MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | December 2022 | Issue #61

ULTIMATE MINCE PIES

LEARN TO CODE FOR MAKERS

Dec. 2022
Issue #61 £6

+ LEARN TO WEAVE

CONTROL YOUR BUILDS WITH RASPBERRY PI PICO

DESIGN YOUR FIRST PCB

SPANNERS MUSIC 3D PRINTING DRAGONS

Welcome to HackSpace magazine

Programming is one of the core maker skills and, like many skills, it doesn't need to be difficult. With wood working, for example, it takes years to master the skills of fine joinery, but you can go from beginner to making something useful in a few hours (even if there are a few gaps in the joints and you're relying on brackets to hold it together). It's the same with programming. You don't have to master everything to create something useful, and in a few hours you can join the bits together to make something that works.

Even if you don't consider yourself a geek, learning to code (even a little bit) **can open up huge new possibilities**

Even if you don't consider yourself a geek, learning to code (even a little bit) can open up huge new possibilities. It doesn't need to be fancy or complex to work and add a little interactivity to your builds.

This month, we're looking at using MicroPython and Raspberry Pi Pico to build simple circuits that you can use to add lights, buttons, and movement to… well, whatever you want to make.

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

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Olivia Mitchell

Photography
Brian O'Halloran

CONTRIBUTORS

Jo Hinchliffe, Marc de Vinck, Andrew Lewis, Rob Miles, Phil King, Nicola King, Rosie Hattersley

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION

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

SUBSCRIPTIONS

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

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

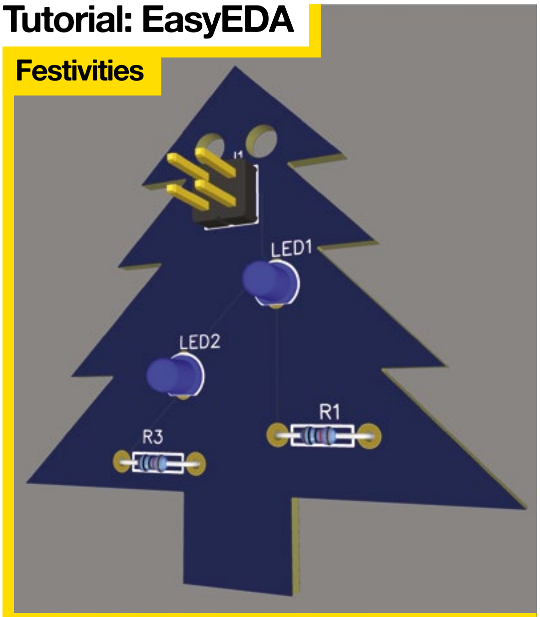
This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

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

Contents

Tutorial: EasyEDA

Festivities



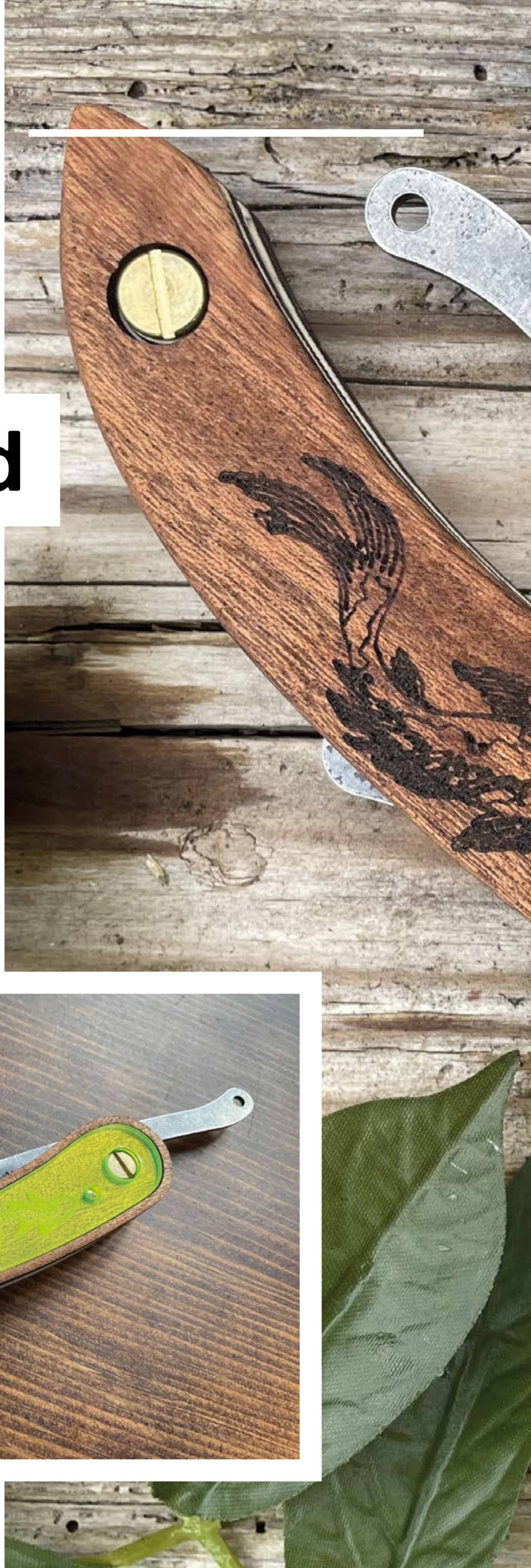
Interview

Pooch

63 FORGE

101 FIELD TEST

REGULAR



Laser-engraved knife-handle

By **Artful Dabbler**

hsmag.cc/PeasantKnife

We'll admit that we hadn't heard of the Svörd Peasant Knife, seen here complete with laser-engraved handle by Artful Dabbler. It's a model of a non-locking folding knife based on designs from 17th-century Bohemia and Bavaria, and in its modern incarnation, it's extremely popular in New Zealand.

Peasants everywhere like to decorate their belongings, so it seems only right to bring the tendency bang up-to-date with a spot of laser engraving. The ancestors would approve.

Right
These cheap, reliable knives are ideal for a spot of customisation


NEW ZEALAND
SVÖRD

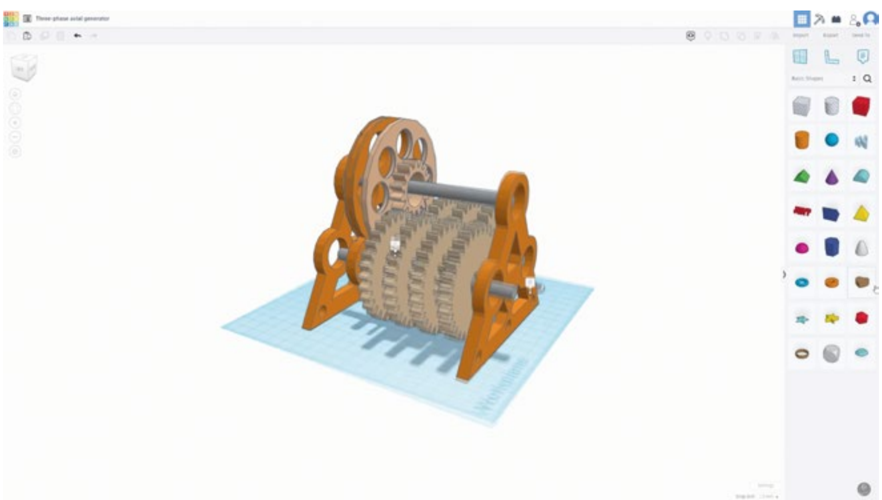
3D-printed generator

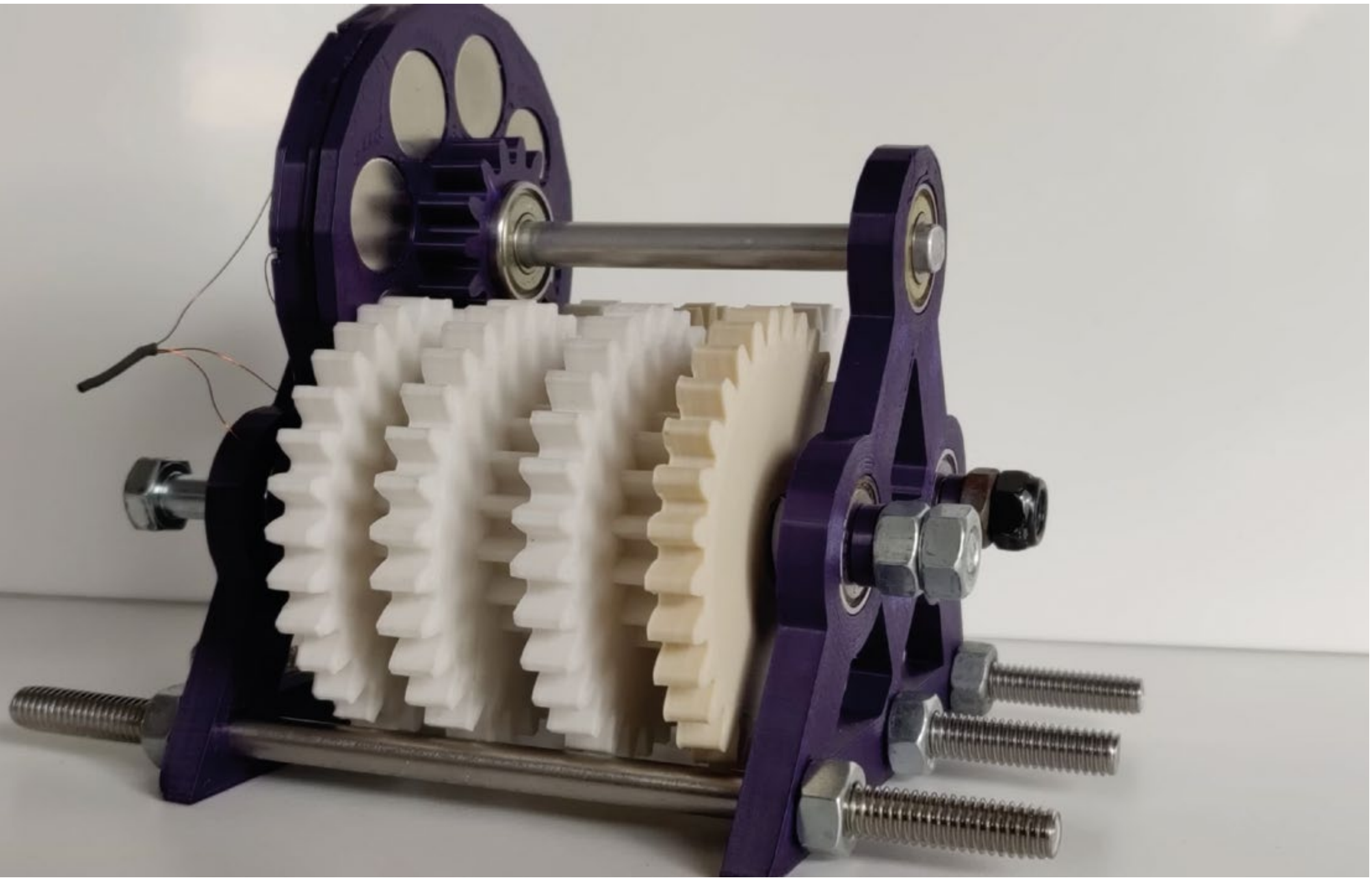
By 1970sWizard

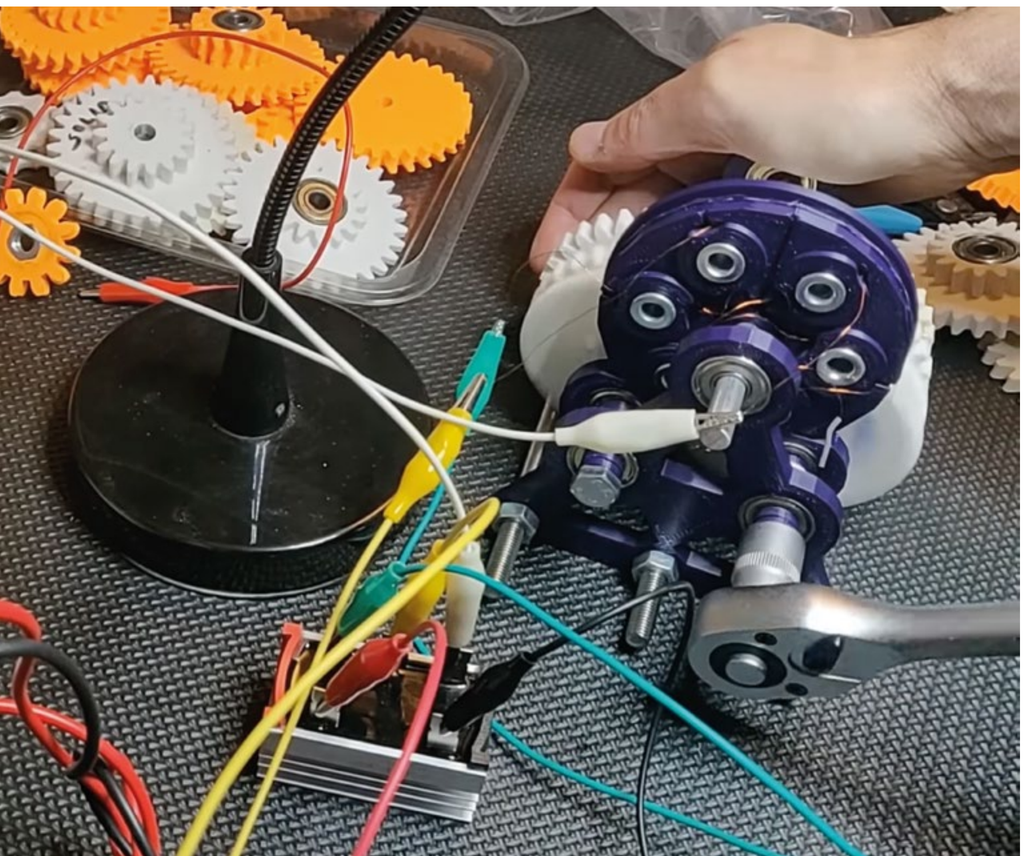
hsmag.cc/3D-PrintedGenerator

We've seen 3D-printed generators before: highly technical, superbly accurate, and made for a specific purpose. This one is different: it's extremely simple. And we say that in a good way; the simplicity is what makes the design so compelling. All you need to make this is a 3D printer and some easily accessible hardware parts, such as nuts, bolts, a wire, and magnets.

The bit that generates the electrical current is actually the smallest part of this model; most of it is a 64:1 ratio gearbox, meaning that for every turn of the handle, the rotor (that's the part with the six magnets attached to it) spins 64 times, which is enough to generate 12 V of current.







Above
Once you've got your own generator, that's where the fun begins: you could attach it to a wind turbine, or a waterwheel, or whatever source of power you wish

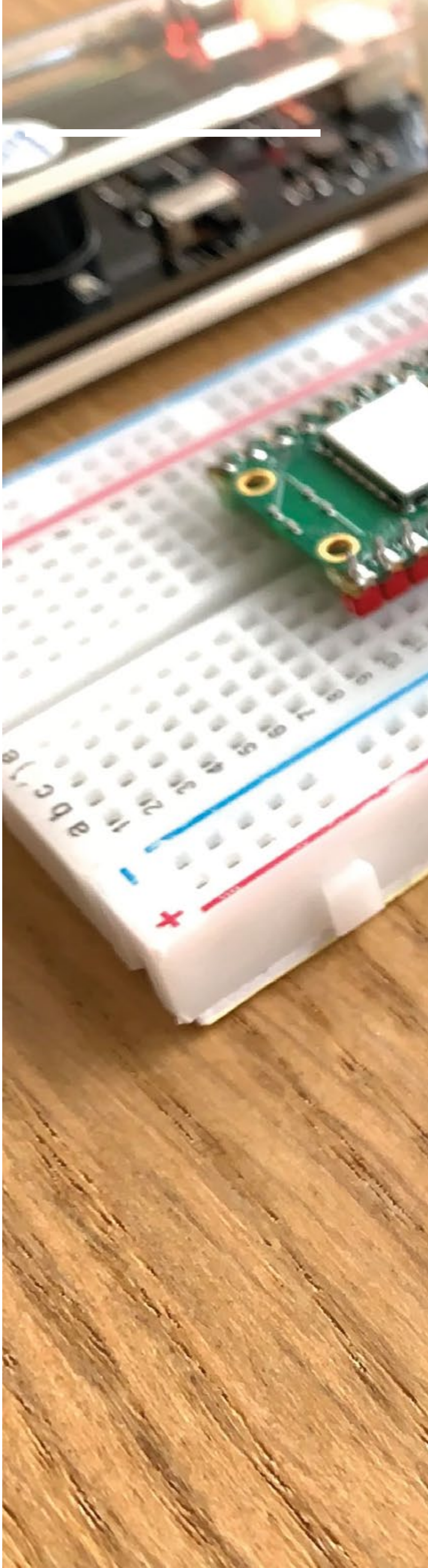
Geiger counter

By Dmytro Panin

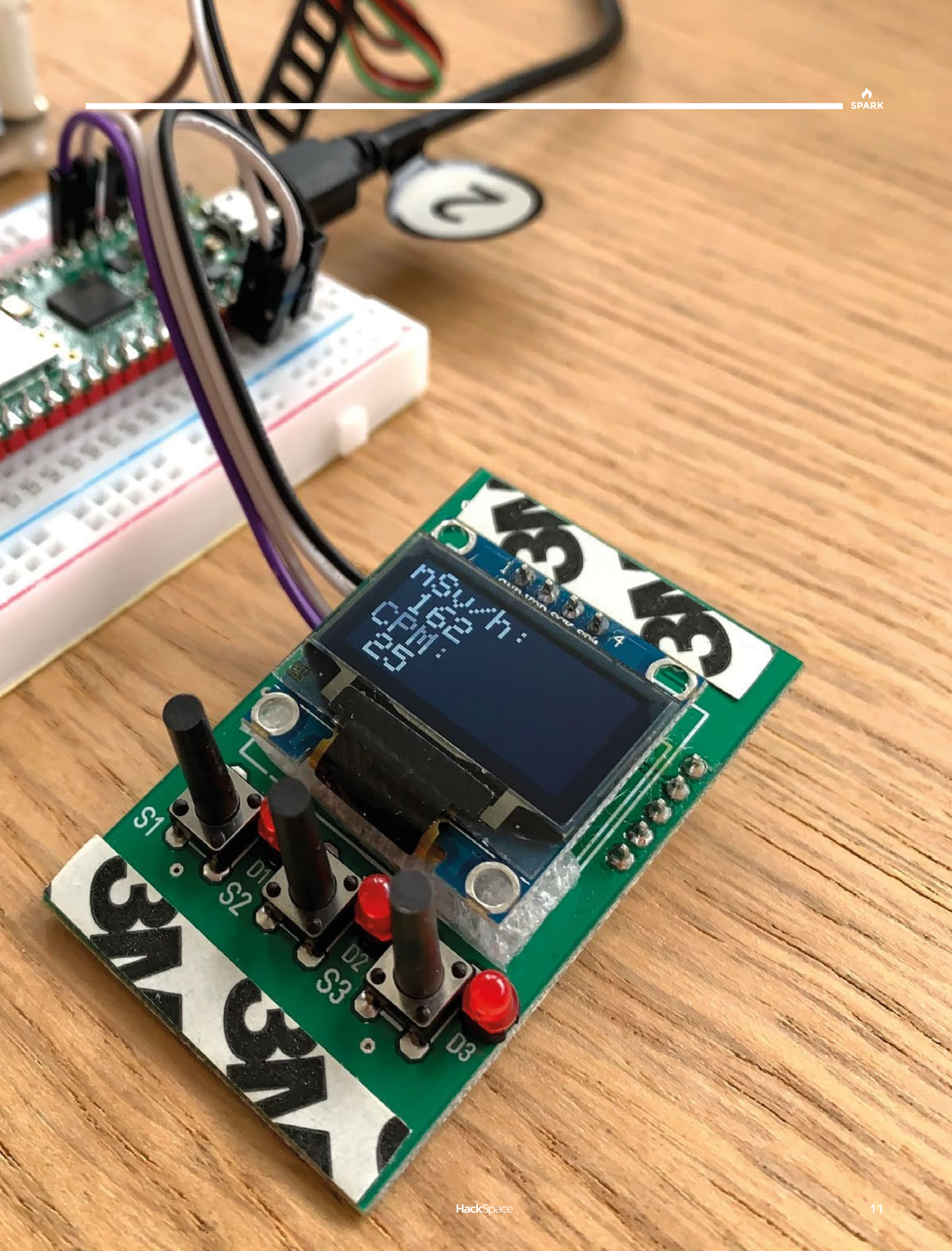
hsmag.cc/GeigerCounter

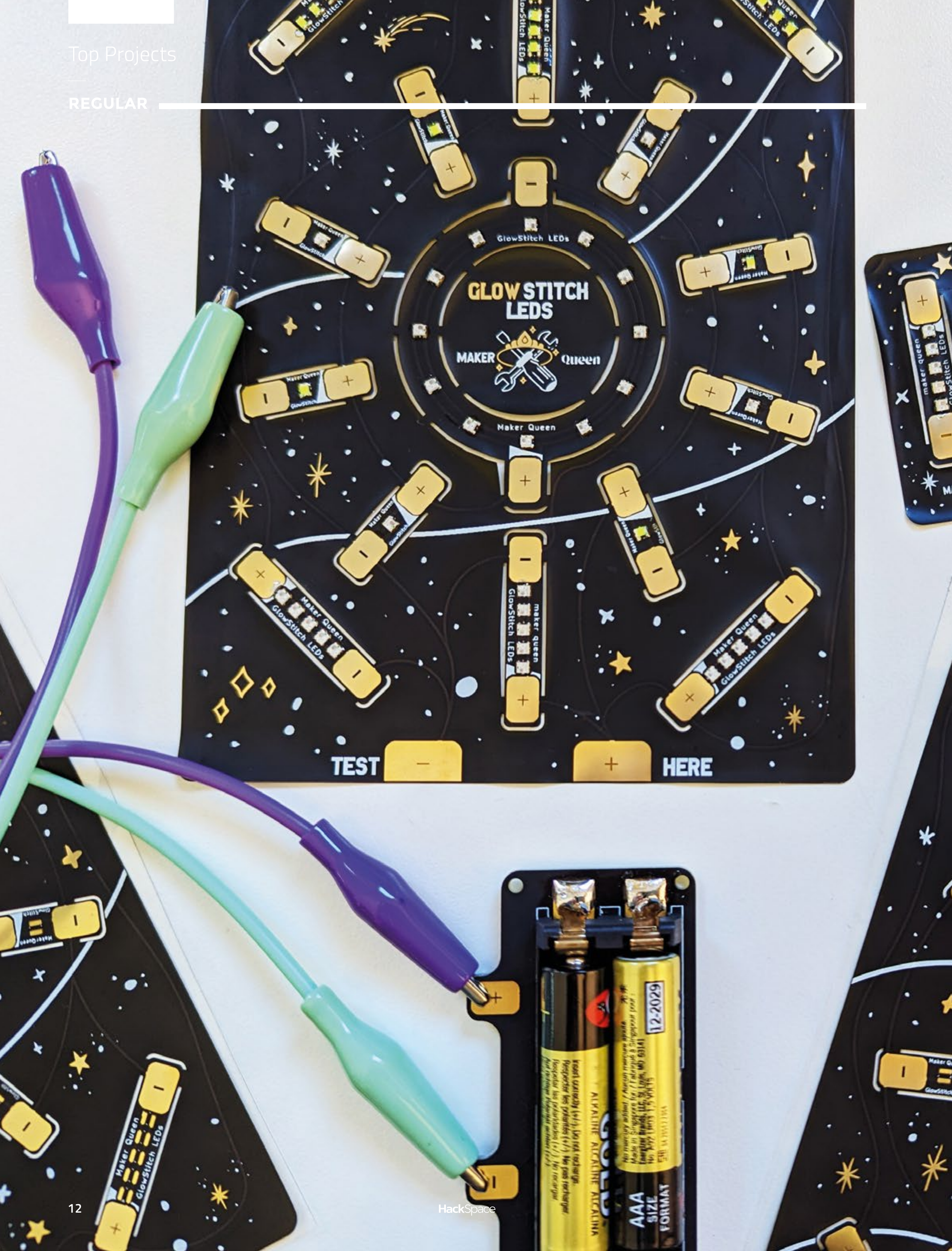
Here's a simple but useful project: an internet-connected Geiger counter. Its creator, Dmytro Panin, lives in Kyiv, Ukraine, not far from the site of the Chornobyl nuclear reactor and its associated fallout zone. Understandably, Dmytro wants to keep an eye on local radiation levels.

The build consists of an M4011 Geiger sensor connected to a Raspberry Pi Pico W, which sends the readings to a server every few hours. Dmytro's also included a 128 × 64 pixel OLED screen to display readings, for those rare occasions when he isn't glued to the internet.



Right The Raspberry Pi Pico's not just for MicroPython: this project is coded in C++


nSv/h:
162
CPM:
25
S1
S2
S3
D1
D2
D3


GLOW STITCH
LEDS
MAKER
Queen
GlowStitch LEDs
Maker Queen
TEST
HERE
12-2029
ALKALINE ALCALINE ALCALINA
AAA
SIZE
FORMAT

GlowStitch LEDs

By Steph Piper

hsmag.cc/GlowStitch

Sewable electronics is an incredibly creative field, with loads of cosplay applications. It's also an area that's attractive to people who might not otherwise be interested in electronics, as it falls so far outside the stereotypical green PCB + wires aesthetic that it would be virtually unrecognisable to an electronics enthusiast from even ten years ago.

One barrier to entry to sewable electronics has always been the difficulty of getting conductive thread to work with sewing machines. It's something you might never think of, but if you speak fluent sewing machine, you're not going to want to drop that and go back to hand stitching just to get glowing LEDs on your clothes. That's where GlowStitch LEDs fit in. They are – and the importance of this cannot be overstated – machine-sewable, making them the perfect first step into wearable electronics for someone who's experienced at sewing but new to electronics. They're non-addressable (though you can code them to turn on and off) and are open-source hardware.

Left
We see a bright future for these machine-sewable glowing things

JägerMachine

By Pegor

hsmag.cc/JagerMachine

The JägerMachine is, according to its maker Pegor, "an IoT shot-pouring machine powered by Raspberry Pi 3 Model B+. It features a full GUI that allows the user to configure the device, pour a shot, and also check previous messages sent to the machine. Once paired with its own Gmail account, the JägerMachine waits for emails containing a secret user-defined subject line and a custom message that gets displayed on the screen. Once an email is received, a shot is poured."

Brilliant. It started as a daft idea for a gift for a friend's birthday (the pair exchange Jägermeister-related gifts as a joke), and has gone from a boxy design in Fusion 360 to a fully functioning oblivion dispenser.

Right Pegor didn't get the JägerMachine finished in time for his friend's birthday this year, but this will make a great Christmas present instead


4:24 PM

Oscilloscope cyberdeck

By Jak_o_Shadows hsmag.cc/OscilloscopeCyberdeck

A non-functioning oscilloscope, plus Raspberry Pi 400, form the guts of this cyberdeck build, one of a number of small, portable computers that have been delighting the internet lately.

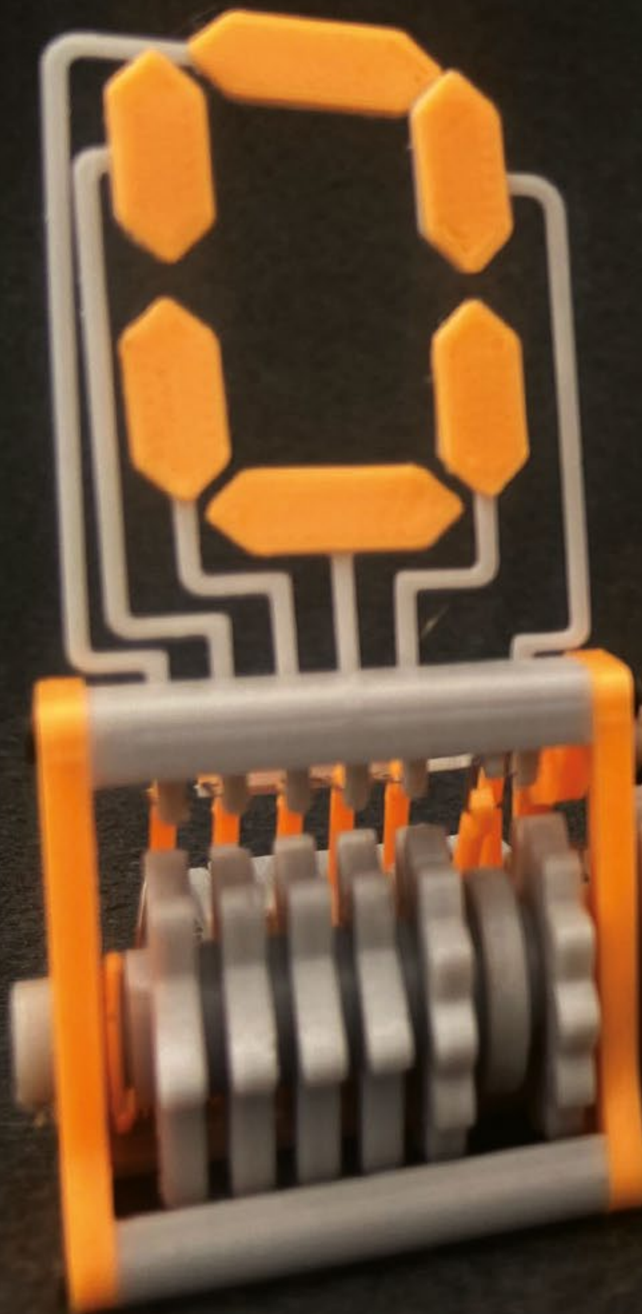
One of the benefits of upcycling old technology is that there's often a lot of room inside it for more modern gear, and that's the case here: the maker has cut into the oscilloscope case and mounted USB sockets, a power distribution board, and an LVDS/HDMI adaptor board, which enables Raspberry Pi 400 to display graphics on the original oscilloscope screen. □

Right The keyboard folds down from the oscilloscope on 3D-printed hinge brackets


Easy hunting
70 MHz
1 GSa/s
Vertical
Horizontal
Trigger

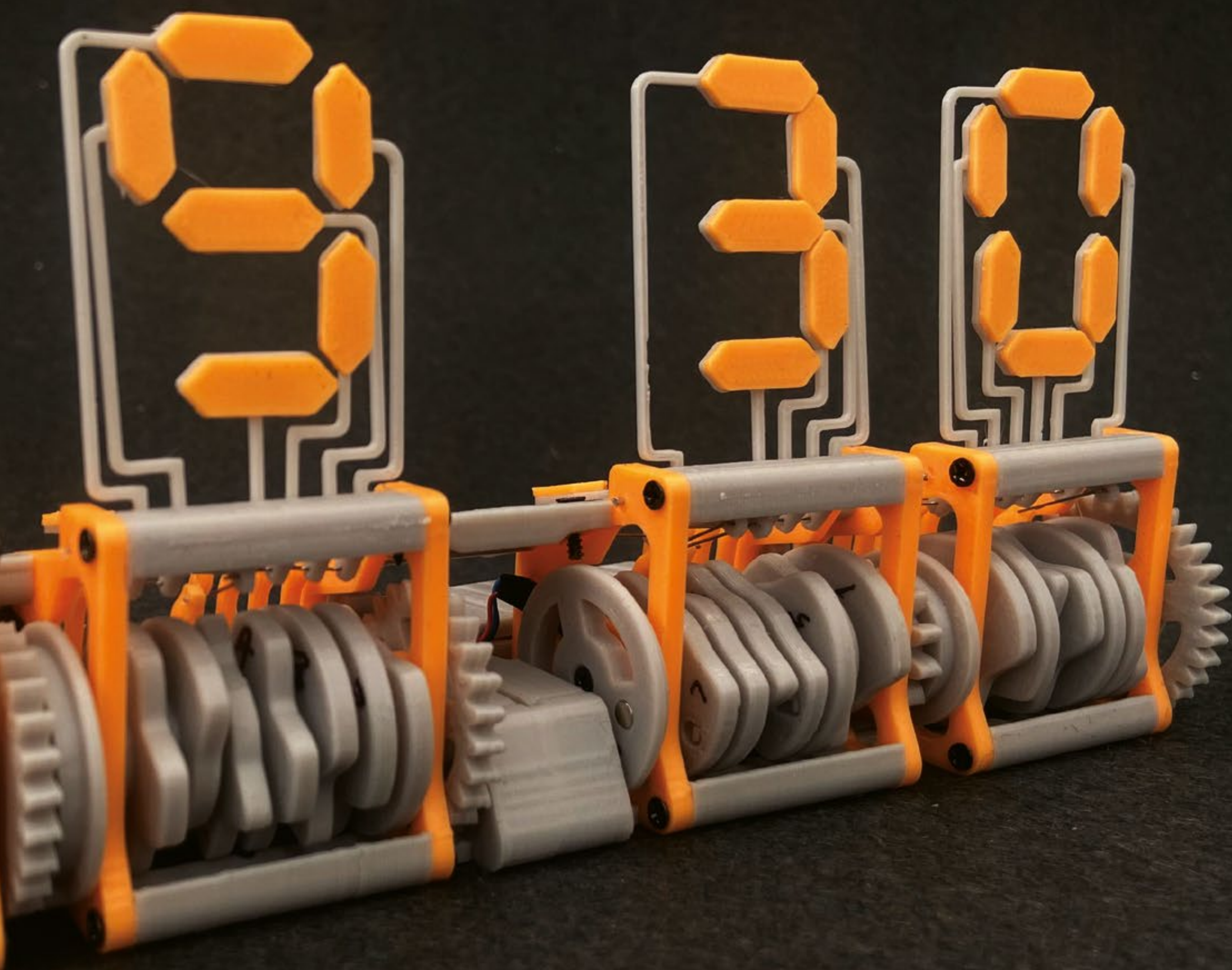
Objet 3d'art

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



Architect, industrial designer, tinkerer, Time Lord — these are the titles enjoyed by Ekaggrat Singh Kalsi, maker of this beautiful seven-segment display clock. It's pretty self-explanatory to look at: each segment of the clock is attached to a cam and a cam follower, which pops it up into a vertical position to display the time. It's the execution of this clock that makes it such a wonderful showcase for 3D printing; can you imagine trying to do this any other way? Wood would be too heavy and bulky; steel would be too big, too heavy, and too noisy. This mechanism, powered by an ATtiny84 and a pair of A4988 stepper drivers, is quiet, light, and printable on a standard printer using a regular nozzle. □

hsmag.cc/Flip7Segment



Meet The Maker: Jonny Taphouse

Building a lightweight stove with an eight-ton press

Jonny Taphouse is a product design engineer, and former wedding photographer, whose wood-burning camping stove recently raised £68,896 on Kickstarter. We caught up with him to chat about how he designed the stove, built his own equipment, and prototyping in Bristol Hackspace. Here's what he had to say.

