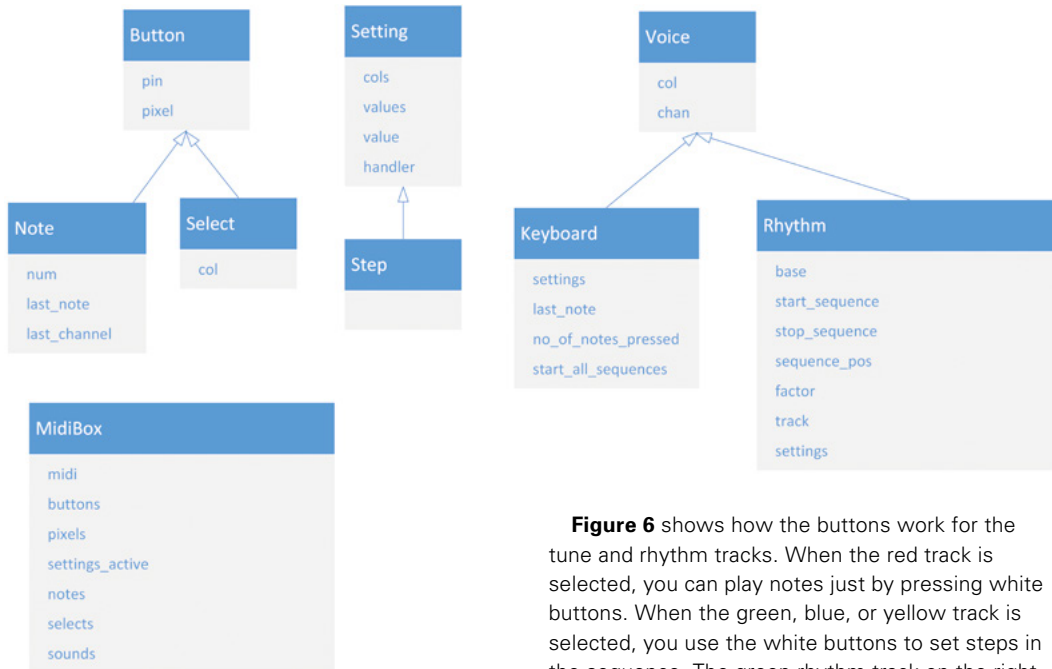
a white note button is pressed and stops the note when the button is released.

When a rhythm track is selected (by pressing a green, blue, or yellow button), the box will display the steps in the track and a moving white cursor that shows the position in the track. The user can select beats to be played at points in the track sequence by pressing the corresponding white button. All the sequencers run in parallel and work through a sequence of up to eight steps. Each step can produce a drum sound or a musical note. The speed of each sequencer can be changed, as can the length of sequence for that track. By making tracks of different lengths running at different speeds, you can create surprisingly complex rhythms as tracks go in and out of time with each other.

Figure 6 shows how the buttons work for the tune and rhythm tracks. When the red track is selected, you can play notes just by pressing white buttons. When the green, blue, or yellow track is selected, you use the white buttons to set steps in the sequence. The green rhythm track on the right of **Figure 6** starts with two beats from the kick drum (the red pixels) and ends with a ride cymbal (the yellow pixel).

The rhythm track functions in the same way for all three tracks. You can step through the options for each step by repeatedly pressing the white button to change the colour of the pixel for that step. A red step means 'kick drum'. When the sequencer reaches that step, it will send a MIDI note to play that sound.

Figure 7 shows the mapping of pixel colours to drum sounds. The number after each sound name is the MIDI note that will be sent to trigger that sound. These notes correspond to the General MIDI standard for musical instruments (**hsmag.cc/MIDI**) so the cheese box will work with most synthesizers. →

Tune Track
press white buttons
to play notes

Rhythm Track
press white buttons
to select beats
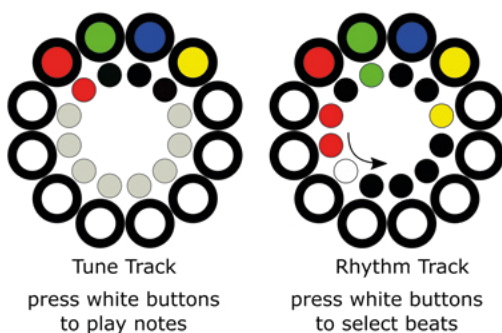
A NOTE ABOUT SCALES

The white buttons on the box don't play successive MIDI note values. Instead, they play the notes from a particular musical scale. This is so that the box sounds musical when you play it. Different scales use different arrangements of gaps between the successive notes. The most common scale is the 'major' scale, but there are others. Some have fantastic names like 'Phrygian' and 'Mixolydian', as you can see in the Scale setting in **Figure 8**. You can experiment with different scales to find the one that sounds best for a particular tune.

Drums

Off	Kick 36	Snare 40	Tomtom 48	Ride 59	Splash 55	Clap 39	Cowbell 56

Figure 7 ◈
Each time you press a step button, the selected beat for that step moves to the next colour. If it reaches the end, it wraps around to off (black)

Figure 8 ◈
The pixel colours in the centre of the settings show the default setting options

SETTING UP

You adjust the settings for the box by holding down the button for the selected track, and then setting the colour of pixels that are mapped onto setting values. If you press the red button when the red track is selected, the tune settings are displayed.

Figure 8 shows the tune settings. There are seven different setting items. Each of them is assigned a pixel and the value of the setting selected by stepping through the colours for that setting. There are also seven settings for the rhythm track…

Figure 9 shows the rhythm settings for the green track. You can set the speed of the track in beats per minute. You can also set the pattern length and the 'speed factor'. The speed factor is used to slow down a rhythm track. A factor of 1 means normal speed. A factor of 2 means half speed, a factor of 3 a third speed, and so on. You can also change the mode of a rhythm track to play notes rather than drums. This lets the track play short musical sequences.

MAKING SOUNDS WITH PURE DATA

You can use the cheese box with any synthesizer, but you can also use it with a specially written Pure Data patch. Pure Data is a programming language for creating music applications. You can find out more about it in the article in issue 47 of HackSpace magazine. Pure Data programs are expressed as a network of connected objects. The connections between the objects can be either individual values or audio streams.

> **You can use the cheese box with any synthesizer, but you** can also use it with a specially written Pure Data patch

Figure 10 shows the Pure Data patch that generates sound output for the cheese box. It works as a 'switching station' which acts on incoming MIDI messages. The 'notein' object at the top left of the patch accepts incoming MIDI messages and sends out the note number, volume, and channel (these are the three outputs along the bottom of the notein object). The patch uses spigot objects to control the flow of the data around the patch. Values will pass through the spigot if the right-hand input to the spigot is true.