"The stove is a USB-powered, forced air camp-stove. It's designed to burn biofuels, so wood, twigs, pine cones, and other things like that, instead of burning the usual fossil fuels, like gas.

"The stove's aimed at lightweight backpacking – a lot of twig-burning stoves are aimed more at the bushcraft market – people in forest schools or fishing hobbyists, that kind of thing, and perhaps people camping from their car.

Right The stove base is cool so it doesn't burn the ground

"I came up with the idea after the other products that I tried just weren't living up to my expectations. There are quite a lot of twig-burning stoves on the market, and I've taken a few out camping with me and ended up being let down. You'll have this huge flame and then the flame would die out, or you end up face down on the ground blowing on it – blowing into these little stoves to try and get them going just so you've got some hot noodles for dinner, and it wasn't as fun as I thought it was gonna be. But I really liked the idea of collecting sticks as I was walking, and just having that abundance of fuel around you – being able to use that instead of carrying it in gas bottles. So I just started here, thinking from that point, how can I improve these stoves? Surely there's got to be a better way of doing this? I got some bellows – it's basically a telescopic straw, you know, aerials from old radios – someone's remarketed those as a telescopic straw to blow in your fire. For me, it was a bit of a no-brainer to find a way of getting a good flow of air into the fire. My first prototype was me thinking, how can I put a fan in this thing to stop me from blowing this fire?

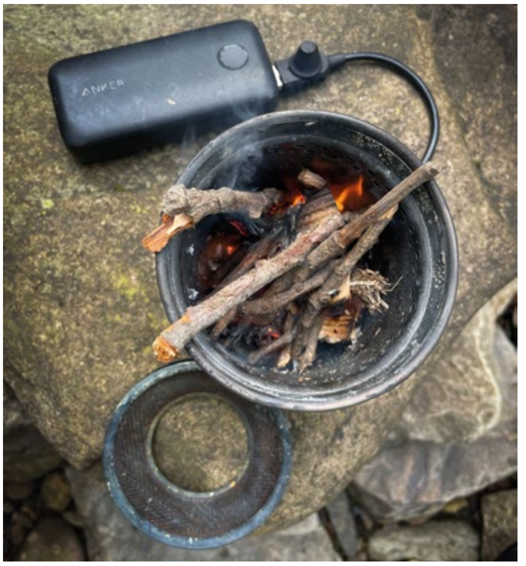
"That was just before lockdown, probably 2019. My first prototype was basically a double-walled titanium camping mug, which I had drilled a bunch of holes in, downstairs in [Bristol] Hackspace. I put a tube feeding into the bottom of it and connected up a computer fan. And that worked really, really well.

"From there, I was like, OK, this has got legs. This is actually doing what I want it to do. How can I turn it into a product? That was before lockdown, where it was just a hobby really – it was just a bit of fun I could do in my spare time. Then lockdown came along and obviously changed everything. I used to be a wedding →


LOFI

photographer, so yeah, overnight, all my work just dried up. I was doing half and half product design, half and half wedding photography, so that part of the work just dried up completely. I thought that I would take that as an opportunity to work on this as a serious idea.

"The prototypes were always titanium. I've got a bit of an obsession with it, and it's incredibly lightweight, it doesn't rust, and I just like it as a material. I think that's because it was my hobby at the beginning. I just thought it's what I enjoy working with, so I'm gonna use it. Thinking back, I was actually really surprised at how easy it was to work with. Being an engineer, you hear stories about titanium being the ultimate material used in aerospace. It's so strong you can't machine it; you can't do anything with it. I didn't find that to be true. Most of my parts are made out of quite thin foil, though – 0.4 mm. It behaves well: you can snip it, cut it, sand it, and drill it OK. It behaves a bit like stainless steel, which is quite difficult when work hardens. But if you're quick with your cuts and you know what you're doing with a pair of snips, it all works quite well.



Right A survey of users showed that the majority of people now take a USB charger camping, so why not use it to power the stove?

Below The LOFI stove has come a long way from its origin as a hacked titanium mug

HOMEMADE TOOLS

"So I ended up making a hydraulic press. The label I made for it says 15 tons, but I'm not sure whether it can quite take that. I've got it up to eight tons. And you could see a little bit of flexing. Yeah, so I'd say I've made an eight-ton hydraulic press. And that was a bit of a lockdown adventure in itself, really. I was looking on eBay as to what kind of presses I could buy to start doing some metal forming. And the eight-ton presses were absolutely enormous, and I have no idea how much they weigh, but they don't look like you could just move them, or put one in the back of my car and pick it up.

"I drew it up in Fusion 360, trying to get an idea of what changing different parameters would mean for the design. Because it's a C-shaped, open-fronted press, the depth of the throat behind this working area is critical. If you make this too big, you suddenly get some crazy forces going on, and the whole thing wants to pull itself apart. Fusion 360 was a really good platform there for just playing around and simulating (FEA) the loads and seeing how it's all gonna work. Then I had to start researching the hydraulics because I didn't know anything about hydraulics. So, fortunately, there are people who have made hydraulic presses and things on YouTube, and I managed to use some of the information from what I found on there to start designing the hydraulic circuitry.

"It's made in the thickest steel that I could get laser-cut online, which I think is 20 mm, and that ultimately dictated the design parameters that I could use, as obviously, the easiest way of making something stronger is just choosing thicker material to make it from, and I couldn't do that. So, by having that fixed, [it] drove the overall shape of the design. It was originally designed to be bolted together because I didn't have a welding facility or know anyone who

could weld. All the forces are designed to push through the main frame, rather than have any tension on any of the joints. The plate that holds the hydraulic ram is pushing against the inside of the C-clamp shape. It's designed to be screwed together, and that's what I did. It's got these massive M14 bolts that go into holes that I then had to tap myself – that was a huge mission.

"I was really pleased with it when it came together. And I just thought, you know what, I'm proud of this, I want to do it properly, I want to get it welded. I happened to be having some welding work done on my boat at the time, so I found this welder who was used to welding thick gauge steel, and he welded it all up for £100, and I put some nice two-pack epoxy paint [on it].

"So, what has the press allowed me to do? I'm trying to think why I originally started using a press! The parts that I had available to me were cut from old camping mugs, cups, and things like that. And I could never quite get the diameter that I wanted to be working with, and I was using the hydraulic press to kind of inflate or enlarge this diameter. I came up with various different 3D-printed tools, some that would splay out in segments to just change, say, a 90 mil diameter to a 91 mil diameter, but that's what I needed. So, I was just flaring out these bits of titanium and forming them into the shapes I needed. I think that's the reason I first started using the hydraulic press, but then, because I had access to it, I realised that I could emboss logos, I could do riveting. And I could do various different sorts of basic sheet metal forming stuff with this press; it was amazing.

"Thinking back, the need for a press actually evolved from me trying to achieve something which I no longer even use in my design. Now, I can just ask the factory to make it in that size. But before you have these conversations and connections with a factory, you need to be able to prove the idea works, and having the press allowed me to prove if these concepts either worked or failed. And in that case, they didn't work at all, but it's now set me up with other ideas and other tools.

"I didn't want to use nuts and bolts in the design of the stove – that was really important to me. I wanted the whole stove to be a toolless design, so that people on the field, camping, didn't need to carry a screwdriver or an Allen key with them. Having the press allowed for my imagination to get carried away with different ideas about what can I do with rivets. The rivets that I needed didn't exist yet, but I could →

> **I didn't want to use nuts and bolts in the design of the stove** – that was really important to me

Above
At under 150g, the LOFI stove is light enough to carry anywhere

draw them up in CAD and get them made for me, try them, and see how they work.

PROTOTYPING

"I'm actually quite embarrassed to say I only made about three or four prototypes. I feel like I should be saying this huge number to get to the design where everything is perfect. The exciting thing about my final design is just how small and compact it is, and that it doesn't burn the ground. In the original design, the stove base itself got very hot, because you've basically got a little micro furnace and it was sat directly on the ground, so everywhere I'd go camping, there was a black circle. That's a massive no-no for leaving no traces after camping, so I had to overcome that.

"Originally, the fan was on the outside of the stove, similar to other twig stoves on the market. My biggest challenge was working out how to get the fan inside the stove. I wanted to have it hidden away, and to make it a bit more robust. That's where the design kind of evolved. And it's a weigh-up between function, aesthetics, and then what is actually machinable or makeable. Those feel like the three main design pivots, because the aesthetic of the LOFI stove is actually quite important to me. I want it to look like a nice streamlined thing. And part of achieving that was going to be putting the fan on the inside.

"One of the prototyping problems I had is that the stove is a twin-wall design. The air that feeds the fire actually goes through the inside wall of the stove. And if you look at the design, it's quite obviously evolved from a double-walled insulated mug. So, to get the airflow running up inside the mug, I had to drill holes from the inside! My first solution for this was to use a 3D-printed Dremel template that I drew up in CAD. I then used a small right-angle Dremel attachment, and I cut down on drill bits so that they were short enough to fit within the inside diameter of

the cup. This wouldn't scale up, so I used a fairly cheap eBay CNC machine that I had bought for another project. I think I picked it up for about £300/400. Then I put an additional [rotational] axis on it, which holds the mug. Then I got a custom arm made. But again, I designed in Fusion. The arm extends the spindle of the CNC machine out to the side – I don't know what the correct term for that is, but basically, it has a belted spindle extension out on an arm. It was fundamentally doing the job of the 90-degree Dremel head, but it was trying to do that with a CNC spindle, which I did manage to do. And there are some quite amusing videos of it on Instagram, where anyone who knows anything about CNCing will just die watching. There are plenty of sparks going on, where titanium sort of ignites on the cutter. So yeah, that was a really fun journey of working out how to get that all up and running and learning the G-coding for another axis.

"I think probably the relationship with the manufacturer I chose has been quite interesting. I didn't tell him what he was making at the beginning, I just asked if he could do it – I didn't want to really give away too much of my IP. So I moved very slowly with updating the factory with the design and what I needed them to make, telling them that they were making a titanium mug at the beginning. For the Kickstarter launch, a lot of the parts weren't factory-made – they were made by me in my little workshop. I designed all sorts of deep drawing tooling, and simple stamping tooling made from PLA. This was 3D-printed tooling, which I then used in the press – I've not really seen many others exploring this. That allowed me to develop the design which I'm going to go forward with – getting the manufacturing tooling made.

"I think that I reached a sort of tipping point, where it is just, yeah, it's now or never – **let's just do the scary thing**"

"I always had the idea that I would have to crowdfund this as a project because I have zero money to invest myself. The only reason that this project existed in the first place was through the payment from the government as support for the self-employed through lockdown. Then there was the bounce back loan – I put this into getting the essential tooling for the titanium parts made in China. I've invested a little of my own money, selling all of my photography equipment to get the prototypes made. I was then stuck with no marketing budget at all, or maybe I had £50 a month I could carefully spend on Instagram ads.

Left
Each load gives you up to 15 minutes of cooking time

"A lot of the decision [of when to put the crowdfunding campaign live] was born out of the fatigue of working, and working, and working on this project with no reward, and me having to keep banging my own drum, constantly telling people about it on Instagram – I had also completely run out of money. Everything was just getting a bit tight, and it just felt like the right moment. I guess it was just getting to a point where, even if this doesn't work, it's not for lack of trying.

"I think that I reached a sort of tipping point, where it is just, yeah, it's now or never – let's just do the scary thing and have a go. I was looking at my mailing list and thinking, well, I've got 2000 people on a mailing list. So, how many of those people will convert? And that's it, that's the scary stuff, because you don't know.

"Everything I read everywhere online says that those conversion rates are very small, like around 1% to 2% of the people will convert to backing the project, but I had a feeling that I had a better relationship with my audience than most people do, and my design was sound, and it was what people wanted.

"Because of the way that I've interacted with them on Instagram, and the support they gave me, it gave me confidence. But it all still felt really high-risk. I guess I felt like I was way out of my depth, just like I was swimming out into this ocean of unknown, and yeah, just taking on the risk, and facing some of those fears. It's really paid off."

Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction), let us know at **hsmag.cc/hello**

PLANT CARE, PART 1

I note from your look at 'compostable' PHA filament that it hadn't actually started to break down by the time you wrote it up. Could you give us an update? Have the worms eaten it yet? Will you be growing your tomatoes in it next year?

Diane
London

Ben says: I'll check tonight [checks; I've gone drastically wrong, and I think the worms have drowned]. Obviously we're only going to get a limited amount of compost out of a spool of plastic, so I'll have to rely on vegetable peelings etc. for the bulk of my compost, but I am hoping that next year's vegetables will have a little bit of 3D-printed magic in them.

INTERNET OF THINGS

Help me out: what is the Internet of Things? I've just read issue 60, so I feel like I should know, but it feels like the terms are just too broad to make any sense. My TV connects to the internet: is that a Thing? My phone? If you have a Tesla, are you driving around in a Thing of the Internet? Help it make sense!

Richard
Berlin

Ben says: It is a pretty vague term, and ultimately, that's helpful. I think the thing that makes a thing a Thing is that it would exist and be useful even without the external data source. So, turn off the data going into your TV, and it's just an expensive black box; turn off the data going into your IoT coffee machine, and it's still a coffee machine; it's just a less useful, less interesting, less hackable coffee machine. That's also what makes it so exciting. You can take anything electric and graft internet connectivity to it, and you've suddenly built a new product.

PLANT CARE, PART 2

Cheers for the Pico plant monitor! It may sound like a back-handed compliment, but I'm not going to build one; I just really like the approach your writer took. I spend all day looking at a screen, so having a physical display rather than a seven-segment display (or even coloured LEDs) makes a welcome change. I'm going to try to incorporate that attitude into my own projects – anything to get away from the dreaded glowing pixels.

Davey
Paisley

Ben says: All it takes is a Pico and a stepper motor and you can turn anything you want into a physical display: an arrow that points up or down can indicate change; a needle moving around a dial can indicate temperature, or moisture, or if you can manage to divide by 60, time. You can rotate colours into and out of a viewfinder to show red/green/amber to show the current terror threat level… OK, maybe not that one, but you get the point. Numbers are good, but they're not the only way of representing data.

CROWDFUNDING NOW

Ukama

Build your own mobile network

From $599 | **hsmag.cc/ukama** | **Delivery:** September 2023

We'll be honest, this is one of those projects we don't fully understand and have absolutely zero use for, but we love that it exists. And there is almost certainly someone out there (quite possibly reading this) who has been looking for exactly this thing: an open-source mobile network. Not the phone – the hardware to set up and run the network that mobile phones connect to.

An obvious problem with this is spectrum licensing – to run a mobile network, you need a little chunk of the radio spectrum that you're allowed to use, and these are sold off for enormous sums of money by the appropriate authorities in each country. In the USA, however, there's a little section that you're allowed to use without a licence – the Citizens Broadband Radio Service (CBRS), which is different from CB. Not all phones support CBRS, but an increasing number of new phones do. For now, this technology is only legal in America.

You can run a network of any size, from one covering just your home, to a bigger building or campus, even a town or larger. The more area you want to cover, the more hardware you're going to need. Users need a SIM (or eSIM) as with a normal network, and you can use roaming if outside your network. Or, you can have an additional plan with a different network for when you're on the move.

For those of us that live in towns and cities, having a mobile internet connection is just a regular part of life, but that's not true everywhere. Ukama lets people take control of the airwaves and build the networks they need where they need them, without relying on global megacorps.

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.

Left
Down with big telco – it's time for small telco

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 32

PROGRAMMING FOR MAKERS

Coding is a skill: learn it, and become a better maker

PG 44

HOW I MADE: A HOT AIR BALLOON

Build a flying machine that'll make you feel like a Victorian gentleman

PG 50

INTERVIEW: POOCH

Inside the infinitely exciting 3D printing industry with Alan 'Pooch' Puccinelli

PG 58

IMPROVISER'S TOOLBOX

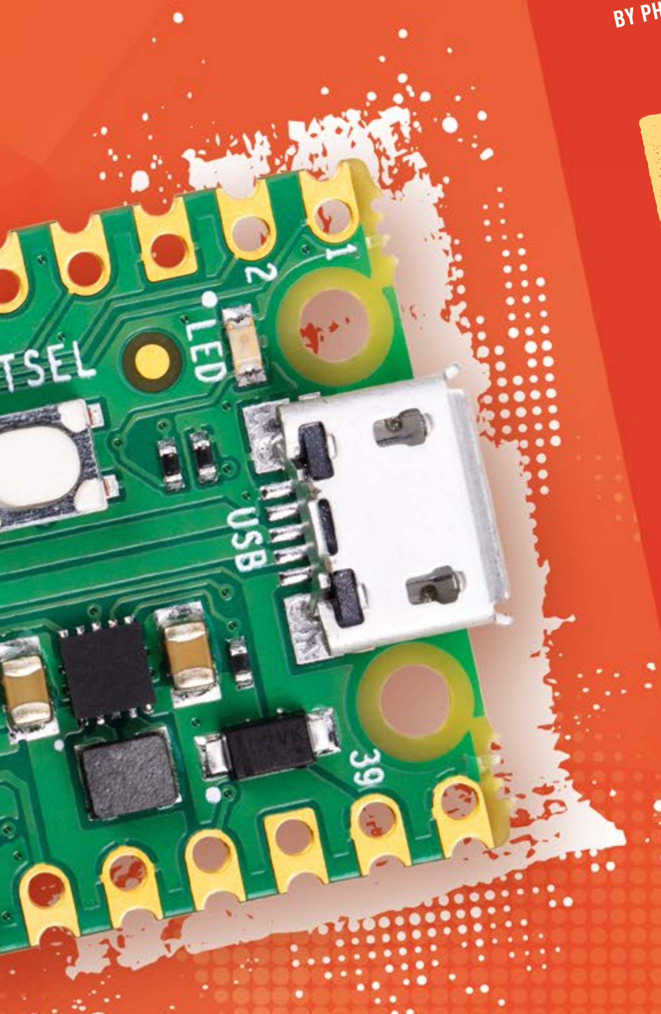
Now that all music is online, why not do something useful with your old CDs?

LEARN TO CODE

FOR MAKERS

Learn to code and bring life to your projects by making them light up, sense their surroundings, and move

BY PHIL KING



Digital making and learning to program go hand in hand. While you can borrow code from other makers to use in projects, and it's often a good starting point for adapting, it's useful and empowering to learn the basics of programming.

In this introductory guide, we'll explore some of the key coding concepts using a couple of variants of the easy-to-understand Python language. Throughout, we're using the Raspberry Pi Pico and mainly MicroPython, but the same techniques can be adapted for use with other microcontrollers or single-board computers (e.g. Raspberry Pi) that support Python or its variants.

We start out simple with the 'Hello World' of digital making: getting an LED to blink. We then add a push-button to trigger it, using a conditional statement. Controlling a multicoloured RGB LED is next on the list, with three GPIO pins used to set its shade. This is followed by a guide to lighting NeoPixel strips using CircuitPython and an LED animations library.

Next up, we move on to reading sensors, starting with a DHT11 to sense temperature and relative humidity. Using a moisture sensor placed in the soil, we then create a plant monitor that sounds an alert when it needs watering. Last but not least, we get moving by controlling motors for a wheeled robot, along with a servo for precision movement.

With a bit of coding knowledge, you can bring any project to life and get it working exactly as intended. Do let us know what you make! →

> "With a bit of coding knowledge, you can bring any project to life"

CODING WITH LEDS & BUTTONS

CONTROL AN LED AND TRIGGER IT USING A PUSH-BUTTON

YOU'LL NEED

Raspberry Pi Pico with soldered headers and MicroPython installed (see hsmag.cc/PicoMicroPython)

Breadboard

1 × LED (any colour)

1 × 330 Ω resistor

Push-button

Male-male jumper wires

Light-emitting diodes (LEDs) are a mainstay of the digital making world, used in many electronics projects. Whichever flavour of Python you're using, they're easy to turn on and off – in the code or by adding a push-button.

Blink an LED

Blinking an LED is the digital making equivalent of a simple 'Hello World' program. For this example, we'll use a Raspberry Pi Pico microcontroller and code it with MicroPython, but you could fairly easily adapt it to use another type of board and/or CircuitPython.

Wire up the circuit on a breadboard as in **Figure 1**. A red LED has its longer, positive leg connected – via a 330 Ω resistor – to the GP15 pin on Pico; its shorter leg is connected to a GND (ground) pin. It won't light up unless we trigger GP15 in our code. We'll use a time delay to turn it on and off every second. First, we need to import a couple of MicroPython modules. The **machine** module enables you to read and control Pico's GPIO pins. The **utime** module is used to measure time and add delays.

```
import machine
import utime
```

Next, we define an object for our LED using the **Pin** class, with the pin number as 15 (for GP15) and setting it as an output with **OUT**:

```
led = machine.Pin(15, machine.Pin.OUT)
```

To turn the LED on and off repeatedly, we'll create a **while True:** loop that never stops running (unlike a **for** loop that runs a set number of times). Indented lines of code form part of the loop and so are repeated in turn.

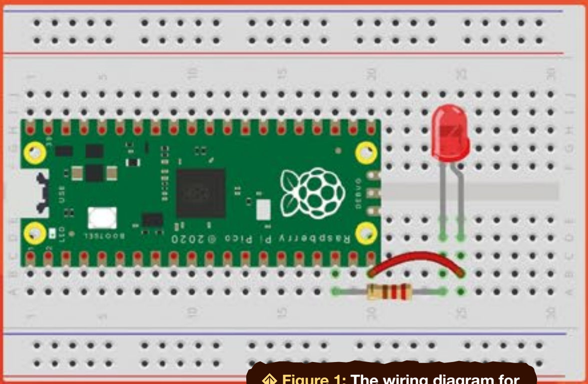


Figure 1: The wiring diagram for connecting an LED. Make sure to use a resistor to limit the current

```
while True:
    led.value(1)
    utime.sleep(1)
    led.value(0)
    utime.sleep(1)
```

Make sure you indent the lines under **while True:** by four spaces. Setting the **led.value** to 1 turns on the LED; setting it to 0 turns it off. The **utime.sleep** command adds a delay of the number of seconds in the brackets. The complete code is:

```
import machine
import utime
led = machine.Pin(15, machine.Pin.OUT)

while True:
    led.value(1)
    utime.sleep(1)
    led.value(0)
    utime.sleep(1)
```

Run the program and your LED with blink. An alternative is to use the **toggle** command to change the LED to the opposite state (on or off): **led.toggle**.

Programming Pico

To program your Pico, connect it via USB to a computer. You'll need a Python IDE to enter and run your code. We're using Thonny (**thonny.org**), with its Python interpreter set to 'MicroPython (Raspberry Pi Pico)'.

Push the button

As well as controlling output devices like LEDs, we can read the signals from input devices such as a push-button. We'll add a button to our previous circuit, with one side connected to GP16 on Pico, and the other to the 3V3 pin (see **Figure 2**) – pushing it completes the circuit.

First, let's try reading the state of the button. As before, we import the `machine` and `utime` modules. We define an object for the button, with its pin number and setting it as an input; we also set it to `PULL_DOWN` mode – see 'Pull up or down'. We then use a `while True:` loop to repeatedly check the button's state; if it's pressed, we print a message to the Shell.

```
import machine
import utime
button = machine.Pin(16, machine.Pin.IN,
machine.Pin.PULL_DOWN)
```

```
while True:
    if button.value() == 1:
        print("Button pressed!")
        utime.sleep(2)
```

The indented lines are only executed when the `if` condition is met. Note the use of == to check if value is equal to a number (whereas = is used to set a variable to a value).

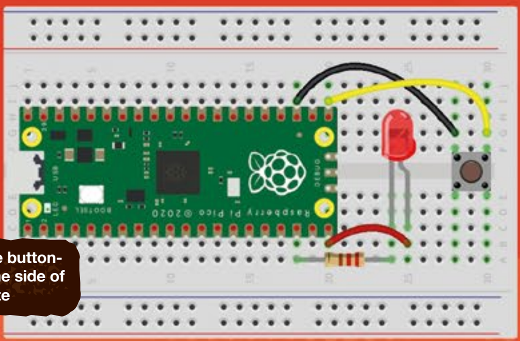


Figure 2: The wiring diagram for the button-activated LED. Use the two pins on one side of the button or those diagonally opposite

Pull up or down

Pico's GPIO pins can have their built-in resistors set to pull-down or pull-up. If the former is used, as in our example, the button should be wired to 3V3 and when not pressed, the input reading is 0. If pull-up is used, the button should be wired to GND and when not pressed, it'll read 1.

Button-activated LED

Now, let's combine the parts of our programs to light the LED whenever the button is pressed.

```
import machine
import utime
led = machine.Pin(15, machine.Pin.OUT)
button = machine.Pin(16, machine.Pin.IN,
machine.Pin.PULL_DOWN)

while True:
    if button.value() == 1:
        led.value(1)
    led.value(0)
```

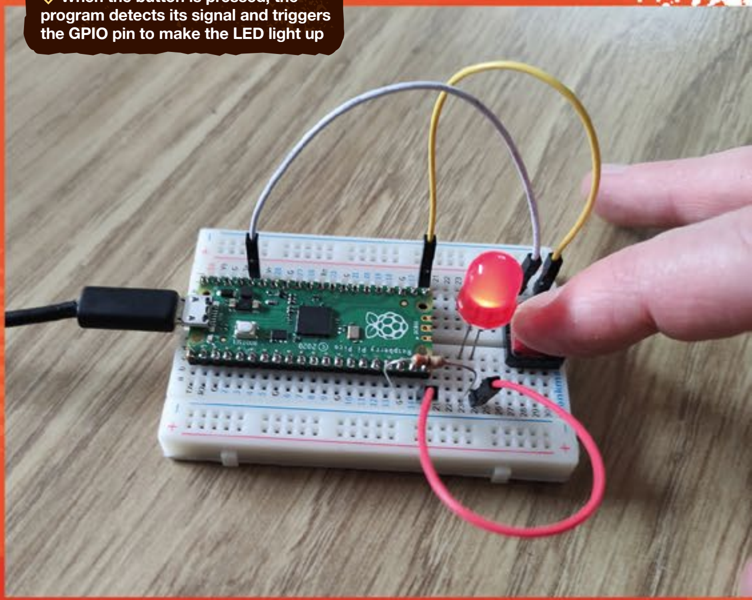
Run the program. Whenever you press the button, the LED will light; when unpressed, the LED will turn off.

Want to toggle the button on or off whenever you press it? Replace the `while True:` loop with this:

```
while True:
    if button.value() == 1:
        led.toggle()
        utime.sleep(1)
```

Now you know how to create code to control LEDs and read buttons, you can add multiple ones connected to different GPIO pins. Ground or 3V3 connections can easily be shared by using the rails on the sides of the breadboard: connect the rail to 3V3 or GND and then wire your components to the rail. You could also add a piezo buzzer that beeps when a button is pressed. →

When the button is pressed, the program detects its signal and triggers the GPIO pin to make the LED light up



"Whenever you press the button, the LED will light"

MULTICOLOURED LIGHTS

CREATING CODE TO CONTROL RGB LEDS AND NEOPIXELS

YOU'LL NEED

PROJECTS 1 & 2

- **Raspberry Pi Pico with soldered headers and MicroPython installed** (see hsmag.cc/PicoMicroPython)
- **Breadboard**
- **1 × RGB LED**
- **3 × 330 Ω resistors**
- **Male-male jumper wires**

PROJECT 3

- **Raspberry Pi Pico with soldered headers and CircuitPython installed** (see hsmag.cc/PicoCircuitPython)
- **Breadboard** (optional)
- **1 × NeoPixel strip/shape**
- **Male-female jumper wires** (or female-female)

You can light an RGB LED in any colour by altering the red, green, and blue parameters. The same goes for NeoPixels (aka WS2812B LEDs), for which we'll use a CircuitPython LED animations library.

Light an RGB LED

An RGB LED has four legs: for red, green, blue, and ground. Depending on the manufacturer, the order of the colours may vary, but the longest leg is for the ground connection. Connect the LED to Pico as in **Figure 3**. We're using GP13, GP14, and GP15 to connect its red, green, and blue inputs, all via 330 Ω resistors to limit the current. Note that each of these GPIO pins uses a different PWM channel (Pico has 16 in total).

This time in our MicroPython code, instead of importing the full `machine` and `utime` modules, we'll just import the classes we need: `Pin`, `PWM`, and `sleep`. This means we can omit the 'machine' and 'utime' prefixes when using them.

```
from machine import Pin, PWM
from utime import sleep
```

We then set each of the LED's components to PWM on the connected pin.

```
red = PWM(Pin(13))
green = PWM(Pin(14))
blue = PWM(Pin(15))
```

PWM is short for 'pulse-width modulation', which is how we'll control the brightness of the RGB components of our colour. This is achieved by pulsing a digital GPIO output pin on and off at high frequency. The duty cycle determines how much of the time it's on, thus setting the LED brightness.

By default, the PWM frequency (of pulses) on Pico is 1907Hz; it's best practice to set a value in your program, however.

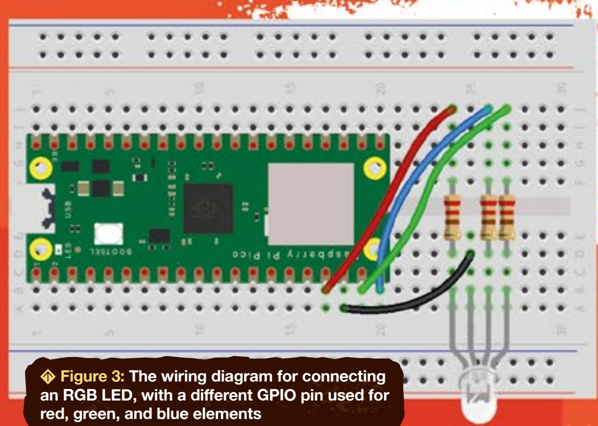


Figure 3: The wiring diagram for connecting an RGB LED, with a different GPIO pin used for red, green, and blue elements

```
red.freq(1000)
green.freq(1000)
blue.freq(1000)
```

In a `while True:` loop, we'll light up the red, green, and blue parts of the LED in turn.

```
while True:
    red.duty_u16(65535)
    sleep(1)
    red.duty_u16(0)
    green.duty_u16(65535)
    sleep(1)
    green.duty_u16(0)
    blue.duty_u16(65535)
    sleep(1)
    blue.duty_u16(0)
```

With `duty_u16`, we're setting the duty cycle with an unsigned 16-bit integer. The higher the duty cycle, the more of the pin's pulses will be on instead of off. Here, we're setting it to the maximum 65535 for full brightness.

Mix RGB LED colours

Let's create different shades by combining red, green, and blue primary light colours. We'll cycle through colours whose RGB combinations are stored as tuples in a list. This time, we'll use a function that we'll call repeatedly in a **while True:** loop. We start with the usual setup code.

```
from machine import Pin, PWM
from utime import sleep
red = PWM(Pin(13))
green = PWM(Pin(14))
blue = PWM(Pin(15))
red.freq(1000)
green.freq(1000)
blue.freq(1000)
```

Next, we create our list of shades. A standard Python list is a range of comma-separated values within square brackets. Here, we're using tuples (in parentheses) for our three RGB values for each list entry.

```
shades = [(255, 0, 0), (127, 127, 0), (0, 255, 0), (0,
127, 127), (0, 0, 255), (127, 0, 127)]
```

For this example, we're using primary and secondary light colours – red, yellow, green, cyan, blue, and magenta – adjusted for equal brightness.

We then create a function to select the colour. Its **rgb** input variable is used to select a tuple from the list – the value of **rgb** is determined when calling the function later in the code. The **[0]**, **[1]**, and **[2]** select the first, second, or third part of the tuple – Python is zero-indexed, so it starts counting from 0. We then multiply the value by 256 to get the 16-bit integer used to set the brightness level of each RGB component.

```
def set_colour(rgb):
    red.duty_u16(rgb[0] * 256)
    green.duty_u16(rgb[1] * 256)
    blue.duty_u16(rgb[2] * 256)
```

We create a **counter** variable that will increase by 1 each time in our **while True:** loop; the latter calls the **set_colour** function repeatedly with a different **counter** value to select a new shade from the list.

```
counter = 0
while True:
    counter +=1
    if counter >= len(shades):
        counter = 0
    sleep(0.5)
    set_colour(shades[counter])
```

Here, **+=** adds one to the counter. An **if** statement is used to check if it is greater than (>=) the length (**len**) of the **shades** list; if so, it resets it to 0. After a short **sleep** delay, we call the **set_colour** function; the contents of the parentheses apply it to the **shades** list with the current **counter** value to select the RGB colour.

> *"We create a function to select the colour"*

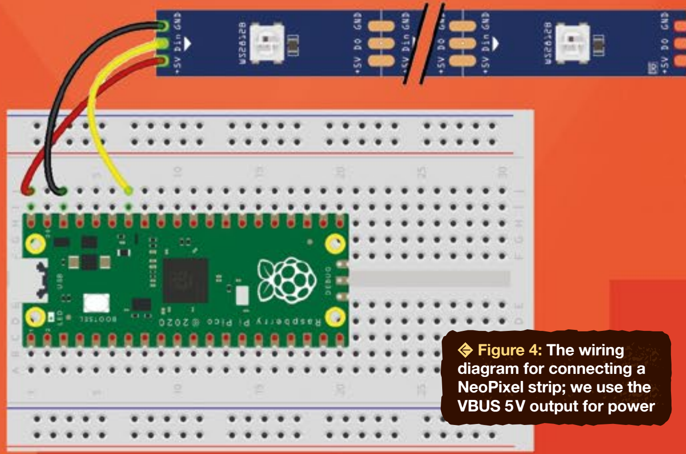


Figure 4: The wiring diagram for connecting a NeoPixel strip; we use the VBUS 5V output for power

NeoPixel lights

For this project, we're using CircuitPython (install from **hsmag.cc/PicoCircuitPython**) so we can make use of Adafruit's LED animations library. You'll also need to download the CircuitPython libraries bundle (**hsmag.cc/CPLibraries**) and copy the **adafruit_led_animation** folder and **neopixel.mpy** file to the **lib** folder on Pico.

Connect your NeoPixel strip (or shape) to Pico. Vcc goes to VBUS (5 V) on Pico, GND to GND, and Data-In to a GPIO pin – we're using GP28 (see **Figure 4**). In Thonny, change the interpreter to 'CircuitPython (generic)' and enter the following program. Note how CircuitPython uses the **board** library, not **machine**, for controlling Pico's pins.

```
import board
import neopixel
from adafruit_led_animation.animation.rainbowcomet
import RainbowComet

pixels = neopixel.NeoPixel(board.GP28, 20)
pixels.brightness = 0.5

rainbow_comet = RainbowComet(pixels, speed=0.1,
tail_length=7, bounce=True)

while True:
    rainbow_comet.animate()
```

Change the 20 parameter for the **pixels** variable to how many NeoPixels you have. Run the code for a rainbow comet trail effect. This is just one example from the LED animations library. You can group animations to run them together or in sequence. For more info, see **hsmag.cc/CPLEDAnimations**. →

Running the rainbow trail animation on a NeoPixel strip

READING SENSORS

CONNECT A SENSOR AND TRIGGER AN ACTION BASED ON ITS READING

YOU'LL NEED

- Raspberry Pi Pico with soldered headers and MicroPython installed (see hsmag.cc/PicoMicroPython)
- Breadboard
- Male-male jumper wires

PROJECT 1

- DHT11 sensor, e.g. hsmag.cc/DHT11

PROJECT 2

- Moisture sensor, e.g. hsmag.cc/moisturesensor
- Piezo buzzer – active type
- Male-female jumper wires

There's a wide range of electronic sensors that you can connect to your microcontroller or single-board computer. Readings can then be viewed and also used with `if` conditional statements in the code to trigger an action such as sounding an alarm.

Digital sensor

First, we'll read a sensor with a digital output signal. For this example, we're using the popular DHT11 temperature and relative humidity sensor – alternatively, you could use a DHT22. Connect it to Pico as in **Figure 5**. Its Vcc pin is connected to Pico's 3V3 pin, GND to GND, and DOUT (data out) to GP15.

Our program will read the digital values for temperature and humidity from the sensor and print them in the Shell pane of our Thonny IDE. At the top, we import the `dht` module, which handles interfacing with the sensor. We also import the `Pin` class from the `machine` library, as usual.

```
import dht
from machine import Pin
```

We then define a `sensor` object for our DHT11 sensor using the `dht` module, `DHT11` method, and `Pin` class, with the pin number as 14 (for GP14). Note that if you're using a DHT22 sensor, you should replace `DHT11` with `DHT22`.

```
sensor = dht.DHT11(Pin(14))
```

We then take a reading from the sensor with:

```
sensor.measure()
```

From that, we create two variables with values set to the temperature and humidity readings.

```
temp = sensor.temperature()
hum = sensor.humidity()
```

Finally, we convert the readings into a string for ease of reading, then print them to the Shell pane.

```
readings = ("Temperature: {}°C   Humidity: {}%
".format(temp, hum))
print(readings)
```

Here, the `{}` represent placeholders to put the data, which is the `temp` and `hum` variables in the parentheses after the `format` method. We've left the placeholders empty, so the variables are taken in order, but you have the option of specifying the variable for a placeholder. You could also set the number of decimal places in the placeholder, e.g. with `:.1f` for one place, but since the DHT11 only gives whole number values, it's not needed here. Run the program and see the reading – we haven't used a loop, so the program ends. Try blowing on the sensor, then run the code again and see the difference in values. You could extend the project by using a mini LCD screen and using a loop to repeatedly update the readings shown on it.

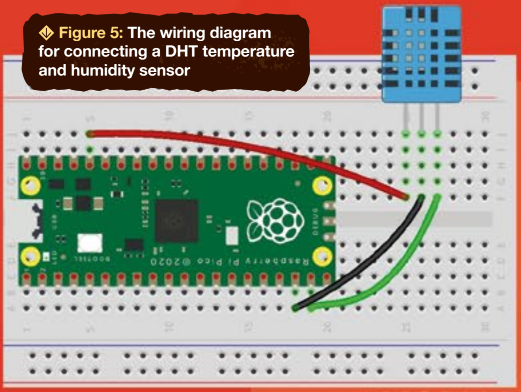
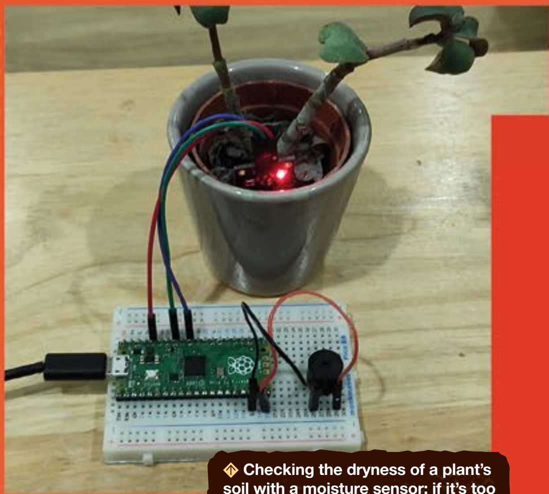


Figure 5: The wiring diagram for connecting a DHT temperature and humidity sensor



Checking the dryness of a plant's soil with a moisture sensor; if it's too dry, a buzzer alert is sounded

Plant monitor

For this project, we'll use an analogue moisture sensor. Unlike a digital sensor, this sends a variable voltage from its signal pin. To read this, we can use one of the ADC channels built into Pico's RP2040 processor. Most microcontrollers have similar analogue inputs; if you were using a Raspberry Pi computer, however, you'd need to add an ADC chip to the breadboard circuit to convert the reading to a digital value.

Connect the moisture sensor to Pico, as in **Figure 6**. Vcc is connected to Pico's 3V3 pin, GND to GND, and AOUT (analogue out) to GP26, which is one of the pins connected to an ADC channel on Pico – the others are GP27 and GP28. Wire the piezo buzzer to GND and GP15 – note that this will only work with an active buzzer, not a passive one.

Along with `Pin`, we import the `ADC` method from the `machine` module; this enables us to take an analogue reading. We also import the `sleep` method from `utime`.

> *"If the voltage falls below 500 mV, it then triggers an alert"*

```
from machine import ADC, Pin
from utime import sleep
```

We then define a `moisture` object set to the ADC channel on the GP26 pin, along with a `buzzer` object as an output on GP15.

```
moisture = ADC(26)
buzzer = Pin(15, Pin.OUT)
```

We create a `while True:` loop to take a reading every two seconds and print it to the Shell.

```
while True:
    reading = moisture.read_u16()
    voltage = 3300 * reading/65535
    print("{:.1f}mV".format(voltage))
    sleep(2)
```

As in our RGB LED project code, the `_u16` specifies that the converted digital value will be an unsigned 16-bit integer. To convert this to the voltage in mV (microvolts), we multiply it by 3300 and divide by the maximum 16-bit value of 65535.

Run the program to see the voltage reading to one decimal place. Try holding the sensor prongs in one hand to see it increase, since you are conducting electricity between them. Wet your fingers and it'll rise much higher. Now place the sensor's prongs in the soil around your plant.

Next, we'll add an `if` conditional statement to the loop: if the voltage falls below 500 mV (or whatever threshold you want to set), it then triggers an alert for dry soil, reminding you to water the plant.

```
if voltage < 500:
    for i in range(5):
        buzzer.value(0.5)
        sleep(1)
        buzzer.value(0.5)
        sleep(1)
    sleep(30)
```

Here, we're using a `for` loop which repeats a set number of times: five in this case. In each iteration, we turn the buzzer pin on and off every half a second. We then wait 30 seconds before taking another reading.

To extend this project, you could trigger a different action. If you have a Pico W, or another microcontroller with an internet connection, you could send an email or other type of notification. You could even set up a relay switch to trigger a pump to water the plant. →

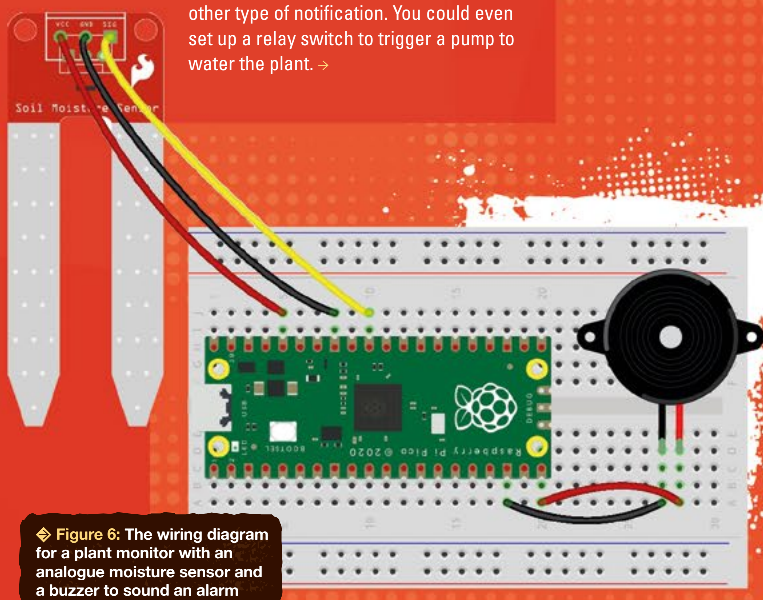


Figure 6: The wiring diagram for a plant monitor with an analogue moisture sensor and a buzzer to sound an alarm

MOTORS AND SERVOS

ADD MOTION TO YOUR PROJECTS BY CONTROLLING MOTORS WITH CODE

YOU'LL NEED

- Raspberry Pi Pico with soldered headers and MicroPython installed (see hsmag.cc/PicoMicroPython)

PROJECT 1

- Dual H-bridge motor driver, e.g. hsmag.cc/MotorSHIM
- 2 × Micro metal gear-motors (with suitable connectors, e.g. JST-ZH)
- Chassis and wheels (optional)

PROJECT 2

- Breadboard
- SG90 micro servo 9g
- 1 × 330 Ω resistor
- Male-male jumper wires

If you want to make the wheels on that robotic vehicle go round, you're going to need some motors. We'll explore how to create code to control them, along with the precision angular movements of a rotary servo motor.

Drive motors

Motors have many uses, but here we'll use them for a simple two-wheeled robot. To be able to drive each motor in forward or reverse, we'll need a dual H-bridge driver which can reverse the polarity – depending on which of two connected GPIO pins we output a signal from.

In this example, we're using a Pimoroni Motor SHIM attached to a Pico, but the basic principles of running motors is the same for any controller. You can run them forward or backward, and adjust their speed using PWM.

For the Motor SHIM, you'll need to download Pimoroni's MicroPython firmware (**hsmag.cc/PimoroniPicoUF2**) to use the firm's Motor library – which works with any RP2040 controller. We'll start by importing the **`utime sleep`** method and various classes from the Motor library (**hsmag.cc/PimoroniMotorLib**).

```
from utime import sleep
from motor import Motor, pico_motor_shim
from pimoroni import NORMAL_DIR, REVERSED_DIR
```

We then set a main driving speed constant and define objects for the left and right motors. In a standard two-wheeled robot setup, one motor will have a reversed direction (as it's mounted at 180° to the other). We also set the **`speed_scale`** for each motor – if the robot doesn't move forward in a straight line, which is fairly common as not all motors run exactly the same, you can calibrate it by adjusting these values.

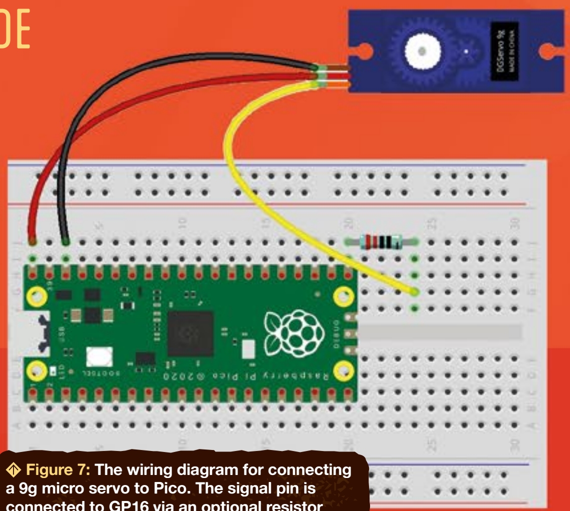


Figure 7: The wiring diagram for connecting a 9g micro servo to Pico. The signal pin is connected to GP16 via an optional resistor

```
DRIVING_SPEED = 5.4
left = Motor(pico_motor_shim.MOTOR_1,
direction=NORMAL_DIR, speed_scale=3)
right = Motor(pico_motor_shim.MOTOR_2,
direction=REVERSED_DIR, speed_scale=3)
```

Next, we define some functions for moving the robot forward and back, by running both motors together at a positive or negative speed.

```
def forward(speed=DRIVING_SPEED):
    left.speed(speed)
    right.speed(speed)

def backward(speed=DRIVING_SPEED):
    left.speed(-speed)
    right.speed(-speed)
```

To make the robot turn sharply left or right, we run one motor forward and the other in reverse.

Control a servo

A servo is a special motor with a feedback element so you can adjust its rotary or linear position – ideal for when you need precision, such as in a robotic arm. Since you don't need to reverse the polarity to operate a servo forward and backwards, there's no need for an H-bridge. You can connect it to Pico (or other microcontroller/SBC) directly and use a PWM output from a GPIO pin, varying the duty cycle to control its angle/position.

We're using MicroPython to control a rotary servo. As in **Figure 7**, the servo's red wire is connected to 5 V, brown to GND, and yellow (signal) to GP16 via a 330 Ω resistor (an optional safeguard). Attach a plastic horn to the servo; you can calibrate its position later. You can attach another part of a project to the horn using tiny screws or even hot glue.

At the start of the code, we import the `PWM` and `Pin` methods, along with `sleep` from the `utime` module.

```
from machine import Pin, PWM
from utime import sleep
```

We define a `servo` object to use PWM on the GP15 pin. We set its frequency to 50Hz, which equates to 20 milliseconds, or 20,000 µs (microseconds), per pulse.

```
servo = PWM(Pin(16))
servo.freq(50)
```

We create a function to set the servo angle, based on the duty cycle of the PWM signal. We obtain the latter (rounded to an integer) by dividing the `angle` input variable by 180, multiplying that by 2000 (for the range of pulse length values in µs), adding 500 for the minimum pulse length, then multiplying by 65,535/20,000 – the maximum 16-bit duty cycle value divided by the number of microseconds per pulse.

```
def set_angle(angle):
    duty = int((2000 * angle / 180 + 500) * 65535
/ 20000)
    print(angle)
    servo.duty_u16(duty)
```

To calibrate the servo, add the following line and run the code. It should cause the servo horn to point straight up (90°) when the servo is laid flat on a surface. If not, remove the horn and reattach it in as close to 90° as possible.

```
set_angle(90)
```

For this simple demo, we'll now use a couple of `for` loops to change the servo position in steps of 10, from 10 to 170 degrees, then 170 to 10. Since many servos of this type struggle to move the full 180 degrees, we're limiting it to this range.

```
for angle in range(10,170,10):
    set_angle(angle)
    sleep(0.5)
for angle in range(170,0,-10):
    set_angle(angle)
    sleep(0.5)
```

Stepper motor

An alternative to a servo is a stepper motor. Whereas a servo uses feedback to control its position, a stepper motor simply moves a precise number of steps. These are determined by the digital outputs from four GPIO pins via a stepper motor controller board; turning each output on in turn moves the motor a step.

```
def turn_left(speed=DRIVING_SPEED):
    left.speed(-speed)
    right.speed(speed)

def turn_right(speed=DRIVING_SPEED):
    left.speed(speed)
    right.speed(-speed)
```

We can turn it more gently by stopping one motor while running the other one.

```
def curve_forward_left(speed=DRIVING_SPEED):
    left.speed(0.0)
    right.speed(speed)

def curve_forward_right(speed=DRIVING_SPEED):
    left.speed(speed)
    right.speed(0.0)
```

For stopping, you can either use the library's `coast` method or the more abrupt `stop`.

```
def stop():
    left.stop()
    right.stop()
```

Those are the basic methods for controlling the motors for a robot vehicle. You can call them in a sequence for a set pattern of movement, with `sleep` delays to set the length of operation, such as in this example:

```
forward()  # Full speed
sleep(0.5)

forward(0.5 * DRIVING_SPEED)  # Half speed
sleep(1)

backward()
sleep(1)

turn_left()
sleep(1)

turn_right()
sleep(1)
```

For a more autonomous robot, you can use `if` statements to determine movement based on external inputs such as sensor readings – an ultrasonic distance sensor may be used for obstacle avoidance, or a bottom-mounted infrared sensor to follow a black line along the floor. □

Untethered robot

To run your robot untethered from a computer, you'll need to use battery power, such as a phone charger. You will also need to save your program as **main.py** so it runs automatically as soon as Pico is powered up.

NEW INTERNATIONAL OFFERS!

Get 6 issues plus a **FREE Raspberry Pi Pico W** for just...

£35 USA and EU

£40 Rest of the world

Head to **hsmag.cc/subscribe** to get yours today!

FEATURE

HOW I MADE

By Jo Hinchliffe

UP, UP, AND AWAY!

Take to the skies with an unpowered flying machine

In Roald Dahl's *Danny, the Champion of the World*, there is a small, almost insignificant, side plot, where Danny and his father make a small hot air balloon. It's been on our list of things to make pretty much ever since. Times change, and it's now not a good idea to launch a flaming ball untethered into the skies, as it's at the very least littering and, at worst, is going to cause a fire. Sky lanterns have now been banned by numerous local authorities, but there is still heaps of fun to be had. With that in mind, we jumped right in!

We'd seen some ideas where aerostats, or balloon canopies, were flown in smaller 'hops', heating the internal air using a device on the ground enabling the balloon to perform a shorter flight. For outdoor hops, people are using camping stoves with a tin can chimney to keep the flames off the tissue or, for indoor flying, people were simply using an electric heat gun. This

What I used

- Glue sticks
- Scissors
- Sheets of tissue paper
- Paper to make a pattern
- Cotton thread
- Thin light wire, (MIG welding wire or similar)
- Heat gun
- Long ruler/straight edge
- Large set square

Figure 1 ↑
The CDAS-2 balloon filled with hot air using an electric heat gun, ready to take flight

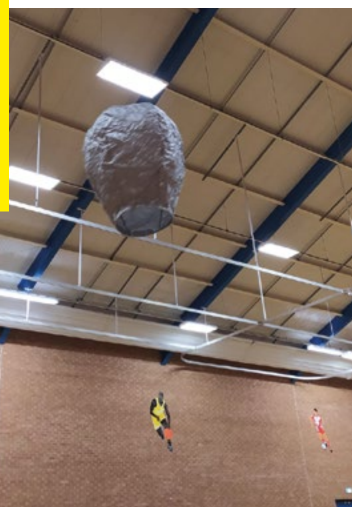
seemed like a safe and easy-to-achieve goal, so we set about making a small balloon example, which quickly escalated to larger balloon designs. We had the idea that we'd like to be able to fly an 'FPV' aerostat, where there was enough lift to haul a small camera, video transmitter, and battery up under the canopy.

As a starting point, we made something around the size of a sky lantern (**Figure 2**). As tissue paper is relatively affordable, it's a pretty good idea to do something small first, as the techniques and gluing sequences are the same on a smaller balloon as they are on the later, bigger designs. In fact, the whole process is similar. You'll need a nice clear table area to work on, and it's worth running your hand over it all to check there are no little bumps or sharp bits that might snag or rip your tissue paper. A bench built out of an old interior door is a great work surface for these projects!

For all the balloon designs, we created a pattern on some brown packing paper, and we've collated all the plans for all three designs as PDFs that you can download at **hsmag.cc/issue61**. All three of our balloon designs are made following the same process, so sometimes we've used pictures from the different builds to illustrate different points. Start by drawing a centreline along a piece of paper to the overall length and then, using a ruler, mark this line at the regular intervals indicated on the plans. For the small balloon, this is every 12.5 cm. Next, use your set square or roofing square to draw a line out from each mark, matching the distances with those on the relevant plan. To keep the patterns simple to draw, we just drew straight lines between each section to create the outer perimeter of the pattern. Having drawn the pattern outline, cut out the pattern.

> **"YOU'LL NEED A NICE CLEAR TABLE AREA"**

Figure 2 ↓
Our first small test design was capable of short hops for a few seconds



A common tissue paper sheet size is 76 cm × 50 cm, and so, this smaller balloon is made by placing the pattern at an angle across the tissue sheets. The larger designs are made from six panels, where each panel is a pair of sheets glued together, so uses twelve sheets in all. Neatly stack six tissue sheets together, and then place the pattern on the top (**Figure 3**). Next, you pin all the →

Figure 3 ↑
For all the balloon designs, we cut the gore panels by pinning the pattern to a stack of six sheets of tissue paper

QUICK TIP

It's very handy to have a desktop fan around as you can use this to blow into the canopies to temporarily inflate them.

way through the pattern and all the sheets in numerous places to keep the pattern and tissue stack all together. Pin close to the edge of the pattern so that pin holes will be folded into your seams. With the pattern and the sheets pinned together, cut around the pattern, cutting all six sheets at once. Don't get too hung up on being incredibly precise through this process – a few millimetres of wavy cut won't make a significant difference to a large balloon construction. As you make your second or third design, you will naturally be better at the process as you refine your techniques.

Before you unpin the panels, just make a mental note of which end is which and, regardless of which design you are building, keep track of which end is where throughout the process. Place the stack away from your workspace, and clean down your bench, ready to begin the assembly process of the gores. This is a complex sequence to describe, so we've also created a short online video tutorial that also explains the process, and you can find it here: **hsmag.cc/GoreAssembly**.

Take a pair of gore panels and place them on the bench with the base end of the gores to the left-hand side. Move the uppermost panel away from you to expose a 10–12 mm edge of the lower panel – this will be our seam (**Figure 4**). Add glue to the closest edge of the upper panel, and then carefully fold over the exposed 10–12 mm seam onto the upper panel and press it down. Once this is done, repeat this with your other cut panels, assembling three pairs.

Stack the three pairs so that they are again base to the left. Pick up the entire stack and rotate them over horizontally so that all the base ends are now on your right and the apex is now on the left. The glued seams should still be at the closest edge of the bench nearest to you. Reach over to the far side of the uppermost panel pair, and move the first single sheet back, folding it over so that you can access the lower sheet of the upper pair (**Figure 5**). Pull the entire upper pair back towards you, around 10–12 mm, exposing the uppermost sheet of the middle pair. Glue the edge of the lower panel of the upper pair and fold towards you the exposed edge that we just created from the upper layer of the middle pair to create a seam (**Figure 6**).

Next, fold the entire seam you just created towards you so that we see the lower layer of the middle pair. Again, pull the upper pair and middle pair towards you by 10–12 mm, which will reveal the upper layer of the bottom pair. Add glue to the edge of the lower layer of the middle pair, and then fold over the upper layer of the bottom pair to create another seam.

Finally, move the seam you just created back towards you and take the very top

Figure 5 ⬗
Having created three pairs of panels, we move on to joining the panels together

Figure 4 →
Creating a seam by offsetting one panel slightly, gluing, and then folding the lower exposed panel over the top

layer of the uppermost pair and move it over all the other layers, to be placed on top of the lowest layer which is on our work surface. Move the upper layer back towards yourself, exposing 10–12 mm of the very bottom layer. Apply glue to the top layer and fold the lowest layer over the top towards you to create the final closing seam.

If you have done this correctly, then all the seams you have created should now match when the balloon is inflated.

You can now open out your canopy and check the seams. Using a desk fan can make this easier – on a low setting, hold the base opening of the canopy open near the fan for it to inflate (**Figure 7**).

Don't worry so much that the top of the small balloon is a bit messy and possibly not fully sealed. For all the balloon designs, we created some kind of closing panel and applied it. For this small one, it's really straightforward. Cut a small disk around 10 cm in diameter of tissue paper, and glue it over the top of the balloon. You can use your hand inside the canopy to press it all together. For the larger CDAS-2 design, seen in **Figure 1**, we created a separate collection of panels to seal the apex of the canopy, which we describe later in this article.

To finish the small example, we need to create a canopy ring. This holds the base aperture open at all times and makes it easier to inflate and fly. For the small balloon, our canopy ring was roughly 22 cm in diameter, and used a single piece of 0.8 mm MIG welding wire, though any thin, stiff, and light wire would do. For the larger CDAS designs, the canopy rings are around 40 cm in diameter. For all the balloons, it's worth making the circle temporarily with some excess available, and test fit it into the canopy, adjusting to each balloon. When you are happy with the size, use a small amount of tape and superglue to create the joint. Insert the ring into the canopy and move it back towards the opening at the base. Position the ring around 25–30 mm inside the canopy. This 25–30 mm seam then gets cut into small tabs that we can glue over the ring (**Figure 8**). Once you have a few tabs holding the ring in place, you can move around the complete seam gluing tabs over the ring.

Once everything is dry, you can test-fly your creation! With a balloon similar in size to the small design, you don't really need to tether it as, in our experience, it will only fly a pretty short hop. Heating the canopy with an electric heat gun, we've found ours will hop the height of a tall room, and pause only for a second or two, before floating back down. It's a perfect project for a lighter-than-air demonstration at a school/youth group.

There was no way our small balloon was going to lift anything, so we knew we had to go bigger! We went on to design two more larger balloons that we called CDAS-1 and CDAS-2 (ConcreteDog Aerostat -1 and 2), and both perform much better. We suggest that you might consider jumping to CDAS-2, as it had numerous improvements over the CDAS-1 design, looks great, flies very well, and, in some ways, is easier to make.

For both the CDAS designs, you need to again cut six gore panels from a pattern. →

Figure 6 ↑
Creating the second set of seams working on the far side of the panels

"YOU DON'T REALLY NEED TO TETHER IT"

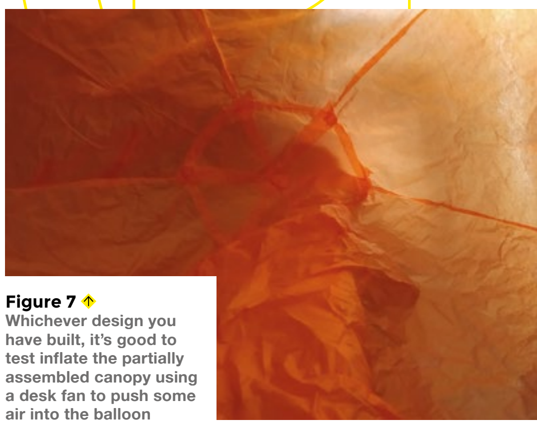


Figure 7 ↑
Whichever design you have built, it's good to test inflate the partially assembled canopy using a desk fan to push some air into the balloon

Figure 8
We used the same technique to create and install a canopy ring for each of our designs

These panels are a lot larger though and, as such, they are each cut from two sheets of the 76 × 50 cm tissue paper stuck together lengthways. The first task, therefore, is to stick together six pairs of sheets. We used a glue stick for this, and overlapped the sheets by around 15 mm. Once your six larger sheets are dry, stack them up, and pin the pattern to the top of the pile. A couple of notes here are to make sure that the sheet seams that you just glued are all aligned, so that you get a nice straight seam around the canopy, and, if you are making CDAS-2, make sure your sheets are stacked accurately, as the gore panel design fills to nearly the edge of each sheet.

You now need to follow the same process as you did for the small balloon to cut the panels in a stack. Glue the first set of seams, forming three pairs of panels, and then assembling those panels together into the canopy. You'll notice that both the CDAS designs don't come to a point at the apex end of the pattern. For CDAS-1, we experimented with adding a larger disk to cap the top of the canopy. We ended up cutting a disk around 20 cm in diameter

QUICK TIP

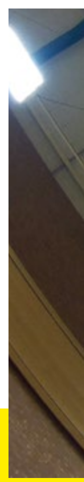
If you snag your canopy and cause a tear, or notice a slight gap in a seam, it's easily fixed with a glue stick and perhaps a small tissue patch.

Figure 9
For the larger CDAS-2 design, we added six smaller triangle panels to create the canopy apex, using an inflated bin-liner inside to act as a form whilst gluing

Figure 10
Adding a thread loop, stuck on with a patch of glued tissue paper, is a handy way to store and hold your balloon

and stuck it to the apex of the canopy. Our approach was to stack some small boxes on the floor and balance a large plastic mixing bowl upside down on top of the stack, and then place the canopy over it, the idea being we could then use the bowl as a surface to press onto and form the curved canopy top. Whilst we managed to achieve this, it's not pretty or particularly easy to achieve. With CDAS-1 together, we added a canopy ring in exactly the same way as we did for the small balloon. We again performed a test-hop in a small room, and you could instantly tell this volume of balloon had a lot more lift. However, we didn't really like the shape of the CDAS-1 canopy – it's a bit too pointy in the top half, and we knew we could do better. One thing we did add to CDAS-1, which is now something we'd add to all our balloons, is another small patch of tissue to the top of the canopy with a double loop of cotton thread sewn and knotted through it. This has two functions; one is you can hang the canopy up to store it, but you can also have an assistant hold the canopy upright, which helps a lot when beginning to fill them (**Figure 10**).

For CDAS-2, we again glued pairs of sheets together but, this time, we created a pattern that tried to use most of the width of the sheet, so we got more air volume in the final canopy. We also went for a different approach to seal the apex which is slightly heavier perhaps, but actually adds more volume to the canopy as well. The apex ends of the gore panels are pretty wide and, instead of a disc to seal it, we created and cut 6 × 25 cm-sided equilateral



triangle panels. With the canopy assembled, we glued one edge onto each of the panels at the apex edges, again using around a 15mm overlap, and making sure that the triangle panels equally overlapped at the edges of each of the main canopy panels. We then, after some head-scratching, inserted a bin-liner into the canopy and blew this up like a balloon and sealed it with some tape. This bin-liner balloon gave us a similar shape and enough resistance to then glue the overlap seams of each triangle to each other, moving around each of them in turn (**Figure 9**). This worked well enough that we only needed to add a small, 8cm diameter circle to fully close the apex of the canopy, and we used this circle to add our thread loop accessory. We finished the CDAS-2 balloon with a canopy ring, and we then used a vinyl cutter to make some small lightweight decals for both CDAS-1 and 2!

There's an excellent model indoor flying group locally, and they agreed that we could fly the balloons in the sports hall that they meet in once a month. This gave us the opportunity to test the balloons up to around 18 metres in height. We needed to tether the CDAS-1 and 2 designs, particularly as there was a lot of metalwork and gantries in the roof of the hall, and we wanted some control! To simply add a tether, and also a structure to attach a tiny FPV camera rig, we sewed a double thread across the canopy ring, knotting it on each side. There isn't masses of force or jerking under acceleration with these balloons, so this proved strong enough. We used a lightweight kite line and a tiny fishing swivel which allows the tether to be clipped on and removed. Both the CDAS-1 and CDAS-2 designs flew very well (**Figures 11 and 12**), and could easily reach the roof of the building and hang there for many seconds before slowly floating down. They are both capable of lifting the tether line and our tiny, low-resolution FPV camera/transmitter. The FPV rig is an Eachine TX06 camera and VTX (video transmitter), and we found a 100 mAh LiPo cell. In all, this weighed around 7 grams, (**Figure 13**). It certainly proved the FPV aerostat concept, but we're planning an upgrade in quality as a future goal. Finally, it's worthy of note to say this is great fun! It's cheap, affordable, and creates a large object that flies well. It's certainly cheap enough that we are pondering going even bigger with our next design!

Figure 12
We were really pleased with the shape that the CDAS-2 design takes when filled and in flight

"IT'S WORTHY OF NOTE TO SAY THIS IS GREAT FUN"

QUICK TIP

Filling these balloons with a hot air gun is an excellent approach, but you must be very careful to not touch the tissue, as it will easily scorch and burn.

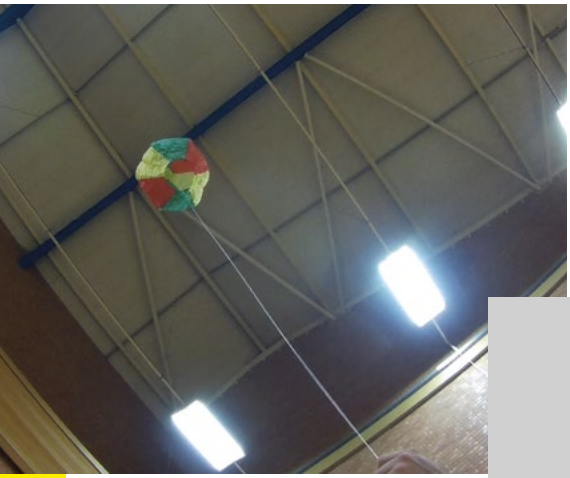


Figure 11
The CDAS-1 design around ten metres up, flying well

Figure 13
CDAS-2 at launch with its tiny FPV camera, video transmitter, and battery payload

HackSpace magazine meets...

Pooch

From the coalface of 3D printing

Alan 'Pooch' Puccinelli: is a 3D printing geek who put his money where his mouth is, and is now making a living from 3D printing via his company, R3PKORD. He's also full to the brim with infectious enthusiasm: we spoke to him when we were full of cold on a grey November evening, and came away from the conversation buzzing with how awesome 3D printing is right now, and how much better it's going to be in future. We hope you do too! →

HackSpace Hi Pooch! First of all, let's have some background: how did you get into 3D printing?

Pooch Like a lot of people, I am a maker hobbyist, and always interested in fun and new and exciting tools that are out there. Around 2015, I started hearing about 3D printing, got curious, and ended up getting my own printer, and got bit by the bug.

You can take something from your own mind and make it tangible so readily, compared with any other tool out there. It's not that it's the best tool for every job, but it's very accessible. And as they become cheaper and cheaper and new tech comes along, we see more and more growth in the space. That's the 10,000 foot view of how I jumped in.

And it wasn't before long that I thought I could maybe make some money with it. It started as a side hustle, and pretty quickly, my little passion project turned into something viable as a way to make a living.

HS So there must have been a market out there for the sort of itches that you wanted to scratch for yourself?

P Absolutely. Isn't that what it all comes down to? We find the things that we love in life and then we try to find ways to legitimise them. I remember saying, 'This 3D printing stuff is great. I'll start a little business, and that'll pay for the printers'. Before long, I started sourcing my own filament and created a brand.

I want to be able to work with my hands. I want to be able to create, and get validation from my ideas. There's nothing more rewarding to me than that feeling of people actually paying for your stuff – that you've had an idea that's so good, it's worth people spending their hard-earned money on it.