Any messages sent to MIDI channel number 1 are passed into the polyphonic object (poly) and

Tune Settings

Rhythm

Off	On

Volume

30	80	120

Scale

Major	Minor	Dorian	Phrygian
Lydian	Mixolydian	Lorian	

Octave

C2	C3	C4	C5	C6

Midi Channel

1	2	3	4
5	6	7	8

Key sharp/flat

Natural	Sharp	Flat

Key

C	D	E	E
G	A	B	

Pure Data patch for the PICO MIDI Cheesebox

Notes to channel 1 are sent to a note synthesizer which is 6 note polyphonic.

Notes to other channels are used for rhythms. The note value is used to select the sound generator.

All the sound generators throw their output here to be converted into an audio signal by the dac

CHEESE BOX

If you don't fancy buying some cheese just to get a box for your cheese box, you can download and 3D-print a custom case. The designs are available at the GitHub site for this project: **hsmag.cc/CheeseBox**

then on to a set of six note synthesizers. Messages sent to any other MIDI channel are stripped of their note off elements (these have no meaning for drums) using the stripnote object and then the note number is used to select the sound generator to be used. You will notice that the note numbers in the select object are the same as those in **Figure 7**. You can find this patch, along with installation instructions, at the GitHub site for this project: **hsmag.cc/CheeseBox**. At the moment, the note generators only output simple sine waves, but in the next issue of HackSpace magazine, we will be discovering how to use Pure Data to produce more interesting sounds and create a crackers controller to go with the cheese.
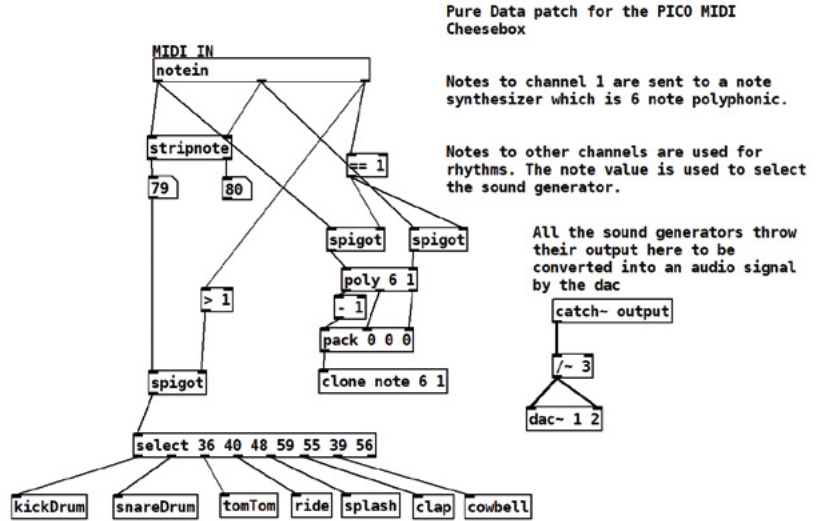
EUCLIDEAN FUN

By changing the speeds of the tracks and playing notes and drums at different points, you can generate interesting sounds that change over time. These are called Euclidean rhythms. You can find out more here: **hsmag.cc/Euclidean**. Don't be afraid to experiment with the sounds. If you change the rhythm mode of a track to 'Euclidian', the cheese box calculates a note frequency by adding together note values from other Euclidian tracks. By careful choice of track speeds, you can create extended sequences of changing notes. Once you have something interesting, try changing the scale and listen to what happens. You can pick different octaves for the note playback so you can generate bass sounds too. Remember that there is no such thing as bad music, just stuff that only you like.

Figure 10
Pure Data programs can run on a huge range of computers, from Raspberry Pi Zero upwards

Figure 9
By default, each rhythm track sends on a different MIDI channel so that the rhythms can be used to trigger different voices. However, you can select the MIDI channel for each track so that you can make the rhythm tracks play on the same channel as the keyboard

Rhythm Settings

Rhythm

Off	On

Volume

30	80	120

Speed (bpm)

60	80	100	110
120	140	160	180

Pattern Length

2	3	4	5
6	7	8	

Rhythm Mode

Drums	Notes	Euclid

Midi Channel

1	2	3	4
5	6	7	8

Speed factor

1	2	3	4
5	6	7	8

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 108

BATTERY SPOT WELDER

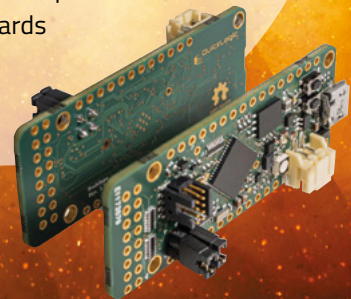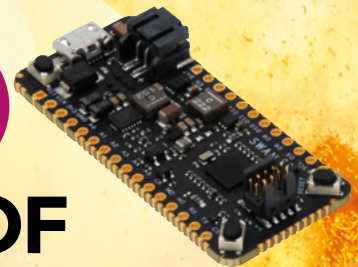**Build your own battery packs**

PG 112

LED MATRIX

Add 117 RGB LEDs to your next project
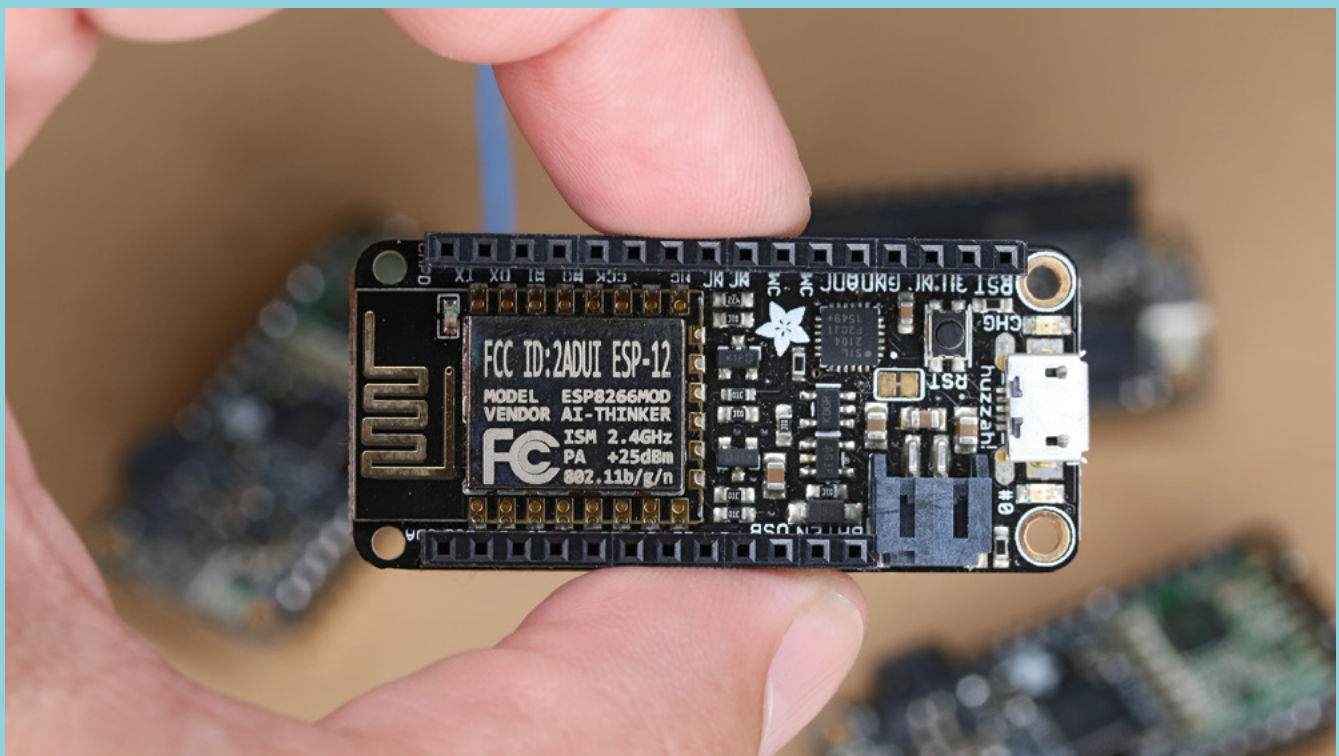
PG 102