HS What sort of stuff do you produce?

P Our flagship product is something called the RepBox, which is one of those funny things where I don't think I really expected much to come of it. I just had a ton of filament and I wanted a nicer way to store it.

It was really born of a need for organisation. I had a partner I was working with at the time, and we both had a similar problem. We were in social media and building and 3D printing and stuff like that and had a lot of filament, and we thought it would be nice if we had a better storage box.

And so it started with storage, then we thought it'd be really great if we could print directly from this box. And so this enclosure became that. And then, before long, people were like, 'Oh, well, if you can make it seal better, it can protect filament from moisture and dust and all that other stuff'. So it's become this interesting amalgam of a number of thoughts and evolutions over time. And when Prusa launched the Multi Material and the Palette and stuff like that came out, people were realising how much space it takes up to try to feed all these little filaments into a different printer. I think that's really where a lot of people realised that we had a nice, elegant solution.

> "My little passion project turned into something viable as a way to make a living"

Now we're doing OEM versions for Prusa and LulzBot® and Printed Solid and other companies as well. We must be doing something right, because we keep selling them.

HS And you have a version of that with lights on it. There's a very pleasing maker tendency to put lights on things.

P There really is. You know, it's funny, because, again a lot of people look at it, and they go, oh, you know, I have questions about your dry box. And it's not a dry box; it was never meant to be a dry box, it was meant to be more of a practical aesthetic.

And so the Lit Kit was kind of leaning into a lot of that. And it just looks really cool, right. So people want to have it up in the background of their videos. And I just love it. I love working with acrylic, and doing edge-lit acrylic and the effects that you can get with that. And really good LEDs are so neat.

HS You've mentioned Prusa already and multifilament printing already. How else have things improved in the 3D printing world since then?

P Oh, man, how haven't they improved? I mean, machines have gotten faster and more reliable, as well as less expensive. If you're looking at a traditional S-shaped adoption curve of there being a bunch of excitement about an industry, then a lot of companies go out of business, and then there's a slow climb back up. I feel like we're coming back out into that slow climb where the players are starting to get pretty established at this point. We're seeing more and more people come into the space, we're seeing it be more affordable, and we're seeing education latch on to it as school curricula get better.

And it's that combination of all those things that makes the community richer, the companies that support it richer, and there's also this interesting segmentation of the hobbyist and prosumer and industrial, and I love all of them. I was kind of born of the hobbyist. But as I go to more of these industrial conferences, that's where I like to try to find the things that we can bring back to the hobbyist space to drive that industry forward in a more meaningful way.

There's a good, vibrant, rich community around 3D printing. Not a day goes by that I don't meet new people that I'm just excited to get to interact with and brainstorm with and all that. →

Above Need a new nozzle? R3PKORD's got you covered

Above
Pooch started his business in 2015 as a side hustle. It's grown a bit since then…

It's funny, I'm literally sitting next to one of the original replicator minis right here, whose heyday probably would have been right around when I got started. 2015, 2016-ish. Maybe a little before that. [There was a tendency to think of the] 3D printer as an appliance that would be as easy to use as a microwave, that's going to be like the replicator that we see from *Star Trek*. I don't know that it'll ever be exactly that.

Just like the tools out in your workshop, your bandsaw, your drill-press, and all that stuff: you can do a lot of really neat stuff with it. But nobody's ever talking about when is a drill-press or a bandsaw going to be an appliance, right.

There's this misconception that 3D printers will be able to do that in the future – maybe in the distant future, when we can start manipulating atomic structures and stuff.

I am really big for advocating the mindset that it is a toolset. And it is a very incredibly capable toolset that empowers some really interesting designs and things that you can't make any other way. And the other thing I'm really passionate about is it enables you to create products and businesses around it in a way that you don't get with a lot of other toolsets.

HS How do you mean? Is that because it's open-source and people like to share their ideas?

P It's because you can actually create a remarkably complete product. It's one of the first toolsets that I've come across in my life where you can take an idea, sketch it out in CAD, and then hold it in your hands in a reasonably short period of time, unlike a lot of other things where you have to do a lot of other processes to get it to be complete.

I mean, we're already seeing this. The collision of being a maker, and being a maker business is really fascinating to me because it combines two of my big passions. I've seen people make some incredible companies out of something very simplistic by creating, like, cookie-cutter businesses, you know, where they're just doing seasonal 3D-printed cookie cutters. They're creating these little niche communities inside of the baking community, or the drone builders, or the foam dart people.

And we're talking about multimillion-dollar companies at some point that you wouldn't think are terribly sexy or glamorous, but they're leveraging this toolset in a way that I get excited about, because I want people to not think of the 3D printer as that useless little box that's printing a bunch of plastic junk that ultimately everybody throws away. We can make practical things that can potentially support us and create a new form of manufacturing. Which, you know, if you want to go into the face shields, thing like that, one of the first things where you really started to see the power of that distributed manufacturing. I've been chasing after that, I think, ever since.

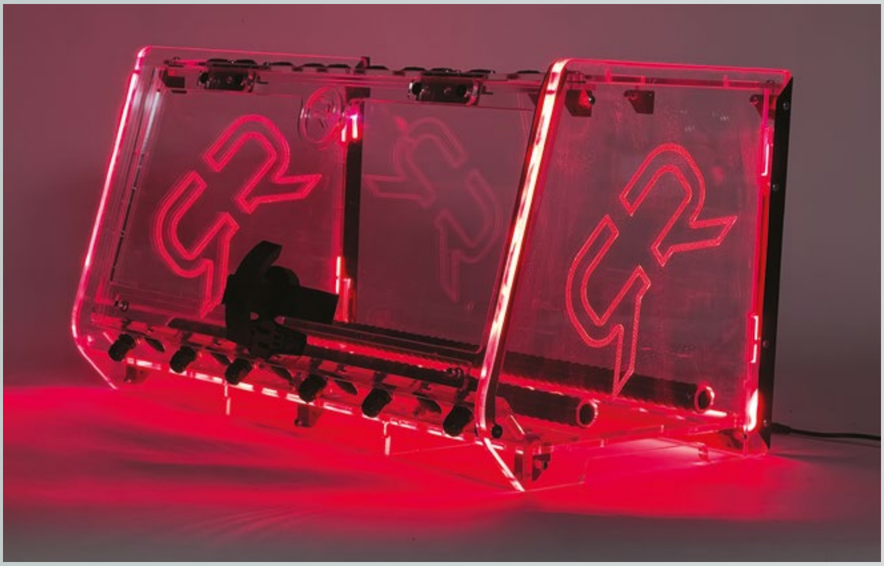
HS You mention objects that get you excited about 3D printing. Would it be unkind to ask you to pick a couple of favourites?

P It wouldn't be unkind. It's just that there are so many!

If, for example, when I've gone and shown off some machines in a school, I'll bring a bunch of models and it'll be like links of chain, or the ball that's inside of the cube; some very simple things where you hand it to somebody and you say, how was this made, and they're trying to piece together how you could possibly get the ball inside of the cube without having to cut it, without seeing a seam and glue and all that stuff. I love little simple examples of the potential of additive manufacturing. I think that's the little base thing that hooks people.

HS You mentioned face shields. It seems a long time ago now, but can you tell us much about the Shields Up project, and how you found yourself making PPE during the pandemic?

P I think, like most, when Covid-19 hit, I was at home looking for some way to help. →



Above The RepBox Lit Kit, for all your edge-lit acrylic filament storage needs

I had literally just gotten out of my garage and said, OK, I'm gonna go full-time with R3PKORD. As a starting point, I'd just got some co-working space and met up with a lot of great makers in this maker space. I had a lot of like-minded people around me who had time because they weren't able to go in to work either. And when I saw what Prusa was doing with the face shields thing… I'm always looking for what can potentially have the most impact, like who's doing something that I can see scaling up or replicating rather than reinventing the wheel.

And there were a lot of interesting projects out there, but I saw that Prusa had gotten a good amount of attention going with what they were doing.

And I thought, well, that's simple. The part that I think was a challenge for a lot of people was cutting the lens piece of the shield out; the 3D-printed frame was easier. But it was the fact that I already had a plastic supplier, and I already had the supply chain secured for materials that I was using for my other stuff. And when I called up my supplier and said, 'Hey, can we get this polyethylene stuff that I need?', I was lucky enough to basically be able to snag a pallet's worth of that material.

And then a groundswell came up immediately when a bunch of the people in the maker space I was with put themselves forward to help. Before we knew it, we were operating out of that maker space and another one and then a church, so we had like three locations. It became this very interesting process control thing where we had to worry about disinfection and all these other things.

I sent out a tweet saying, 'here's what we're doing. If you're interested in helping, help me print this thing, and send it to us here, because we're trying to make as many of these face shields as we can'. And there were some contacts that I had made through this maker space that were plugged in with some pretty high-up individuals in our local health care system. They basically already had a lot of the contracts with the hospital systems. And so, by partnering with them, they already had the right context to say, 'hey, we've got these face shields'. And at the time, nobody could get any of this stuff – everybody was trying to get as much PPE as possible. So like I said, it was a perfect storm of just right people, right space, right time, right tools, right materials, and just doing it. And when people saw that, we were actually making headway. That's when we got news coverage, donations, substantial donations, I think we ended up raising north of, we're gonna say $120,000, and we made more than 36,000 face shields.

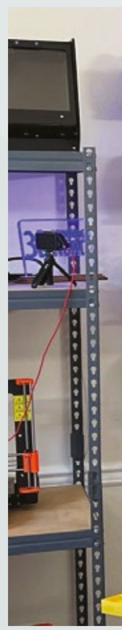
Compared to Prusa, we didn't make nearly as many as they did, but for my little small rag-tag army of makers and the support that we had, it was amazing. And it was an amazing proving ground for this whole concept of distributed manufacturing, where it's like, you can print this at home and send it here, and we were able to scale that to that number in three months – we went for about six months in total, but the core of the production was probably about three months. It was a bit of a blur at this point.

HS Now that is awesome. You couldn't build a factory in that time, and certainly not for $120,000.

P Not even close. And we were like, I mean, we only had one product, but a lot of it came down to the logistics. I always thought it was a logistics problem; not a making problem. We were successful because we had supply and distribution. There were loads of little questions, like, how are we going to package this? How are we going to put together assembly guides for the user, or give the hospitals that were getting them the confidence that they were going to be sterile? And so we had to have messaging around how we use these materials, that we've sterilised these to the CDC standards and what have you, so the hospitals were confident that it was not just some crummy thing that was made somewhere in a potentially dirty environment.

HS There is an extremely strong sense of friendliness and helpfulness, I think, about 3D printing people.

P I think that's the maker spirit in general. Yeah, 3D printing is an interesting subset of that. I love that I get the chance to go to a number of different Maker Faires and RepRap festivals; that's kind of where my roots are. Getting to go to some of these cons and things like that, where you can see people that are yes, leveraging 3D printing, but then doing just these amazingly creative things out of foam and electronics and all of this other stuff is like, that's my vocation, that's my happy place… being able to get to go to those and to be able to legitimately justify it as a business expense, you know, to get to go to these things.



HS What should we be looking for in the future of 3D printing? What's coming down the rails, do you think?

P There's a number of incredibly exciting things in 3D printing that are coming up that I am particularly fired up about. There's the introduction of AI: we're starting to see that. This is really fascinating to me, because if you've seen any of the two-dimensional AI artwork, where you can feed it text, and it'll start giving you imagery and stuff like that – I mean, just imagine what the power of that is when you can start to generate three-dimensional models, when you can prompt a computer, like 'I need a gear that has x, y, and z'. Right there, without the ability to do CAD, you can start to create unique, interesting products.

As a toolset, part of the challenge with 3D printing right now is that it still requires you to have some skill for the input, right. Yes, you can go and download things that other people have created for you on Thingiverse, Printables, and whatnot. And we're seeing more and more robust libraries there, which is really exciting, you know, better utility, searchability, and all that stuff. But if you don't know how to model CAD, you're kind of at the whim of whatever someone else is providing. If we want to see this thing really take off, giving people more toolsets to make this a usable tool to them means that we have to do more training on computer-aided design for people in schools and whatnot.

I love seeing the crossover in the gaming industry – it's like watching kids start to see the value of learning CAD, whether they want to be a drafting engineer, or if they want to be in game design, or in cinema, or anything in between, is an incredibly, incredibly desirable skill set moving into the future. But anyway, AI is exciting.

We're starting to see multi-axis systems. So right now, we're not even leveraging the power of 3D: we call it two-and-a-half D [2.5D], because it's layering up as we go. But you have to do a layer at a time and then you move up, right, and so when we start to see four-axis, five-axis machines where we can get geometries that aren't currently possible. We're really limited still by the software that supports the whole thing. And so when I see systems coming into place that support new software, like, whether it's AI or whether it's human-generated for enabling this, that's exciting.

Materials science! I mean, some of the materials that are becoming available now – foaming PLAs like polyurethanes, for example. We're starting to see that used in shoes – I have a pair of shoes with soles 3D-printed by Carbon 3D with this incredible lattice structure; they're the most comfortable shoes I've ever had. That's a fantastic example of mainstream industry, Adidas, partnered with Carbon. And it's something that I can show people. I'm like, here, this is 3D printing; I'm wearing it on my foot right now. And they start to see that and realise that you couldn't make it any other way.

Conductive filaments, where we can start to crossover electronics in a different way. Materials science is getting a real blast from all of this too. There's a lot of support from industry. And then, I guess the other thing is automation. So I'm really into belt printing, because I'm in a very expensive labour market. And the more I can automate process, and start to see things like vision systems come online to do my quality control, potentially move it down the line to handle packaging, the better.

There are a lot of support systems that can pair very readily with 3D printing, where you've got validation, so you print out the part and now you have a system that can take photos of it and compare it to the model that you put in and say, 'Yes, this is good output'. And those are the kinds of things that I get really excited about when I go to a forum next or [3D printing trade show] TCT, where you see on the industrial side, because they have the funds to actually fund development of it, but we'll start to see that trickle down into our space.

There are AI systems that are doing visual validation, detecting if you have a failure, you know, hopefully to save on waste and stuff like that. And so we're just gonna see more and more of that kind of thing. The future is bright, my friend. ◻

Left
The writer of this caption would appreciate the image a lot more if he weren't colourblind

FEATURE

IMPROVISER'S TOOLBOX

CDs

Maligned as expensive coasters, compact discs have a whole heap of other potential reuses once you tire of their contents, finds **Rosie Hattersley**

Rosie Hattersley

@RosieHattersley

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

In the late 20th century battle of the global consumer brands, Sony took the crown for the Walkman but shares the glory with Philips with the compact disc, which it launched in 1985 with Dire Straits' *Brothers In Arms* and claims of audio superiority. Despite the success of iTunes, iPods, and iPhones, plus streaming services such as Napster, Spotify, and the like, the compact disc is in rude health. In fact, offline music consumption is making something of a comeback as spoilt-for-choice music lovers embrace albums and plump for directly supporting artists they enjoy, rather than gifting most of the proceeds of their digital music choices to the streaming giants. Savvy folks often then sell off their CD collection to the likes of Music Magpie.

Walk into any charity shop however, and you're almost certain to find dozens of CDs going cheap, having fallen foul of domestic declutterers and lacking the warmth attributed to vinyl. Issue an appeal for unwanted CDs on a local social media group, and you're nearly certain to be offered an embarrassment of riches. Great news if you're thinking of a creative upcycling project that requires hundreds of the shiny silver platters such as the *Waste Landscape* **elise-morin.com/WASTE-LANDSCAPE-1**, which saw 60,000 unsold CDs hand-sewn into undulating hillocks and displayed inside in a Parisian hall in a visual protest of how much waste consumers generate. "Made of petroleum, this reflecting slick of CDs forms a still sea of metallic dunes," commented artist Elise Morin.

> "Backing up to another medium or the cloud thus seems de rigueur"

Being fed up with shuffling CDs and dumping them in favour of digital music isn't the only issue, of course. They are easily scratched, chipped, or broken if not kept in their protective jewel cases, while their contents also degrade over time. A report by the US Library of Congress, **hsmag.cc/DataDiscShelfLife,** showed that CD-Rs (discs containing data) had a shelf life of as little as a few years, rather than their intended lifespan of 30, when humidity and heat accelerated the rate of deterioration and degradation.

Backing up to another medium or the cloud thus seems de rigueur, though many millions of mid-1980s CDs surely play perfectly, and the internet is not awash with apocryphal tales of critical data lost to disc degradation.

As ever, one person's trash is another's treasure, right down to the hardware that plays the discs themselves. This project details how to make a charming doodle printer from the guts of an old CD/DVD drive: **hsmag.cc/CD-DrivePrinter**. Generally, however, it's best to reuse the discs themselves, collecting a decent number in order to create a pleasing visual effect.

CD SHARD SCULPTURES

Project Maker
SEAN E AVERY

Project Link
hsmag.cc/CD-Shard-Sculpture

Sculptor and illustrator Sean E Avery hit a rich vein when he first began snipping at old CDs with a pair of kitchen scissors a decade ago while still at university in Perth, taking inspiration from the wildlife of Australia. He arranges the resulting shards according to colour and hot-glues them to a wire mesh that he has already created. Burning himself on the glue is more of an issue than stabbing himself with the sharp CD pieces, he says. Almost all of the pieces Sean has made feature feathered or furry friends, so the challenge

"BURNING HIMSELF ON THE GLUE IS MORE OF AN ISSUE THAN STABBING HIMSELF WITH THE SHARP CD PIECES"

is to make his creations look as natural as possible despite the shiny medium in which he works. A scroll through his portfolio reveals a delightful menagerie of birds from hummingbird to horn-rimmed owl, while chinchillas, panda, and pangolin up the ante on the 'aw, so cute!' front. While Sean has made larger works including a dragon and a very large purple bird, as well as commissions for public display, most of his sculptures are between 15 and 25 cm tall and take only a week to make. →

Right
Blending writable CDs with pressed ones gives a dramatic effect

CD CIRCUIT BOARD CLOCK

Project Maker
PIXELTHIS

Project Link
etsy.com/uk/shop/pixelthis

CDs repurposed into clock faces are two a penny, since there's little more to them than adding plastic hands, a battery and, optionally, some form of indicator of the hour. However, US upcycler pixelthis, aka Allan Young, caught our eye with a nifty selection of timepieces based on various items of no longer needed tech. Objects that have been refashioned include a CD-ROM drive, motherboard, record turntable, bicycle wheel and gears and, as here, a circuit board. In fact, a passion for taking things apart to work out how they work is what led to his very first upcycled clock, which he made from an old hard drive. Allan has barely looked back since assembling his first clock from discarded electronics back in 2007, and his work has proved sufficiently lucrative that it's now his full-time job. "I have met some incredible people who eagerly donate materials that they would typically throw in the trash, and I have seen the inside of more dumpsters than I could have ever imagined," he explains.

Right
CDs are the best-looking digital storage medium

CNC MACHINE

Project Maker
TINKERNUT

Project Link
hsmag.cc/CNCNC

As CD and DVD drives began to be phased out on laptops and desktop computers, enterprising hacker Tinkernut hit on the imaginative wheeze of repurposing their drives into a more modern piece of kit: a CNC machine. Noting that "CD-DVD drives have a lot of really cool parts inside": a laser diode that reads the discs, a DC motor to open and close the bay tray, a brushless motor that spins the CD, and a stepper motor that moves the laser back and forth. Tinkernut takes these parts to create a CNC (computerised numerical control) machine for plotting or 3D printing. Using stepper motors, the XYZ axis movements can be controlled. The project uses three desktop computer disc drives, three stepper motor drives, an Arduino, and a power supply plus various nuts, bolts, and cables. The detailed two-part video shows how to dismantle the old drives and scavenge useful components before repositioning the three motor drive casings as walls to enclose the working parts of the three axis motors in the DIY printer. Before mounting the drives perpendicularly with L braces, carefully check the travel for each motor plate to ensure they won't crash into each other. Part II of the YouTube video outlines the Arduino setup used to control the self-assembly 3D printer.

CD LAMP

Weighing little and having reflective surfaces, compact discs are a great material to use as the basis of DIY lighting. With an abundance of internet setup discs lying around back in 2011 when he first came up with this ingenious upcycling idea, the marvellously monikered fissionchips decided to create a mega stack of around 120 of them and illuminate the lot. He began with a 12-inch fluorescent tube lamp, "shaved it a little" so it would fit inside the centre of the discs which he then piled up around it. The electronics, on/off switch, and ballast to stop the lamp toppling over are inside the back support, while the stack of CDs is held in place by two small notched rails that were glued in place. However, the assembly stand is separate from the lamp tube so everything can be disassembled. Fissionchips comments that the design worked well "because all the old AOL CDs and old CD-ROMS came in various hues, with colourful prints on top (though the tops face down in this version)." He also created a slight zig-zag to the CD arrangement to let more light through.

Project Maker
FISSIONCHIPS

Project Link
hsmag.cc/CD-Lamp

Above
This takes a lot of CDs, but some of us still inexplicably have a lot

DIY DUMB-BELLS

Fitness fan Sydney Capello fully embraces his love for all things 1990s, which is, of course, the "magical" decade in which compact disc sales were at their peak (hsmag.cc/CD-Sales). His CD-based weightlifting gear is the exercise hack we never knew we needed. "Who doesn't love the iridescent look of CDs?", he asks."If you don't, there's always duct tape or, hey, turn that CD around so the artwork side is showing!"

Sydney says you need approximately 150 CDs for a 10lb dumb-bell; 75 for each side. This should create a stack roughly four inches thick. Thread them onto an 18-inch threaded rod and secure them with nuts. It's critical that you use plenty of duct tape with the threaded rod "to avoid shredding your hands while doing curls!" warns Sydney.

This surprising, but genuinely useful, method of reusing old compact discs – even the ones you burned yourself and scrawled on with Sharpies – has a side benefit of helping you keep warm, as well as fit, as the temperatures plunge and energy prices continue to spiral.

Project Maker
SYDNEY CAPELLO

Project Link
hsmag.cc/CD-Dumbbells

Left
Turn garbage into gains

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

Ultra-light 3D printing

Create models with foam to save weight

Ben Everard

@ben_everard

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

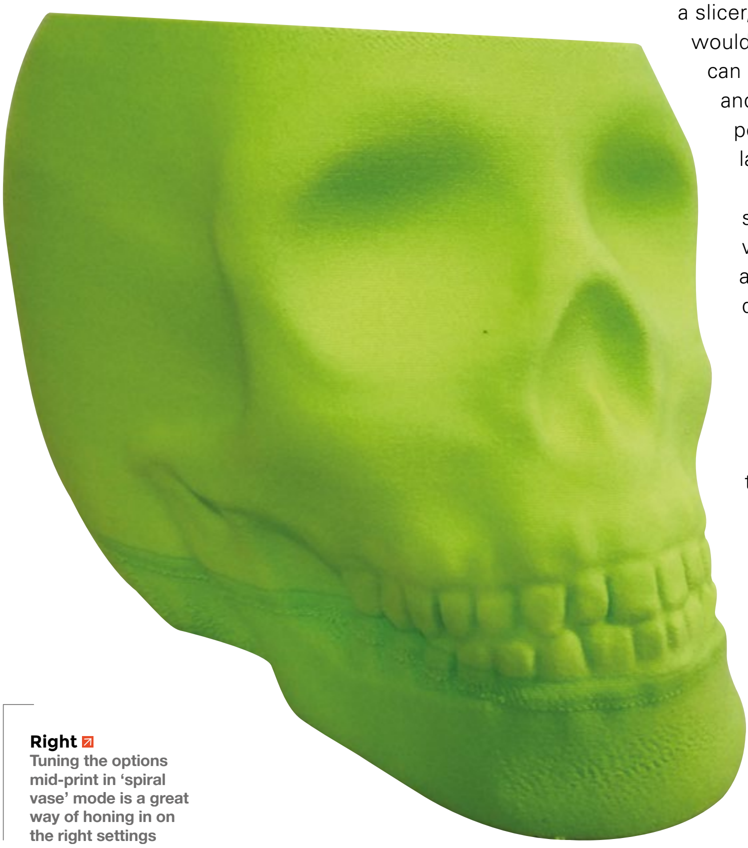
PLA is by far the most popular plastic for 3D printing. LW-PLA takes this base and adds a little magic in the form of foaming agent. This is activated by the heat of the hot end and creates bubbles of gas inside the plastic as it comes out of the nozzle. This foaming causes the plastic to expand, which means that you need a much lower flow-rate to get the same volume and extrude properly.

The foaming is a little less predictable than regular extrusion because it depends upon several printer-specific factors, such as how large the heated part of the hot end is and exactly how hot it is. This means that you need to calibrate the slicer settings for your printer. We did this following the guide on ColorFabb's website. While it's not really hard, it's also not entirely straightforward. Step one is: "Print cubes at a set speed in temperatures varying from 200 °C up to 280 °C."

A few things about this – the cubes should have no bottom because, until we have the flow-rates dialled in, the bottoms don't print properly. There's no information about how to print a cube with no bottom. Another crucial bit of missing information is that it should be single-walled. This isn't particularly hard if you know your way around a slicer, but a bit more information would be useful (in PrusaSlicer, you can right-click on the preview pane and add a filament, then set the perimeters to 1, top and bottom layers to 0, and infill at 0%).

The following steps are to select the temperature you want to use based on the amount of expansion and then dial down the flow-rate until you're getting the volume of extruded material your printer expects (by measuring the thickness of the single perimeter). Once you have those, you can set your filament profile and then slice and print as usual.

The process works reasonably well and gives you a filament that can give you the same wall thickness with much less plastic than traditional printers. For projects where you need large parts that are very lightweight (such as aircraft parts), this works really well.



Right Tuning the options mid-print in 'spiral vase' mode is a great way of honing in on the right settings

Left These two prints are both one perimeter wide, but one had foamed up much more due to higher temperatures

GOING FURTHER

There are undoubtedly interesting things that you could do with this filament if you could easily change the temperature and flow-rate at various point in a build. Infill and support material could be super-light while the main body could still be strong, for example. Alas, unless you're interested in spending a lot of time in the guts of slicing software, you're not going to be able to do this.

However, with the software we have now, there are still interesting things to do with LW-PLA. A key one is in the name – lightweight (LW). By foaming up, you can fill the same volume with less plastic. For things like model aeroplane parts where weight is absolutely critical, this is a huge advantage. To a certain extent, you can get the same effect by switching nozzles, as a smaller hole lets you create thinner walls and, therefore, lighter parts. However, that requires extra hardware, fiddling around, and much slower print times, whereas with LW-PLA, you just pop it in and out come the extra parts.

There's also an environmental benefit with using LW-PLA in that you can instantly drop 50% of the plastic on any parts that don't need to be strong (and also 50% less plastic in support material).

In terms of cost, LW-PLA can seem expensive (at around £40 per kg), but bear in mind, a kilogram creates about twice the volume as a kilogram of regular PLA, and the cost per part can be very similar (again, depending on the strength you need).

LW-PLA is a really interesting filament, but it's not as easy as most to use. It takes some dialling in the settings to get it right, but if you want the lightest possible 3D-printed parts, then this is how you do it.

> "If you want the lightest possible 3D-printed parts, **then this is how you do it**"

SPEED **FREAK**

Usually, the maximum layer height you can have with a printer is limited by the nozzle diameter. However, this filament foams as it comes out of the nozzle, which, in theory at least, means you can have layers thicker than the nozzle diameter. Thicker layers mean faster print times, so we decided to push our Prusa MK3S as fast as it could go by dialling up all the speeds and setting the layer height to 0.8 mm (despite a 0.4 mm nozzle).

The results were horrendous. Yes, they just about hung together, but only just. The problem is at high speeds, the filament didn't have very long in the hot end and foamed up inconsistently. In the parts of Benchy where there's not much on each layer (like the cabin), the filament foamed up properly, but in other parts, it's barely more than 0.4 mm.

Mince pies

How to make Christmas's most delicious treat

Ben Everard

@ben_everard

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

Christmas is all about food and, if there's one food that we can't do Christmas without, it's mince pies. These little sweet fruit delights are so emblematic of the season. However, too often we see sub-standard mince pies churned out by the million by supermarkets with more of an eye on shelf life and profit per unit than quality. In this article, we're going to go through our technique for making mince pies that we've honed over a couple of decades.

Like all pies, the first important thing to get right is the filling. You can buy mincemeat, but we've never found any shop-bought mincemeat that's particularly good. Good mincemeat should be mostly fruit with a rich syrupy sauce. If it's mostly a vaguely spiced, unidentifiable brown spread, then we'd definitely recommend avoiding it. Fortunately, however, while you can't buy good mincemeat, you can make your own.

The basic idea behind making mincemeat is to start with a spiced flavoursome liquid, and then use this to rehydrate some dried fruit. It's really hard to give a recipe for this because it's heavily dependent on moisture levels, and different batches of dried fruit suck up different amounts of liquid. You just have to play it by ear and adjust as you go.

A good starting point is a small bottle of wine – that's just 187ml. You can swap this out for a roughly similar amount of fruit juice if you prefer. To that, add plenty of sugar.

This is one of those traditional recipes where ingredients are 'whatever you have to hand' rather than being strictly prescribed. Darker sugars add a great flavour, but don't feel you have to get them just for this recipe. A glug of treacle, if you have some sitting on the back shelf of your store cupboard, can also taste great.

To this, you want to add some spices. The traditional spices include clove, nutmeg, allspice,

Right
Your pies are done when they're just starting to go golden

Left The goo awaiting the fruit

Below Use whatever dried fruit you like. This is our selection

and cinnamon. Fortunately, these all come in a blend called mixed spice, and we find it easier to use this. We also add an extra dollop of allspice because it's a particular favourite of ours. About two teaspoons of mixed spice and one of allspice is a good starting point (we can add more/adjust later).

The next step is probably the only odd one in the process. It's the only part that this author can

> Some people add chopped nuts – we're really not fans of the texture of this, **but if it works for you, go for it**

remember from the original recipe he followed years ago, and it really does add quite a lot to the final product: grate in an apple. This should ideally be about half a cooking apple, but we didn't have a cooking apple to hand, so we used an eating apple. Cooking apples will disintegrate completely into the sauce, whereas eating apples will (depending on the variety) stay a bit more solid. Either way, they add a lot of flavour and a bit of sharpness.

Now, heat the sauce as gently as possible. You want to cook in the spices and give the flavours time to mingle. Get it onto as low a boil as possible for about ten minutes. While it's boiling away, it's time to prepare the dried fruit.

Some form of dried grape is traditionally the main part of this – raisin, sultana, or currant – or a mix of lots of them. The other staple is citrus peel, usually in the form of mixed peel. However, you can add whatever you like. We've added apricots, and glacé cherries, but this author had good results with mango, dates, and figs (though this did add a slightly unusual texture from the seeds). Crystallised ginger can also taste delicious. Everything needs to be cut up to a small size – 5 mm-sized cubes are about →

Above You can make your own pastry, but this author's not very good at it

Left Mixed spice has all the traditional spices, but we like a little extra allspice

EXTRA ADDITIONS

We've described a fairly basic mincemeat recipe, but you can make it in all sorts of ways. As the name suggests, the Christmas treat originally contained meat. Some recipes include suet (shredded animal fat). This author isn't personally a fan of this, but add it to the cooled mincemeat if you like.

A few other things we've tried in experiments over the years:

- **Chocolate** – while the result tasted good, everything seems to be going chocolatey, especially at Christmas, so we keep this out of our recipe.
- **Soy sauce** – we wanted a slight tweak on the traditional sweet and savoury mixture this pie originated with. The results were quite pleasant, but it's perhaps telling that we haven't continued with it.
- **Chilli** – a little pinch to give a bit of background warmth can work well, if you like chilli. You will, however, forever become known as the person who adds chilli to mincemeat.

Right Fundamentally, mince meat is just fruit and goo

right, but there's no need to be too strict on this. Some people add chopped nuts – we're really not fans of the texture of this, but if it works for you, go for it.

Once, when money was tight, this author bulked out the mix with finely chopped carrots. The mincemeat wasn't quite as nice, but still pretty good. They do need a bit more cooking than dried fruit, and obviously don't soak up as much liquid.

Below There's no need to blind bake pies this small, so just fill them up and cook them

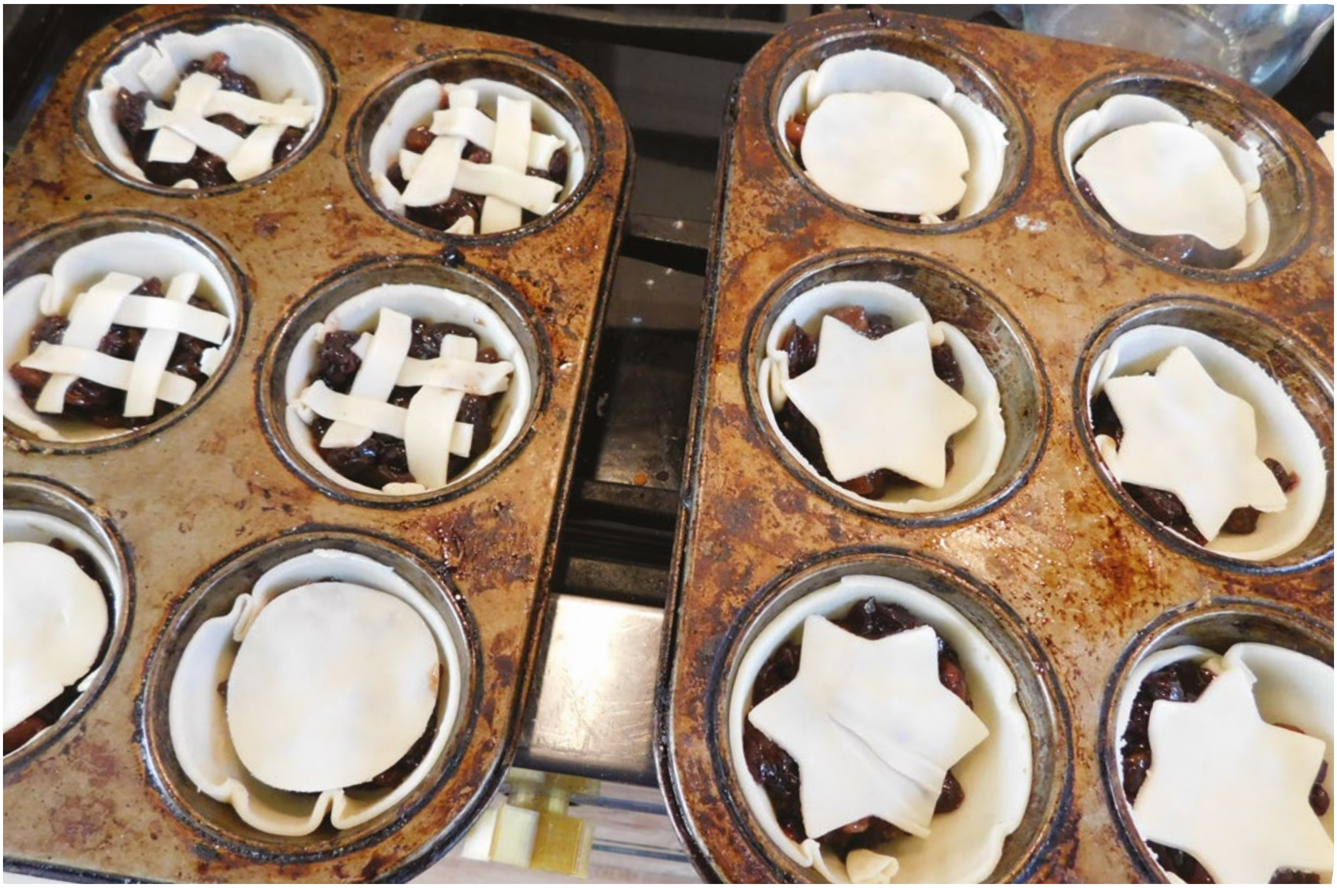
At this point, we should give quantities, but it's basically impossible in this recipe. Dried fruit aren't dried to a particular standard. Sometimes, you get little shrivelled raisins that can absorb huge quantities of liquid. Other times, they come out of the packet quite plump. The difference gets even starker when you're adding different types of fruit. We'd recommend starting with a couple of handfuls of dried fruit. If you end up with too much liquid, you can chuck a bit more fruit in later.

Now, add your fruit to the liquid. You want to keep this warm, but there's a risk of burning as the fruit

> **You can store the mincemeat in the fridge for a week or so,** or freeze it if you want to keep it for longer

soaks up the liquid. Depending on how low your hob can go, it might be a good idea to turn off the heat and just give it occasional 'blasts'.

Now is the point where you need to adjust the liquid levels. As the fruit soaks up the liquid, your mix should get drier and drier. A test for this is to pull a wooden spoon across the base of the pan. The goo



should very slowly creep back into the line it scrapes. If it doesn't, the mix is too dry. If it flows back in then the mix, it's too runny (it'll continue to thicken for the next few hours, so don't worry too much if it's too runny, but add more dried fruit if you're concerned).

What liquid can you add? Whatever you have to hand! Wine, port, brandy, or fruit juice are all good options. We went with brandy and orange juice. At this point, you can also taste for spicing and sweetness. Add more of either if you think it needs it. Bear in mind that it'll taste less sweet and spicy once it's encased in pastry, so don't be afraid to go hard.

Keep going with the gentle heat and liquid until the fruit feels mostly hydrated. This might be 30 minutes to an hour. After this, leave it to cool – it'll still thicken a bit as it does.

Your final mincemeat should be runnier than what you find in most mince pies. Partly this is because it'll thicken a bit as it cooks, and it's also because commercial mince pies have to be kept dry to have a long shelf life. More moisture in them makes a more delicious pie, and ours all seem to disappear in a day or two anyway, so shelf life isn't a big problem.

You can store the mincemeat in the fridge for a week or so, or freeze it if you want to keep it for longer.

MAKING PIES

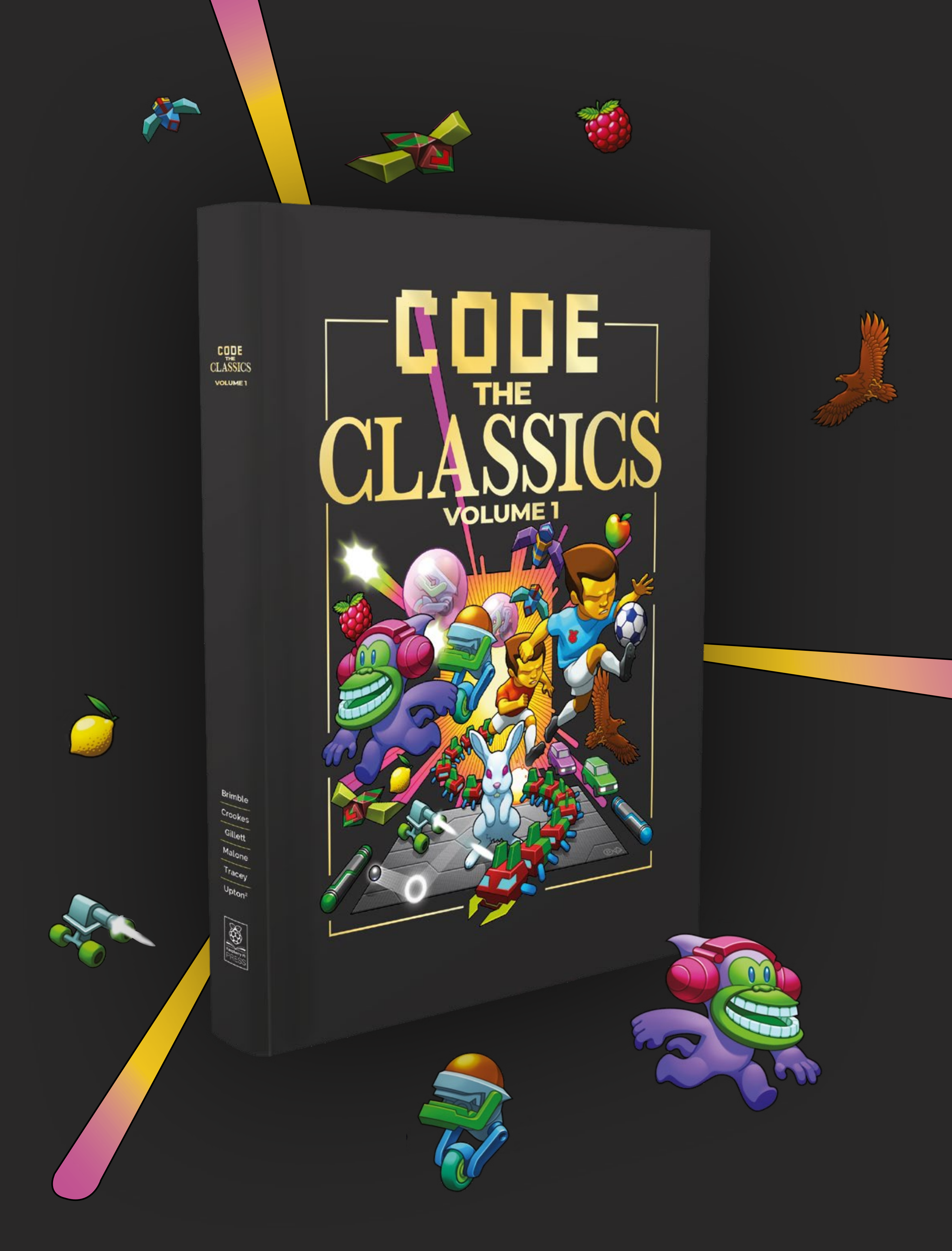
Now that we've got our filling, it's time to make pies. This is just about the same as any other pie recipe. We always start with shop-bought shortcrust pastry. Homemade pastry might give a better result, but it's one area of baking that this author has never been particularly successful with, and shop-bought works perfectly well.

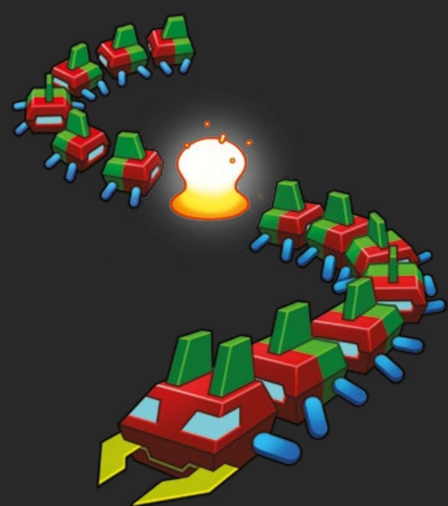
We like to make the pastry as thin as possible. Commercial mince pies often have a thick crust to make them robust in shipping and have a long shelf life – neither of these things are a problem for us, so we'll focus on taste instead. A thin, crispy shell gives a much better mouthfeel.

Roll it out, then cut out circles for your pie tray or muffin-tin. Fill these mostly (but not completely) full of mincemeat, then top them. Traditionally, mince pies are completely encased in pastry with just a little slit to let the steam out, but there's no reason to stick to this. We like to do a bit of a mix, with some stars and some lattice. Any exposed fruit will caramelise slightly in the oven – mince pie purists may baulk at this, but we think it's delicious.

Finally, bake in a 180°C oven until done (slightly golden on top). Take the pies out of the tin straight away, and leave to cool.

Above
You can top your pies with whatever shape you like


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



Making a dragon's nest

Make a misty forest dragon's nest, complete with a flaming egg

Dr Andrew Lewis

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

Above
By modifying a flame-effect light bulb and an ostrich egg, this project would make a great background for reading children stories that have dragons in them, or a Halloween decoration

There's a staggering amount of mythology surrounding dragons. From giant creatures in ancient tales to the vacuum cleaner-sized yapping flamethrowers of the Discworld, they're a staple of the fantasy genre. In this article, you'll get the opportunity to create your own dragon's nest prop with firelight effects and a mysterious haze. You'll also see how to use a motion sensor module and a Raspberry Pi Pico to turn on the electronic effects when someone approaches.

This dragon's egg prop has two practical effects built into it to give the impression of fire and smoke. On first glance, it's fairly obvious that flickering LED lights are used to give the effect of dancing flames, but the source of the 'smoke' is perhaps a little more difficult to discover. It's become quite popular to hack electronic cigarettes (vapes) to create smoke effects in models – for some projects, it's a good choice. A vape works by heating a mixture of propylene glycol and glycerin with a coil, the vapours of which are then inhaled. To create a smoke effect from a vape, you need to find a way to keep the heating element active while you use an air pump to push the vapour out of the way of the coil. The dragon's egg doesn't use this approach because there's a risk that a vape coil could overheat, boil the liquid, catch fire, or burn out in continuous use. This is combined with the fact that a vape needs a special liquid to work. From an aesthetic standpoint, the vapour from an e-cigarette needs to be kept in motion to keep the heater coil clear, and the vapour tends to rise, rather like a plume of smoke from a chimney.

For this project, a low-hanging mist is more visually effective, evoking memories of the alien nest in the movie *Alien*. To achieve this look, a more tried-and-tested technology comes into play. An ultrasonic pond mister or humidifier creates an eerie, rolling mist when it's activated, and it only needs a few inches of water to work. While

DRAGON SAFETY

It does not do to leave an electrically live dragon out of your equations, if you live near him. This project mixes water and electricity, and although the voltages involved here are low and should be safe, you should always make sure that your projects are properly fused and protected with a residual current device where necessary. Make sure water cannot travel anywhere in your project where it might cause a danger. In particular, ensure water cannot escape from your project and pool anywhere near household electrical outlets or extension cables.

pond misters are effective, they do have a few drawbacks that make them unsuitable in some situations. They're quite large when compared to an electronic cigarette, being generally about the size of a cupcake. They also tend to need an AC voltage input, because they work by oscillation of a piezoelectric element, and it's generally cheaper and easier to generate that oscillation with an AC voltage source. That means that it's less easy to get an ultrasonic mister to work from a battery, and, at the same time, it's more difficult to control the mister with a solid-state device like a transistor. Luckily, none of these drawbacks are a problem for the dragon egg project. We've used a 24 V DC mister that comes with its own power supply. These are widely available online.

Begin your project by building a nest. You can use a variety of different methods for this, including weaving branches together into a bird's nest, using a ready-made basket or bowl, using chicken wire and plaster bandages, or gluing together clay beads over a form and then using resin to strengthen the structure.

After building your nest, you'll need to put in a reservoir to hold your mister. A large spray can lid is useful for this, but be careful that you choose one that's deep enough to submerge the mister without creating a large amount of splashing, or you'll lose most of your water in just a few minutes. You can disguise the containers used for the reservoir with moss and stones.

LOW LIGHTING HIDES MANY CRIMES

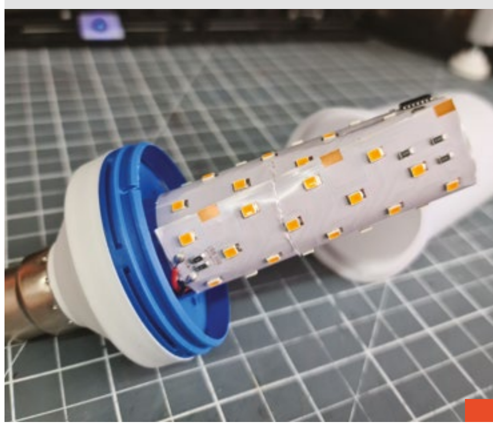
Route the power cable for the pond mister out of the back of the nest. It's time to feather the nest →

Above You can really let your imagination run riot and include whatever you want when you're designing your nest. This nest uses an antler and some old keys in addition to crystals, pine cones, and lichen

> "After building your nest, **you'll need to put in a reservoir to hold your mister**"

FLAME EFFECT

One of the main features of this project is the flame effect generated inside the egg. While the effect might look daunting to create, the circuit is actually salvaged from a cheap flame-effect light bulb. It's possible that you might have considered using these flame-effect light bulbs in the past but were discouraged by the practical or safety concerns arising from including a mains-powered bulb in your project. The good news is that the while the bulb uses 240 V, the circuit that controls the light effects inside these bulbs actually runs somewhere between 3.3 V and 5 V.



Below There are lots of different brands of LED flame-effect light bulbs, but they all look very similar inside. This particular version has the PCB held in place with a few dabs of hot glue, an elastic band, and double-sided tape

Right For the ultimate in flexibility when making, you can glue beads together on a plastic form (a plastic mixing bowl, for example), then remove the form when the glue is dry. The set PVA will be flexible rather than rigid, so you can manually tweak the shape of your nest and then use superglue to make the shape permanent. Finally, you can paint the whole nest with a casting resin to make it rigid and waterproof

USER SENSING

If you want to save power and water by only triggering your dragon's nest when someone is nearby, then you can do this with a Raspberry Pi Pico, a motion sensor, and some relay modules. Connect the motion sensor power to the 3.3V output of the Pico (VSYS), and the sensor output to GP0. GND on the sensor connects to GND on the Pico. Connect the relay modules' power to 5V (VBUS, assuming that you are powering your Pico from USB), and the modules' triggers to GP1 and GP2, with the modules' GND connected to GND on the Pico. A simple program can be used to trigger the relays on GP1 and GP2 when the Pico receives a high signal on GP0, then turn them back off if no motion has been detected for some time.

to disguise your cable and make it look lived in by a dragon. Add in a layer of lichen, some stones, pine cones, leaves, and any other detritus that you think a dragon might stuff into a nest. Once the nest looks nice and comfy, you can move on to the egg. If you can't find a genuine dragon's egg, you can substitute an ostrich egg or a papier-mâché egg instead. Ostrich eggs are generally available in

> "Ostrich eggs are generally available in the immediate **vicinity of an ostrich, or from online auction sites**"

QUICK TIP

If you'd rather have more control over the flame effect, you can use WS2812 LEDs (aka NeoPixels). These come in many forms, but a chain should fit inside the egg and then you can control the lighting of your egg from a microcontroller. You could even use different colours as notifications.

BULB SURGERY

Extracting the flame circuit is pretty straightforward, but the process will vary depending on the exact bulb you are working with. The general process involves removing the outer cover from the bulb with a plastic spudger (or a similar tool such as a screwdriver or sledgehammer) to reveal the electronics inside. The inner part of the bulb contains a flexible circuit board with a microcontroller and LED lights, powered by a small transformer circuit in the base of the bulb. This flexible circuit is usually glued or taped around a cylindrical support in the centre so that the flame effect is visible for 360 degrees. Carefully peel the flexible circuit board away from the central support, and cut the wires that lead to the transformer. You can now connect the board to a DC power supply and test it at 3.3V. If it doesn't light up or flicker properly, increase the voltage until it becomes stable. Choose a suitable DC power supply for your board, and connect to the flexible circuit with an in-line plug and socket.

Left Some flame-effect light bulbs use 3.3V internally, while others use 5V. If you find the bulb stops working after a few seconds, try reducing the voltage. You'll probably find something is overheating. The circuits can be quite fragile when it comes to electrical spikes, so be careful when connecting and disconnecting wires

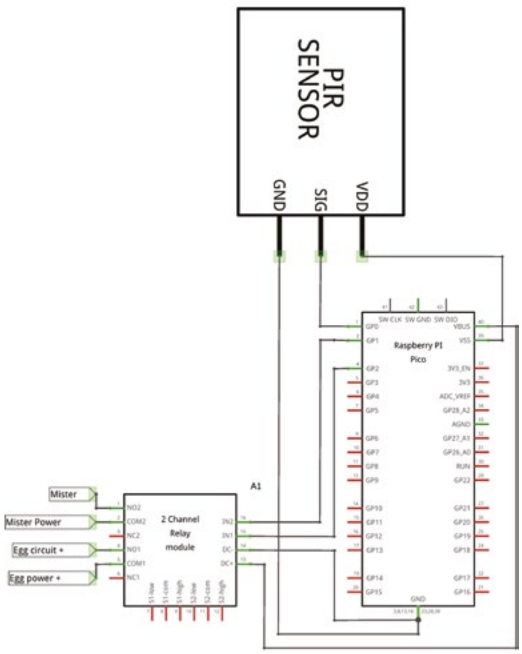
QUICK TIP

If you're interested in hacking vapes as a source of practical smoke, take a look at Dave Bodnar's notes at **hsmag.cc/AnimationSmoke**

the immediate vicinity of an ostrich, or from online auction sites. It's easier to get a blown egg (which is just an empty shell) than a whole egg, unless you need to make a really big cake and are willing to carefully blow the egg yourself without cracking it.

A blown egg will have a hole in the base, and you can use this hole to insert the flexible circuit from a flame-effect light bulb. Attach long wires to the flexible circuit, and carefully roll it up with the LEDs on the outside of the roll. Push the rolled-up circuit into the egg, and then unroll it so that it sits with the LEDs facing upwards. This step is quite fiddly, and you may find some pea-pod forceps or tweezers useful. Fix the circuit in place with a dab of hot glue, and then close over the end of the egg with glue and tissue-paper to disguise the hole.

Place the egg into the nest and route the wires under the lining so that they are hidden from sight when you're looking in. Fill the mister reservoir with water and plug everything in. After a few seconds, you should be rewarded with the sight of a fiery dragon's egg in a sea of mist.



Above Connecting PIR sensors and relay controllers to your Pico is a simple way to control the practical effects in your props

QUICK TIP

If your ultrasonic mister isn't deep enough in the water and fires water out of the nest, use a plastic lid at an angle to catch any splashes and divert them back into your reservoir.

Weaving: A beginner's guide

It's easier than you think to get started in weaving: you can even hack your own loom!

Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor. She's creative, but you can't expect her to be tidy too…as a peek at her craft room will attest.

The craft of weaving is thousands of years old and basically involves the creation of a fabric by interlacing threads. In its most elementary form, even before weaving was used to make cloth, the interweaving of plant leaves and branches was practised in order to make baskets, shelter, fences, and so on. There's evidence that weaving was used as far back as Neolithic times. The ancient Egyptians were enthusiasts of the craft, as were other cultures around the world, including the Amazonians and Native Americans.

Back in issue 57 (**hsmag.cc/issue57**), we discussed how the craft of weaving and, in particular how the development of the Jacquard loom and its use of punch cards, played a significant role in the early development of computers. That famous loom enabled more complicated patterned cloths to be woven and paved the way for computer coders. It's been a fibre art that this author has wanted to try for some time. So, in this tutorial, we will be attempting to grasp the very basics of weaving and we'll create a straightforward wall hanging with a simple loom and some weaving materials.

TOOLS OF THE TRADE

It's worth noting, at this point, that weaving is a pastime that you can begin without spending too much cash. Obviously very large looms are expensive and you would need to be serious about your craft to invest in those, but the fundamentals of the art can be learned on a very basic loom. We've used an inexpensive wooden handheld frame loom from a craft store (30 cm by 39.5 cm), and pretty much everything we needed in terms of tools came in the box with the loom. Shop around, but you should be able to purchase a similar loom for under £20. We're starting at an elementary level in order to begin to understand the principles and some bedrock techniques. Plus, we want to feel inspired to continue with this craft, so we need to set a realistic and achievable goal.

We'd be remiss, as this magazine is entitled HackSpace, if we didn't also mention the hackability of loom frame-making, as we're sure many of you have the wherewithal already in your maker spaces. This one's made from wooden stretcher bars, but you could use an old picture frame – **hsmag.cc/MakeaLoom**. You can even fashion your own simple loom from a basic material such as strong cardboard: **hsmag.cc/CardLoom**.

One very important thing to be aware of is that the size of what we're able to create on a

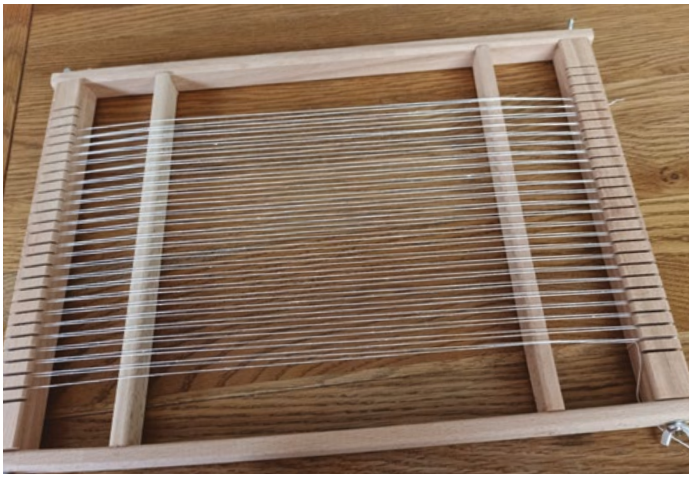


Figure 1 The loom is now 'dressed' and ready for action. We're utilising most of the frame for our wall hanging – if you were making a bookmark, for example, you'd use fewer grooves or pegs on the frame, and string up a much narrower area

RESOURCES

If this article on weaving has piqued your interest in taking up a very stress-reducing and satisfying pastime, then there are plenty of resources for you to investigate. Learn how to weave a thing of beauty, and hopefully not a tangled web, with these beginner-friendly sources of weaving wisdom:

- Read a decent tome on the subject. *Weaving within reach: beautiful first projects with and without a loom* (Anne Weil: 2018) – **hsmag.cc/WeavingBook**, or *Little Loom Weaving* (Fiona Daly: 2018) – **hsmag.cc/LoomWeaving**.
- Research and maybe order some appropriate tools online, as we did: **hsmag.cc/UsefulWeavingTools**.
- Watch a video of someone who knows what they're doing; in this case, weaving tips and tricks – **hsmag.cc/WeavingTips**. YouTube is awash with such handy insights. Plus, take a look at this site with its contemporary approach to lap loom weaving – **theweavingloom.com**.
- Wallow in the history of weaving and peruse some informative sites such as **hsmag.cc/JacquardWeaving** and **hsmag.cc/BriefHistory**.
- Instructables is always a reliable source of inspiration, and a search on 'weaving' (**instructables.com/howto/weaving**) produces projects ranging from woven bacon (!) to how to weave a paper basket. Explore and appreciate the inventiveness.
- Seek some inspiration from some weaving blogs: **laurasloom.blogspot.com** or **curiousweaver.id.au**.

small frame loom like this is constrained by the size of the frame that we are using. See the 'Types of weaving looms' box – but, we're using a very simple loom and it's not one that can produce a continuous piece of fabric, such as the fabric that can be woven on a rigid heddle loom, for example. However, as we're weaving novices, we'd argue that's really not a problem. We will still be able to create things like coasters, place mats and, if you sew pieces together, even items like small bags or pouches.

The great thing about weaving is there are no real rules in terms of what you should weave with (see 'What materials can I weave with?' box), and you can keep costs to a minimum. We've purchased a low-cost loom and some warp thread, but this stash-owning author has plenty of suitable weft yarn/fabrics to use in the body of her weaving, and we're pretty sure that most people will have access to something

to weave with too, even if it means unravelling an old sweater – recycle and reinvent.

WEAVING BASICS

First, let's familiarise ourselves with a few weaving terms that will be useful:

- **Warp thread** – the strong, tight threads stretched vertically up and down the loom.
- **Weft** – The fibres and yarns woven onto the warp to create a fabric. These are woven horizontally across the warp.
- **Tapestry needle** – This is what we will attach most of our weft threads through and is what we'll feed over and under the warp thread. For big sections of weaving, you can also use shuttles to wrap your yarn around, two of which were in our kit.
- **Selvedge** – The sides of the woven fabric that run parallel to the warp.
- **Tension** – The degree of tightness of the warp and weft threads.
- **Beating** – Where the weaver uses a comb (we've even seen people using forks for this too!) to push down the weft threads to compact them and create a more dense fabric.
- **Plain weave** – The simplest of weaving techniques and the one we are going to use. Each weft thread passes alternately over and under each fixed warp thread. Note: there are many, many other weaving techniques.
- **Twining** – A way of preventing you beating your weaving too far down. For example, you can insert a ruler, as a spacer, at the base of your weaving to act as a guide. →

Figure 2
Over and under, under and over… the mindful act of weaving is very rhythmic

YOU'LL NEED

- **A loom** (we used a kit which included most of what was required: hsmag.cc/LoomKit)
- **A shuttle or large tapestry needle**
- **A comb** (hacker's alternative: a fork)
- **Heddle bar** (optional)
- **Scissors**
- **Tape measure/ ruler** (for checking width of work)
- **Warp thread** (we used hsmag.cc/WarpThread)
- **Weft fibres and threads** (e.g. yarn, rope, roving, ribbon, fabric strips)
- **Hot glue gun/ glue sticks and embellishments** (optional)

Above A selection of colourful scarves made on a rigid heddle loom, which enables the weaver to create pieces with a long length to them

STEP 1 WARP FACTOR

So, before we can weave anything, we need to add the warp threads to our loom so that we have something to weave through, commonly known as 'dressing' the loom. The warp thread used needs to be fairly strong – the one we chose is 100% cotton. If we learnt anything from this process, it's that preparation of a well-tensioned loom is everything in weaving; so, when warping up, try to ensure that the tension is tight and even. In many ways, it's similar to how a guitar is strung: the warp threads need to be taut, but they also have some bounce to them.

There are different ways to warp your loom, but we have chosen to use a continuous double warp for ours which means that the warps will be strung more closely and densely together than, say, a single warp. Begin by tying the end of your warp thread in place over the first groove, or possibly peg depending on the loom being used, and make sure it is secured with a double knot. Then take the warp thread to the opposite end of the loom, keeping tension fairly taut, and wrap it around the opposite peg or groove. Now, take the warp thread back down to the groove you first used, wrap it around, and then take it back up to where the warp string was before and so on. So, you are basically following the grooves on the loom but going back to where the warp string was before each time. This short YouTube video is really useful in showing you how to warp your loom in several different ways – **hsmag.cc/WaystoWarp**.

Note that if you have a heddle bar, you can place that in the centre of the loom and string the loom through the grooves of this, as well as the grooves at the top and base of the loom. We had a heddle bar in our kit, but we haven't yet used it. What the heddle bar does is create an opening, or 'shed', which makes it really easy to pass your needle or shuttle through the warp threads as you are weaving.

> **"Hopefully, you have kept your tension reasonably** taut and regulated throughout"

When you get to the other side of the loom and the final warp string is in place, you can tie off the end in the final groove, again using something secure, such as a double knot (see **Figure 1**). Hopefully, you have kept your tension reasonably taut and regulated throughout but, before you tie off, check for any slackness and adjust any strings that need tightening. The loom that we purchased actually has adjustable top and base ends with wing nuts, so we can easily twiddle the top and bottom bars to tighten and adjust for any loose strings, which is a helpful feature to be aware of.

STEP 2 LET'S WEAVE!

Once you're strung up, so to speak, you're ready to weave. One thing that we came across in our research was the suggestion of doing something called a 'twining', mentioned in 'Weaving basics' above. You can use a ruler or spacer to stop you

MODERN WEAVING AND TECHNOLOGY

We mentioned earlier how the Jacquard loom was a stepping stone to the invention of computers. It can be argued that we certainly owe a great deal to the craft of weaving.

So many of the things that we use on a daily basis are woven – home linens, clothing, and so on – and most are obviously now produced by machine, not by hand, in order to meet society's increased consumption. When computer-aided design and manufacture (CAD/CAM) enabled further automation of the manufacturing process, weaving evolved with the technology. These days, designers are able to create their design on a screen, and these are then transferred directly to the controls of computerised looms. The process of design and manufacture has been sped up – what once took days now takes hours, and designs can be stored digitally.

But many artisan weavers, producing handmade pieces, have also embraced technology to help them design their work. There are plenty of software applications available to assist them, such as **weavepoint.com**, software for advanced weaving design; **weaveit.com**, a weaving design app that also helps weavers calculate how much yarn they'll need; and **pixeloom.com** which, among other things, allows a weaver to create patterns and experiment with colourways.

It's also worth remarking that, if something can be improved with a Raspberry Pi, you can just bet someone will improve it, and we saw such a weaving-related project back in 2016 in issue 52 of our sister magazine, The MagPi (**magpi.cc/52**). Fred Hoefler built a four-harness tabletop loom, with four 12-volt DC motors, and it was all controlled by a Raspberry Pi, thus making the loom physically much easier for his wife to use.

We've said it before in these pages, but many fibre artists – including weavers, knitters, and crocheters – are embracing technology, fully recognising how it can improve the creative process.

Above
Using a comb to push down and compact the rows of weaving, maintaining a constant tension and shape

from pushing your weaving too far down, or you can do a twining stitch at the beginning and end of your weaving to help lock things in place. Given this is a beginner's tutorial, we've not done this for our wall hanging, but this useful video explains why this might be handy: **hsmag.cc/Twining**. Instead, we used the two wooden bars on our loom as a visual guide for the 'top' and 'bottom' of our work. We wanted a short length of warp fibre on either side of the actual wall hanging. Specifically, the top warp threads will be turned into a hanging loop to hang on our small branch, while the bottom threads will be tied off.

Begin by selecting your first weft thread or yarn. We used an Aran weight yarn as our first weft, and threaded it through our needle. We're using a one-to-one ratio as we weave, which means that we will work our way across the loom, going behind one warp thread, then over the next thread, all the way until we reach the other side (see **Figure 2**). Clearly, different ratios will produce different textured effects. When you reach the other side, if you have gone over the last thread, on the way back you will go under that thread and over the next, and vice versa. →

QUICK TIP

As a side hustle, weaving is a great one. For example, this author recently saw a small, very basic woven wall hanging on sale in a local coffee shop for around £40.

WHAT MATERIALS CAN I WEAVE WITH?

The answer to this is simply pretty much whatever you want! Consider yourself completely free to use your imagination, and you can recycle, weaving-wise, whatever you have to hand – weaving is basically a fantastic hobby for someone who likes to recycle and not throw anything away! It's also a way of holding onto memories, perhaps incorporating bits of a loved one's old clothing item in your work, so you can see them every day when you glance at your finished work of art. This is a creative process, so give vent to your creativity. Delve into your maker's stash, loft, or drawer where you keep those really 'useful' things, and see what you can find. Here are just a few ideas to set you in motion:

- Make your own yarn – for example, find some old T-shirts and cut them up into strips with scissors or a rotary cutter, and sew them together for longer lengths. This will have great texture, doesn't fray, has a stretch to it, and is thick, so your weaving project will work up quickly. This useful video (**hsmag.cc/TShirtYarn**) shows you how to make yarn from the whole T-shirt – no wastage there!
- You can basically cut up any fabric item into strips, and this author is planning to use some of her son's old shirts in her next project. Ditto, timeworn sheets, tea-towels, tired jeans, or how about unravelling a fatigued knitted item that's seen better days and weaving with its yarn? Or, consider using some organza fabric cut into strips which gives a very floaty feel to your work.
- Use any haberdashery-type items you might have lingering – ribbons, lace, and trimmings all add something unique to a piece and a lot of texture.
- Wool roving – this is a wool fibre that has been processed but not yet spun into yarn, and is great for adding some bulk and interest. For example, white/cream would make great clouds if you were trying to weave a landscape and sky scene, and even sheep in a field scene, with a tiny bit of black roving for their heads.
- Plastic bags cut into strips and woven through your work will give it a very individual look, as will weaving with any kind of paper – **hsmag.cc/PaperWeaving**.
- Use *any* yarns that you have, thick or thin – variety is the key to giving your weaving interesting textures and depth.
- If you are making a wall hanging, hot-glue your finished piece with pebbles, shells, stones, leaves, or twigs that you picked up on a walk, weave through some LED lights at the end, sew on some pompoms or buttons, create some tassels… you get the idea. Sticking embellishments to your weave can also hide any areas you are less than happy with!

Right
With a sister who teaches Design and Technology/Textiles at a local grammar school, and who has been weaving for some time, this author was lucky she had someone she could ask her weaving-related queries. Here's her portable rigid heddle loom with a scarf 'in progress'

TYPES OF WEAVING LOOMS

The word 'loom' comes from the Middle English word 'lome', meaning 'tool', and there are a vast array of looms out there. If you are thinking of trying weaving for the first time, consider the size of the items that you might want to make (a small wall hanging, or maybe a long scarf if you are more ambitious), and how much free space you have to house a loom. Most beginners will invariably start with a smaller piece of weaving equipment, as we have done here. This enables you to begin to understand the principles of weaving, grasp the basics, and decide if you actually enjoy the process of making fabric on a loom, before progressing further and maybe investing in a larger piece of equipment. Here are some of the many types of looms that are in use today:

- **Industrial weaving loom** – as the name suggests, this is not a loom for a hobbyist. From the power looms of the Industrial Revolution to modern computerised industrial looms, these are generally enormous pieces of machinery intended to produce a large amount of fabric.
- **Floor loom** – the largest of the home-weaver looms, it contains multiple shafts. Shafts are frames that hold the warp threads, and treadles control the movement of the shafts. These looms are free-standing, and intended for weaving larger items such as rugs.
- **Handheld frame loom** – sometimes called a lap loom, and perhaps the most beginner-friendly of looms, hence why we've chosen it for our tutorial. Inexpensive and portable.
- **Back strap loom** – this is an uncomplicated loom developed by ancient civilisations and still used today. The warp is tied around a stationary object on one end and the weaver at the other, the weight of whom keeps the warp taut.
- **Inkle loom** – this is a compact and portable loom, designed to create long, strong braids such as belts, dog leads, trims, and straps.
- **Rigid heddle loom** – this one is great for beginners too. It's portable and can be used with or without a stand. If you want to make a long scarf, cowl or wrap, cushions, or bags, this is the perfect option and won't break the bank. A great step to take after learning the ropes on a basic, handheld piece of kit.
- **Table loom** – these enable multi-shaft weaving without the expense required for a floor loom. Levers control the shafts on this type of loom. It's still a pricer loom than, say, a rigid heddle loom, and you're probably a fairly seasoned weaver if you intend on purchasing one of these.