BEST OF BREED

Feather development boards

ONLY THE BEST

The Feather board ecosystem continues to grow

What's been happening with Feather?

By **Marc de Vinck**   🐦 @devinck

HackSpace magazine is celebrating its fourth year in print, and I'm happy to have been a little part of that history, having written over 40 Best of Breed roundups, and even a build here and there. I decided to look back at some of my early reviews, and I thought it was about time I revisited my favourite microcontroller ecosystem, the Adafruit Feather. It's been years since we last did a dedicated roundup, and there have been so many new products launched that it warrants a second look. In fact, I could do several Best of Breed articles and I'd still only cover a small percentage of the new products!

And although it has been a while since there was a dedicated roundup, I have covered a few Feather boards in various reviews over the years, as there are so many varieties of boards and accessories out there that it's hard not to include them when writing about microcontrollers and sensors. The most recent board I reviewed, just a few months ago, was the Adafruit Feather RP2040 which features the all-new microcontroller from Raspberry Pi, the RP2040. I've played around with the board since then, and I still stand at giving it a 10 rating, as it's a great board at a great price.
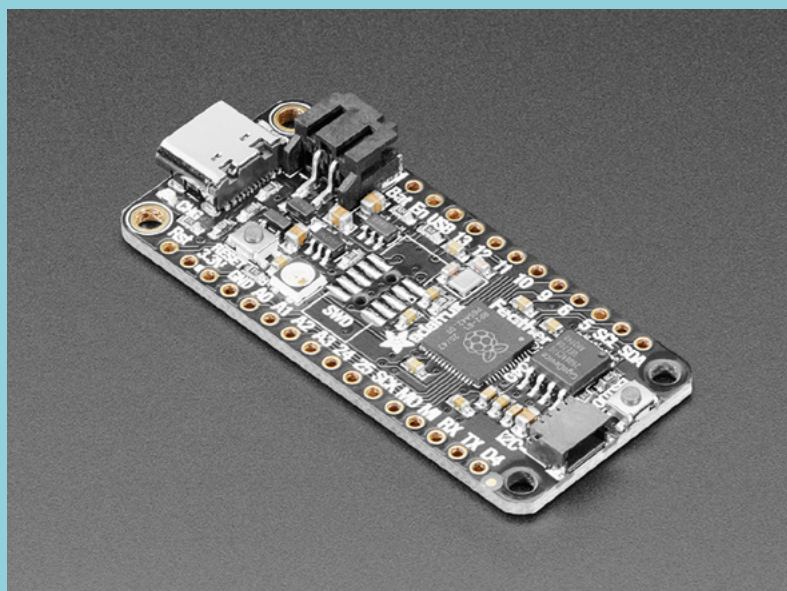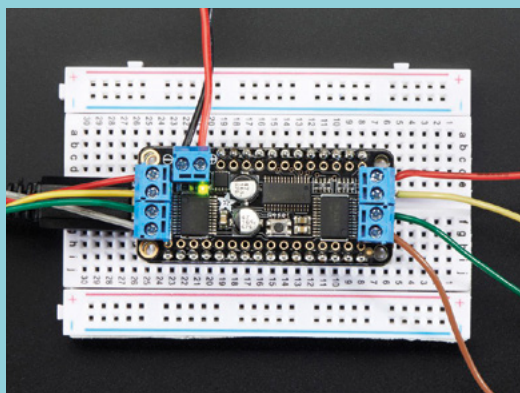
"Wouldn't it be great if, instead of collecting shields, HATs, PMODs, Clicks, Booster Packs, Props… you could have a cross-compatible platform? Something that doesn't force you into one chipset?"
– Limor Fried (Lady Ada)

But what else is happening? And who better to ask than Limor Fried, the creator of the Feather platform and founder of Adafruit. I've known Limor for more years than I can remember. I bought one of her first kits when she launched Adafruit. We even took some early road trips back in the day, looking at equipment and visiting some tech companies. Since then, we have kept in touch, and crossed paths every so often, so I decided to reach out to her and ask about the Feather ecosystem and what she thought was interesting.

Limor sent me some interesting links, and in true inclusive Adafruit fashion, it included a variety of Feather boards from a wide range of companies – indie makers, startups, and larger established brands. Limor continues to be the catalyst for growing and celebrating the community. It's just another reason why so many people love Adafruit's products, myself included!

Thanks Lady Ada! →

OLD FEATHERS

You can learn more about the history of the Feather platform by visiting the Adafruit Learning System. And don't forget to check out all the Feather-related tutorials too!

The Feather board ecosystem continues to grow

—

The Swan MCU vs QuickFeather

BLUES WIRELESS ◈ $25 | blues.io   **QUICKLOGIC** ◈ $49 | quicklogic.com

The Swan is a low-cost, embeddable STM32L4-based microcontroller, specifically designed for battery-powered IoT solutions. Its Feather form factor enables it to be used with a variety of add-on FeatherWings and sensors. You can power the board via USB, a battery, or line power. It also features a software-switchable 2 A regulator for powering external sensors. The Swan only draws 8µA when in low-power mode, making it an ideal development board for remote and long-term sensing.