This is by no means an exhaustive list, but you can see from the variety of looms listed just how wide an assortment there is, and how different looms will create different sizes of fabric.

One thing that we found we needed to do on every row was use the comb to 'beat' down the yarn. At the end of each row, ensure that the yarn at the opposite selvedge is not too tight, gently push the yarn down towards the row beneath it with your comb, and lightly pull the yarn end so that the selvedge straightens. One of the key skills in weaving is ensuring that edges are straight, which is why tension is so important – if you pull the yarn too tightly, you will find the piece loses its shape and you get an hourglass appearing in your work. As long as you are aware of this potential mishap early on, keep your tension even, and regularly use a ruler or tape measure to check width, you can work to ensure that this doesn't happen.

In terms of changing weft threads, we wove as far as we wished with one thread, then dropped the tail end, ready to be woven into the back of the work later with a needle. The same applies to the beginning of a weft thread – we found that you can let it drop and come back to it later. When it comes to weaving in things like roving or thicker fabrics, we didn't use our needle for this: we just used our fingers, as it was far easier.

We tried to use some different textures to draw the eye to different parts of the piece. We also used some rya knots at the beginning of the process, which you can see at the bottom of our hanging – these are a type of tassel and very easy to add. You are essentially creating your own picture in threads and fabrics, so you can be as creative as you wish. We were aiming for an autumnal theme to our piece, so the colours are quite muted and, considering this was our very first attempt, we are really happy with the results (see **Figure 3**).

STEP 3 TAKING IT OFF THE LOOM

When you have completed the weaving element, and are happy with your work, you can remove the piece from the loom. As mentioned, we wanted to use the loops at the top to hang our work, so we simply slipped the top warp strings off of the loom. When these are removed, you'll find it simple to slip the bottom warp threads off the loom too. Take two bottom warp threads and tie them together with an overhand knot to secure them, and do that all along the bottom. You can sew them into your piece to hide them using

QUICK TIP

You can use all sorts of things to hang a woven wall hanging, including a piece of driftwood, a small woodland branch, or even a piece of copper piping. Upcycle, recycle, reclaim!

A WEAVING PIONEER

Theo Moorman (1907–1990) was a British textile artist who developed her own very individual weaving technique named, perhaps obviously, the Moorman inlay technique. This technique uses tie-down warp threads to lock-in supplemental weft yarns so that the weaver can create freehand designs. It's a weaving technique that has been likened to painting with yarn as you weave a background cloth underneath.

She taught a large number of workshops in the US, and she also wrote an important book entitled *Weaving as an Art Form: A Personal Statement* (1975), detailing her technique and sharing her weaving insights. She created some huge commissioned pieces, which can be found in cathedrals and churches in the UK and US.

Theo's technique is not perhaps something for a beginner to try and grasp early on in their weaving career, and wouldn't be possible to do on a handheld lap loom as you'd need something like a multi-shaft floor or table loom, but it is certainly something for a newbie to aspire to using at some point on their weaving journey.

Take a look at this link for further inspiration: **theomoormantrust.org.uk**.

Above
Here's an example of the Moorman technique. This author's sister made this using the inlay technique – the back of the scarf is plain weave, but the front has a design woven in

a needle, but ours were disguised by our tassels, so we just left them. We then threaded the top loops onto our branch, sewed in the weft thread ends with a tapestry needle, embellished with a few finishing touches using some hot glue and, hey presto, one wall hanging!

WEAVE YOUR MAGIC

So, why not buy or hack together some kind of loom and lose yourself in the repetitive and mindful craft of weaving? On more than one occasion this author was saying to herself, 'I'll just go back and forth one more time'… half an hour later, she was still there. Thread your way through the learning process and we're sure you'll be glad you gave it a go. As with anything in life, success won't be yours overnight, so practise… then practise some more, as we certainly plan to do. Enjoy it and see where you end up – maybe friends will be so blown away by your talent, they'll commission pieces from you. Be unique and experiment on your blank canvas – believe in your weave!

" Thread your way through the learning process and we're sure you'll be glad you gave it a go "

Figure 3
The fruits of our labours – one wall hanging ready for… well, hanging. We've used different fibre textures to add some interest and depth, used a foraged woodland branch to hang it on, and hot-glued some autumnal adornments

QUICK TIP

If you are confident with a sewing needle, how about embroidering a design onto your finished piece?

Fun with sound trackers

Discover sounds, how to use them, and how to make the one that all the cool kids are using

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** and follow him on Twitter at **@robmiles**

YOU'LL NEED

- **You can get sound tracker programs for all computers**
- **We are going to use the MilkyTracker program you can get from:** milkytracker.org/downloads/
- **If you want to have a go via your web browser, you can find a tracker here:** stef.be/bassoontracker/

Sound trackers are cool again. They were present at the very dawn of digital music. Now, devices like the Dirtywave M8 (**dirtywave.com**) and the Polyend Tracker (**polyend.com/tracker**) are finding their way into the hands of musicians who are rediscovering the joy of making music "the spreadsheet way". In this article, we are going to examine the origins of sound trackers, have a go with one, and discover how to create our own version of one of the best sound tracker devices on the market today.

SOUND TRACKER ORIGINS

Computers are now a routine part of making music. But it wasn't always so. The first computers had hardly any sound-making abilities. But then came the Commodore Amiga series of machines which, courtesy of a custom chip called Paula, could produce four channels of 8-bit audio with 6 bits of level control. Software and games could use this hardware to produce vaguely orchestral sounds. In 1987, German programmer Karsten Obarski developed the first Amiga sound tracker as a tool for creating game soundtracks. The program was modestly called 'The Ultimate Soundtracker'. This turned out to be an ironic title as its code was subsequently taken to bits and used to make improved versions that were widely distributed for free.

Figure 1
MilkyTracker runs fine on a Raspberry Pi

Later improvements to sound tracker programs added four more channels, and the Amiga scene also included cheap audio digitisers. It was now possible to sample a sound and then sequence and play it back in any order on multiple tracks. Music makers piled into the Amiga. Lots of household names got their start in music production by playing around with screens full of numbers.

Nowadays, we have much more polished ways to make computer music. As well as digital audio workstation (DAW) software for our computers, we also have dedicated hardware sequencers. However, as far as this author is concerned, there is a lot of fun (and learning) to be had messing around with patterns of notes that scroll up the screen as they are playing. Let's look at an example of a sound tracker and have a go with it.

GETTING STARTED WITH MILKYTRACKER

The MilkyTracker program is free, available on lots of platforms, and makes great sounds. It also plays a wide range of tracker files. You can download it from

Figure 2
The setting values are saved automatically

milkytracker.org/downloads. There is no installer for the program; you just run it from the folder that is created when you unpack the archive. You can make your life easier by creating a short cut on your desktop that refers to the MilkyTracker program file.

Your computer might baulk at running raw binary code downloaded straight from the internet, but you can usually persuade MilkyTracker to run eventually. Just make sure you have downloaded your program from the official link above. If you are concerned about installation, you can use a sound tracker that runs inside your web browser. The one at **stef.be/bassoontracker** works well, and you can load song files from your local filestore into it. MilkyTracker comes with a simple manual, which is a great thing to keep open when you are using the program.

Figure 1 shows the MilkyTracker program playing music. Each of the eight vertical columns represents a sound channel. Each row across all the channels represents the notes to be played for that 'tick' of the music. Above each channel, you can see a tiny waveform display of the signal coming out of that channel. On the top left are controls, and on the right, there are windows for Instruments and Samples. It all looks rather complicated, but it does make sense once you understand what each part does. But before we make any sounds, we need to do some configuration.

"It all looks rather complicated, **but it does make sense once you understand what each part does**"

SIZING UP THE DISPLAY

The first time you run MilkyTracker you might be a bit underwhelmed by the tiny window it displays. However, you just have to adjust the scale and resolution values to fit your screen. To do this, click the Config button in the controls to open the configuration window and then click Layout. You should see the setting screen as in **Figure 2** above. Use Scale to make the text larger and Resolutions to control how much you can fit on the display. Click OK to apply the settings. You may need to exit the program and start it again for the changes to have an effect. Make sure that you can see all eight channels without having to pan the display inside the screen, and use the scale adjustment to make the text in the display a usable size.

Once you've got your display sorted, you can think about the audio output from the program. MilkyTracker will try to use the default audio device on your machine, but you can change this if you →

QUICK TIP

You don't have to create any music to have fun with sound trackers. There are thousands of songs (or MOD files) you can listen to. The author likes watching a tracker player chugging through a song, rather like you would watch a player piano or steam organ. You can find loads of songs here: **hsmag.cc/aminet**

Figure 3
If you change the audio configuration of your device (perhaps plug in a USB sound device), you will have to restart the MilkyTracker application for the new device to appear

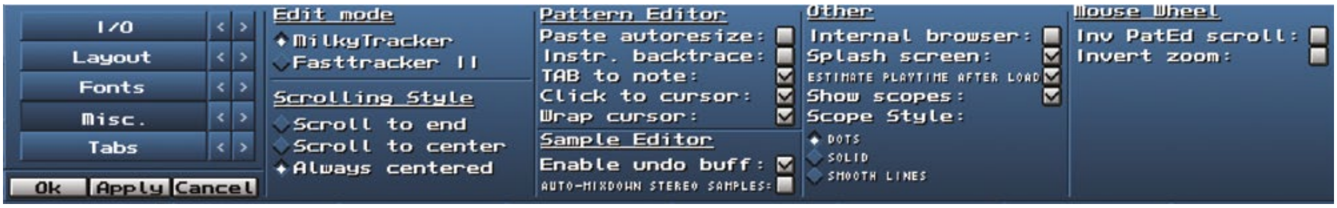


Figure 4 Setting to MilkyTracker edit mode turns on the all-important CTRL+Z delete and CTRL+Y redo keys

want to use different devices. Use the I/O menu shown in **Figure 3** overleaf. Click Select Driver to select the driver on your PC that you want to use.

PICK YOUR INTERFACE

The final thing you need to do is make sure that you are using the MilkyTracker user interface rather than the Fasttracker II one. Use the Misc menu in config and set the Edit mode to MilkyTracker (**Figure 4**). Click OK to exit the configuration pages. Now you might want to reward yourself with a chiptune while you have a nice drink. Click the Load button and open the file dialog to browse the Example Songs folder. Open and load *theday*. It has a nice bouncy tune and uses eight tracks. Press the Play Sng button. Watch what happens as the music plays. You will see rows of characters flowing up the screen. You can click the instrument edit (Ins. Ed.) button while the music is playing to display the instrument settings and see a keyboard playing along with the tune.

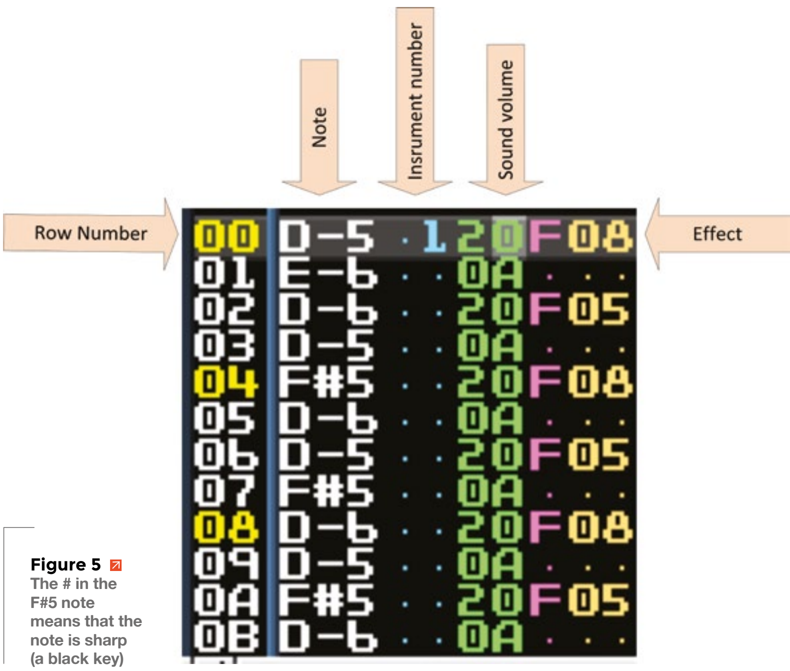


Figure 5 The # in the F#5 note means that the note is sharp (a black key)

To discover how the tune fits together, you can click inside the waveform displays at the top and mute individual tracks. This gives you a feel of how the sound is constructed. Mute all the tracks except track 1, and then press the Stop button to stop the playback. If you hold down the **SHIFT** key and press the **SPACE** bar, MilkyTracker will play the track one row at a time so you can listen to individual notes.

THE ART OF SOUND-TRACKING

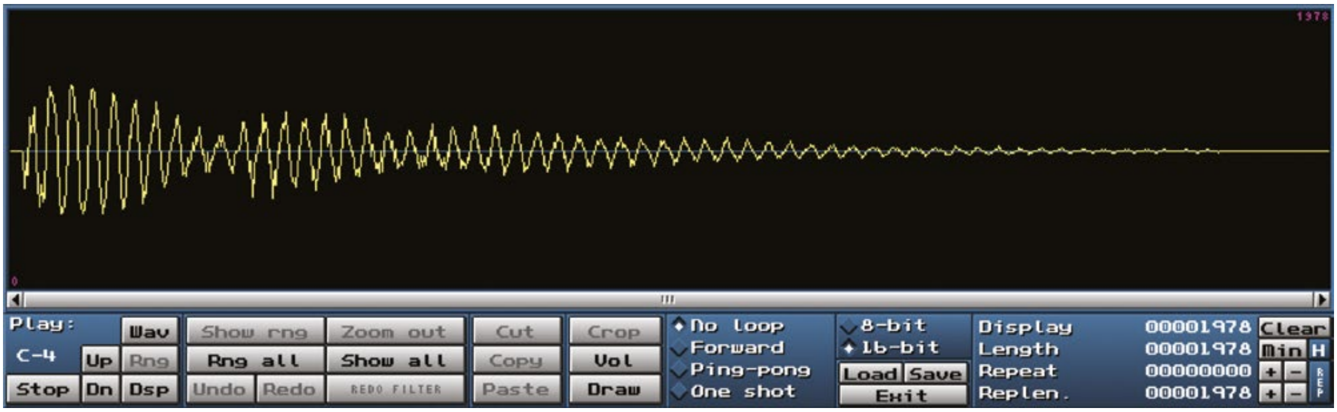
Figure 5 shows part of the song track for channel 1. There is a little row of data for each note. The first part of the row is the pitch of the note, indicated by the letter of the note on a standard piano keyboard. The part of the pitch is the note (the first note is D), and the number gives the octave to use. The octaves are numbered starting with the leftmost octave at 1, ending with the rightmost (highest pitched) at 8. The note indicated is at octave 5.

The next number, 1, in the case of the row in **Figure 5**, specifies the instrument to be used. An instrument can be either a wave sound you've designed or a sound sample you have loaded. A given track can switch between eight different instruments. When an instrument has been selected, it will be used until another is specified.

The fourth value in the topmost note description, the green 20, specifies the volume level of the note. This position can also contain commands to change the volume as the sound is playing.

The final item, F08, is the effect. There are effects for vibrato, tremolo, portmanteau, and lots of other things that change the sound. There are also commands that can create playback loops or transfer playback to a different row. The F08 command doesn't affect the sound; instead, it sets the speed at which the tracker moves from one row to the next as it plays the tune by setting the number of 'ticks' that the playback will pause between each row.

The sound tracker calculates the length of a tick based on the beats per minute (bpm) setting for a song. The song we have loaded runs at 160 bpm, making each beat last 160th of a minute, and a tick is a fraction of this, usually 6. You can change the bpm value in the menu. The first note sets the speed to 8



ticks, which means that the note will play for slightly longer than the second note, which sets the speed to 5. This is one way to change the intervals between notes. Another way is to leave gaps in the sequence of notes. Once a note starts playing, it will continue until either a different note is played, or a note is reached that contains a stop note instruction which stops playback for that track. The track in **Figure 5** plays a sequence of notes at different levels to get an echo effect which works very well.

INSTRUMENTS OF TORTURE

There are two types of instruments available for tunes. One is a sound sample, and the other is made up of waves generated programmatically. You can see the instrument bank at the top of **Figure 1**. The creators of the tune have used the instrument names for a spot of self-promotion – normally, these names would identify each type of sound. A given instrument can be associated with a sound you design from wave shapes or from a sound sample. You can load a WAV file, select it, and press Smp. Ed to work with it.

Figure 6 shows a sound sample being edited. You can crop the sample, change its volume and how it repeats. Once you've done that, you can press

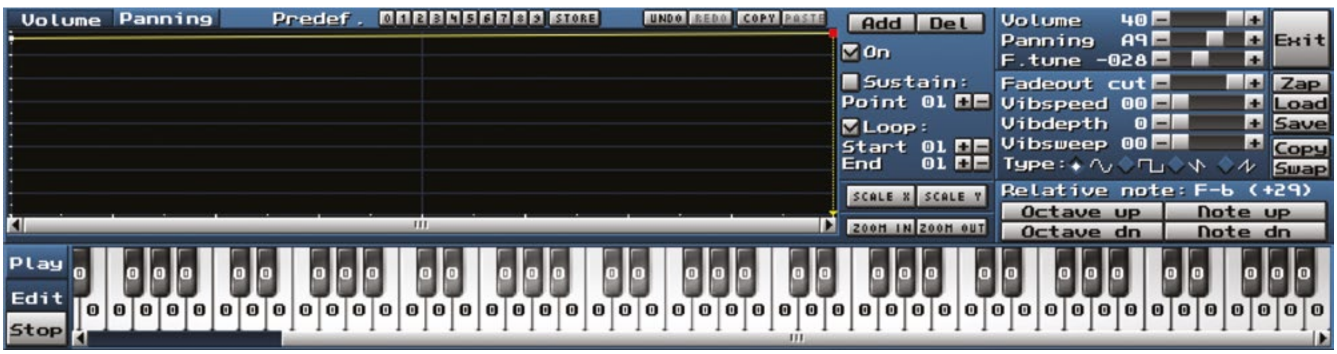
> "The author spent a happy ten minutes creating **strange sounds with the cow-bell sample**"

the Ins. Ed. button to edit the instrument setting. These control how the volume of the instrument changes over a note and let you add vibrato effects. If you want to create your own samples, you can use the Audacity program, which you can get from **audacityteam.org**.

Figure 7 shows the instrument editing page. You can change the way the volume of a note changes as it plays. You can also pan the note left and right and make other adjustments. At this point in the writing of the article, the author spent a happy ten minutes creating strange sounds with the cow-bell sample and then playing *Twinkle, Twinkle, Little Star* with them. This recording has been mercifully lost, however. You can modify the settings as the track is playing, which is great fun. →

Figure 6
This is actually the sound of a cow-bell. Surprisingly musical

Figure 7
You can't use the mouse to play the keys on this screen, but the keys light up when you press notes on your computer keyboard. The bottom row of your keyboard enters the white keys and the next row up the black ones



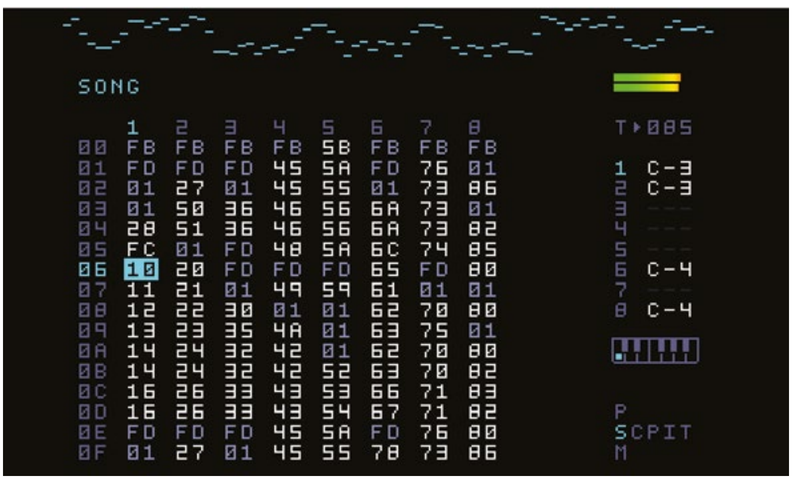
MAKING MUSIC

If you want to record some sounds of your own, you can get a clean slate by pressing the Zap button to clear the tracker. Then load some samples, or design some sounds, put them in instruments, and start entering notes on the keyboard. Pressing the **SPACE** bar toggles between editing settings and storing notes as they are typed on the keyboard. Click a note in a track and then start entering notes to put them into the track. If a note sounds a bit wrong, just go back and enter new values over the top of the existing ones. You can use the undo key (**CTRL**+**Z**) to make changes and then restore the original if they don't work. Blocks of notes can be copied and pasted into new parts of the tune. You can use this to get echo effects. Blocks can also be transposed (moved up and down the scale) so that you can make two tracks that play in harmony.

The manual does an excellent job of describing the various commands. However, the best way to discover what vibrato, tremolo, and portmanteau do is to put some notes into a channel, apply the effects, and listen to what happens. You can create more complex tunes by creating multiple blocks of notes and changing them together. You could make a symphony with a slow movement, a fast movement, and a house movement if you like. And remember what you are doing is art. The only person who needs to like it is you. If anyone else likes it, that's a bonus.

Figure 8 The Dirtywave M8

Figure 9 Raspberry Pi makes a fine Dirtywave host



MAKE YOUR OWN DIRTYWAVE M8

The Dirtywave M8 is a pocket-sized device that brings sound-tracking bang up to date. It is powered by a Teensy 4.1 microcontroller and lets you compose and play sound tracker-style songs anywhere on something not much bigger than an old-school Nintendo Game Boy. It also contains several built-in synthesisers and signal processors. It sounds amazing. People have created complete albums on it. It is such a popular device that, at the time of writing, it is just about impossible to buy one. However, the creator of the Dirtywave M8 has done something very nice. He has made a 'headless' version of the device software which runs on a bare-bones Teensy 4.1 device. All you need is a Teensy (search for Teensy 4.1) and a 32GB microSD card (check for compatible devices here: **hsmag.cc/DWmicroSD**) and you can make a Dirtywave of your own to hang off a USB port on your computer. Your computer provides the screen, keyboard, and audio output, and the Teensy runs the Dirtywave code. There is a web page you can visit to open a connection to the Teensy and you use your computer with it. There are also free-standing programs that do the same thing.

Figure 9 shows the Dirtywave playing a tune. It looks complex, but the user interface is surprisingly intuitive. Although you do need to work in hex. You can find a full description of how to copy the firmware onto a Teensy 4.1 device here: **hsmag.cc/M8HeadlessSetup**.

DIRTY KEYS

The author has created a tiny unit to hold the Teensy and Raspberry Pi Pico, and provided a keyboard which implements the Dirtywave keys.

Figure 10 shows the keyboard. It has two USB sockets which are used to connect it to the host computer. One connects the Pico keyboard and the other the Teensy. The keyboard uses individual key switches which are wired directly to Pico pins. The Pico runs a CircuitPython program that emulates a USB keyboard. Each of the physical keys is mapped onto a particular input:

```
key_switches = [
        Switch(pin=board.
GP18,name="up",ch=Keycode.UP_ARROW),      # up
        Switch(pin=board.
GP19,name="option",ch=Keycode.Z),  # option
        Switch(pin=board.
GP20,name="Edit",ch=Keycode.X),    # edit
        Switch(pin=board.
GP13,name="Left",ch=Keycode.LEFT_ARROW),    # left
        Switch(pin=board.
GP12,name="down",ch=Keycode.DOWN_ARROW),    # down
        Switch(pin=board.
GP11,name="right",ch=Keycode.RIGHT_ARROW),   #
right
        Switch(pin=board.
GP21,name="shift",ch=Keycode.LEFT_SHIFT),   #
shift
        Switch(pin=board.
GP22,name="play",ch=Keycode.SPACEBAR)   # play
        ]
```

The code above shows how the mapping works for each pin. You can change the numbers if you want to connect the keys to different pins. You can find full details for the project, including 3D-printable designs for the case, on the GitHub site **hsmag.cc/USBkeyboardDW**.

Figure 11 shows the contents of the device. The intention was to make the slimmest possible unit. It uses right-angled headers for the connections on the Pico and the wires are wrapped and then soldered to individual pins on the key switches. Something you almost certainly aren't supposed to do in real life. However, it works well enough and is a nice way to practice making music until the real device comes back into stock.

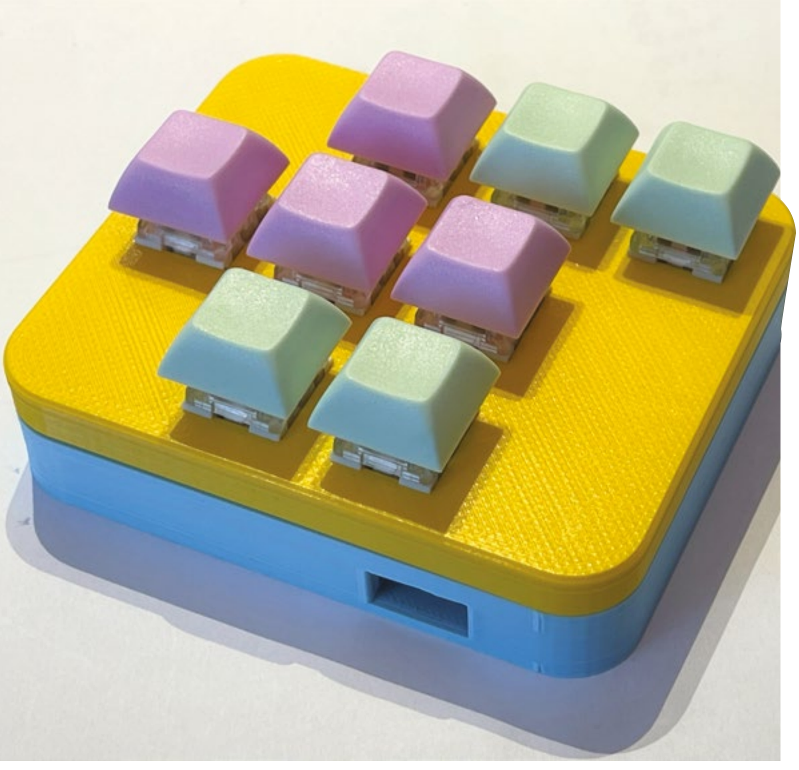


Figure 10
You could use a Pico like this for any situation where you want a custom USB keyboard that only needs a few keys

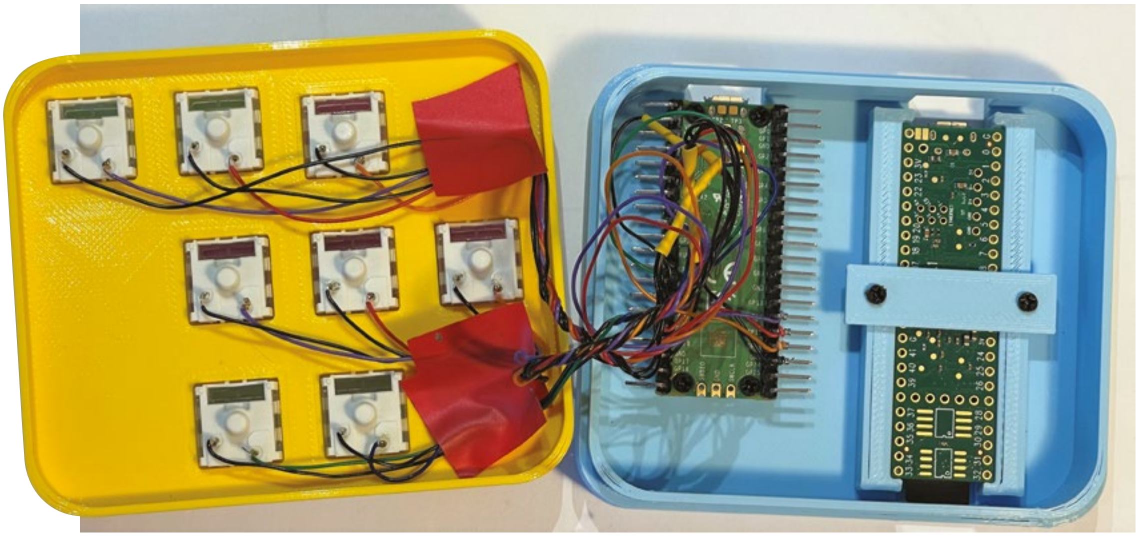


Figure 11
The device on the right is the Teensy 4.1. The one with all the wires is the Pico. There is a slot in the front of the keyboard to provide access to the SD card in the Teensy which is used to hold all the sound files

Design your first PCB

Create some Christmas decorations and learn a new skill

Ben Everard

@ben_everard

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

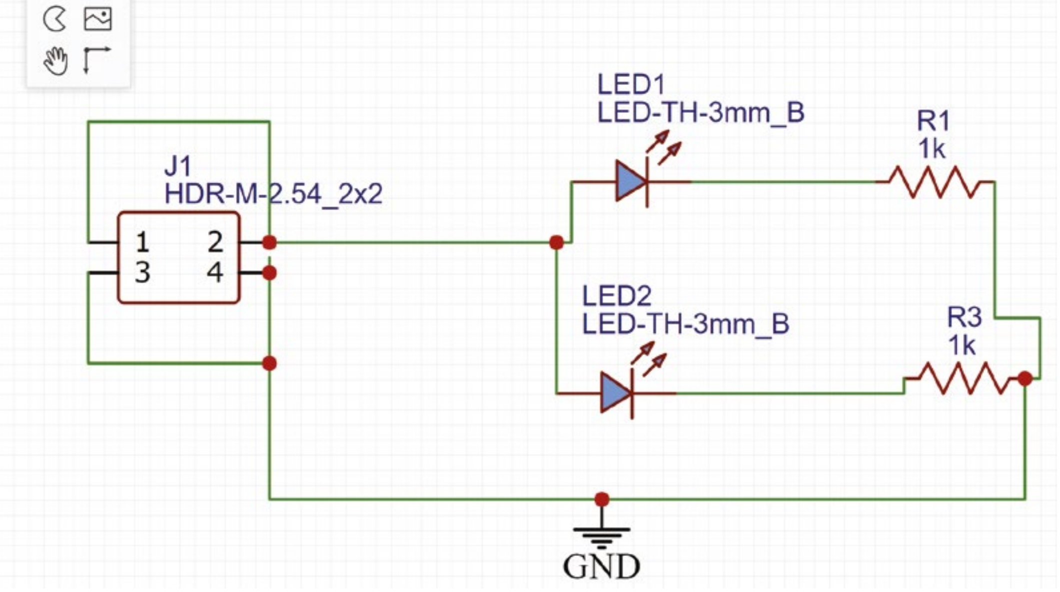
One of the biggest challenges with learning a new skill is having a project to start with. Too complex a project and it can be hard to know where to start, and you can end up going round in circles and giving up before you've really started. Too simple, and it can seem like you're not really getting anything out of it.

In this article, we're going to pick our favourite 'simple but not too simple' PCB project – Christmas lights. It'll be a little PCB that holds one or more LEDs that you can solder up. If you want to, you can add a bit more sparkle by adding addressable LEDs, but the point here isn't to flex your electronics muscles, it's to get used to PCB software and the process of ordering a PCB from a manufacturer.

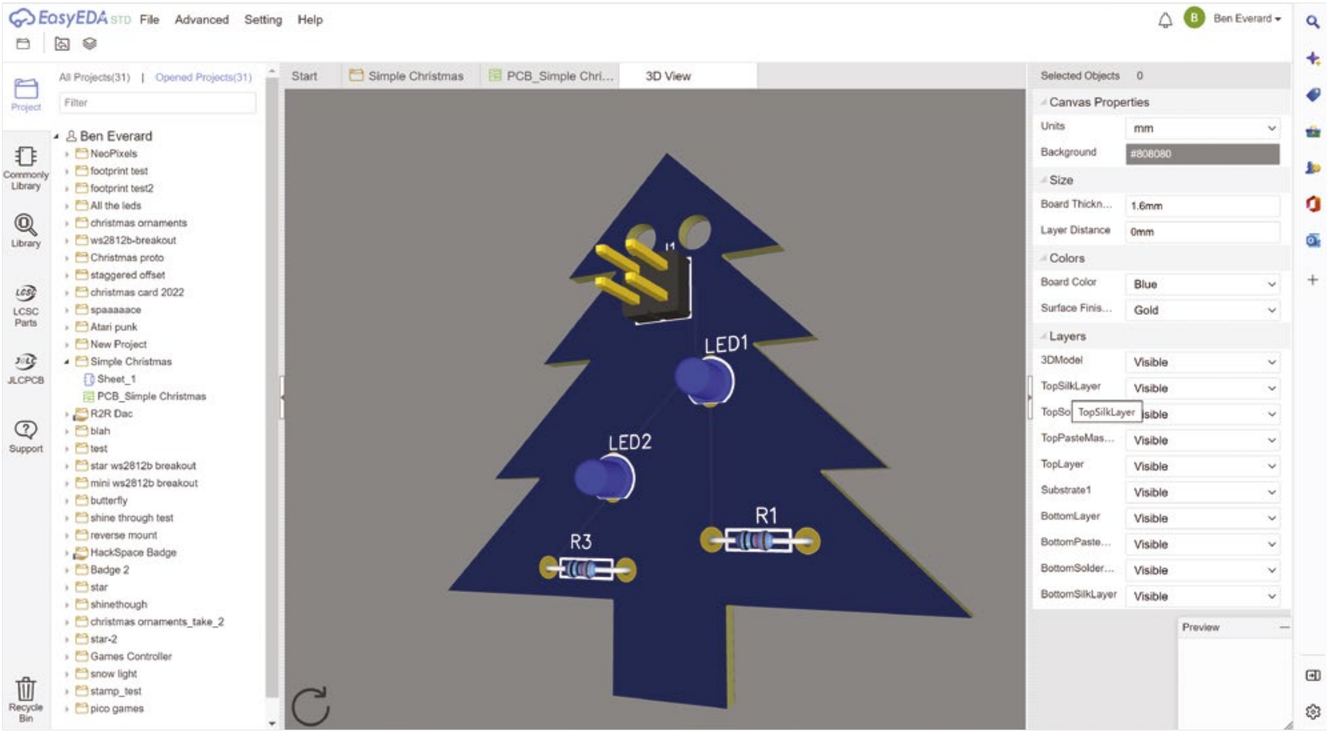
There are a few PCB design tools available (Electronic Design Automation, or EDAs). The open-source KiCad is popular with makers and powerful enough for professional use. EAGLE is a commercial tool with a slightly limited free tier for makers. However, we've found that the simplest tool for getting started is EasyEDA. It's web-based, so there's no need to install anything, and hits the Goldilocks of being simple enough to be easy to understand, yet powerful enough to actually be useful. Most EDAs follow a roughly similar workflow, so if you decide to try others later, your experience with EasyEDA should make that easier.

First, you'll need to head to **EasyEDA.com** and log in, (or create a free account if you don't have one already). Click on EasyEDA Designer, then select 'STD edition' in the pop-up. This will take you into the main window.

There are two parts to creating a PCB. First, you have to design the schematic, and then you have to design the PCB to match this schematic. EasyEDA will let you just create a raw PCB without a matching schematic, but it's hard to make sure you've



Right Figure 1. The simple schematic just connects up the LEDs



Above EasyEDA's 3D function lets you see your designs as they will finally look

connected everything up properly, so we wouldn't recommend this method.

Go to File > New > Project and create a new project. This will create a folder in the project's list on the left-hand side, add a schematic, and drop you into the schematic's editor.

We're going to use the 'commonly used' components, so you can click the Commonly Library on the left-hand side. This gives us quick access to standard components. All these are available from multiple manufacturers and aren't tied to a specific thing. For example, you can buy 3 mm LEDs or 0805 LEDs almost anywhere and they'll fit.

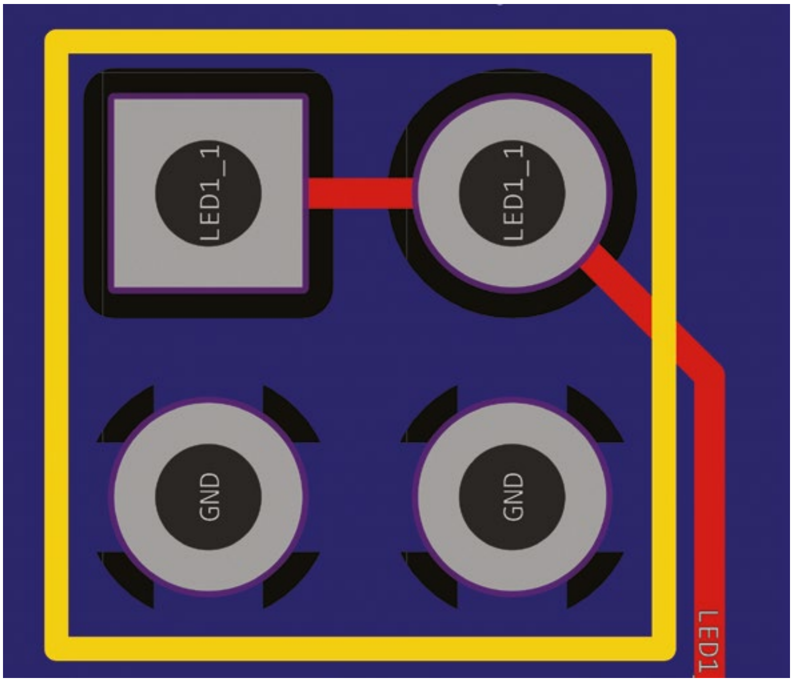
Each component needs a layout on the PCB to accommodate it. This might be pads for surface-mount parts, or holes for through-hole. It might also include PCB cut-outs to make space for the physical shape of the part. This PCB layout is known as the footprint. In EasyEDA, you select the footprint you want when you place a component. For example, we won't just add an LED, we'll add a 3 mm LED. Scroll down until you find the LED symbol, and select the footprint LED-TH-3mm_B (the final _B just means that it's a blue LED and this only impacts the look of the project if in the 3D viewer – you can equally select red or green and it won't affect our PCB). Feel free to use a different LED footprint if you prefer.

Actually, add two LEDs. We also need two resistors. The resistors are a little confusing. You can select either US (zigzag) or EU (rectangle) schematic symbols, and then any footprint within each option. It doesn't really matter which symbol you select. The through-hole options are each given a number from 1.2 to 0.3 – this is the size (in inches) between the two holes. Most 0.125-watt resistors will fit into the 0.3-inch footprint, but it can be a bit of a squeeze with some 0.25-watt resistors. Depending on what you have to hand, you'll probably want one of those two (bear in mind that anything that fits in a 0.3-inch footprint will also fit in a 0.5-inch footprint, but the reverse isn't true).

> **"You can set the value by selecting the resistor and changing the value in the Name box"**

Obviously, you can put any value of resistor into the same footprint, but setting the correct value means that it can appear on the PCB and then make the PCB easier to assemble. You can set the value by selecting the resistor and changing the value in the Name box.

The final piece of the puzzle is a way of attaching our power source. We'll add a header to solder in power leads. How you get the power is up to you. →



Above The bottom pads join to the copper fill, while the top pads are separate

It could be from a bench power supply, or any other source of power approximately 3 to 5 V.

In the connectors section, you'll see a range of different options with different numbers of pins. HDR stands for header, and that's the type we want. Not because we'll actually be adding the headers, but because they give us connections we can solder into. There are socket (F) and pin (M) versions of each, but it doesn't matter which one we want.

We want two connections – one for power and one for ground. However, it can be good to chain these PCBs together like fairy lights, and this is easier if there are two of each, so the HDR-M-2.54_2x2 footprint is a good choice. 2.54 refers to the pin spacing in millimetres, 2.54 mm being 0.1 inches. No, we don't know why resistors are sized in inches and headers in millimetres.

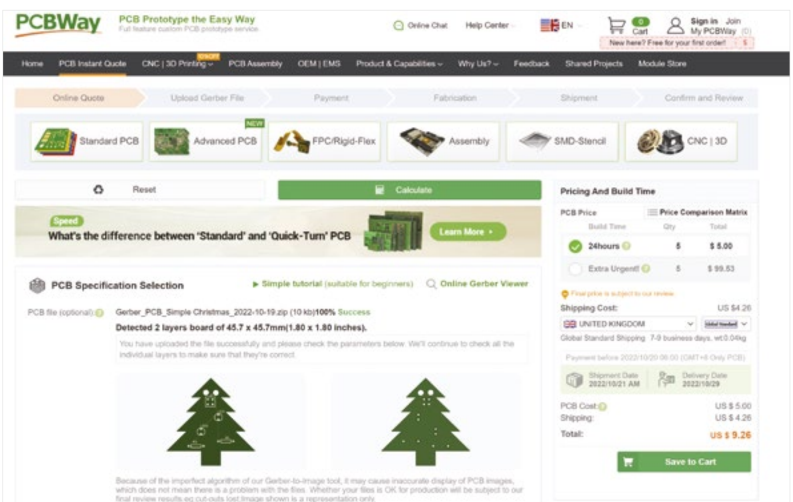
Those are all the parts we need, now it's time to connect them together. There should be a box labelled 'wiring tools' that you can use to select how you want to join it all together. We're going to use the simplest option and use wires. When your mouse is over a component's connection point, you should see a black circle appear. Make sure that you see that when you start connecting everything up. You can also join wires to wires, and if you do so, you'll see a red dot to indicate that the wires are all joined.

The only thing we've not added so far is the net flag. Again, you can select this from the wiring toolbox and add it in the place shown in **Figure 1**.

The connections on a PCB are known as nets. Sometimes a net will just join one component to another – as we see with the LEDs and resistors – but sometimes they're more complex. In our case, there's a single net that joins the output connections from both resistors to the header pins, and this is what we've added the ground flag to. The reason for this will become clear when we design the PCB.

That's our schematic created. Now you can save the file, and we'll start making our PCB.

Below Most PCB manufacturers let you preview the PCB to check it's correct



PUTTING THINGS INTO PLACE

Go to Design > Convert Schematic to PCB. There'll be a pop-up asking if you want to check nets. This just basically looks to see that everything is connected to something. This is just a quick sense check – there are many perfectly valid schematics where not all things are connected to things, and there are plenty of ways of messing up a schematic that this won't pick up. However, it's a good sense check.

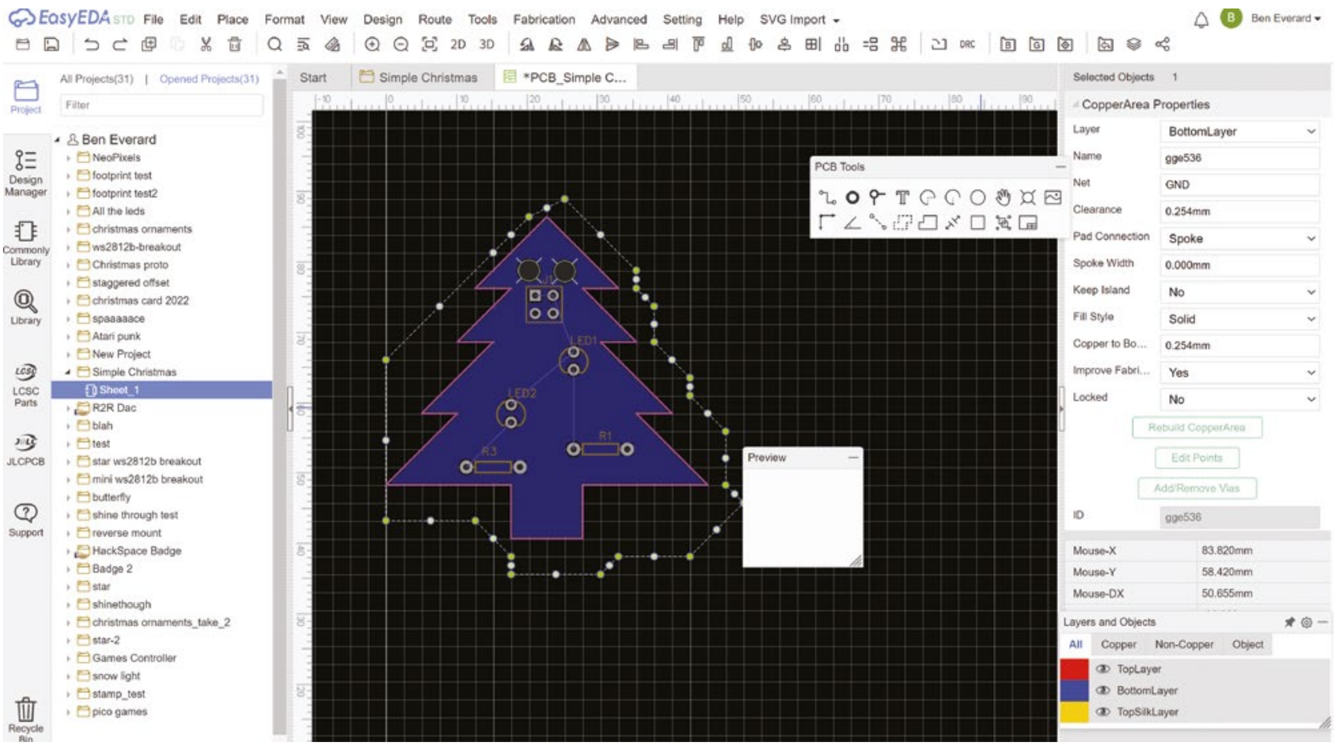
If there are any problems, it won't generate the PCB and it'll show you a list with yellow warning triangles on the unconnected nets. Click on them for more info.

Once you've sorted this out, you can click on Design > Convert Schematic to PCB again and generate your PCB.

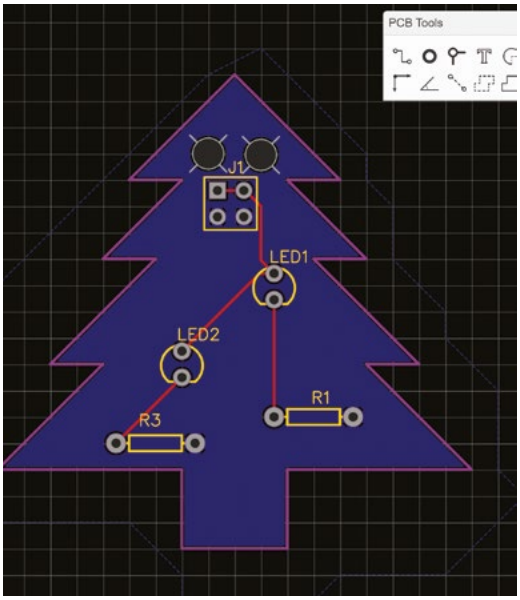
You'll get a pop-up suggesting making a board outline for the PCB. Usually, this is pretty useful and it'll take a guess at the right size. However, we're going to create ours manually a bit later, so click on Cancel.

If it does create a board outline (a rectangle in purple) anyway, highlight it and then press **DELETE** to remove it.

Next, click on the background and this will let you set the canvas properties. You can set the units to either thousandths of an inch (mil) or millimetres



(mm). Pick the one that works for you. You can also set the snap size. When doing larger placements, we often use a snap size of 2.54 mm (100 mil). This lines up with the default grid squares and is the same as the pin spacing on many PCBs. For finer things, you might have to set it smaller.



Above You should be able to route the whole PCB without needing to cross wires

Now, draw the outline of your PCB. In the Layers and Objects box, make sure the pencil is on the board outline layer, then use the Track tool to draw the shape of your PCB. We've opted for little Christmas trees, but you can do whatever shape you want. If you want to get really artistic, you can import images

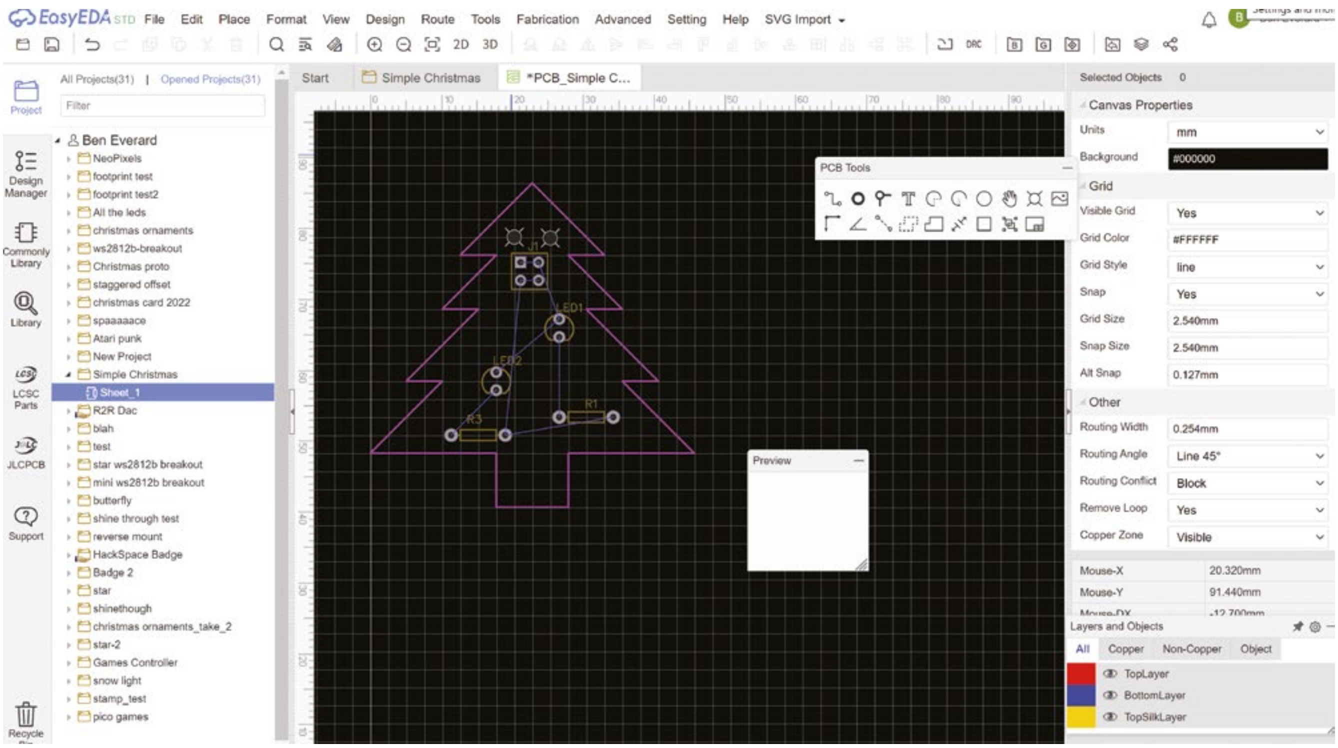
> **We've opted for little Christmas trees,** but you can do whatever shape you want

– there is the ability to do this built into EasyEDA, but it doesn't work very well. The SVG Import Extension (**github.com/xsrf/easyeda-svg-import**) is a much better option.

Other than the components that should have been pulled in from the schematic, the other thing we want on the PCB are two 3 mm holes near the header pins. These are to let us use a cable tie to attach the power cables to the board – this makes the power cables much more secure. Use the Hole tool to place two holes, then press **ESC** to revert to the Selector tool, and then select each hole and adjust the diameter to 3 mm (or whatever size cable tie you have to hand).

Now, place your components wherever you want them. For a simple PCB like this, you can probably put them anywhere, but for more complex PCBs, you →

Above The copper should include the whole of the PCB



Above The different colours represent different aspects of the PCB design

might have to think a little about how you can route all the connections.

We're almost ready to start connecting up our PCB but, before we do, we're going to convert the entire bottom layer of our PCB to a ground plane. This is a big copper area that's all connected to ground. We do this for a few reasons – but the main one for PCBs is a little complex. If you have an electrical signal that changes very rapidly, it can emit electrical interference that can affect bits of the PCB that aren't connected to it. Adding a ground plane helps mitigate this. That's a very simplistic explanation of a complex phenomenon that we're not going to delve into further, but ground planes are good and while not really necessary on this PCB, they tidy up the traces and are a good habit to get into.

> **Ground planes are good and while not really** necessary on this PCB, they tidy up the traces

We'll add ours to the bottom layer, so select the bottom layer in the Layers and Objects box, then select the Copper Area tool (it looks like two dashed rectangles). Draw around the edge of the PCB. It doesn't have to be precise, just make sure your entire PCB is inside the copper area. Once it's done, it'll calculate the copper area and you should see your PCB fill with blue. If you look very closely, you'll notice that some of the pads are connected to the blue copper area while some aren't. You should also note that some of the light blue 'rats nest' lines that indicate our connections have disappeared because those connections are now 'routed' by the copper fill.

This copper area won't automatically recalculate if you move components, or do anything that will affect it. If you need to at any point, select the copper area and click 'rebuild Copper Area' in the right-hand panel.

The final thing to do is route the rest of our traces. First, make sure that the top layer is selected, then use the Tracks tool to join up all the necessary pads (that are linked by pale blue lines). You'll probably find that you can do all this in just the top layer, but if you've got a bit complex, you might find that you can't do this. If so, you need to switch to the bottom layer and add some track there. All our components are through-hole, so you can connect to them on either layer.

You can also link top and bottom layers using a 'via'. These are holes through the PCB that you can connect to from either layer.

Once you've fully routed your design, save your work (if you haven't already), and you can have a look at a 3D render of it by pressing the 3D button.

ORDERING YOUR PCB

Now it's time to order your PCBs. This is the part that many people find daunting, partly because there are lots of options, and partly because it involves parting with real money.

We need to generate a 'Gerber' file, this is a standard set of files in a zip archive that is supported by almost every PCB manufacturer out there. There are some slight differences in how Gerbers can be packed, but the output from EasyEDA should work with most fabricators.

Go to Fabrication > Generate Fabrication Files (Gerbers). There'll be a pop-up asking if you want to check DRC (Design Rules Check). This is a way of checking that your PCBs are actually manufacturable and don't include things like traces too close to holes. It's a good idea to check this, but bear in mind that we haven't taken the time to set the design rules that it's checking. There are plenty of options and we're just going with the defaults.

Hopefully, DRC will pass and you'll be taken on to the 'download Gerbers' page. Click on Generate Gerber and you should download a zip file.

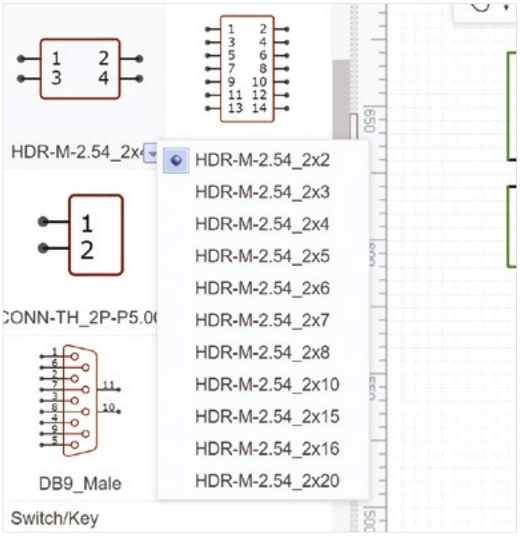
There are hundreds of PCB manufacturers to choose from, and they vary quite a lot in price. Most will offer an online quote where you can upload your Gerbers and it'll tell you straight away what the price will be. Do bear in mind that the shipping can often cost more than the PCBs.

On the PCB order form, there will be a huge range of options and you can usually accept the defaults on all of them. The one exception is that many default to a leaded surface finish, while we'd recommend a lead-free option.

Typically, you'll have to have at least five copies of your PCB, but see what it says on your manufacturer's website.

A few options you might see on the PCB order page:

- **Material:** FR4 – this is the standard type of fibreglass PCBs are made from.
- **Thickness:** It doesn't really matter for this – about 1.6 mm is normal.



Left Use the drop down arrow to select variations on components

- **Solder mask colour:** The majority of your PCB will be covered in this. Green is the most common option, but they all work the same.
- **Minimum track spacing:** This is unlikely to be a problem with a simple PCB (and DRC will complain if you've done anything strange).
- **Minimum hole size:** This is unlikely to be a problem.
- **Surface finish:** Copper corrodes very quickly, so any exposed copper is usually covered with a finish that's either silver-coloured (HASL) or gold-coloured (ENIG). HASL – which stands for Hot Air Solder Levelling – is far cheaper. We'd strongly recommend going with lead-free HASL.
- **Via process:** If you've not added any vias, this is irrelevant, but if you have, you probably want them tented (i.e. covered in solder mask).
- **Copper weight:** This is how much copper is on one square foot of PCB. 1 oz is pretty common, and perfect for our use case.

With this selected, you can place your order. Perhaps the hardest part of ordering a PCB is waiting for delivery. This can often take two or three weeks. After this, you can solder them up and see if they work. With the double-header on the top, you can daisy-chain them together to make a Christmas decoration to show off your new-found PCB design skills.

Build a MicroPython handheld

Jump into hardware hacking and gamedev at the same time with a purpose-built handheld that uses MicroPython as its game development platform

MAKER

K.G. Orphanides

KG is a writer, developer, and software preservationist with a particular interest in projects that allow everyone to get hands-on with their own games and technology.

magpi.cc/kgorphanides

You'll Need

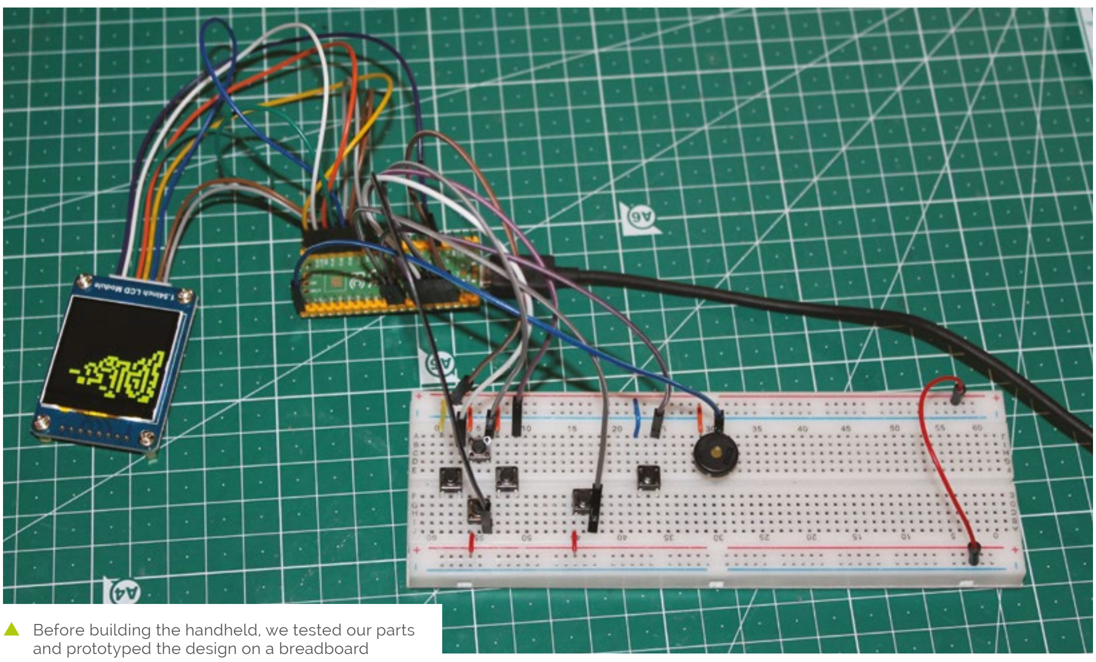
- Rotary tool with cutting disk and 2 mm drill
- Soldering station
- Flat face cutters
- Wire strippers
- Optional: 3D printer
- Parts: See list

YouMakeTech's Vincent Mistler has produced some of our favourite Raspberry Pi Pico builds. This month, we're going to take a step-by-step look at building the 'Raspberry Pi Pico GameBoy' – a MicroPython-based handheld console that provides huge scope to develop your own games.

Vincent details the equipment you'll need and provides a video of the assembly process at **magpi.cc/buildpicogb** that you'll definitely want to look at. We've made a few tweaks and broken down some undocumented steps to make the build extra easy, even if it's your first time making a project like this.

01 Prepare your hardware

To build our Pico-powered handheld, we'll use a 1.54-inch 240×240 Waveshare display module with an embedded ST7789 screen controller and a 3D-printed case designed by



Before building the handheld, we tested our parts and prototyped the design on a breadboard

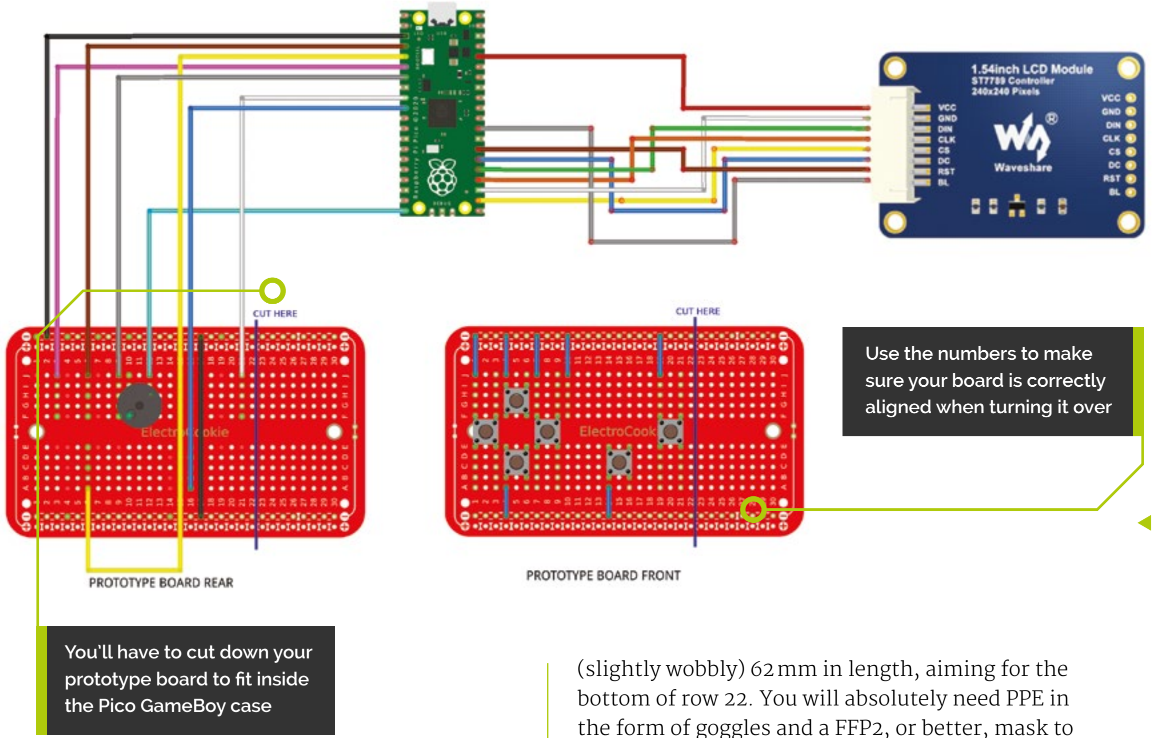


Figure 1 The wiring diagram for the handheld

YouMakeTech. We'll build our own control input and audio board. This calls for six 4-pin 6×6×6 mm tactile button switches and a piezo buzzer to make noises with, plus solderable prototyping breadboard and wire – see our shopping list (overleaf) for details.

02 3D-print YouMakeTech's case

This project starts with YouMakeTake's STL files (**magpi.cc/picogbstl**) for the handheld's case, available to buy for £4 from Cults3D. The STL files include the top and bottom of the case, a D-pad, and a button - remember to print two of those!

If you're going to print these yourself, you'll find recommended settings at YouMakeTech's site (**magpi.cc/buildpicogb**). If you don't have a 3D printer, you can order them from a printing service. We ordered from a local printer via Cults3D's partnership with Craftcloud, which allowed us to directly send the STL models to the service before choosing the materials we wanted. We opted for PLA with a standard finish.

03 Cut your prototyping board

Once you've got the case, you can cut your prototyping board to fit the available space in the handheld's case. We cut our board down to a (slightly wobbly) 62 mm in length, aiming for the bottom of row 22. You will absolutely need PPE in the form of goggles and a FFP2, or better, mask to help protect you from particulate matter. The best tool for the job is a Dremel with a cutting disc. A clamp isn't essential for this cut, but it'll make your life safer and easier. We used uPesy's medium solderable prototyping board, which measures 87 mm × 53 mm. ElectroCookie's solderable breadboard has similar dimensions, and is more readily available in the UK.

04 Measure and drill

You'll almost certainly need to drill additional 2 mm holes. This is so that the board can be mounted to a set of four pillars located inside the top half of the handheld's case, just below the hole for the display. Your breadboard will slightly overlap the back of the display when both are fitted. We needed two extra holes, positioned just at the number markers for row 21 on our board. This will be different on other brands, and you may need to drill all four holes on some. We took measurements against the 3D-printed case with the display in situ, made a template, and marked up the board with a paint pen. Wear your mask and goggles when drilling.

05 Buttons galore

We're using both sides of our solderable breadboard. On the front, we'll position six 6×6×6 mm, 4-pin, tactile button switches to line →

Danger!
Cutting tools

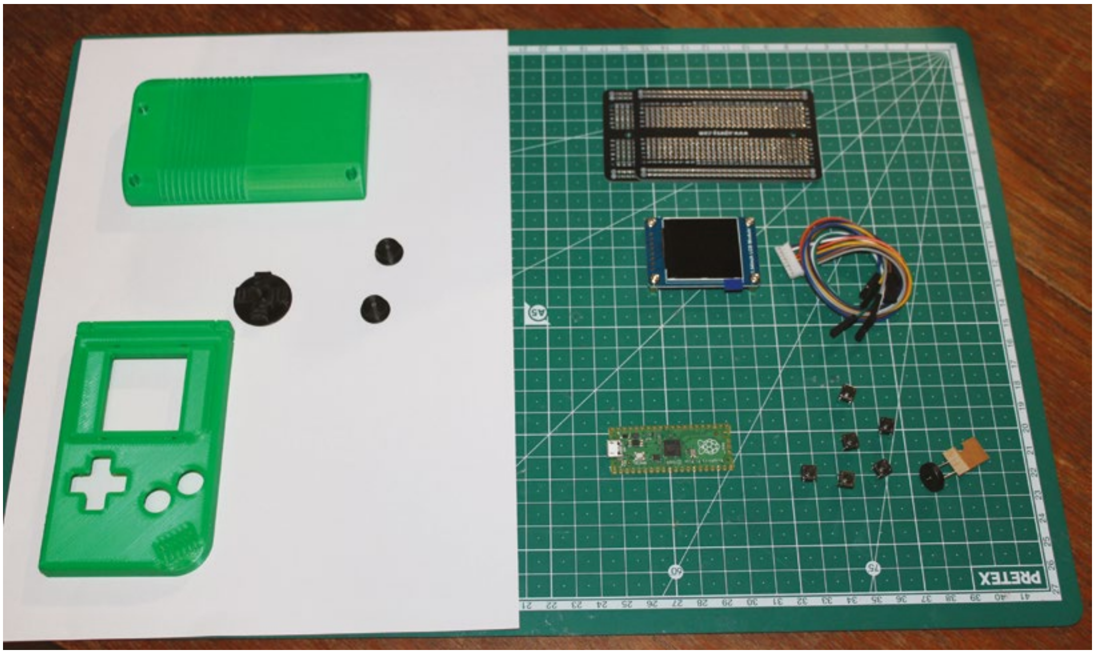
When cutting and drilling PCBs, wear protective goggles and an FFP2 or better mask.

magpi.cc/cuttingtools

Top Tip

Tape up

When cutting and soldering, use heat-proof masking tape to mark and protect adjacent parts of the boards.



This build doesn't use that many parts, but it's a good idea to order extra prototyping boards in case of cutting errors

Top Tip

Prototype and test

We prototyped this build with a Pico H on standard breadboard before we started, and tested at various points during assembly to be sure everything was correctly connected.

up with the up, down, left, right, B, and A buttons. We strongly recommend carrying out fit tests before soldering to ensure that the buttons and D-pad on your board will line up with the holes in the case. The front of the board is also home to most of our ground connections, including for the piezo buzzer fitted to the rear of the board. We used short pre-cut jumper wires for these, but you can cut your own from solid core wire.