You can pick up the Swan as a standalone board, or as part of Blues' starter kit that features the Swan board and the firm's Notecard and Notecarrier boards that add additional features, and make connecting to all 55 pins of the board simple and fast. Head over to **blues.io** to learn more.
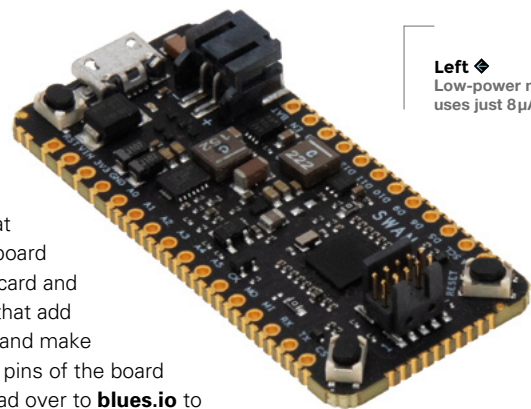
The QuickFeather Development Kit from QuickLogic is a Feather form factor system that the manufacturer says is ideal for enabling the next generation of low-power machine learning-capable IoT devices. What makes it so special? Not only is it open-source, as it's built around the open-source SymbiFlow FPGA Tool, but it also features a Cortex-M4F MCU and integrated FPGA. You'll also have access to the on-board GigaDevice 16-Mb of flash memory, mCube accelerometer, Infineon pressure sensor, and Infineon digital microphone.

Being that this is cutting-edge technology, you'll really need to head over to the product page (**hsmag.cc/QuickFeather**) to learn more about how you get started with powerful tinyML applications, sensors, and programming an FPGA. They have a lot of videos and PDF instructions that will certainly help.
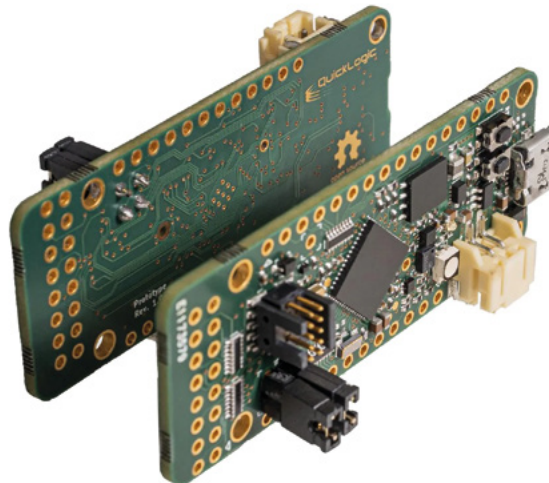
Below
There's an accelerometer and microphone on board

VERDICT

The Swan MCU

Low power is great for IoT.

9

QuickFeather

Incredible power in such a small form factor.

9

FeatherS2 – ESP32-S2 Feather Development Board

ADAFRUIT ◈ **$22.00** | adafruit.com

T**he powerful FeatherS2, designed by Unexpected Maker, features an ESP32-S2 development board with 16MB of flash memory and 8MB of QSPI-based external PSRAM.** Yes, those are in MB, not kB. You've got a lot of memory to play around with, along with a host of other features.

This FeatherS2 has two 3.3 V regulators, allowing you to not only power the board itself, but external sensors as well. It also features the STEMMA QT connector, making adding those sensors as simple as plug and play. It's a great addition to the ESP32 and Feather family! →
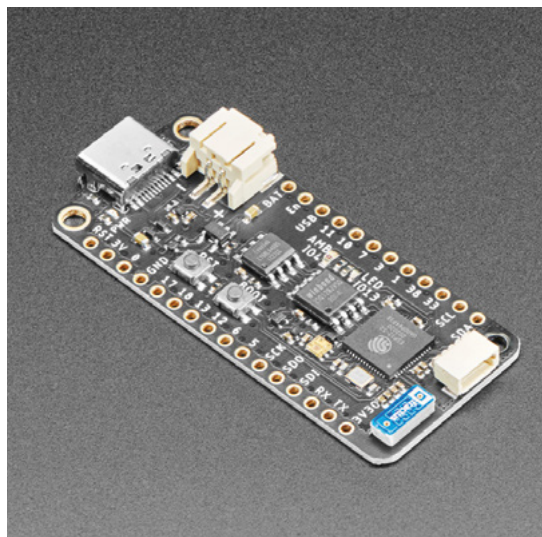
Left ◈
8MB of RAM for data-intensive projects

VERDICT

FeatherS2 – ESP32-S2 Feather Development Board

So much memory and the familiar ESP32.

10/10

OPENWEATHER DESK DISPLAY

ADAFRUIT | hsmag.cc/DeskDisplay

A while back, I created a non-obtrusive, and simple, connected weather display for a HackSpace magazine build. And you guessed it, inside was a Feather HUZZAH. I just love how easy it is to get it up and running. And several years later, it's still running perfectly, notifying me of the weather in a quiet and passive way.

—
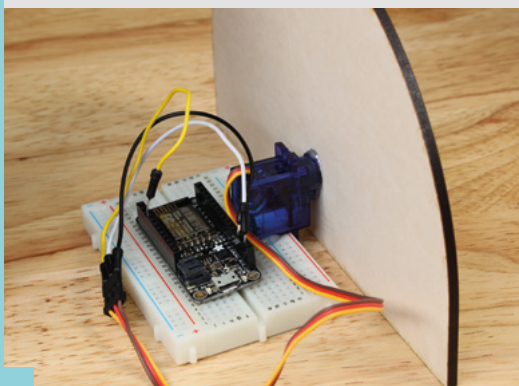
SparkFun Thing Plus – XBee3 Micro

SPARKFUN ◈ $52.50 | sparkfun.com

S parkFun has entered the chat with its Feather form factor boards and Wings that it dubs 'Thing Plus'. The company has a large and growing ecosystem, ranging from SAM-based boards to the new RP2040 and ESP32s. It also has shields, or Wings, to add extra functionality.

In this review, we'll be looking at the Thing Plus - XBee3 Micro. The XBee wireless system has been around for a long time, and SparkFun created a Feather-compatible board that features the new XBee 3 Micro Module. With its near plug-and-play 802.15.4 2.4GHz wireless connectivity, using the Zigbee 3.0 Protocol, it makes a great choice for so many home automation setups and wireless network projects. You'll also get 20 I/O pins to play with and a JST connector for battery power. And, best of all, you can program it with MicroPython! □
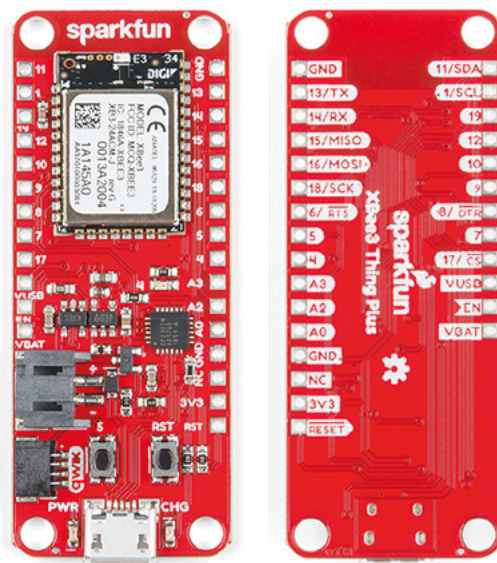
Left ◈
A microcontroller for wireless applications

VERDICT

SparkFun Thing Plus – XBee3 Micro

If you're using XBee already, this is the board for you.

8/10

CHEERLIGHTS

ADAFRUIT | hsmag.cc/CheerLights

Yes, another internet-connected device that I created a few years ago using the Adafruit Feather HUZZAH, my all-time favourite microcontroller board. Again, the board is so small, reliable, and easy to get up and running that I just can't think of using anything else. This build is part of the CheerLights project. Originally developed by Hans Scharler, CheerLights allows people to control the colour of lights all over the world and synchronise them to one colour set by a specific Twitter feed. Head over to the Adafruit Learning System if you want to learn how I built my own version of the CheerLights, or if you just want to change the colour of the light on my desk!
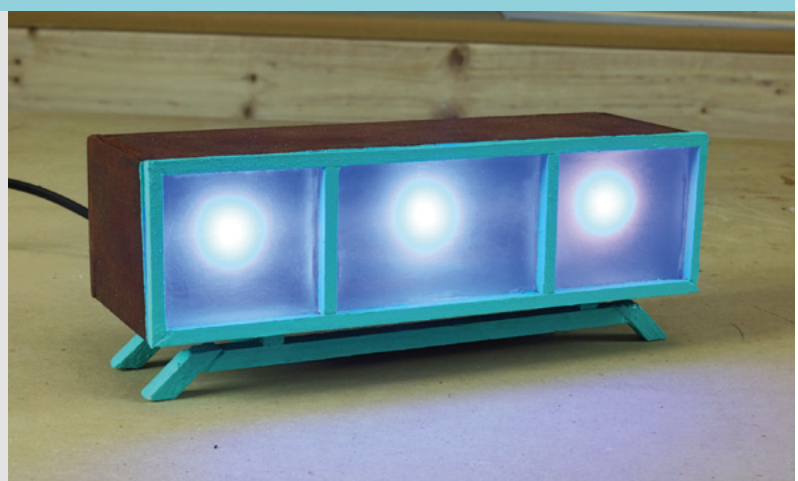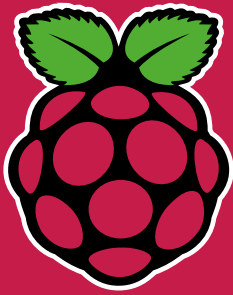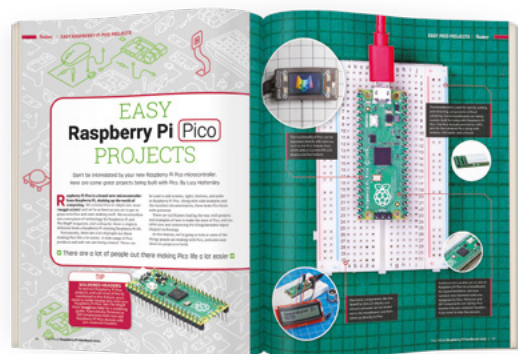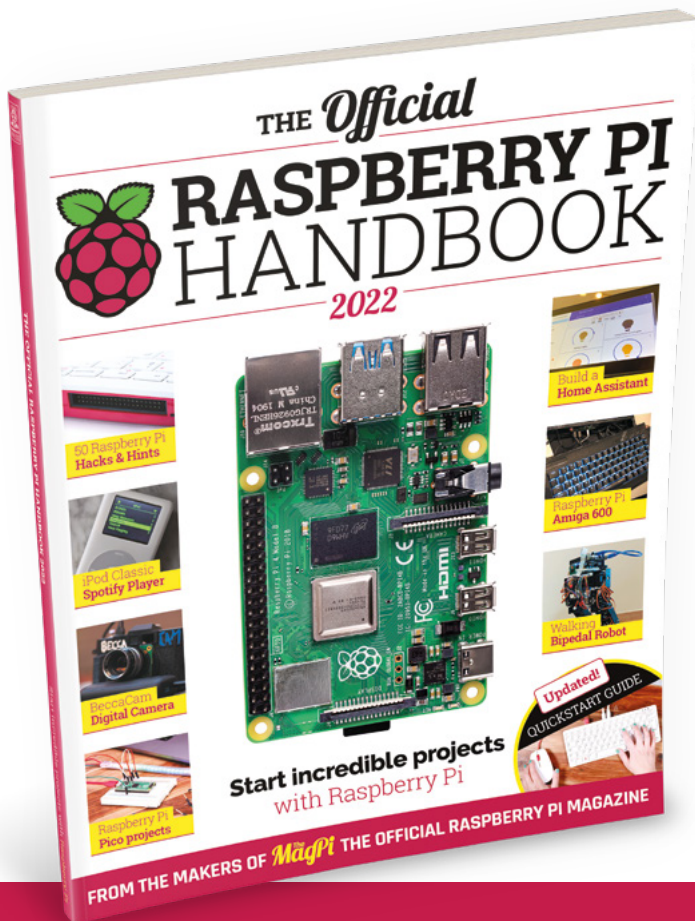
SUNKKO 709A battery spot welder

Practise your spot welding with this feature-laden piece of kit

SUNKKO ◈ **£169** | **hsmag.cc/sunkko**

By **Jo Hinchliffe**   🐦 **@concrete_d0g**

If you are interested in building your own Li-ion battery packs, perhaps for drone use, e-bikes, hacky racer or for a power wall project, you have probably considered a battery spot welder. Like many tools, there's a range of offerings available, from cheap PCB-based spot welders you run from a well-specified car battery, through to very automated high-end battery production systems.

Sat toward the budget end of the spectrum, you quickly find SUNKKO branded machines with a range of features. We opted to take a look at the SUNKKO 709A machine, which isn't the cheapest model at £169, but offers the range of features we wanted.

The 709A is primarily a spot welder designed to spot-weld thin nickel strips to steel battery terminals. The 709A also, however, has a soldering iron channel and attachment. Soldering directly to battery terminals
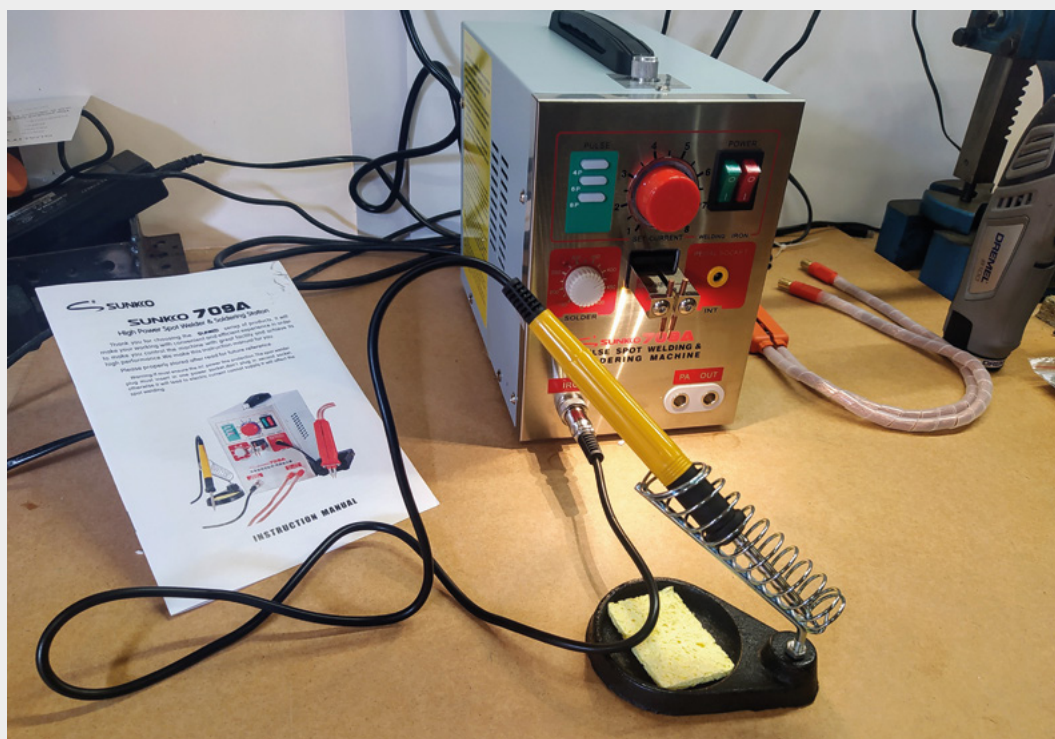
Right ◈
The machine set up with the soldering iron connected. It's possible to run the internal spot welding system whilst the soldering iron is connected

can be dangerous and the heat and the duration of heat from soldering directly to battery terminals can damage some batteries and is generally best avoided, and hey, that's why we have a spot welder, right? The inclusion of a soldering iron may not be an important feature for many who have a choice of irons already, but often you may want to solder wires onto the nickel tabs you have created to finish the pack, and an all-in-one system might save you some desk space.

The unit arrived very well packed, and inside the box, there is a pleasing range of accessories, including two bags of nickel-plated iron strips, a 18650 battery holding jig, spare welding probe tips, the soldering iron, soldering iron stand, and more. Also included is an external pen probe welding attachment and a foot switch.

The standard way that this machine works is using the on-board welding system. In the centre of the machine are two metal arms into which you insert the welding probes. On our machine, one of the holes to receive a probe had a tough burr inside and needed some work with a tiny file to allow us to insert the probe. Once set up, the idea is that you press the work against the probe tips and push upwards. At the top of travel of the two welding arms, click a switch to fire the welding system and create the weld; this system means that a good amount of pressure is applied to the weld and the level of pressure can be adjusted by turning a knob on the top of the unit. You can adjust the power and the pulse count for each weld on the
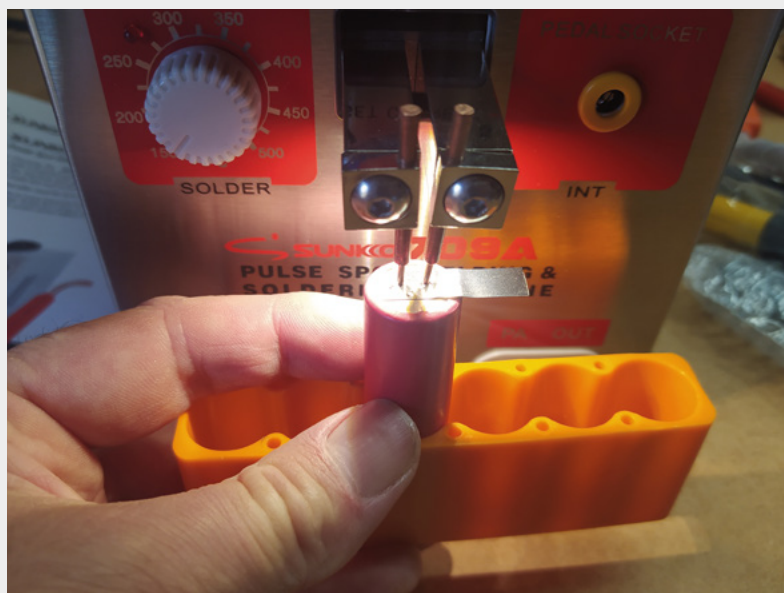
machine using simple radio buttons. You can then adjust the power with the current adjustment dial. We found that when using the thicker nickel-plated iron strips (0.15 mm), we would get a decent weld at four pulses with the power dial set up at 7. We imagine this would be different with each machine and also with different materials. Often people get better results with pure nickel strips rather than the plated nickel-iron strips.

One of the reasons we were interested in this particular model is that it offers the pen probe welding attachment to enable you to assemble larger pieces further away from the machine.

To use the pen probe, you have to also attach the foot switch to trigger the spot weld rather than any automated system. Whilst our pen probe does work, we have a pretty major flaw on our pen probe attachment. Our probe cables are covered in the spiral plastic wrap that is often sold or bundled with machines as a cable tidying system. Unwinding some of the spiral wrap revealed a situation that our copper cable underneath was not insulated in any way, confirmed with some continuity testing with the multimeter. Whilst the output through these cables would be quite low voltage, it would certainly be possible to spot-weld the cables together if they touched through the gaps in the spiral wrap. We haven't seen or read discussion of this problem, so it could be a 'Friday afternoon' machine that we acquired and others have better insulation but, as ours stands, we have marked it down heavily for this issue.

In use, we have been able to create strong welds with this machine, and it's worked reliably with repeated sessions of spot welding. The soldering iron system also works well and is certainly usable for the scale of work intended: soldering battery and balance cables to battery packs. ▫

VERDICT

A good machine marked down for the slight fault in the hole to attach the on-board probes, and the issue with insulation on the pen probe attachment.
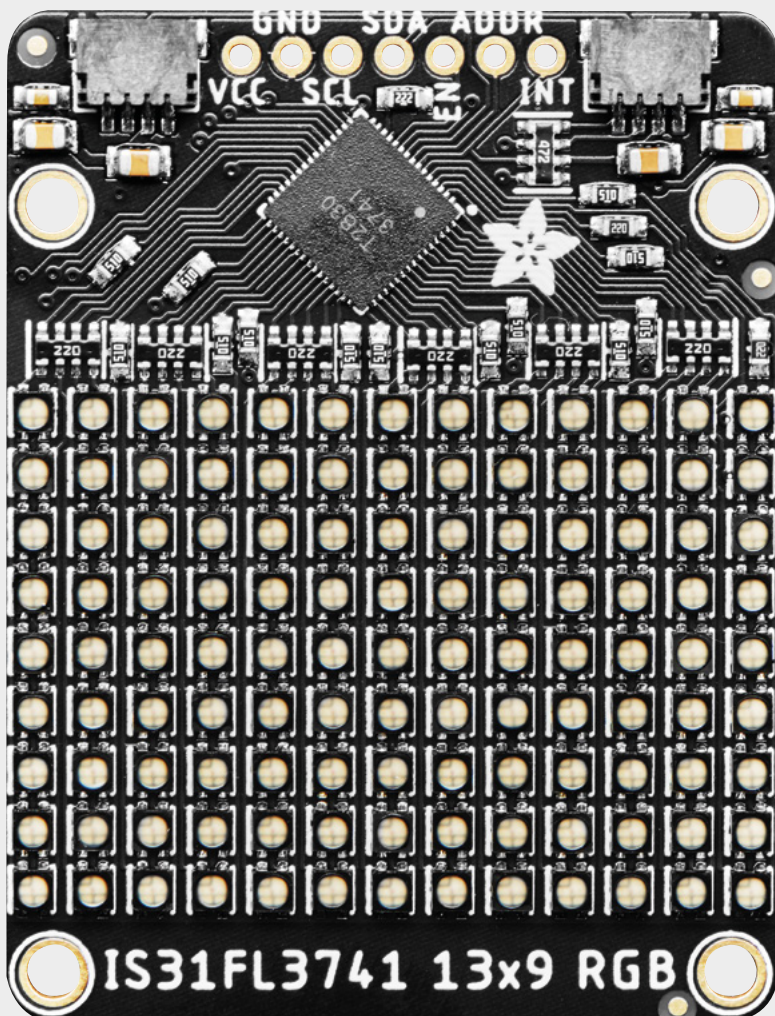
6/10

Adafruit IS31FL3741

Let there be light

ADAFRUIT ◆ $14.95 | adafruit.com/product/5201

By **Ben Everard**   ◉ **@ben_everard**

There are lots of ways of adding programmable LEDs to your project. NeoPixels are probably the classic, but they're not without their pitfalls – they are expensive if you need a lot of them, can be slow to update, and don't have great brightness control at the low end. HUB75 LEDs can be cheap and fast, but can also be awkward to wire up and control. APA102s are fast and look great, but they also get very expensive very quickly if you're using a lot of them.

This board uses an LED driver – the IS31FL3741 – which controls the 13×9 RGB LED matrix. This driver is controlled over I2C with a selectable address, so you can control four of these boards using a single two-wire I2C connection. You can connect them up either with the pin headers or the STEMMA QT/Qwiic connection, meaning that you can get started without any soldering – just plug it in and go. The LEDs are mounted flush with the side, so you can connect them horizontally to make a continuous display.

There are libraries to work with Python, CircuitPython, and Arduino, so you should be able to get up and running quickly with almost any microcontroller as a host.

You can run this board at either 3.3 V or 5 V – both power and logic. While most modern microcontrollers are 3.3V, you can power the board at 5 V and still use this logic level. Adding the extra 1.7 V of power makes a big difference to how the LEDs look. 3.3 V isn't much above the forward voltage of the green and blue LEDs, and they struggle to get up to full brightness at the lower voltage.

Left ◆
13 × 9 gives enough space for some complex effects
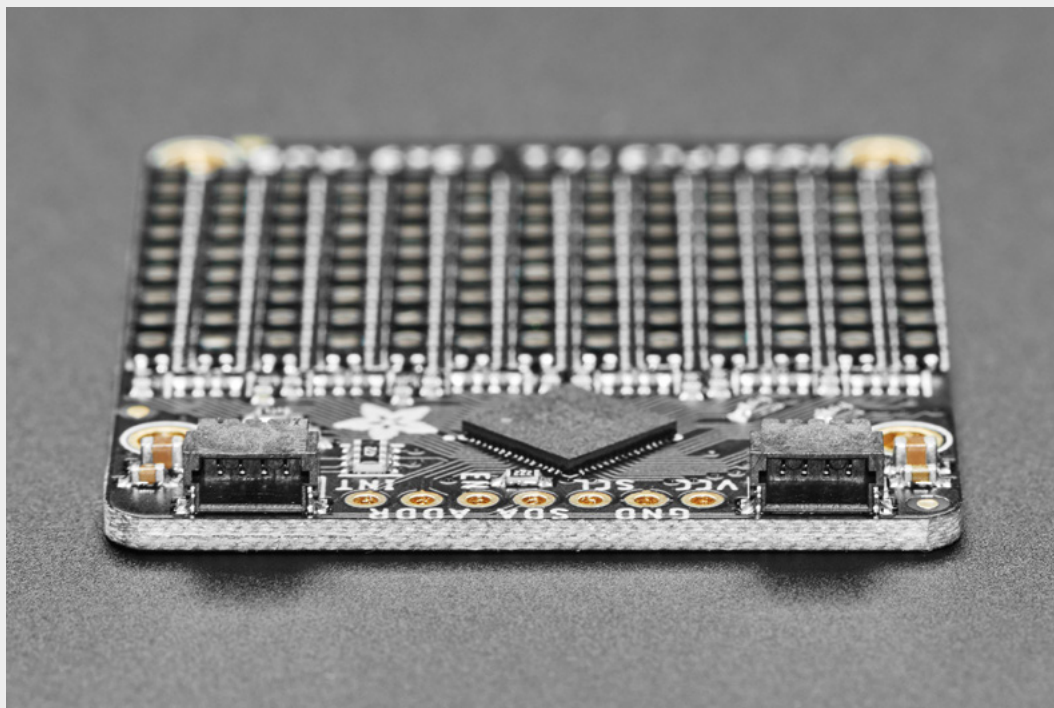
BRIGHTNESS CONTROL

One feature we particularly like on this board is the global current scaling setting. When you give an LED a value, the brightness is controlled through pulse-width modulation (PWM), which means that the LED is flickered on and off faster than the eye can see to make it appear bright (if it's mostly on) or dim (if it's mostly off). The addition of the current scaling means that you can shift the brightness range up or down. By turning the current scaling down, you can have the full 8-bit brightness range on the dim side, or by shifting it up, you can have the range on the brighter side. This means you can adapt your display to the environmental conditions and still have a wide range of brightnesses to play with.

One bugbear of ours with many LED matrices – particularly cheap ones – is the lack of mounting holes. Yes, you can hot-glue them down, but that's not a particularly good solution. There are four mounting holes on this, so you can attach it securely to your project.

Another thing to really like about this display is the price. At just under $15 for 117 LEDs, this is good value, especially if you're getting a few of them.

Of course, there are trade-offs for this approach. While I2C is a reasonably fast protocol, it's not super-fast. Each LED is controlled individually, so if you're trying to change large numbers of LEDs quickly (such as if you're doing full-display animations), then you might struggle.

The other significant trade-off is that while you can tile these displays horizontally, you can't tile them vertically – well, OK, you can turn them on their side. You can't tile them both horizontally and vertically at the same time. However, it's unlikely that you'd want to. If you're looking to create a huge display, you'll probably want to use hardware that you can update faster than you can with this matrix.

There is no such thing as the perfect LED solution for all maker projects. While the Adafruit IS31FL3741 does have some limitations, it's also a great fit for many projects. ◻

VERDICT

Cheap, easy-to-use lights that you can tile horizontally.

9 /10
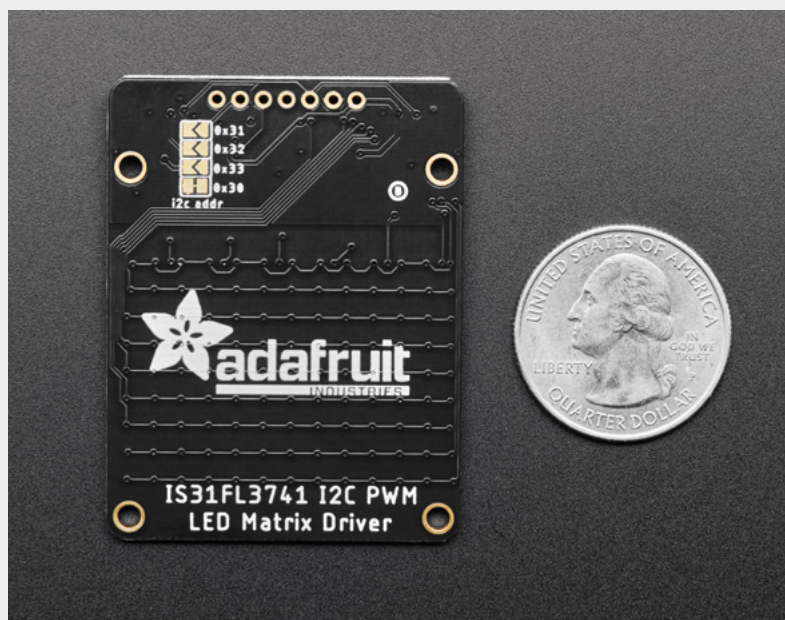
next month

issue

#49

ON SALE
18 NOVEMBER

INTERNET OF

LEGO ®

ALSO

→ **3D PRINTING**

→ **PICO**

→ **SOLDERING**
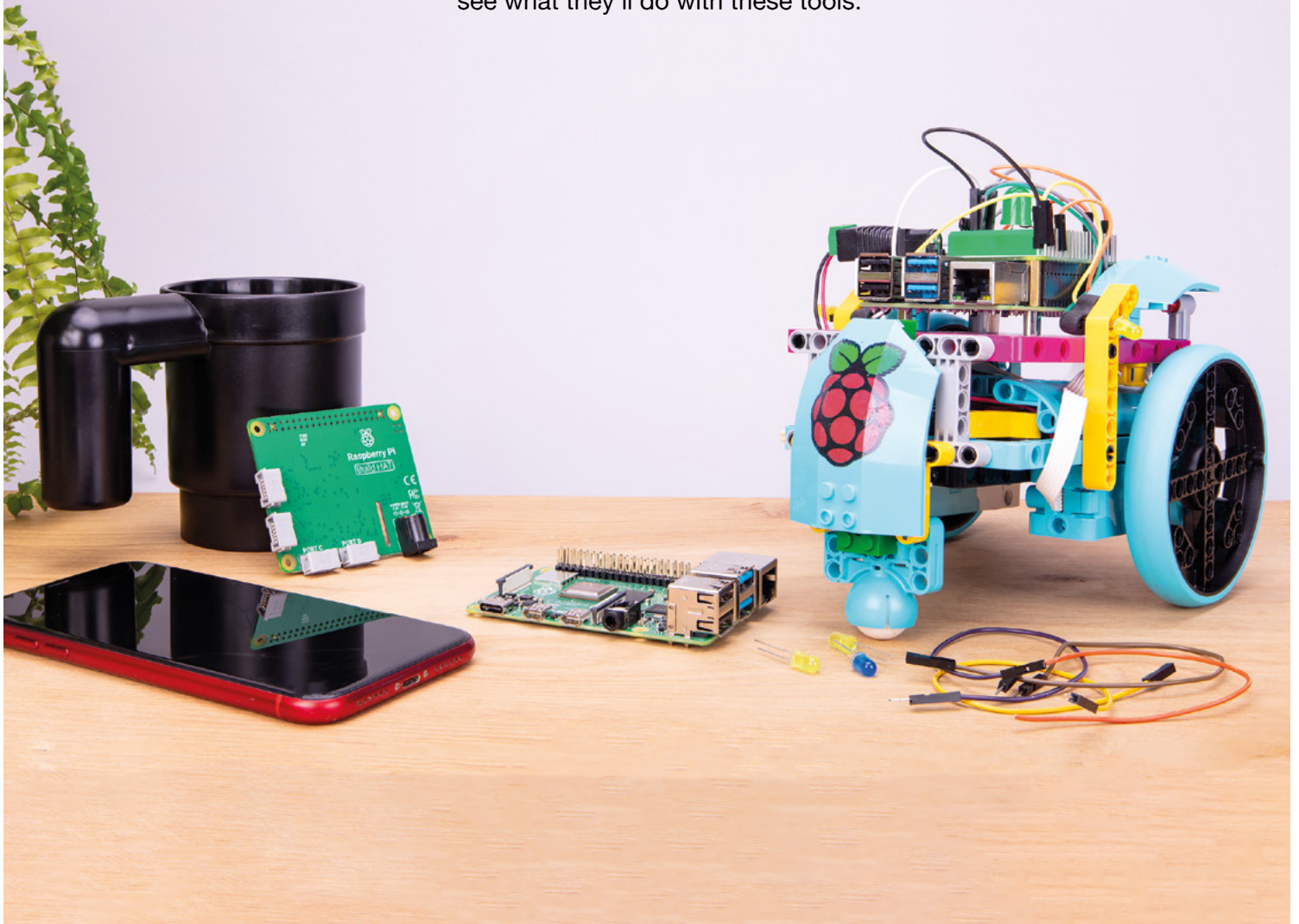
→ **ELECTRONICS**

→ **AND MUCH MORE**

DON'T MISS OUT

hsmag.cc/subscribe

Raspberry Pi Build HAT

Many makers got their start with Lego. There's something about the fact that it's easy to put two bricks together, yet this simple process can lead to huge and complex builds that fire the imagination. This author vividly remembers the Lego Technic kit he had as a child. There wasn't much technology to the Technic back then – a pair of motors and buttons – but that was enough to let the creations come to life. These days, technology is more accessible than ever. The Raspberry Pi Build HAT lets makers add a huge amount of processing power to their Lego builds quickly and easily. We're simultaneously jealous of the young makers of today and excited to see what they'll do with these tools.