06 Connect controls to Raspberry Pi Pico

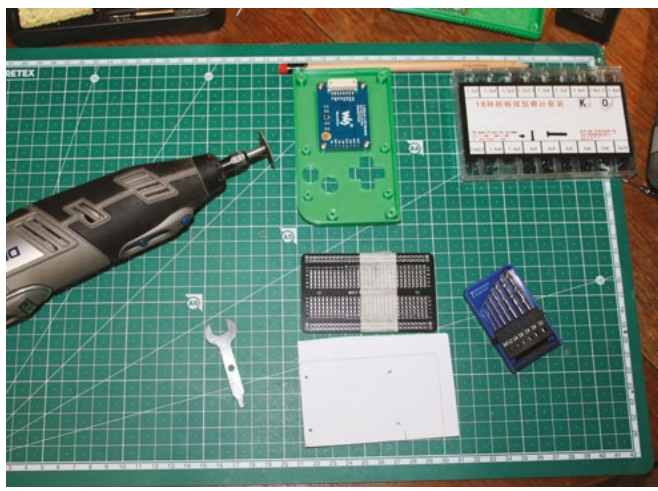
We show both sides of the board separately in our diagram (**Figure 1**). The piezo buzzer goes on the rear so it doesn't obstruct assembly, although we've connected it to ground on the front. Also soldered at the rear is a wire bridging the upper and lower ground rails on the prototype board, soldered between the two negative rails on row 17. Most importantly, all the wires connecting the prototype board to Raspberry Pi Pico are also soldered at the back. We used solid-core wire to connect our input controls to Pico, at a length of around 80 mm. A different wire colour for each connection will help keep track of them, and you can use a helping hand to hold components in position while soldering, but our build went well without either.

07 Connect the display

Although the Waveshare display we're using has though-holes for soldering, we're going to use its on-board connector, and the connector cable it ships with, for use with headers. Snip the individual DuPont female connectors from the end of each wire and solder them to Raspberry Pi Pico. These wires are stranded, so you'll want to twist or lightly tin them with solder to make them more manageable before soldering them to Pico. Follow our Pico pinout connection chart (**Figure 2**) for where to solder each wire.

08 Mount the display

The Waveshare display ships with four bolts and four brass standoffs attached to each of its



Carefully measure and mark up the board before cutting, and don't forget to wear goggles and a mask to protect you against particulate matter bolt holes. Remove these and hang on to the bolts. Turn the top half of the Pico GameBoy case upside down and line the display up with the case's four integrated plastic standoff pillars. The LCD display should face outwards, through the hole in the case, and the connector should face upwards. Use the bolts we set aside earlier to fix the display to the inside of the case. If you have any issues with standoff holes being misshapen or obstructed as a result of the 3D printing process, use a 2 mm drill bit to open them out before inserting the bolts.

09 Mount Pico and its controls

The mounting standoffs for Pico are on the rear half of the case. Use four 4 mm M2 bolts to screw the microcontroller into position, with its micro-USB port lined up with the hole in the case. With both display and Pico in position, it's time to mount the handheld's controls. If everything went well in step 3, our prototype board should be perfectly cut and drilled to fit the four standoffs on the case, just below the display. Drop your 3D-printed D-pad and buttons into their holes, then lower the control board into position so that its buttons line up with them. Ensure that none of the wires connecting the board to Pico are caught against the front of the case, and screw the board into the standoffs using four 6 mm M2 bolts.

10 Seal the case

Plug the display connector that you soldered to Pico into the Waveshare display, then carefully route the wires from this and the controller board to allow the rear shell of the case to be closed shut against the front half of the case. A lip creates a firm seal here. With the two halves properly fitted, drive four 8 mm M2 bolts into the holes that connect the back of the case bottom to the case top.

11 Transfer the files

We have a handheld! Now we need some software to run on it. Connect the micro-USB port to a Raspberry Pi or other computer with Thonny installed. Download MicroPython for Pico from **magpi.cc/rp2pico**, and copy it to the handheld using a file manager. At the command line, `apt install git` if you don't have it, then: →

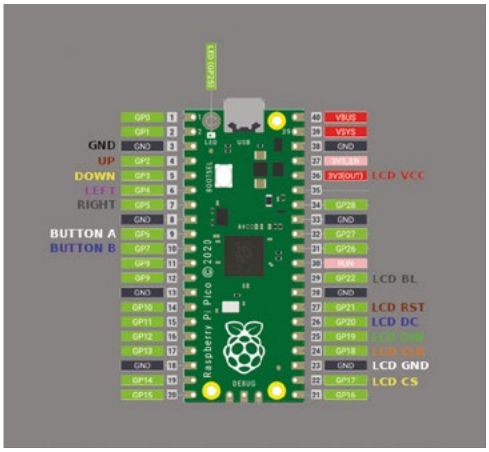
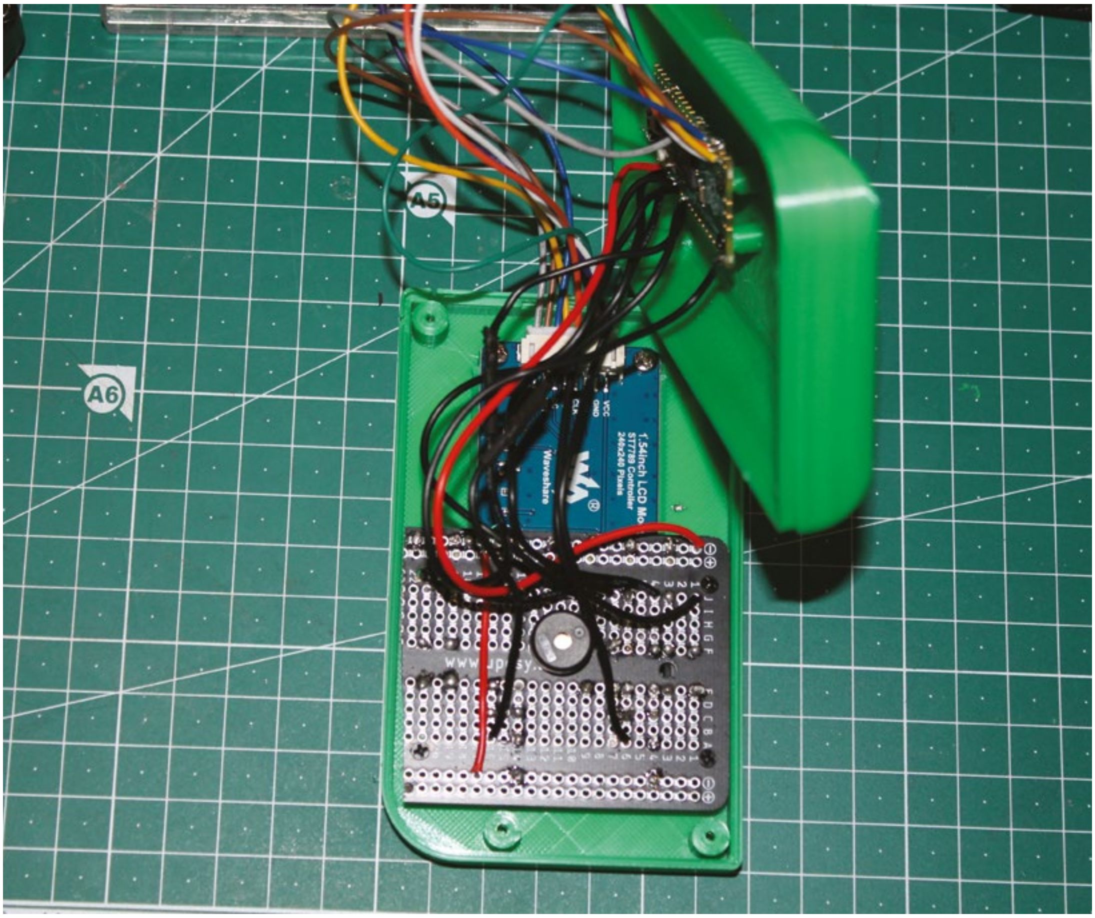


Figure 2 Both the controller board and LCD display are connected to Raspberry Pi Pico

Shopping list

Item	Price
Raspberry Pi Pico GameBoy STL (£4). Only available from: **magpi.cc/picogbstl**	£4
Solderable prototyping breadboard. Get one that matches traditional breadboard configurations. Medium-sized uPesy and ElectroCookie boards meet the spec. **magpi.cc/prototypeboard**	£12 for five
Waveshare 18079 1.55-inch LCD Display Module. **magpi.cc/154lcd**	£10
4-pin tactile button switches 6×6×6 mm. You'll need six. **magpi.cc/tactilebutton**	£2 for 20
Piezo buzzer. We used a TDK PS1240P02BT. **magpi.cc/piezo**	£5 for ten
Solid core electronics wire. Multiple colours are useful, but not essential. **magpi.cc/elecwire**	£15 for multipack
Lead-free solder. 0.5 mm or your preferred thickness.	£7
M2 bolt kit. You will need four 4 mm, four 6 mm, and four 8 mm M2 bolts.	£10
Optional: Pre-formed jumper wire kit. **magpi.cc/jumperwire**	£10 for 560 of various lengths
Optional: 3D printer filament. Sunlu PLA+ is recommended.	£16 per 1 kg spool



▶ With all the parts bolted into the case, all that remains is to carefully tuck your wires inside before closing and bolting it together

▼ The complete handheld requires external power, but it looks and feels very polished and is ready to develop games for


```
git clone github.com/YouMakeTech/Pi-Pico-Game-Boy
```

This repository includes library files to support development for the handheld, a version of Tetris by Vincent of YouMakeTech, and versions of the Game of Life and Flappy Bird by his son Matthieu Mistler.

12 Time to play

Open Thonny, making sure that it's in regular rather than simple mode, and that you've selected the MicroPython interpreter for Pico at bottom right. In the View menu, select Files, then use Thonny's file manager to drag all the files from the repo over to Pico. Open **tetris.py** from Pico and re-save it to the microcontroller as **main.py.** Press play to try the game. Pico's limitations give you clear parameters for developing your own MicroPython handheld games. YouMakeTech's games provide an example of what you can do with the console, from defining game areas and making sounds, to loading smart-looking title graphics like **tetris_title.bin**. □

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 110

ENGINEER SPANNERS

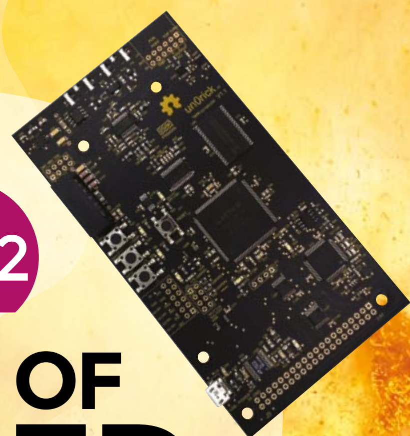
Small spanners for fiddly gadgets

PG 112

KOBRA NEO

A feature-packed printer with an entry-level price

PG 102



BEST OF BREED

Expensive dev boards for your Christmas list

ONLY THE BEST

Expensive and exotic development boards

A collection of dev boards for advanced DIYers and professionals

By **Marc de Vinck** @devinck

In this month's Best of Breed, I'm going to look at a few of the more exotic and expensive development boards and kits on the market. I know what you're thinking! Just about everything on the market is expensive these days! Sad but true. Especially when buying from third-party vendors. In fact, it seems like you've hit the lottery when you are able to source a microcontroller board for the manufacturer's suggested retail price. I just hope that trend changes soon.

Unlike many of the affordable development boards out there, these boards are generally available, but at a very different price point. And just to clarify, I'm not talking about the expensive development boards like a VORAGO Technologies RH-OBC-1 dev board that comes in at around $11,000, or a Virtex UltraScale+™ VCU129 board for around $20,000. I'm looking at significantly more affordable development boards and kits that all come in at roughly $500. Still expensive, but all within reach for more advanced DIY electronic enthusiasts or professional developers.

And if you're wondering why some of these boards are so expensive, I think what you really need to ask yourself is why are some of the other boards so incredibly inexpensive. Companies can spend thousands and thousands of hours developing a new hardware platform, and they only end up selling a few hundred development boards. How do you

> "Many of the people who buy such expensive **dev boards still find them to be of great value**"

justify a dozen engineers' time over the course of months or years to develop a new product? Those boards are going to cost a lot! And believe it or not, many of the people who buy such expensive dev boards still find them to be of great value. Especially when you consider the amount of time they will save you when it comes to developing new electronics.

DIGI ConnectCore® 8M Nano Development Kit vs Open ice40 Ultrasound Imaging Dev Board

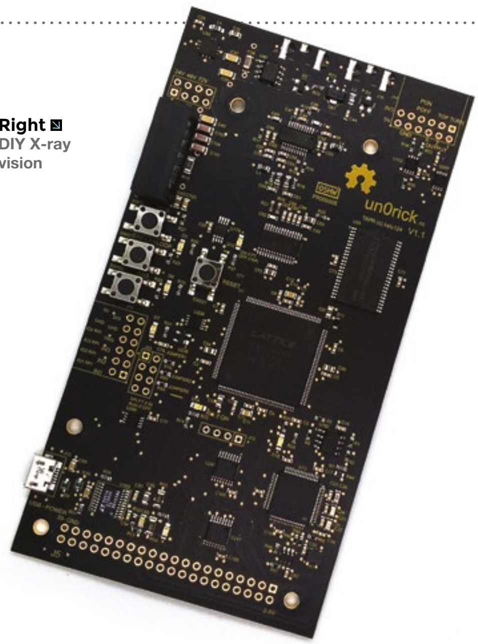
DIGI CONNECTCORE® $658.95 | sparkfun.com

OPEN ICE40 $489 | tindie.com

The Digi ConnectCore® 8M Nano Development Kit allows users to design and build a custom connected System-on-Module (SoM) device that is robust enough for industrial applications and in a single-board computer platform format. It features scalable Arm Cortex-A53 and Cortex-M7 processors, capable of rich multimedia output and processing via various connected devices.

The development board also includes two Digi XBee® connectors to enable cellular, short-range, or long-range wireless connectivity. It features multiple connectors for adding cameras, displays, interfaces, or external audio devices. It's a very powerful board that can be used as the basis for medical devices, executing machine learning code, or edge computing tasks, all without the risk and time of developing your own board from scratch. If you are familiar with Linux, especially the Yocto Project, be sure to check out this dev board.

Above Design your own computer



Right DIY X-ray vision

Coming in at just under the target price point of $500, the Open ice40 Ultrasound Imaging Dev Board is the perfect example of cost versus value of a dev board. This board is a bargain! Why? Well, first, how many other boards are out there that allow you to do non-destructive ultrasonic testing? Let's just say not many! And how long would it take you to develop a board on your own? A long time!

The Open ice40 Ultrasound Imaging Dev Board is a single-channel, ultrasound, pulse-echo design with high-voltage pulser, TGC and ADC, plus 8MB of on-board storage. It's part of a larger kit with several other specific hardware components required before you can do any non-destructive testing. Head on over to the link to learn more about this unique board. →

VERDICT

DIGI ConnectCore® 8M Nano Development Kit

Got Linux? Then check out this dev kit!

9/10

Open ice40 Ultrasound Imaging Dev Board

A unique board that's hard to compare.

9/10

BEST OF BREED

Intrinsyc Open-Q™ 865XR SOM Development Kit

INTRINSYC OPEN-Q™ $1228 | sparkfun.com

Intrinsyc Open-Q™ 865XR SOM Development Kit is an exposed board platform, powered by the ultra-compact Open-Q 865XR and based on the Qualcomm SXR2130P processor. It breaks out everything you would need to develop new hardware based on the popular Qualcomm processor for your next cellular or Internet of Things project.

The board is laid out in an open-frame Mini-ITX form factor and has been designed for developing projects around cameras, voice control, machine learning, radio frequency, and thermal power optimisation. Qualcomm has a long history of developing innovative electronics and has a robust ecosystem and documentation. It would be hard to not recommend this dev board for someone who is working on developing hardware that runs Android. With this board and every other product in the round-up, you really need to head over to the website for more information.

Left Design your own Android

VERDICT

Intrinsyc Open-Q™ 865XR SOM Development Kit

A robust ecosystem for SOM developers.

10/10

MYIR Tech FZ3 Deep Learning Accelerator Card

MYIR TECH $494 | sparkfun.com

AI and deep learning is certainly a hot topic in the electronics community. The MYIR Tech FZ3 Deep Learning Accelerator Card allows you to explore the incredibly complicated world of object detection and identification. And speaking of complicated, keep in mind this is not really a very beginner-friendly piece of hardware. But if you are a more advanced programmer, it does have some interesting-looking specs.

The MYIR Tech Card features a Xilinx Zynq® UltraScale+™ ZU3EG MPSoC, a 1.2GHz quad-core Arm Cortex-A53 64-bit application processor, a 600MHz dual-core real-time Arm Cortex-R5 processor, and a Mali-400 embedded GPU. It's ready to run PetaLinux and supports Baidu's PaddlePaddle deep learning AI framework.

The card also has integrated 4GB of DDR4 RAM, 8GB of eMMC, 32MB of flash, and 32kB of EEPROM. You can expand the board with USB 2.0, USB 3.0, a DisplayPort, Gigabit Ethernet port, a PCIe interface, JTAG, and more. It's a lot to digest, but if you know what you are doing when it comes to deep learning, this just might be a board you should investigate. Make sure to check out the MYIR website for some example code and more documentation.

Below Built to learn

VERDICT

MYIR Tech FZ3 Deep Learning Accelerator Card

Get lots of AI in a small space.

8/10

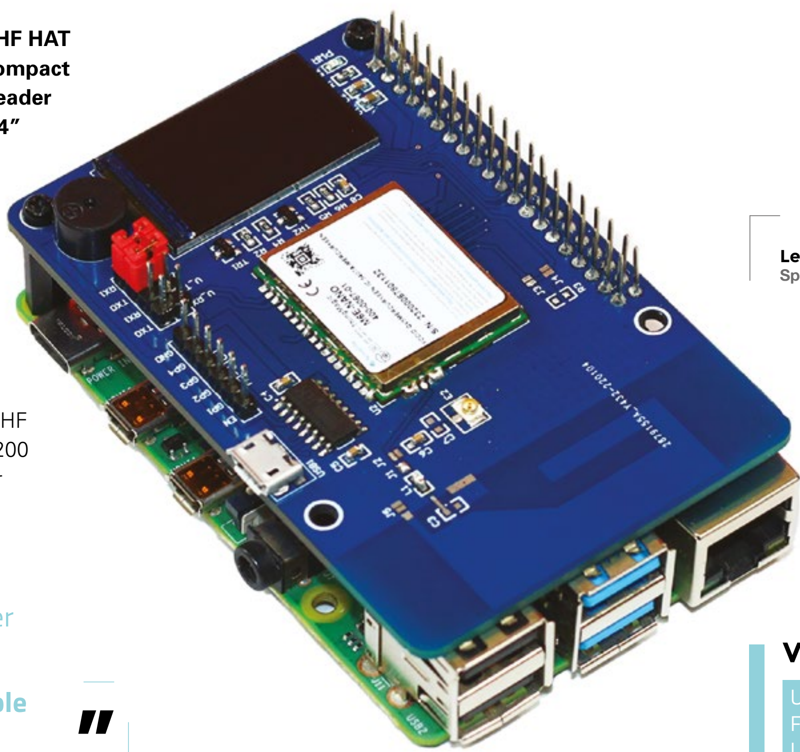
Ultra High Frequency (UHF) HAT for Raspberry Pi

SB COMPONENTS ◈ **$459** | **tindie.com**

Ultra High Frequency UHF HAT for Raspberry Pi is a compact and low-power RFID reader with an integrated 1.14” LCD display that was designed by SB Components in the UK. You may be thinking, “why is this RFID reader so expensive?”And the answer is simple. It’s not just an ordinary RFID shield for your Raspberry Pi. This is an ‘Ultra High Frequency’ RFID reader that integrates the powerful ThingMagic® M6e Nano UHF RFID reader module, allowing for up to 200 tags per second to be read. Yes, 200 per second! That’s not something you can typically do with a more DIY-oriented and less expensive RFID tag reader.

> “You may be thinking, ‘why is this RFID reader so expensive?’ **And the answer is simple**”

The board also features an on-board programmable 1.14” LCD display, a small buzzer, a jumper that allows you to switch between GPIO and USB communications, and an on-board antenna. The board can be configured for frequencies compatible with the Americas, European Union, India, Korea, Australia, China, and Japan. It supports both the EPC Gen2V2 and ISO 18000-63 standards. →



Left ◪
Speed tagging

VERDICT

Ultra High Frequency UHF HAT for Raspberry Pi

Industrial performance RFID.

8 /10

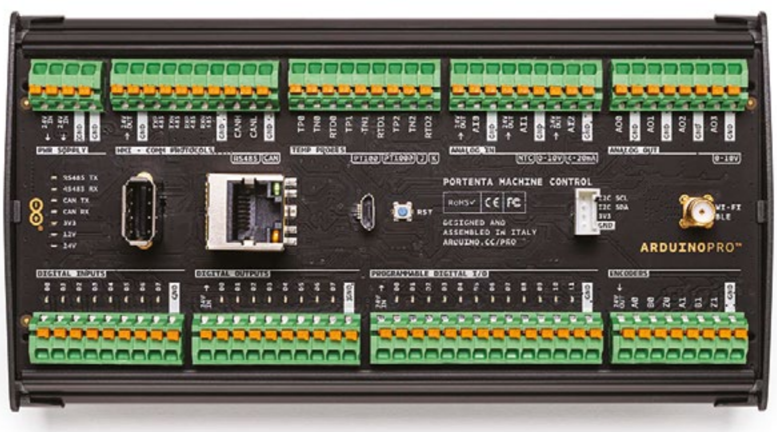
Portenta Machine Control

ARDUINO ◆ **$358** | **store-usa.arduino.cc**

When you think of the Arduino environment, you don't necessarily think, expensive. In fact, you most likely think the opposite. And in most cases, you'd be correct. Arduino boards are generally inexpensive, but they get a bit pricier when moving to industrial applications.

The Portenta Machine Control by Arduino allows you to add industrial Internet of Things (IIoT) capabilities to standalone industrial machinery with the same logic and language you've come to love when programming your Arduino microcontroller. The system features an STM32H747XI dual 32-bit Arm Cortex-M7 and Cortex-M4 MCUs with eight digital 24 V inputs, two channel encoder readings, three temperature probes, and three analogue inputs. It also features eight digital outputs, four analogue outputs, and twelve programmable digital I/O at 24 V logic.

The designers have loaded it with multiple communication capabilities too. It has on-board CAN bus, serial ports (232/422/485), Ethernet, USB, WiFi, and Bluetooth Low Energy. It's a low-power industrial control unit that allows for a wide range of applications. You can leverage AI to do predictive maintenance, collect real-time information on mechanical processes, or control machines via the cloud. ◻



Above ◆ All the power of Portenta and easy access to the I/Os

VERDICT

Portenta Machine Control

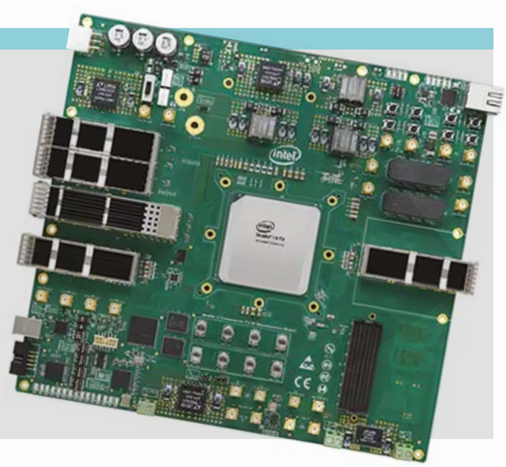
Add Industrial IoT capabilities with the ease of Arduino.

8/10

DK-SI-1STX-E-A

INTEL ◆ **$12,000** | **hsmag.cc/stratix**

If you think these dev boards are a bit expensive at around $500, have a look at what you, or more likely your company, will have to spend to do a little dev work on a Stratix® 10 TX FPGA. At $12,000, you might think that it's too expensive, but if you want to be at the leading edge of developing applications in 5G, you very well may need to pick up this dev kit. And why so expensive? Try to imagine all the engineers who spent countless hours developing the electronics to make this board possible. It might start to sound like a bargain!




FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
NEW 2022 UPDATE
RETRO GAMING WITH RASPBERRY PI
2ND EDITION
164 PAGES OF VIDEO GAME PROJECTS
BUILD AN ARCADE MACHINE
PLAY & CODE GAMES!

REVIEW

Engineer TS-04 Spanner Set

Can the humble spanner be improved?

PIMORONI ◈ £24 | pimoroni.com

By **Jo Hinchliffe** @concreted0g

Nuts and bolts are a mainstay of making, and whilst we may have a sizeable collection of adjustable spanners, socket sets, and more, it's often the smaller end of the scale that gives us a challenge to work with. Hailing from Japan, the Engineer brand of tools has, for a while now, been garnering a reputation for quality precision tools, and they are becoming more available in the UK via a range of vendors. Whilst not the cheapest of tools, they are generally considered tools worthy of investment, with good quality construction and decent quality control. We found that a set fell into our shopping basket during a recent order from Pimoroni, and we wanted to share our thoughts on them.

The TS-04 combination spanner set is described as a "light duty" set of tools, which largely pertains to them being very thin, increasing their usefulness in tighter projects. We'd slightly argue that with a size range of 3.5 mm to 8 mm between the spanner jaws, you're dealing with M1.6 to M5 nuts and bolts, so you shouldn't really be trying to apply too much torque anyway. However, it's fair to say that improper use could cause these spanners to bend.

They arrive well-packed on a cardboard backing with a plastic cover, and they are attached with a short length of ball chain, which makes them almost a key chain item. The spanners feel comfortable in the hand – they are made from stainless steel and finished to a high standard. As combination spanners, they have an open end and a ring end, with the ring

Right ◈
The slim packaging makes these easy to send in the post

Left The 30-degree bend makes it easy to access tight spots

Below Having different sizes on each end means you have two of each size

end being set at a 30-degree angle. Each end of the spanner is a different size, which means you get the benefit of having an open end and a ring end of the same size on different spanners, so you can use two spanners of the same size to hold and tighten a nut and bolt, which is excellent.

In use, they are incredibly handy, as often at this scale, access to a nut or bolt can be a challenge. We have another set of small, steel-forged spanners which are great, but they are a little thicker in standard spanner proportions. This sometimes renders them unusable in a tight spot, leading to improvisation. For example, the budget TP101 tracked robot chassis kit we have used for numerous projects, like our MTV robot (issue 32), or the sheet metal rover (issue 53), uses a crude pair of nuts on a captive bolt to create some axles. Due to assembling this axle system with the wheel in situ, we often had to use fine needlenose pliers to tighten the nuts on the axle; however, the diminutive Engineer TS-04 spanners, at just 1.5 mm thick, handle this task with ease.

> You can use two spanners of the same size to hold **and tighten a nut and bolt, which is excellent**

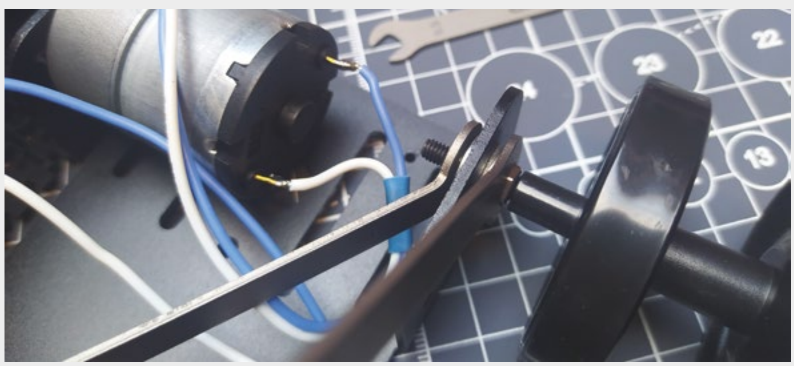
The TS-04 spanners are clearly labelled, so it's easy to grab the correct size from the pile. We aren't massive fans of the storage chain, as you have to undo it to remove what you want, you then end up removing them all, and then you have to thread them all back on when you want to store them. That said, it does work to keep them together. At some point, we can imagine finding or making a little tool pouch to store them in securely. That tiny niggle aside, these are excellent tools. We can imagine exploring more Engineer tools in the future.

VERDICT

A lovely set of small and useful tools; handy for very tight spaces.

8/10

Below These 'light duty' spanners might sustain damage if you have to whack them to remove a stuck nut



REVIEW

Anycubic Kobra Neo

Can you get into 3D printing for £200?

ANYCUBIC $235 | anycubic.com

By **Ben Everard** @ben_everard

Assembling the Anycubic Kobra Neo is straightforward, and shouldn't take too long. There were about a dozen screws to join the parts together. The aluminium seemed a little prone to cross-threading, so don't rush. Once the machine's all together, the first step is to level the bed using the auto bed levelling feature. This is done automatically, but only when you instruct it to (rather than happening by default each print, as some printers do). The levelling makes sure that the print bed is the same distance away from the print head across the whole bed, but doesn't actually make sure that the print head is the right distance away, that's the job for Z-levelling, which comes next.

This is a pretty straightforward process, where you start a print and adjust the height until it's correct. It's made a little more difficult than it needs to be because none of the pre-sliced G-code that came on the SD card is particularly good for it. They're all smallish models that make it hard to see underneath. There are plenty of better models available online, so it's easy enough to grab one and slice it (there's a Cura profile on the SD card). This lack of a model for Z-height is probably our biggest gripe with setting up the printer. Yes, it might sound small, but to someone new to 3D printing, this will be the first time they've squirted hot plastic through a nozzle. They'll be excited and a little nervous, and not quite sure what they should be looking for. Trying to make out whether the underside of an owl is the right distance from the build plate is not a nice way of experiencing this for the first time. What's particularly frustrating is that this isn't a hardware issue, it just needs an extra model on the SD card and it'll make peoples' first impressions much better.

Z-height set, it's time to start printing. We found the print quality was fine. Nothing spectacular, but nothing terrible. Certainly well within the acceptable range for a budget printer.

The Kobra Neo has a 250 × 220 × 220 mm print volume – very similar to the majority of hobbyist printers. You can print PLA and PETG without problems. Anycubic also says that you can print ABS and TPU, but these will cause some problems. ABS really needs an enclosure to be reliable and, while there's nothing stopping you putting this in an enclosure, it doesn't come with one.

While the issues setting the Z-height are a little annoying, the process for getting up and running was simple and, we think, that's the beauty of this printer. For its price – it's available for $235 direct from Anycubic – that's an astoundingly quick and simple setup.

We also liked the control box. It's not a touchscreen, but what's the obsession with touchscreens anyway? Perhaps we're just old folks who dislike change, but having a knob to twiddle generally gives a better, more precise interface. The colour screen looked pleasant, and we always found what we were looking for quickly.

At the budget end of the spectrum, most printers use Bowden extruders, which mean that the motor that pushes the filament is separate from the hot end and connected by a PTFE tube. The Kobra Neo, however, uses a direct drive system where the motor is in the hot end. The advantage of this is that you get more precise control over the filament, particularly with retractions. We found that this worked well with our test prints, but there are some plastic gears in there which could be a worry for longevity.

The biggest issue we had with this printer is that the build plate is not as sticky as some others we've used. We had a few prints come detached while printing, but this did happen when we weren't quite as diligent about wiping the bed down with isopropyl alcohol after every print as we should have been.

We're also not particularly keen on the fact that you have to kick off the self-levelling manually. It's really not clear how often you should kick it off. We found that the printer would be fine for a few prints, but slowly drift out of level and need to be re-levelled when a print failed. It's probably not a bad idea to simply run it every print to avoid this problem, but then why not just do it automatically rather than add another manual step? That said, it does work well when you actually run it.

Budget 3D printers seem to get both better and cheaper every year. If you'd gone back just a few years and suggested that you'd be able to get a printer with automatic bed levelling and a flexible magnetic bed for around £200, it would have been considered impossible, yet here we are.

While this isn't the only machine on the market at this price point, it's the first one we've had in to test, and it delivers well. With the Anycubic Kobra Neo, we got a machine that we found easy to set up and use at a price that's very attractive.

Above The Kobra Neo has a direct drive extruder

Left Both belts have easily accessible tensioning knobs

Below The user interface is tidy and easy to use

VERDICT

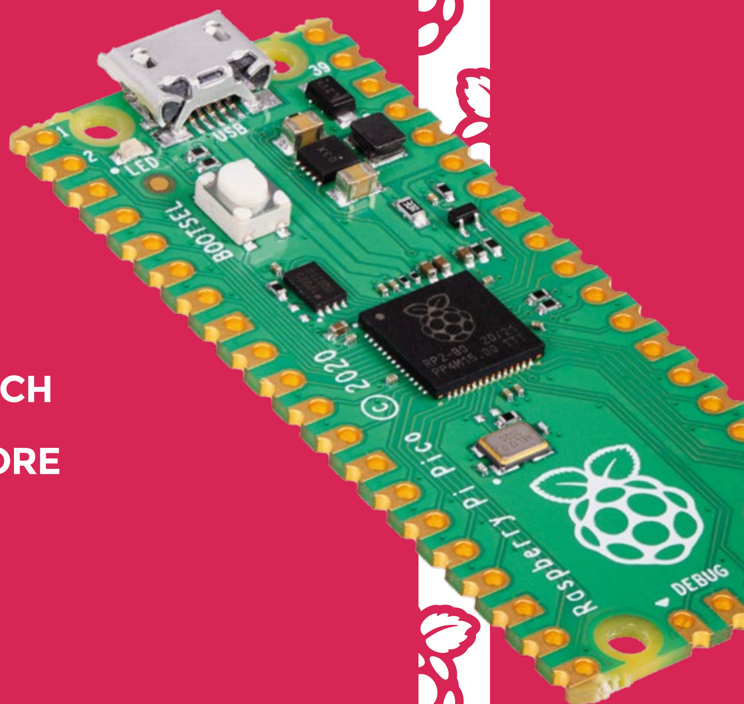
A great printer for beginners at a great price.

9/10

next month

issue #62 ON SALE **15 DECEMBER**

RASPBERRY PI PROJECTS



ALSO

- → 3D PRINTING
- → ELECTRONICS
- → THE LATEST TECH
- → AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe