

MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc

February 2022

Issue #51



FREECAD

Digital modelling using frames

EGG ART

A CNC egg-drawing machine

TECHNOLOGY FOR
EVERYBODY

People are different
Machines should be too
DIYers are making this happen

LIGHT-UP JEWELLERY

Add LEDs to your earrings

CIRCUITPYTHON

Create animations with Raspberry Pi Pico

Feb. 2022
Issue #51 £6



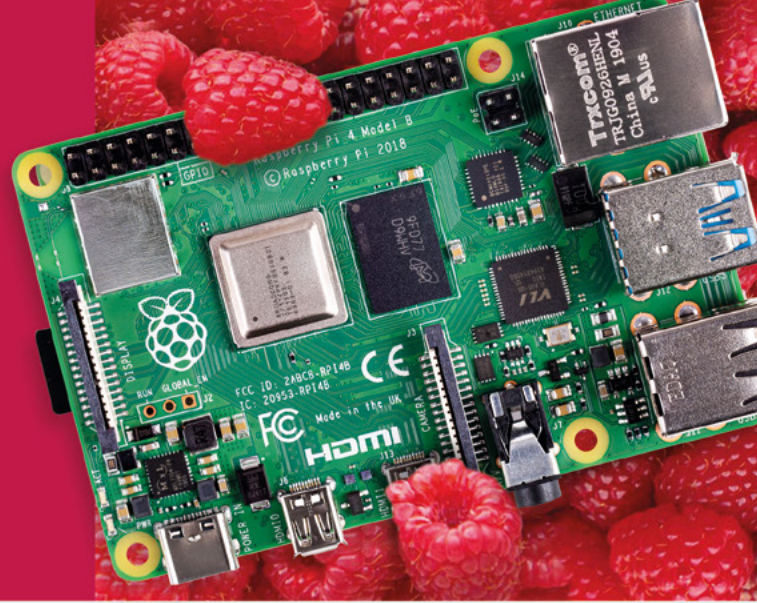
51



9 772515 514006

STRING PICO LASERS BADGES

American Raspberry Pi Shop



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

Partner and official reseller for top Pi brands:



and many
others!

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

USA



PiShop.us

Canada



PiShop.ca



Raspberry Pi

APPROVED RESELLER



Welcome to HackSpace magazine

A few years ago, I broke my arm. In the grand scheme of things, it wasn't a huge injury and I was all healed within a few months. However, while healing, my arm was in a cast from hand to shoulder and I could barely use a computer. That's a bit of a problem for someone who makes a living working at a computer. I tried out some of the various accessibility options, including speech-to-text, but none

You may not need it now, but at some point in your life, **you'll probably need a little help to perform everyday actions**

of them worked particularly well. At the same time, it was a struggle to get dressed and generally perform daily chores. It was

a little reminder that assistive tech is for everyone – just not all the time. You may not need it now, but at some point in your life, you'll probably need a little help to perform everyday actions. We, as makers, are better placed than most to help make sure our world is accessible to everyone. In this issue, we're taking a look at some projects that are helping to make sure that the future is for everyone, all the time. Turn to page 34 to find out more.

BEN EVERARD

Editor ben.everard@raspberrypi.com

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

get in touch at hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.com

[hackspacemag](#)

[hackspacemag](#)

ONLINE

hsmag.cc



PAGE **44**
FREE PICO
WHEN YOU
SUBSCRIBE

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,

Ty Logan

Photography

Brian O'Halloran

CONTRIBUTORS

Jo Hinchliffe, Marc de Vinck, Rob Miles, Andrew Lewis, Rosie Hattersley, Nicola King, Phil King, Mike Bedford, Robert Hamilton-MacLaren

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

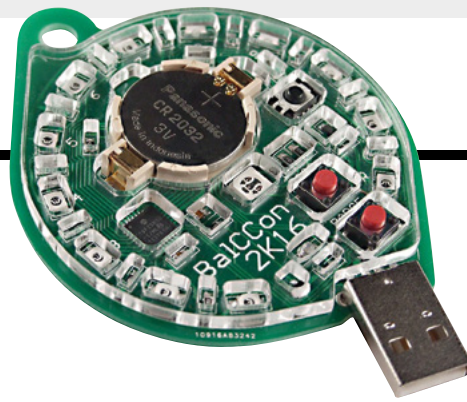
hackspace@subscriptionhelpline.co.uk



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 (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



06

SPARK

- 06 Top Projects**
Brilliant artefacts made by real human beings
- 20 Objet 3d'art**
Capture time in 3D printing
- 22 Hexaflexagons**
Walk in the footsteps of John von Neumann
- 28 Letters**
Imagination: the most important part of any build
- 30 Crowdfunding**
Electric power for your Italian scooter

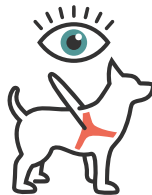
33

LENS

- 34 Assistive technology**
How makers are creating technology for all
- 46 How I Made: Raspberry Pi Pico racetrack**
See who's got the fastest Hot Wheels in town
- 52 Improviser's Toolbox String bags**
Keep this versatile material out of landfill
- 58 In the workshop Blinkenlights**
Make the world brighter with Direct Memory Access

102

Cover Feature



TECHNOLOGY FOR
**EVERY
BODY**

Tutorial

Build a Pico
Pomodoro timer



88

Program a piece of silicon to shout at you every 25 minutes



...
People are different
Machines should be too
DIYers are making this happen



34



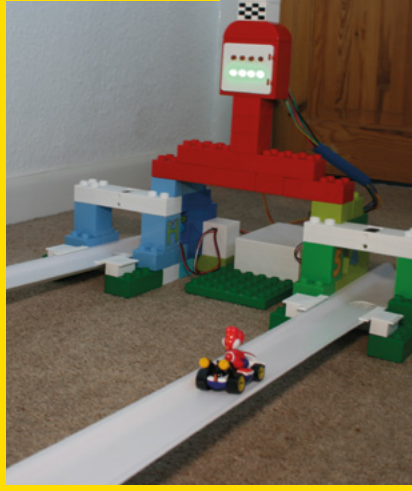


22



How I Made

Pico racetrack



112



30

Tutorial

Jewellery



82

Prepare for next Christmas with homemade jewellery



06

46

Who's faster: Yoshi or Luigi? Find out with a Raspberry Pi Pico

63

FORGE

64

SoM CircuitPython
Animate your LEDs

66

Tutorial Monitor air quality
Keep an eye on workshop dust

70

Tutorial Translucent PCBs
Shine light through your electronics

72

Tutorial Raspberry Pi
Work with sensors

76

Tutorial FreeCAD
Design with tubes and beams

82

Tutorial Jewellery
Fashion a pair of LED earrings

88

Tutorial Pomodoro timer
Work harder with a Pico Pomodoro timer

94

Tutorial Beyond K40
Build a laser cutter from scratch

101

FIELD TEST

102

Best of Breed
Conference badges – they're mini works of art

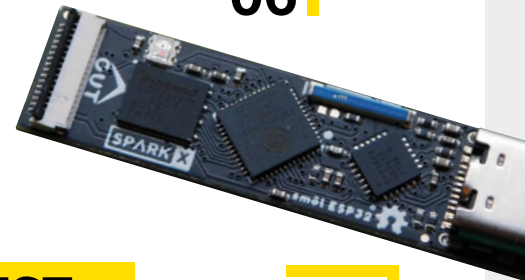
108

Review SparkFun smôl
A tiny, battery-friendly dev board

112

Review Troika pens
All-in-one tool and literacy aid

108



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

Lab Vial Lamp

By Tjiani Lucht

hsmag.cc/TestTubeLamp

This project by Tjiani Lucht is incredibly simple – the excellence lies in the way the idea has been executed. There are almost no fancy electronics to it, other than a step-down converter to reduce the voltage from the 15V coming from an existing power supply to the 3V required by the LEDs, and a switch to turn the lamp on and off.

What stands out about this build is the use of materials: instead of light bulbs, the maker has used 24 LED filaments housed in glass test-tubes, with additional brass bits to extend the LEDs further into the tubes. The electronics are housed on a piece of MDF, and the test-tubes are held together by transparent acrylic. It's incredibly simple, but no less striking for that. □

Right □
After years in the doldrums, science is cool again. Thank heavens for that



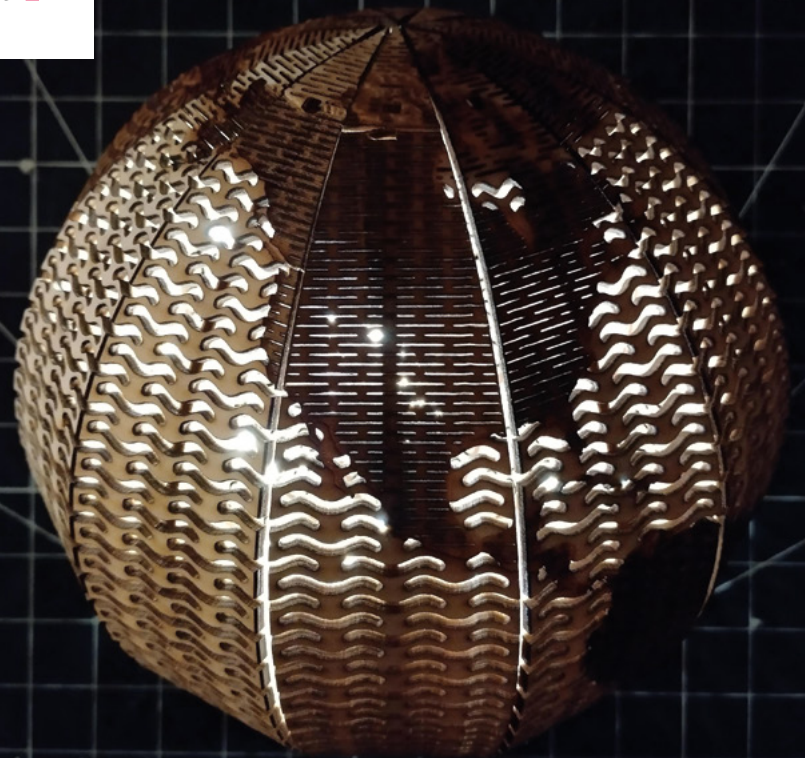
Laser-cut globe

By [deelstradesigns](#)

hsmag.cc/LaserGlobe

There are loads of different ways you can laser cut plywood to make it flexible; [deelstradesigns](#) has used two in the same model to create a contrast between land and sea in this globe design.

It's made from 3mm poplar and 6mm Baltic birch plywood, and the design is all the maker's own work, using Inkscape, LightBurn, and mapping software called QGIS. ▣



Right ▣
We love the smell of a laser cutter in the morning





WeatherBot

By Lewis Aburrow

hsmag.cc/WeatherBot

When we spoke to Lewis Aburrow, creator of the DIY Machines YouTube Channel, he was still working on the prototype of his weather forecasting machine. It's such a great idea that we had to get it into the magazine, so here it is.

The WeatherBot uses an ESP32 to take data from OpenWeatherMaps and display it as a constantly updating diorama, showing various weather conditions as an overlapping series of scenes that rotate to show wind, rain, snow, and other weather conditions. We're sick of looking at computer screens, so it's a nicely analogue way of getting information, but if you want more detail, Lewis has added an e-ink screen showing temperature, precipitation, and wind speed, so you're not forced back to the computer if you want more information. **□**

Left **□**

If you want to make your own (or build on the work already done), the design files are all available at diymachines.co.uk

Machined Pi Enclosure

By AppliedSBC

hsmag.cc/NickelEnclosure

As much as we like the bare PCB aesthetic of the Raspberry Pi, it does tend to attract dust, and is a magnet for spilled coffee. Third-party cases for Raspberry Pi have been around almost since the dawn of time (at least, since the Raspberry Pi was launched way back in 2012).

However, we've seen none so shiny, so simple, nor as robust as this one from AppliedSBC in Plainview, New York.

As you can probably guess by looking at it, the two halves of the case are carved out of slabs of aluminium, which are then nickel-plated to keep it shiny. There's a rubber base to stop it from sliding around, or you can screw it to a piece of wood. There is a black version, but we find that the bare metal look fits our cyberpunk ethos better. ■



Right ■
You can also get a version of this case for the BeagleBone Black



Analogue spectrum analyser

By emdee401

hsmag.cc/FreqAnalyser

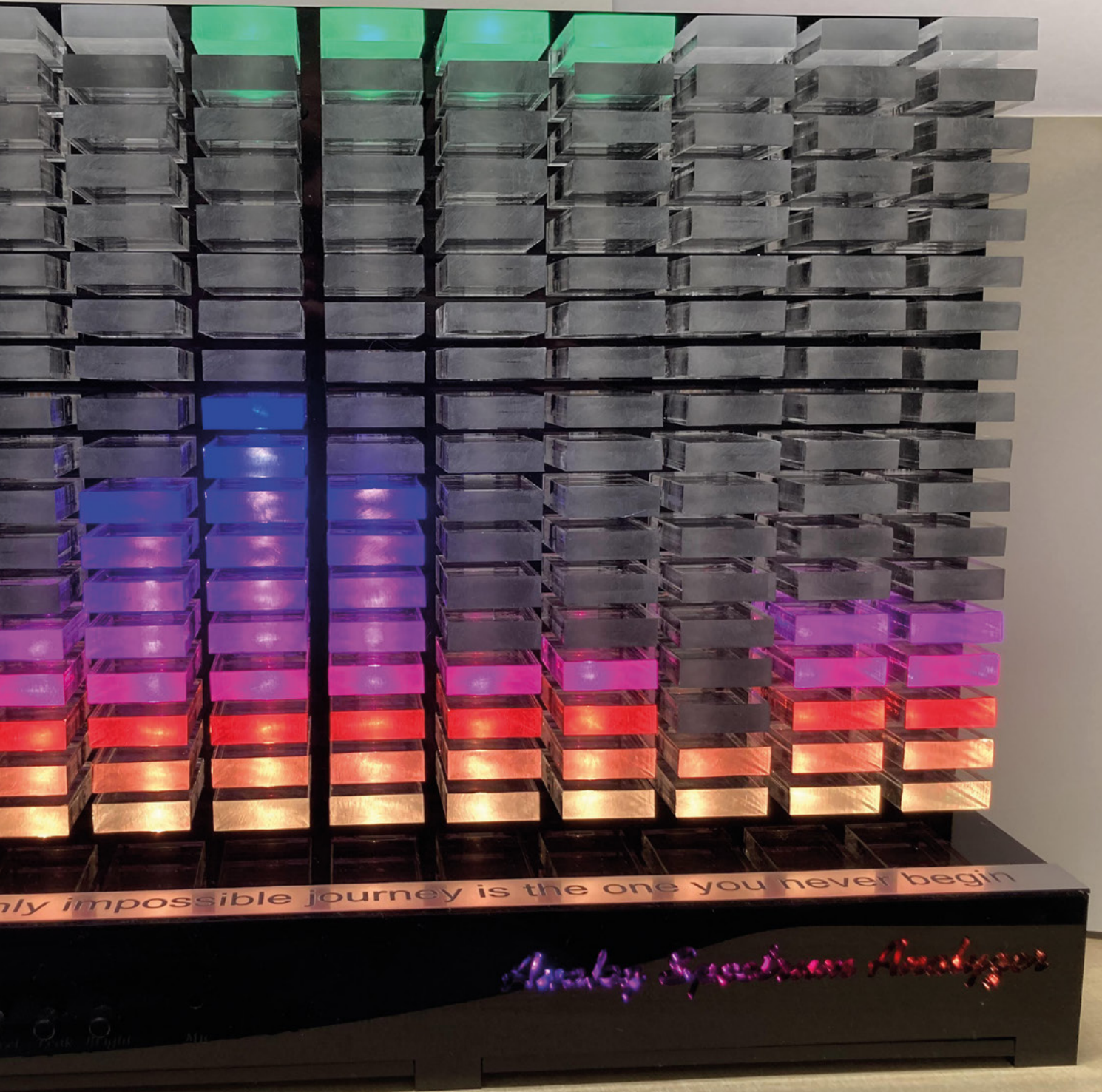
A

udio is a huge, deep, twisty rabbit hole of expensive gear, performance optimisations, and indecipherable jargon. And that's what makes it fun. If you're at all into audio, this ten-band frequency analyser provides an easy, graphical way to show you exactly what's going on when you're listening to the latest Taylor Swift LP.

You can roughly divide this project into the pretty bit, the clever bit, and the very clever bit. For the pretty bits, the maker used addressable RGB LEDs, 10mm thickness transparent acrylic to diffuse the light from the LEDs, and black acrylic to provide a background. The clever bit is the ESP32. The very clever bit is the custom PCB designed by emdee410, which makes everything possible. The exact details are beyond us, but we do know that the circuit is analogue, with each frequency band having been given its own op-amp rather than having frequencies broken out digitally. □

Right □
Visualising frequencies in music is a great way to pretend you're a studio engineer





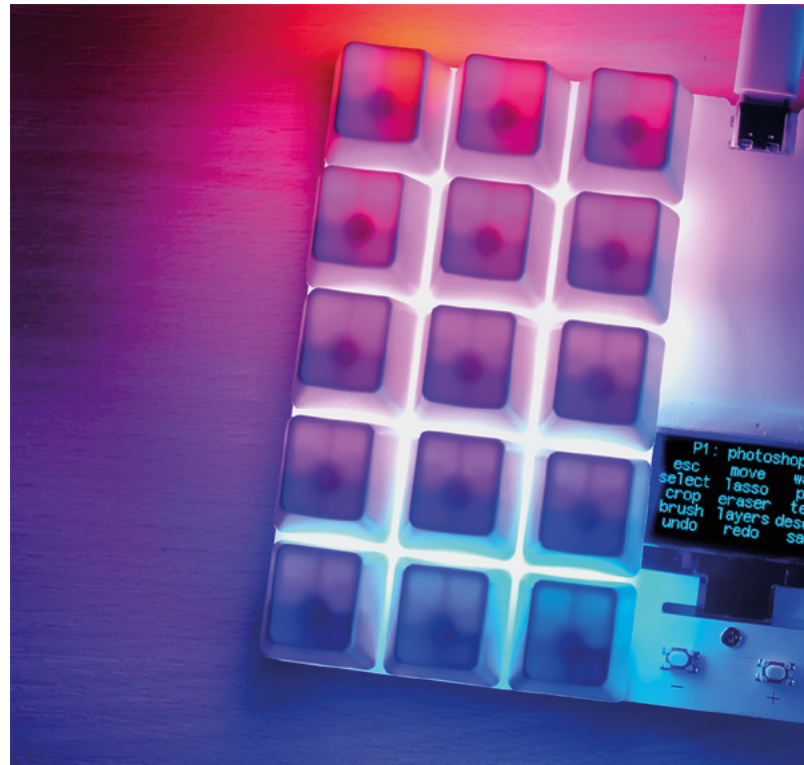
duckyPad

By dekuNukem


hsmag.cc/duckyPad

The standard keyboard, as it exists on most devices, is a slow, archaic thing. The QWERTY layout wasn't invented with speed or ease of typing in mind, but to ensure that the most frequently used keys were far apart in order to reduce the risk of jammed typewriter mechanisms. Once that became entrenched, computers were forever hampered. Useful features such as special characters and actions such as copy, paste, zoom, etc., are treated like add-ons, often accessed by contorting the hands to press two or three keys at once, twisting our precious carpal bones until they ache.

Into this workflow and anatomical mess comes the duckyPad. You can buy this from its creator, dekuNukem, in a complete form or as a bare-bones assembly to which you add your own keys, allowing the user to customise the keys' exact amount of clickyness. Even better than this, the device incorporates an OLED screen showing what each key does, so if you forget, you won't have to go diving into configuration files to find out. Oh, and did we mention it looks lovely? ▣





Below  The duckyPad works with MacOS, Linux, and Windows



Steampunk desk clock

By [TheTimeOnTheWall](#)

hsmag.cc/NixieDeskClock

Nixies pop up occasionally in these pages, thanks to the incredible way they look. They're not the most practical things – they're expensive and fragile, and you need to be comfortable with high voltages if you want to incorporate them in your projects. But when they're used right, as in this clock by Valery, they look like nothing else.

All the way from Lipetsk, Russia, this steampunk-themed clock/phone stand is made from brass, copper, bronze, and wood. It tells the time, looks incredible, and that's about it. Feast your eyes. □

Right □
Captain Nemo would love this steampunk masterpiece





Objet 3d'art

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

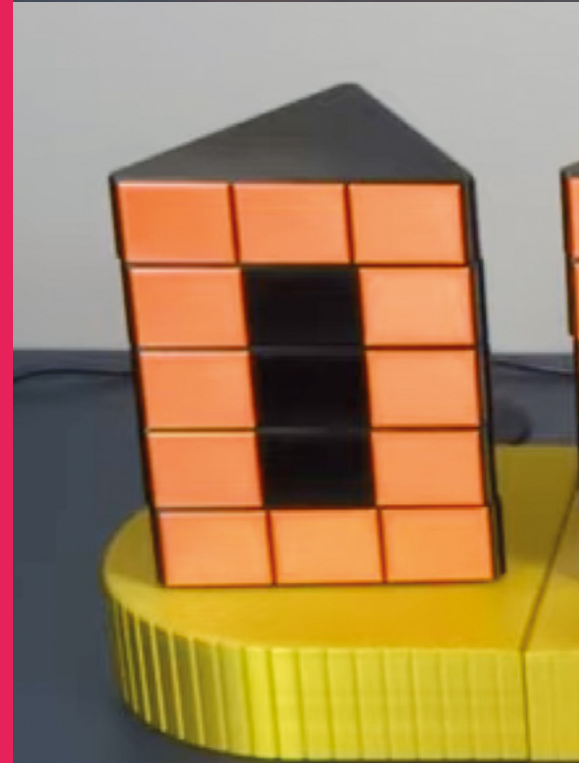
W

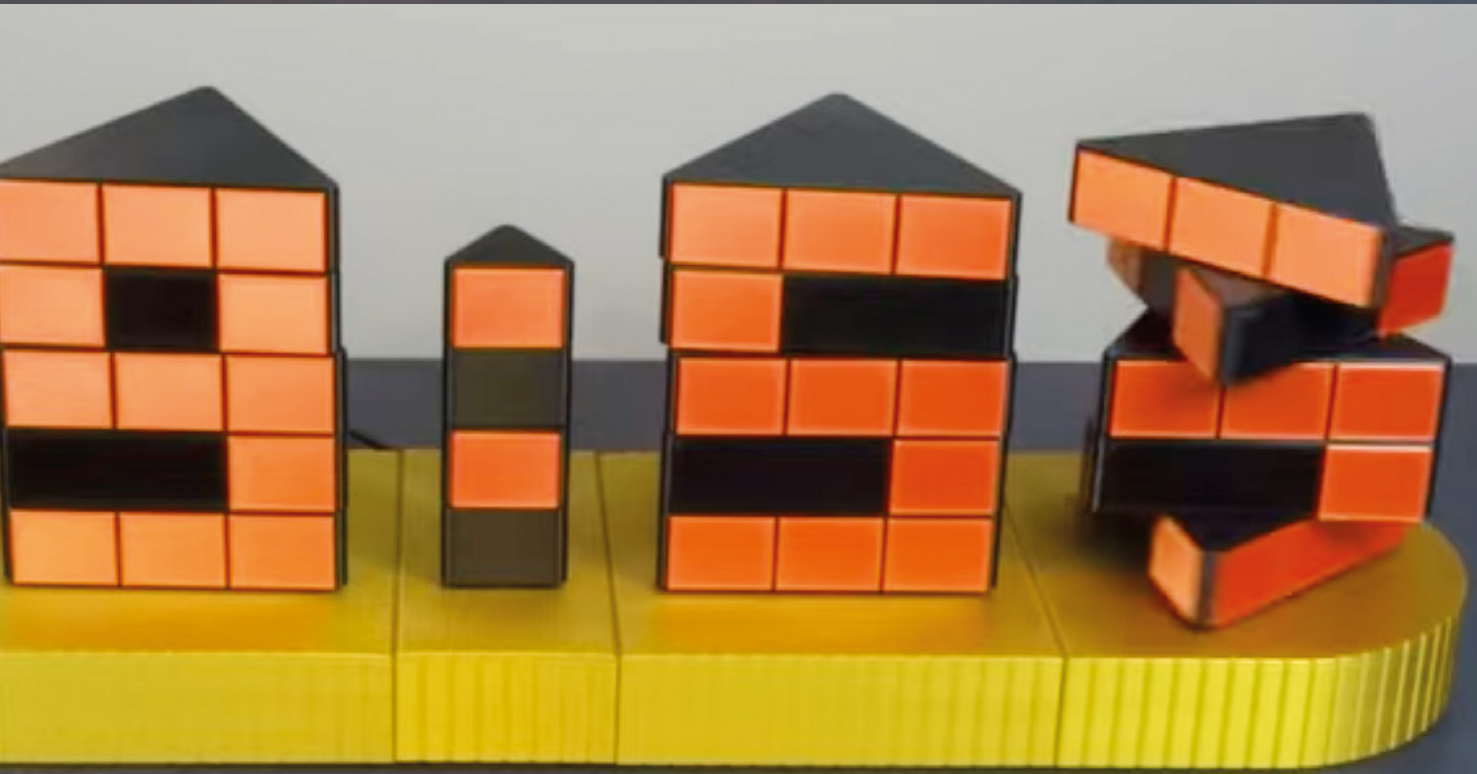
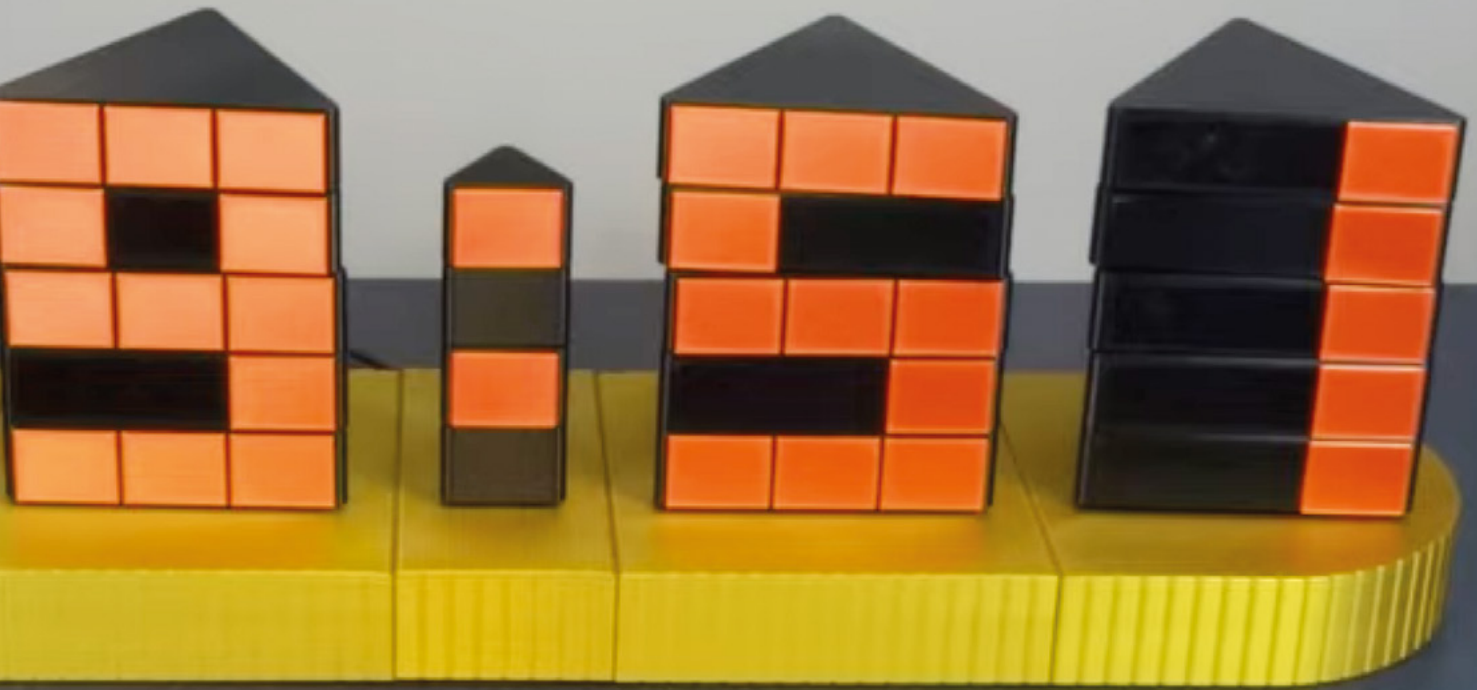
e've seen many implementations of the classic seven-segment display over the years, but this one gives it – if you'll

permit us a small pun – a new twist.

It's hard to get a sense of what's going on when looking at a static photograph, so it might help if we tell you that the creator of this project, Hans Andersson, has also built a Rubik's cube solving machine. Each digit is comprised of 3D-printed triangular segments that rotate around a vertical axis; by rotating the slices that make up each digit, it's possible to display numbers 0–9, controlled by a system of gears, servos, and an Arduino Mega. □

➔ hsmag.cc/TimeTwister





The strange world of hexaflexagons

Mike Bedford shows you how to make a mathematical curiosity called a hexaflexagon



Mike Bedford

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

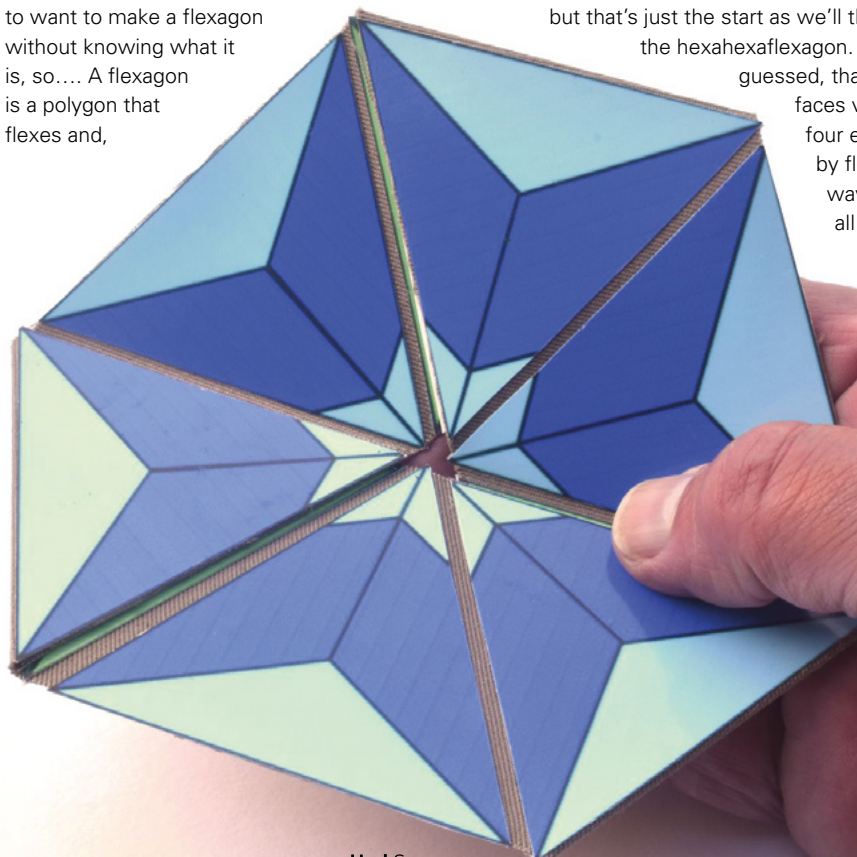
Here, we're going to show you how to build and investigate something you've probably never heard of – a flexagon.

Even so, we trust that you'll find this unusual mechanism fascinating, in a mathematical sort of way. Don't let the maths connection put you off, though; we promise you that the experience will be a lot more enthralling than solving differential equations.

Chances are you're not going to want to make a flexagon without knowing what it is, so... A flexagon is a polygon that flexes and,

in particular, a hexaflexagon is the six-sided flexing polygon. And to answer the question that raises, flexing a flexagon involves manipulating it so that a different face appears. An ordinary hexagon – not a hexaflexagon – has just the two faces, of course, and you can see both just by flipping it over. The simplest hexaflexagon is the trihexaflexagon and, just like a regular hexagon, two of those faces are always visible. However, if you flex it, the third face, which was previously hidden inside, will become visible. We'll see how to make a trihexaflexagon, but that's just the start as we'll then move on to the hexahexaflexagon. As you've probably guessed, that always has two

faces visible with another four embedded inside, and by flexing it in a particular way, you'll cycle through all those faces.



Right

An ordinary paper hexaflexagon will soon wear out or tear, but this one with fabric hinges should keep you amused for much longer. It's also much easier to flex than a paper hexaflexagon too

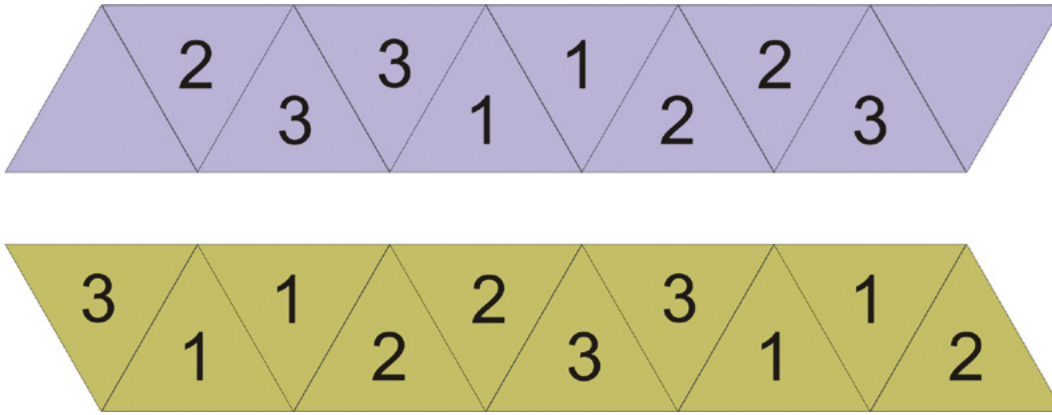


Figure 1 ♦ The two sides of your strip of paper should be marked like this before folding it into a trihexaflexagon

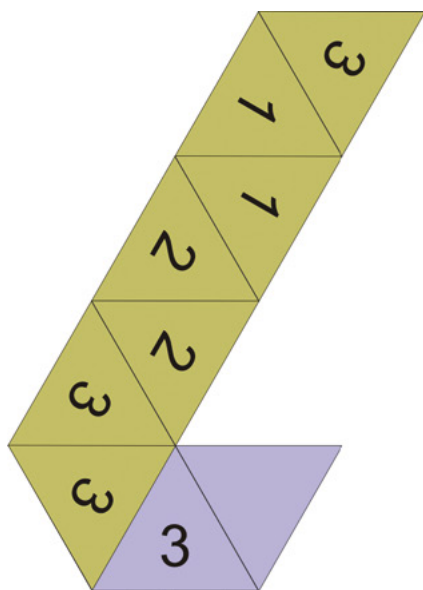
Hexaflexagons are undoubtedly mathematical – some people have described them using pages of equations that are impenetrable to most of us – but they’re truly captivating. They can also be quite attractive, depending on the patterns you use to adorn their multiple surfaces. What’s more, they’ll be quite a talking point if you leave a few scattered on your coffee table for your guests to play with.

THE TRIHEXAFLEXAGON

To make a trihexaflexagon, start by drawing a row of ten equilateral triangles on a strip of paper. It’s easiest to use a 60 degree set square, otherwise you’ll have to work out how long each of the edges should be for a particular width of strip. Alternatively, you could design it using drawing software, or download one of the many templates you’ll find online, and print it out.

For trihexaflexagons, a strip cut from a sheet of A4 or Letter size paper, lengthwise, will be large enough, but make sure it’s at least seven times longer than its width. With your triangles drawn, cut off any surplus at the ends so it comprises just the ten triangles. For your first trihexaflexagon, we suggest that you number the triangles as in **Figure 1**, both on the top and the bottom of the strip, leaving just the two faces blank as shown. Note that the top and bottom

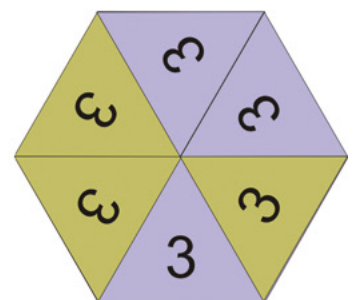
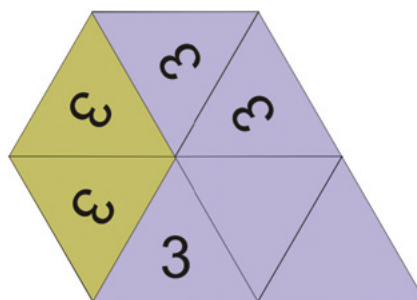
“ They can also be quite attractive, depending on the patterns you use to adorn their multiple surfaces ”

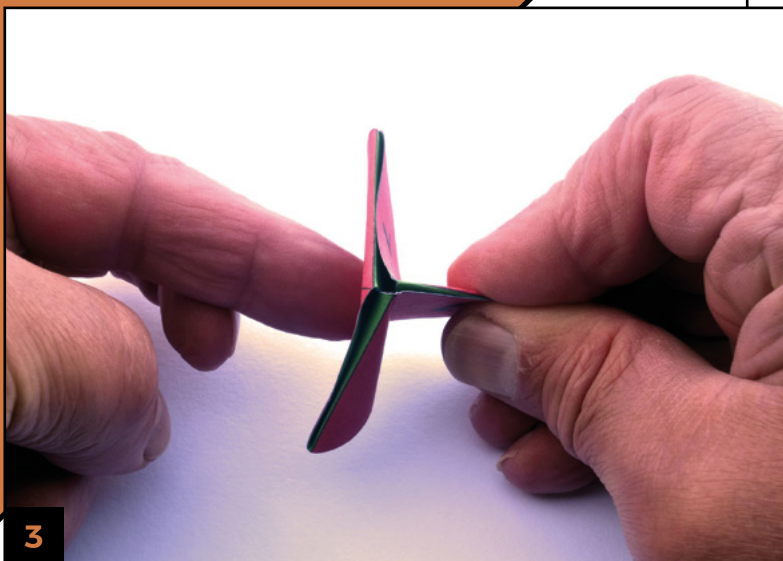
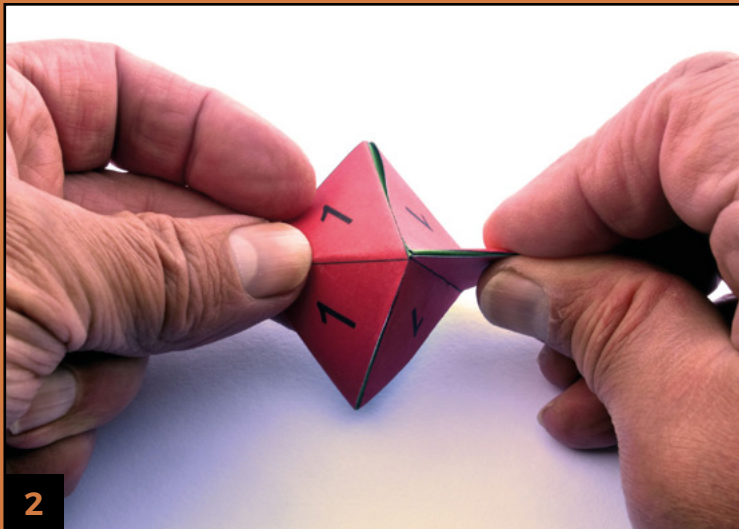
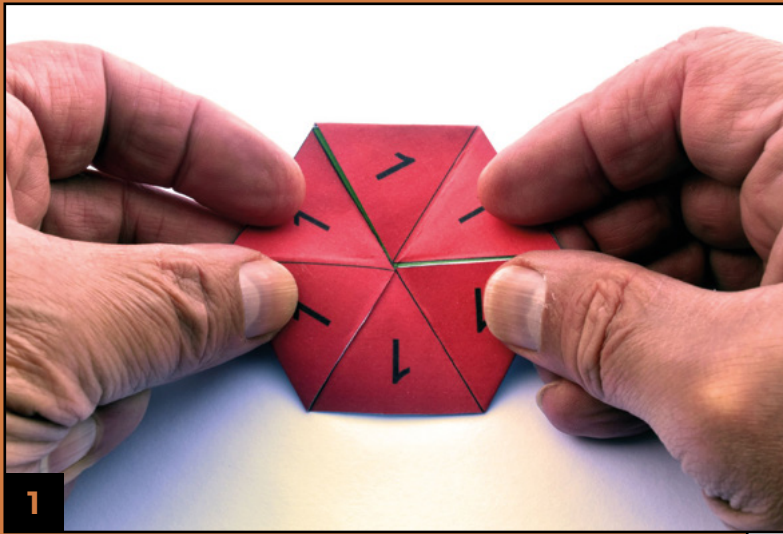


of the strip are shown as if the strip was turned over widthwise, not lengthwise. It’ll be easier to get the numbers on the back in the centres of the triangles if you also mark the triangles on the back. For clarity, this diagram and the folding diagrams use different colours to differentiate the two sides of the strip.

Next, it’s time to fold the paper strip into a trihexahexagon. First of all, though, for easier flexing, bend along each of the dividing lines between →

Figure 2 □ Turning the numbered strip of paper into a trihexaflexagon involves folding it three times and gluing together the two unnumbered triangles





the triangles in both directions. Now, starting with the strip as shown in the top part of the numbering diagram, make a forward fold (i.e. fold the long end toward you) between the third and fourth triangles from the right so that it looks like the first part of the folding diagram. Now, make a fold between the sixth and seventh triangles, again forward, but tuck the penultimate triangle under the first one so that it looks like the second part. Finally, fold the last (unnumbered) triangle forward and glue the faces of the two unnumbered triangles together (**Figure 2**).

You'll notice that the completed trihexaflexagon has the face with 3s on one side, and the one with 1s on the other side. The face numbered with 2s will be invisible though, so let's get flexing. To flex, hold the trihexaflexagon in both hands and rotate it until one of the divisions between triangles that has an exposed paper edge is running from the centre to the right-hand edge of the trihexaflexagon, because



You might want to make some with more attractive and colourful patterns



you can't flex from the positions with no exposed edge. With the thumb and index finger of your right hand, squeeze together the two triangles at the right side and then, using the index finger of your left hand, push the opposite two triangles to the centre. You'll notice the trihexaflexagon starts to open at the centre, so pull out the top corner of the two left-most triangles outwards while releasing your grip on the two triangles you originally pinched together. You'll find that it opens out to show the face numbered with 2s. If you now rotate the hexaflexagon by 60 degrees, either clockwise or anti-clockwise, it'll come to a position where it'll flex again and, having flexed at that position, if you rotate it by another 60 degrees in the same direction, you'll be able to flex back to the initial configuration.

Now you've seen how trihexaflexagons work, you might want to make some with more attractive and colourful patterns, although you might want to include the numbers, even if only discretely. You can pattern the triangles by hand, using a marker or paint, or create the design using software. In deciding how to adorn your trihexaflexagon, bear in mind that there are two different ways in which a particular face can

appear. You can easily see that if you draw a circle in the centre of each face on your first trihexaflexagon. You'll find that the next time you see the same faces after flexing, the circle will appear as three partial circles on the outside of the face. Make sure you keep an eye on both sides of the trihexaflexagon to see both arrangements of all three faces.

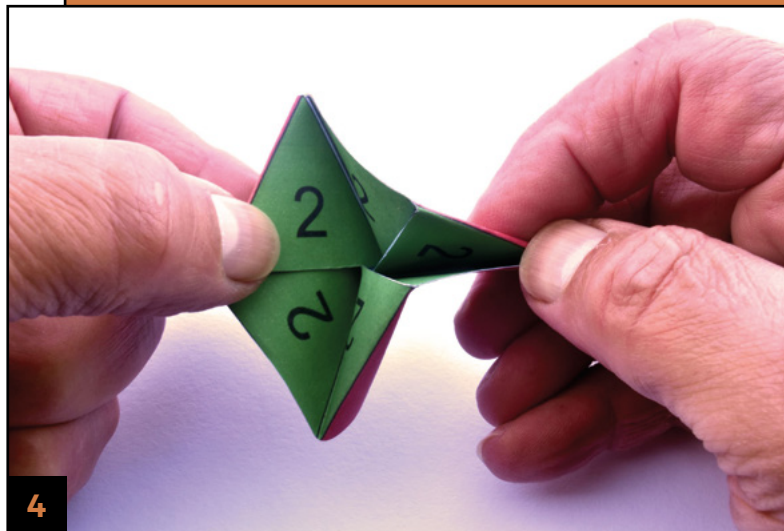
HEXAHEXAFLEXAGONS

Next up, we're going to see how to make a hexaflexagon with more than three faces. We could have chosen the tetrahexaflexagon or the pentahexaflexagon – and you might want to read up on those later – but we decided to go with the hexahexaflexagon because it's more similar to the trihexaflexagon in its construction.

This time we're starting with a strip divided into 19 equilateral triangles. It'll be too small if you try to make it from A4 or Letter size paper so, unless you have a larger sheet of paper, glue together two shorter strips. Now number the triangles as shown in **Figure 3**. Folding starts with the strip as shown in the bottom part of the numbering diagram. Starting at the left, fold the strip so the two triangles numbered with 4s are facing each other, then fold it so the 5s are facing, then the two 6s, and continue until you get to the end. The end result as shown in the folding diagram is a flattened spiral, and you'll see that it looks almost identical to the strip that we used to make a trihexaflexagon. From here on, we don't need to provide any more folding instructions because they're identical to those that you followed to make a trihexaflexagon (**Figure 4**).

The hexahexaflexagon is flexed in exactly the same way as the trihexaflexagon. If you flex more or less at random, you'll find that 1s, 2s, and 3s appear more commonly than 4s, 5s, and 6s (**Figure 5**). The quickest way to progress through all its states is to flex at one position until you can't flex again, rotate 60 degrees, and flex there as many times as you can, and continue in this way. This way, you'll find that it takes twelve flexes to get back to the starting configuration. This can be summarised as a state diagram, and we're providing the one for our hexahexaflexagon. Depending on whether you rotate clockwise or anti-clockwise, you might proceed in the opposite direction to the one shown in the diagram.

Now you've got to grips with a hexahexaflexagon, you might be interested to know that there's not just one but three hexaflexagons, and they flex in different sequences, so they have different state diagrams. We've provided diagrams showing the starting pattern for the other two, but we'll leave →



4



5

OTHER FLEXAGONS

Although we've only delved into hexaflexagons, and while they're the most popular, they're only the tip of the iceberg. So, if you get bored with six-sided figures, you might like to investigate the wider world of flexagons. Probably the next most common types are the tetraflexagons (tritetraflexagons, tetraflexagons, etc.), but you'll also find templates for lots more. Pentaflexagons, heptaflexagons and the like are obvious, and not too dissimilar to hexahexaflexagons, but some of the more recent discoveries have faces that are subdivided and which allow flexing along these subdivisions.

Above ♦
It might seem strange to start with, but flexing a hexaflexagon will soon be second nature

TUTORIAL

Figure 3 ♦

To make a hexahexaflexagon you'll need a longer strip of paper than for a trihexaflexagon, which should be numbered as shown here

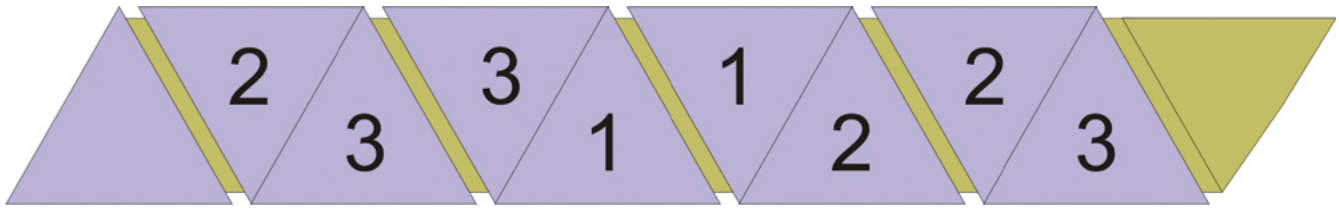
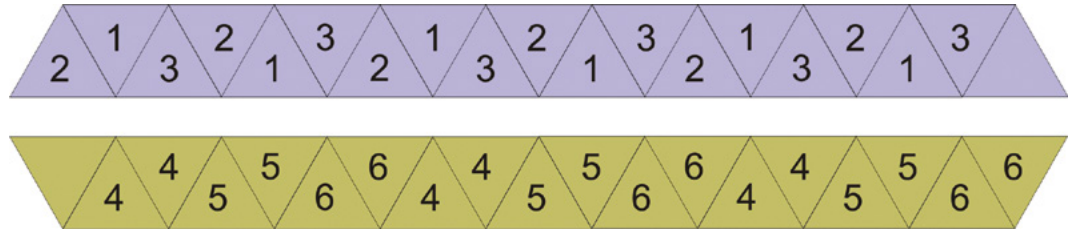


Figure 4 ■

This is how the hexahexaflexagon looks after you've rolled it into a flattened spiral. From this point, fold it just like a trihexaflexagon

you to figure out how to number and fold them up. If you struggle with the alternative hexaflexagons, you might like to try a dodecahexaflexagon. In fact, there are no fewer than 733 different variants, but the version we recommend is the one made from a straight strip – just like we used for the trihexaflexagon and the hexahexaflexagon – but with 37 triangles. The first job is to roll it into a flattened spiral that looks like the starting point for the hexahexaflexagon. You can probably guess how to complete it from that point, although it might take

a bit of practice to get it right. As a word of advice, though, it would be a good idea to make it with quite large triangles, otherwise it might be difficult to flex.

BEYOND PAPER

Paper is good for learning about hexaflexagons but, in time, your creations will start to look shabby and, depending on how much you've flexed them, the folds might start to tear. If you want something that'll look good and work well for more than a few days, therefore, a different method of construction would be preferable. 3D printing something that's normally

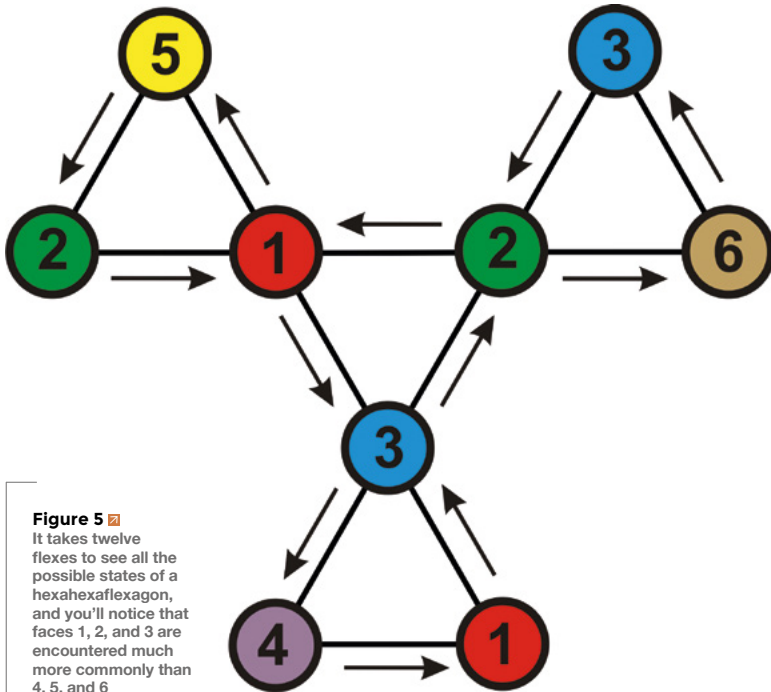


Figure 5 ■

It takes twelve flexes to see all the possible states of a hexahexaflexagon, and you'll notice that faces 1, 2, and 3 are encountered much more commonly than 4, 5, and 6

“ If you don't have access to a 3D printer, there are other ways of making a more durable hexaflexagon ”

made by folding paper might not seem obvious, but it has been done and you'll find designs online (Figure 6). However, those designs only work for the trihexaflexagons, not hexahexaflexagons.

If you don't have access to a 3D printer, there are other ways of making a more durable hexaflexagon. Probably the best approach is to somehow arrange for the hinges to be made from a more flexible material than the triangles. We did this by printing the pattern onto thin card, cutting out the individual triangles, and gluing them onto fabric, with a slight gap between them (Figure 7). PVA-based glue is suitable. You don't need to glue the two blank triangles onto the fabric because you can glue together the two exposed



Figure 7 ♦ You can make a more durable hexaflexagon by printing the triangles onto thin card and gluing them onto a strip of fabric

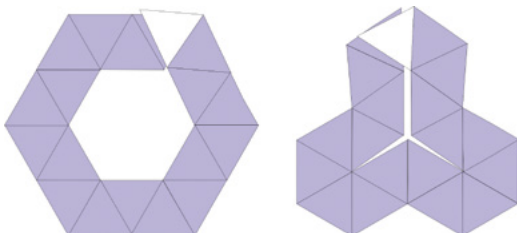
Figure 6 ♦ If you have a 3D printer, you might like to try printing a hexaflexagon. You'll find several designs on Thingiverse or Shapeways like this one by Henry Thomas

fabric triangles. Be sure to avoid applying the glue to those parts of the fabric that form hinges, because that would make them much less flexible. Other approaches come to mind which you might like to try, or you could dream up your own ideas. We did try gluing paper triangles onto polythene sheets of the type used in building and gardening and then, to make sure the triangles don't come loose and to protect them, we applied a coat of varnish. We found that polyurethane varnish is partially absorbed by the paper, resulting in a blotchy look, but there are probably more suitable types of varnish. Another idea that has promise is to sandwich the paper triangles between two sheets of adhesive transparent film of the type that's used for protecting books. We note that fully fabric hexaflexagons have been made by sewing fabric triangles together, and some people have even knitted hexaflexagons. □



Below ♦

Our detailed instructions show you how to make one of the three possible hexahexaflexagons. Here are the starting patterns for the other two – the unnumbered triangles which are glued together are white, but we'll leave you to figure out how to fold and number them



SIMULATING HEXAFLEXAGONS

There's something special about playing with a real hexaflexagon that an online image will never capture, but if you enjoy coding, perhaps you'd like to create a flexagon simulator? You'll find one of a dodecahexaflexagon hsmag.cc/Flexagons, but how about making a 3D one that fully captures the way a hexaflexagon turns inside out when it's flexed? This approach could also allow you to play with flexagons that would be challenging to make from a paper strip.

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

LIGHT PAINTING

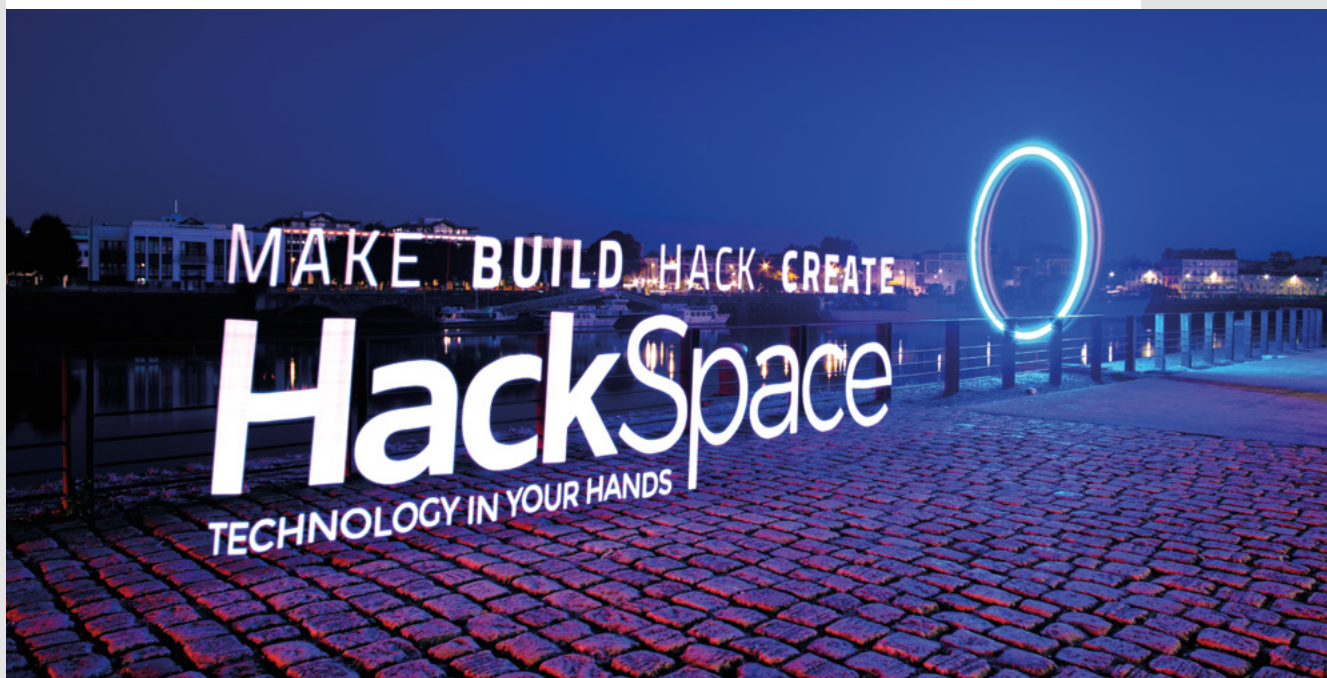
Holy smokes! The images from the Raspberry Pi light painting project, Pilightpaint, last issue are mind-blowing! Give us more!

James

Glasgow

Ben says: That was short and sweet! The Pilightpaint project is indeed mind-blowing. Light painting is awesome anyhow, but when you factor in how well Didier Briand executed his project, it

gets even more incredible. The amazing and humbling thing about it is that the components he used are things that most of us will have lying about (or at least within a click of buying cheaply): some DotStar LEDs, a small screen, and a Raspberry Pi. It's like a Neil Young guitar solo: there's so little to it, but it's magnificent in its grandeur. Of course, this means that the only thing that's stopped us from doing something similar all these years is a lack of imagination, which is either inspiring or terrifying.





DIY MACHINES

Can I add a vote of confidence to the DIY Machines YouTube channel? I've been racking my brains for a way to describe it, but the best I can come up with is 'clear'. I've tried following a few makers and [Lewis Aburrow, host of DIY Machines] is right when he mentions feeling cheated sometimes: if I've trawled the internet for a set of clear instructions, I want to see every step. If someone's showing me how to make a thing, and they miss a step out, it's jarring at best, and confusing at worst. If you want to actually learn how to make things, rather than be impressed by the shiny things, DIY Machines is brilliant.

Dan

London

Ben says: If we ever feel that things are falling apart, the centre cannot hold, etc., the two things we turn to are woodworking videos and Lewis's YouTube channel, DIY Machines. Every step of every build is explained so clearly, you can feel yourself learning as you watch. Even better, all the builds are open-source, with every source file shared – as Lewis told us, he learned by following others' web instructions, so it's only fair that he shares as much as he can.

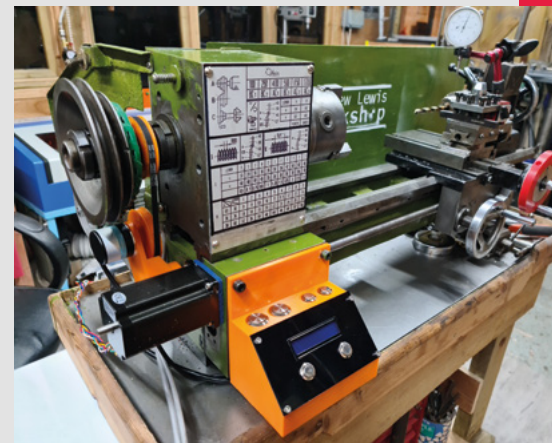
PICO LATHE

I loved the description by your writer, Andrew Lewis, of his old lathe as "really should be retired from service" and that, "keeping it alive is not a commercially sensible decision". I've got loads of stuff like that, though whereas Andrew admits to a sentimental attachment, I am openly a hoarder. I love the idea that a big, old machine, that could have come from the dirty old days of the industrial revolution, can be resurrected by a tiny microcontroller from the modern age. There's no need to throw things out when we can modify them to work just as well as they ever did – sometimes better.

Amanda

Greater Manchester

Ben says: As the storming success of the BBC's *The Repair Shop* shows, repainting a broken old thing is immensely satisfying. So you're absolutely right, but just as with Didier Briand's light painting device also mentioned in these pages, the key ingredient in that lathe restoration project isn't the £3.00 worth of Raspberry Pi Pico; it's the hours of graft and the imagination of Andrew Lewis that made the project happen and produced such a pleasing result.



CROWDFUNDING NOW

Electric motorbikes

New electric drives for old Vespa scooters

From £3495 | hsmag.cc/evespa | Delivery: August 2022

Internal combustion engines are a technology of the past. Much like steam engines, they had their time in the limelight. They represent some of the very best engineering of the 20th century, but the 21st century belongs to drive systems that can run on clean power. What does that mean for classic cars and motorbikes? Should we keep them running, consign them to museums, or send them to the recycling plant?

While there's certainly a place for unmodified classic cars to help the young folk understand why older people go 'brum brum' when playing with their toy cars, it's possible to keep the aesthetics and feel of a classic going without the constant drip-feed of squished ancient plankton. There are a few conversion kits for different vehicles around. This one will convert a Vespa smallframe into an electric

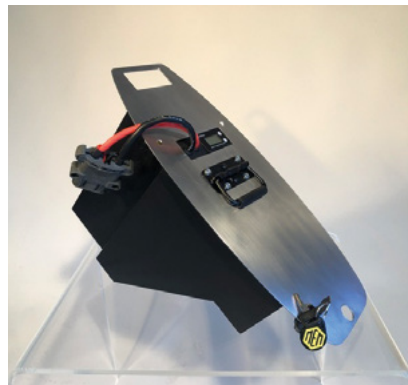
motorbike. You'll save on fuel costs, save the environment and, as an added bonus, you'll get more power as well. We've not tested this out, so can't comment on whether or not it lives up to expectations, but the small and light frame of a Vespa should make a good base for an electric motorbike.

The laws on such conversions vary from country to country, so make sure you check your local laws before adding a new power source to your hog. □

Make sure you check your local laws before adding a new power source to your hog

BUYER BEWARE

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



Below  Vespa is named after its famous sound: it means wasp in Italian



Start Your Concept.

Start With Digi-Key.



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



More Products

More Inventory In-Stock

More New Technologies

More Name-Brand Suppliers

More Technical Resources



DIGIKEY.CO.UK



0800 587 0991
DIGIKEY.CO.UK



*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of \$18.00 USD will be billed on all orders of less than \$50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future



PG
46

HOW I MADE: A PICO RACETRACK

An ingenious gadget to settle arguments over who has got the fastest Hot Wheels car



PG
52

IMPROVISER'S TOOLBOX: STRING BAGS

Frugal, environmentally friendly fun with this space-saving shopping bag



PG
34

ASSISTIVE TECHNOLOGY



How makers are bringing technology to everybody



PG
58

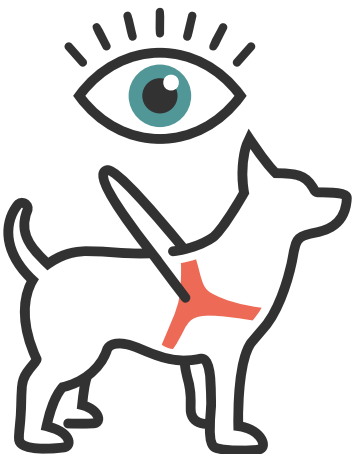
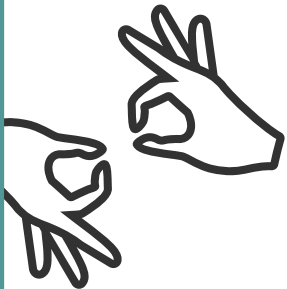
IN THE WORKSHOP: DIRECT MEMORY ACCESS

Get your Raspberry Pi Pico to power LEDs without taxing its little brain too much



ASSISTIVE TECHNOLOGY FOR ALL

Advances in assistive technology have improved many disabled people's lives. Open-source projects can bring the technology to many more people around the world who need it, discovers **Phil King**





By 2030, the World Health Organization estimates that two billion people will require access to at least one piece of assistive technology. While rapid advances have been made in the field over the past few decades, only one in ten disabled people currently have access to the assistive technology they need. Commercially available solutions tend to be

expensive, so the role of open-source projects is crucial in bringing assistive technology to everyone around the world who needs it.

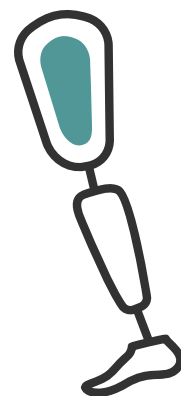
Bionic and prosthetic limbs and hands are life-changing for amputees and people with limb differences. So the existence of an online community like e-NABLE, whose members develop low-cost, 3D-printable prosthetic hands, is vital.

Wheelchair users benefit from a wide range of projects to help them enjoy greater mobility and accessibility, from the Eyedrivomatic eyegaze wheelchair controller to the Wheelmap crowdsourced mapping project. Meanwhile, projects such as the Freedom Wing Adapter and Walkin' & Rollin' Costumes bring greater accessibility to video gaming and cosplaying.

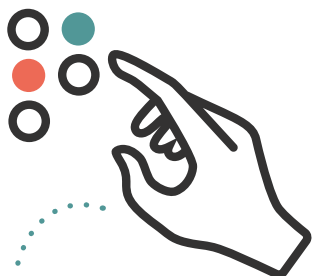
A raft of accessibility aids are available for visually impaired people and others with sensory loss, helping them to identify objects, navigate, and read online news. And with the Blind Arduino Project, there's nothing stopping them getting involved in digital making too.

Electronic speech aids give a voice to people who lack one, so they can communicate more easily and naturally. Digital accessibility is also essential so that people with disabilities have equal access to computing and the internet and the opportunities that offers.

While we only have space to highlight a select few in this feature, many more open-source assistive technology and accessibility projects are helping to create a truly inclusive world for everyone. →



The existence of an online community like e-NABLE is vital





BIONIC BODY PARTS

The open-source development of replacement limbs and hands can reduce costs and encourage co-operation and further innovation

OPEN-SOURCE ROBOTIC LEG

opensourceleg.com

A

ll over the world, tens of millions of leg amputees have a reduced quality of life due to persistent mobility limitations. So the

potential of an open-source project like this one, from the Neurobionics Lab at the University of Michigan, is vast. Its purpose is to unify the research of prosthetic leg controls. This is achieved with the use of a single, ubiquitous robotic leg system so that the results of research teams can be easily compared.

Through the project's website, researchers have access to downloadable hardware and software files so that they can enter the research field without having to design a leg themselves. Featuring a robotic knee and ankle system, the open-source leg can be easily manufactured, assembled, and controlled – using a Raspberry Pi 4 running Python code. The advanced AI-based control can automatically adapt the actions of the bionic leg to seamlessly switch activities, such as going from walking, to going up stairs, to down a ramp.

While the total cost of the parts required is around \$28,500, it's really a research tool rather than a build-at-home solution, with the ultimate goal of developing a bio-inspired blueprint for improved control of assistive technologies. So it may well eventually lead to the ability to manufacture low-cost, high-performance robotic legs to help many more people.



Right ♦ Research is carried out with amputees to test the AI controls of the open-source robotic leg



E-NABLE

enablingthefuture.org

Founded in 2011, the e-NABLE Community is a global network of around 40,000 volunteers who are using their 3D printers, design skills, and personal time to create free 3D-printed prosthetic hands for those in need – with the goal of providing them to underserved populations around the world.

Community members – including engineers, scientists, medical professionals, and tinkerers – collaborate on ways to help improve the open-source 3D printable designs for hands and arms for those who were born missing fingers or who have lost them due to war, disease, or natural disaster.

While the hands are not fully functional prosthetic devices, as the fingers don't have individual movement, they are helpful for children, in particular, who might not otherwise have access to a prosthetic hand. Children can use them for simple tasks like holding water bottles, using two hands to ride a bike, holding sports equipment like a cricket bat, and catching a ball.

The e-NABLE Hub has 3D printing files and detailed assembly instructions for a variety of hand, arm, and shoulder designs, some with specialist use cases such as a guitar pick holder. Raw materials for a project can cost as little as \$35.

OPENBIONICS

openbionics.org

Originally inspired by the Yale OpenHand Project, OpenBionics is an open-source initiative developing affordable, lightweight robotic and bionic hand devices that can be easily reproduced.

Low-cost devices include a robot hand, prosthetic hand, and exoskeleton glove, all of which can be modified by developers for specific types of tasks. With a modular design that can accommodate a total of four fingers, and a differential mechanism that allows for independent finger movements, the robot hand is very capable of grasping various everyday objects.

The prosthetic hand has a bio-inspired design that structurally reproduces the movements of human fingers.

All the designs, electronics, and software required to replicate and control the devices can be found in the project's GitHub repository: github.com/OpenBionics.



Above ♦ The body-powered exo-glove was designed to enhance the grasping capabilities of the user

Left □ Different sizes of the 3D-printed Kinetic Hand designed by e-NABLE community member Mat Bowtell

Below ♦ The Brunel robot hand offers precision control and can be programmed via the Arduino IDE



OPENBIONICS LABS

openbionicslabs.com

While the team behind this project has gone on to create a commercial company (openbionics.com, not to be confused with openbionics.org, above) which produces the robotic Hero Arm – with the option of some cool-looking superhero and Star Wars covers – their original open-source designs for robotic hands remain available to download.

Although the kits are sadly no longer available from the site's shop, they live on through this project. There are 3D-printable parts and PCB designs that come together to make an Arduino-compatible appendage. →



WHEELCHAIR WONDERS

Numerous projects enable wheelchair users to enjoy greater mobility and accessibility, and have fun



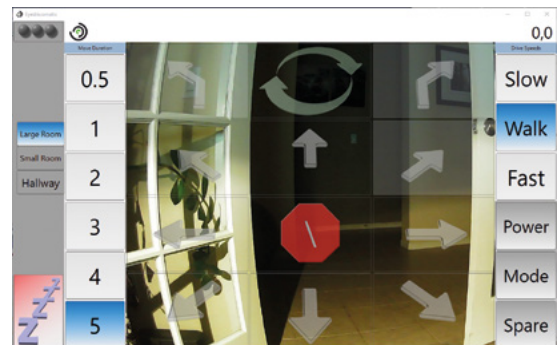
EYEDRIVEOMATIC

eyedrivoatic.org

Whether caused by spinal injury or diseases such as ALS, quadriplegia makes it almost impossible for people to control their wheelchairs due to the inability to produce controlled movement in any of their limbs or head. Eyedrivoatic solves the problem by using an eyegaze system to relay controls to a motorised wheelchair's joystick, so the user can make eye movements to control the chair.

Below ♦ Taking commands from the eyegaze system, the 'Electronic Hand' moves the chair's joystick

As someone living with advanced ALS, Eyedrivoatic inventor Patrick Joyce understood



the mobility issues facing people in his position. Like many others, he didn't actually own either the wheelchair or the eyegaze system that he used. So he needed a solution that would interface the eyegaze system and physically move his chair's joystick without requiring any modifications.

Eyedrivoatic is a two-part system: a Windows application to interface with an eyegaze system, and a device, dubbed the 'Electronic Hand', which fits over the chair's joystick and contains servos to move it, along with three relays for switch activation. Taking commands from the software, it works just like a hand on the joystick.

Crucially, there is no need to make any modifications to existing user hardware, and the system can work with virtually any wheelchair and eyegaze software combination.

WALKIN' & ROLLIN' COSTUMES

walkinrollin.org

Every kid deserves a cool costume: that's the concept behind the Walkin' & Rollin' Costumes project, which will provide any child using special-needs equipment with a unique and custom-built costume, free of charge to the family. We first covered it back in HackSpace issue 21 (hsmag.cc/issue21).

The project's origins date back to 2008, when Lon Davis created a WALL-E costume to fit around his three-year-old son Reese's wheelchair for Halloween. After making further costumes for Reese, which were shared on social media, Lon began receiving requests from people all over the US to build their child a wheelchair costume. Walkin' & Rollin' Costumes was born.



Above ♦ Who ya gonna call? Ghostbusters!

Of the many costumes created so far, notable builds include the Ghostbusters Ecto-1 car (which even played the theme tune), the DeLorean from *Back to the Future*, and an Ant-Man costume.

Since every handmade costume costs around \$250 to create, this non-profit organisation relies on donations from the public. As demand is high, and shipping costs are also an issue, new 'chapters' of the organisation have been created in different cities by volunteers. Why not visit the website to find out how to start your own?

FREEDOM WING ADAPTER

hsmag.cc/freedomwing

The first system that enables you to control Xbox games by plugging in a motorised wheelchair, the Freedom Wing Adapter is a boon for wheelchair users who enjoy

video gaming since they are already adept at moving their own mini joystick on the chair.

Created by AbleGamers COO Steve Spohn in collaboration with Bill Binko from ATMAkers, the device connects the wheelchair's joystick to an Xbox Adaptive Controller with no modifications required. It works by reading the input from a standard proportional mini joystick, via its nine-pin connector, and feeding it into a very simple JoyReader microcontroller board that converts it so it can be fed into the Xbox Adaptive Controller.



Furthermore, since the JoyReader fits the Adafruit Feather standard, it could also be used to control devices via Bluetooth or WiFi, enabling a great deal of flexibility. Even when fitted with a 3D-printed enclosure and Feather, the bill of materials for the Freedom Wing Adapter is only around \$30 to \$35. →



Left ♦ Connecting the Freedom Wing Adapter to an Xbox Adaptive Controller



SENSORY AIDS

For people with visual impairment or other sensory loss, there's a raft of accessibility aids and projects

TALKING BARCODE READER

hsmag.cc/TalkingBarcode

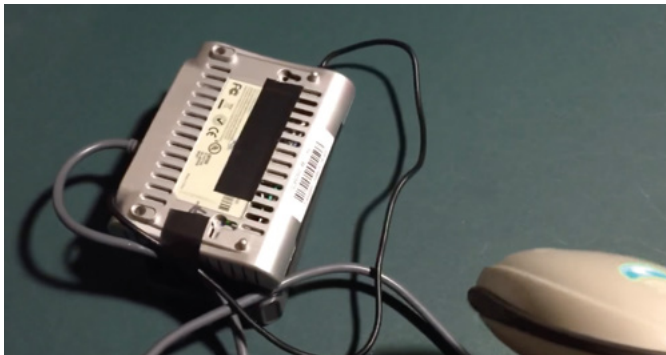
P eople with visual impairment can find it difficult to read the labels on groceries in stores or in the cupboards at home. That was the

case for Russell Grockett's mother, so he decided to create a DIY talking barcode scanner to help her. When an item's barcode is scanned, the device looks it up in an offline database and reads out its name.

To make the device, Russell adapted an inexpensive standard laser barcode scanner by adding a Raspberry Pi Zero W. It uses the Flite text-to-speech tool to speak the item's name through a tiny speaker.

Right Motors in the backpack's straps vibrate to tell the user which way to turn

Below Everything is self-contained in a small case, no internet connection required



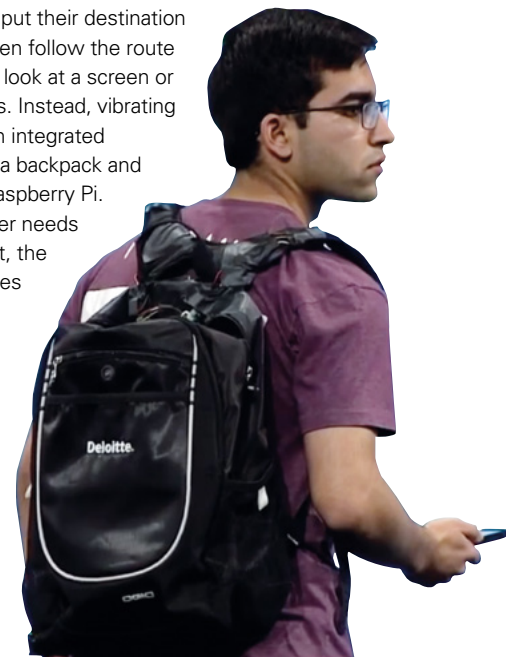
BACKMAP

hsmag.cc/BackMap

H ackathons are great for coming up with new technology ideas and solutions, and that was the case with Disrupt NY 2017, where a team including Shashank Sharma

invented the BackMap. It uses haptic feedback to help visually impaired people navigate to a location without needing to look at a screen.

The user can input their destination via an app, and then follow the route without having to look at a screen or listen to directions. Instead, vibrating motors have been integrated into the straps of a backpack and hooked up to a Raspberry Pi. Whenever the user needs to turn left or right, the Raspberry Pi makes the respective motor vibrate. It can even be used indoors using beacons. For more details, see hsmag.cc/BackMapGH.

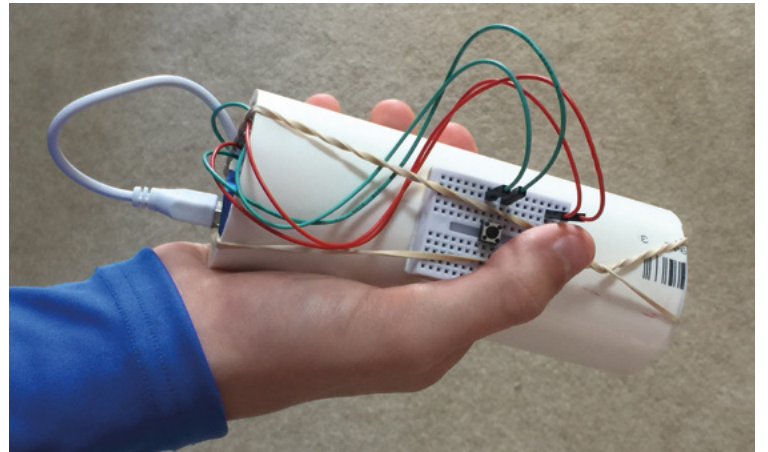




BLIND ARDUINO PROJECT

hsmag.cc/BlindArduino

Blind people can be hackers and makers too. The Blind Arduino Project grew out of a community effort led by Dr Joshua Miele to better understand barriers faced by blind and visually impaired people wanting to participate in the vibrant global culture of DIY hardware prototyping. Based at the Smith-Kettlewell Eye Research Institute, the project organises occasional hands-on workshops for children and adults on Arduino, coding, soldering, and other concrete skills for blind makers. Its ongoing activities and tutorials are documented at the Blind Arduino Blog (hsmag.cc/BlindArduinoBlog), which includes numerous articles documenting the tools and techniques of the blind Arduino maker.



SEEING WAND

hsmag.cc/SeeingWand

Inspired by a blind cousin, Robert 'Hobbes' Zakon built a Seeing Wand that can speak the name of whatever it's pointed at, akin to a magic wand. Housed in a makeshift PVC tube, a Raspberry Pi Zero W is connected to a Camera Module that takes a photo when a small push-button is pressed. The image is sent to Microsoft's Cognitive Services Computer Vision API to obtain a description, which is then spoken – using the eSpeak speech synthesizer – through a Pimoroni Speaker pHAT. While the prototype design is rather makeshift, with an exposed mini breadboard and wires, the wand really works and could prove a more tactile, lower-cost alternative to smartphone apps with similar functionality.

Above ♦ This DIY Seeing Wand could be built in a matter of hours

Left ▣ A project showcased on the Blind Arduino Blog

Credit: Larry Rippel

Below ♦ The BrailleBox relays online news stories via a dynamic Braille display

BRAILLEBOX

hsmag.cc/BrailleBox

This project is by Joe Birch, whose family has a history of a genetic condition known as Retinitis Pigmentosa, which causes eyesight to deteriorate over time. The BrailleBox fetches the top news article from a news API, converts the string representation into a series of Braille sequences, and plays them one at a time on a refreshable Braille display.



The low-cost device makes use of a Raspberry Pi and solenoids to trigger the six points (wooden balls) on the Braille display. You can read more about the build at hsmag.cc/BrailleBoxBuild. Joe has also created the Chord Assist (chordassist.com), an accessible smart guitar for blind, deaf, and mute people. →



DIGITAL ACCESSIBILITY AND SPEECH AIDS

Everyone can take part in computing with digital accessibility aids, while speech aids give a voice to many people

THE A11Y PROJECT

a11yproject.com




A 11Y stands for 'accessibility' (11 characters between A and Y), and this project is a community-driven effort to make digital accessibility easier.

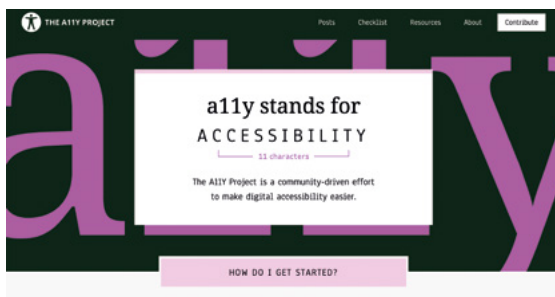
While vital for so many disabled people, accessibility is something that's often overlooked in traditional digital design and development education. The A11Y Project strives to be a living example of how to create beautiful, accessible, and inclusive digital experiences.

So, if you want to ensure your website or app will work with assistive technology and therefore be accessible to all, The A11Y Project site is the place to come. Along with a blog and accessibility checklist, it features a comprehensive resources section.



Right  SuperNova is a combined screen reader and magnifier

Below  The A11Y Project site is a treasure-trove of information on digital accessibility



SCREEN READERS

hsmag.cc/ScreenReaders



A screen reader enables blind or visually impaired people to use a computer or smartphone and access digital content. In short,

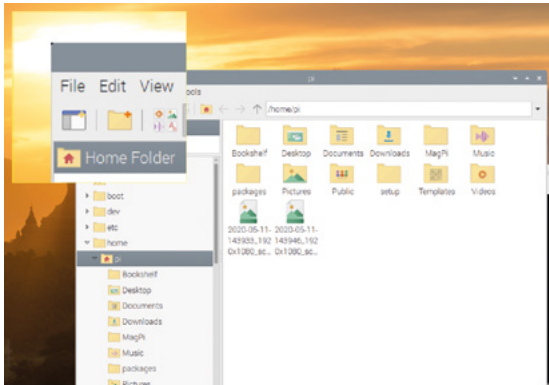
it reads out loud what is shown on screen, enabling people to navigate websites and apps via the speech output.

Settings enable users to adapt a screen reader for their own needs, such as slowing the speed of speech or changing the language. Depending on the screen reader, it may also be possible to use it with a dynamic Braille display.

There are numerous screen readers available, including those built into operating systems.

For partially sighted people, some screen readers and operating systems include a magnifier tool to enlarge a selected area of the screen.





RASPBERRY PI OS

hsmag.cc/RaspberryPiOS

The default operating system for Raspberry Pi computers, based on Debian Linux, Raspberry Pi OS (formerly known as Raspbian) now includes a couple of useful tools to aid accessibility.

The Orca screen reader is a standard application that uses synthesised speech to read out on-screen items such as window titles, menus, and button labels on the desktop. It will also work for the Firefox and Chromium web browsers. Its settings can be accessed by pressing the **INSERT + SPACE** keyboard shortcut.

Raspberry Pi OS also has its own purpose-built Magnifier tool that can be installed from the Recommended Software menu. You can then click a magnifying glass icon in the taskbar or press **CTRL+ALT+M** to activate the Magnifier. See the blog post (hsmag.cc/RPiMagnifier) for more details.

EARSWITCH ✕

hsmag.cc/Earswitch

This ingenious project by Nick GPx enables the user to control assistive technology software by contracting a muscle in their middle ear. More than 75% of people are able to contract their tensor tympani (making a rumbling sound or muffling of the hearing, such as when yawning or tightly closing their eyes), and more than 20% can do so in isolation from other movements.

EYE TRACKER INDICATOR FACTORY

hsmag.cc/ETIndicator

Eye-tracking communications software is life-changing for people with speech and mobility issues. When engaging in a conversation,

however, there's no easy way of others knowing whether they are operating their communications system to prepare a synthesised speech response.

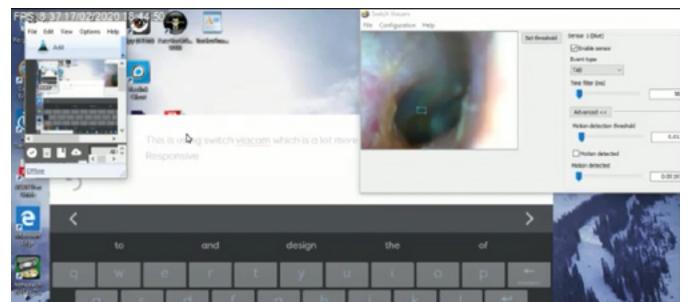
Matt Oppenheim's project addresses the issue by creating a visual indicator for when the communications software is in use, to enable a more natural two-way conversation. You talk to the person who is using the communication software; when you see the light flashing, you know that a reply is being prepared.

A script running on the communication device detects a change in the number of black pixels in the speech area of its interface. It then triggers a micro:bit to flash its LEDs. Simple but effective.



Above ⬆ The micro:bit flashes when the person is preparing a response with their communication device

Left ⬅ Magnify a selected area of the Raspberry Pi OS desktop with the Magnifier tool

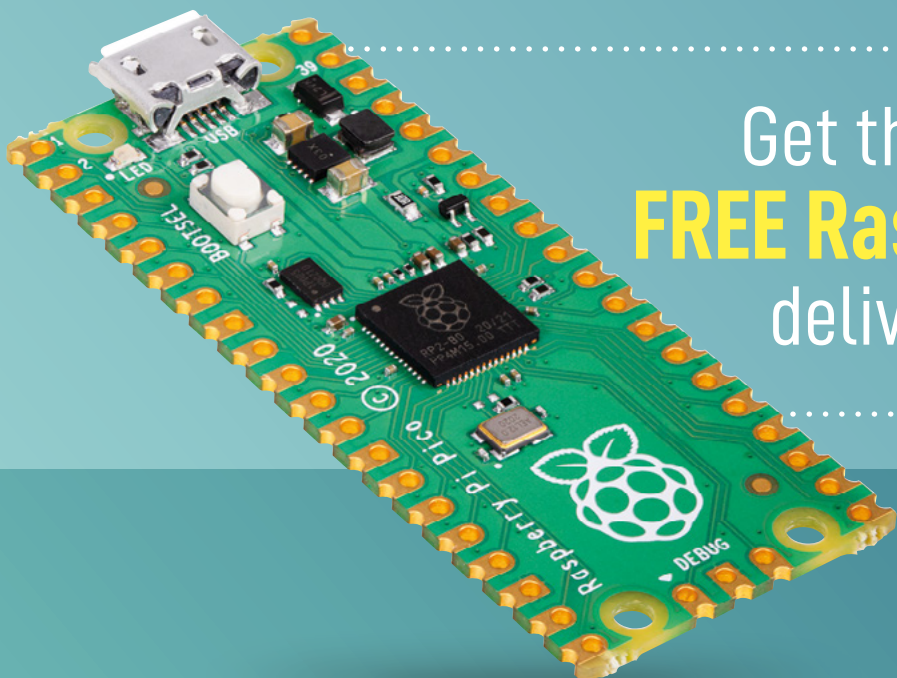


This causes movement of the eardrum which can be seen using a USB auroscope (camera) placed in the ear. By using iSpy or Switch ViaCam software to detect motion, Nick has successfully used the proof-of-concept system for hands-free control of assistive software (Grid3), an on-screen keyboard, and to function as a 'left click' for head tracking. □

Above ⬆ Using the Earswitch to type on Grid 3, configured to use a TAB key press when the switch is triggered

SUBSCRIBE TODAY

FOR JUST £10



Get three issues plus a
FREE Raspberry Pi Pico
delivered to your door

..... hsmag.cc/FreePico

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

Subscription will continue quarterly unless cancelled

SUBSCRIBE on app stores

From £2.29



Buy now: hsmag.cc/subscribe

Free Pico with print subscription only



MAKE | BUILD | HACK | CREATE

HackSpace

MAKE | BUILD | HACK | CREATE

TECHNOLOGY IN YOUR HANDS

hsmag.cc | February 2022 | Issue #51

FREECAD
Digital modelling using frames

EGG ART
A CNC egg-drawing machine

FreeCAD
Model objects from real life

GO
Build your bricks with Technic Hub Raspberry Pi

Security
Make your own lock picking tools

DATA GRAFFITI

TECHNOLOGY FOR **EVERY BODY**

People are different
Machines should be too
DIYers are making this happen

LIGHT-UP JEWELLERY
Add LEDs to your earrings

CIRCUITPYTHON
Create animations with Raspberry Pi Pico

STRING PICO LASERS BADGES

VINYL CUTTING
Customise your mug

MA
LEARN
Improve your building with artificial intelligence

LIGHT PAINTING
Take photography to the next level

WRAPPING 3D PRINTING

SAVE 44%

HOW

By Robert Hamilton-MacLaren

MADE

Last February, my children rushed in with a copy of the latest HackSpace magazine for me, excited because of the gold lettering and free gift on

the cover. The gift was a microcontroller, and my children's enthusiasm was a catalyst for me to come up with a project that would appeal to and involve them. This would be the first project that I had done with a microcontroller. I picked the Python route as it seemed the closest step from BASIC that I had used many years ago at school. I worked through the tutorials published in HackSpace issue 39 (hsmag.cc/issue39), and these gave me a good understanding of how to flash the Pico with Python and connect components. If you are starting out with the Pico, I recommend this as a great place to begin.

With small children, we have an abundance of toy cars and building bricks and, whilst rolling cars down improvised tracks built from books and supported by DUPLO, I thought it would be great if we could set up a race track and use the Pico to determine the winner. After research on YouTube, watching videos of Hot Wheels cars being raced down roof sheets and guttering, I was left wondering how I could transport such a large track. I needed to find a more portable solution, and inspiration arrived in the form of plastic conduit. I took a trip to the local electrical wholesaler to find

RASPBERRY PI PICO HOT WHEELS DRAG RACE TRACK

A microcontroller for micro racing


What I used

- Raspberry Pi Pico
- 4 × red LEDs
- 4 × green LEDs
- 8 × 330Ω resistors
- Transistor and resistor
- Piezo buzzer
- 2×3mm break beam sensors
- Push-to-make button
- Electrical wire
- Stripboard
- Race track
- Toy cars

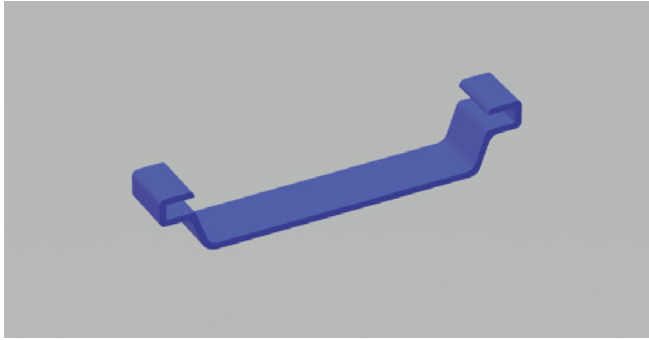


the optimum width for the cars to run along (38mm) and the 2m lengths easily flexed to fit into the car. Hot Wheels track would be another option but, for a fair race, the competing tracks would need to be identical and quite a large collection would be needed. A 3D-printed adapter could probably be made to incorporate loops and jumps to add to the excitement.

In terms of components and materials for the build, this is a low-cost project (£30 or less, depending on components you already have). Of course, you will need a soldering →

Above  The final race track in action

FEATURE



Above ♦
Custom brick to hold and connector to join the track

iron and a computer to program the Pico, but other parts can be adapted or improvised using whatever you have available. I chose to house the sensors in 3D-printed parts that fit to DUPLO because I could save a lot of printing time. All that was needed was a few bespoke bricks. Thingiverse proved a great starting point with bricks that could be modified to fit the chosen sensor. On Thingiverse, 'klamowk' has published a customisable DUPLO-compatible brick:

[thingiverse.com/thing:4815088](https://www.thingiverse.com/thing:4815088). The required size can be entered, which takes the hard work out of making a brick that will fit to others. Other benefits of using DUPLO are that you don't need much space to store the completed project because it can easily be disassembled, and the layout can be varied every time.

Don't be put off doing a similar project at this point if you haven't got access to a 3D printer – you could easily mount the sensors in wood or even cardboard.

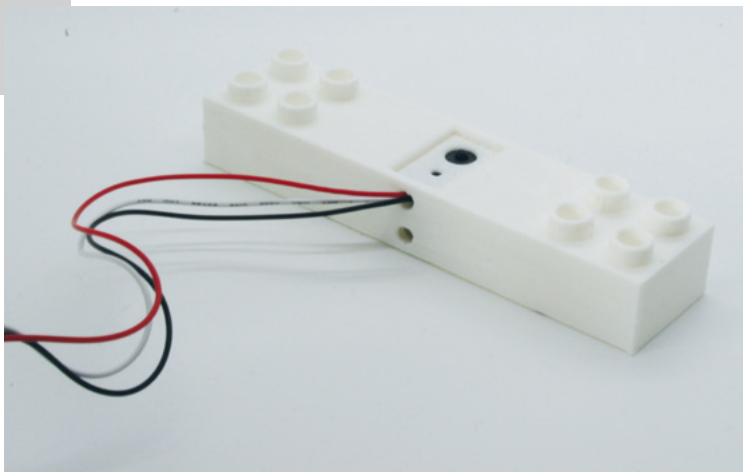
THE TRACK

The track on its own is quite flexible so does need supporting. I designed a track connector that slides over two pieces of conduit to join them. If you don't have a 3D printer, the track could easily be supported and joined by fixing it to a batten (this would be very rigid, but not so easy to collapse to store). DUPLO towers were used at intervals to support the track along its length up to the finish line, with custom bricks holding it in place. I got the children involved in building the towers, and we were having fun racing cars even before I'd started on the electronics.

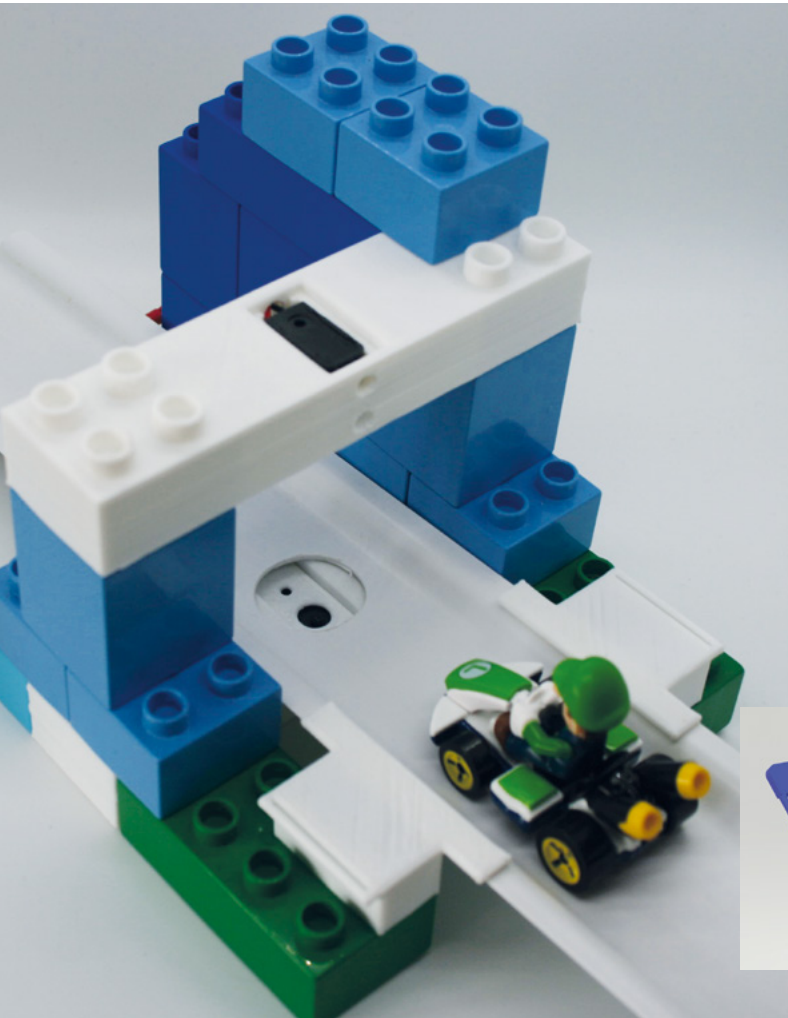
BREAK BEAM SENSORS

We tried timing the races manually, and the children realised how difficult it was to tell by eye which car had won the race. Browsing the available sensors, I came across the Adafruit Breakbeam sensor ([hsmag.cc/BreakBeam](https://www.hsmag.cc/BreakBeam)), which will operate at the Pico's 3.3V (certainly over the distance needed for the project). There are two parts (an infrared LED and a sensor) and, as both components are the same size, I was able to design a brick that could be used for both and know that they would be perfectly aligned. I chose to place the sensor vertically above and below the finish line. The conduit track can easily be drilled to allow the beam to pass through without compromising the car's path. Once connected to the Pico, the pin is high until the beam is broken. I set up a trigger event to capture the pin going low (see **Code 1** listing). The code run by the trigger event is really simple: if an `in_a_race` flag is set to

“THE TRACK ON ITS OWN IS QUITE FLEXIBLE”




Left ♦
Light and Sensor fits the same sized 3D-printed housing, ensuring they align

**CODE 1**

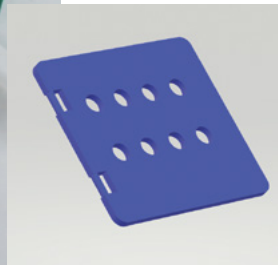
```
breakbeam1 = Pin(5, Pin.IN, Pin.PULL_UP)

def beam1broken(pin):
    #set beam broken flag only if in a race
    global beam1_broken
    if in_a_race == True:
        beam1_broken = True

breakbeam1.irq(trigger=Pin.IRQ_FALLING, handler=beam1broken)
```

Left  Assembled finish line, ready for track to be fitted

Below  Modified DUPLO window to house LEDs



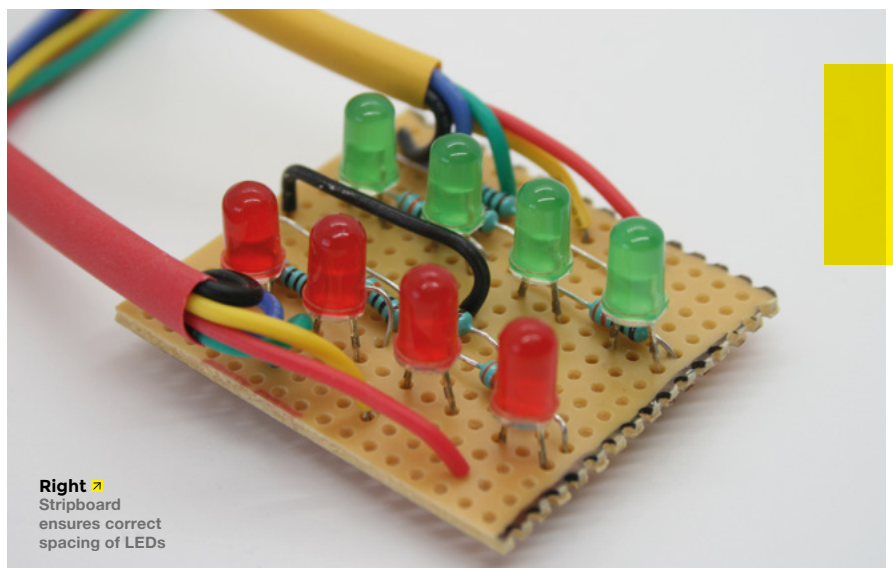
I did start looking at options such as adapting an SPI module which would have only required a few wires. I opted to keep things simple, and connected each LED to a pin on the Pico as there were plenty available for this project. The eight LEDs were soldered onto a stripboard with appropriate resistors, as this ensured the correct spacing for the printed door. The finished circuit fitted nicely into the window casement.


We thought it would be fun to have an audible start to the race, but the first piezo buzzer I used was far too quiet. I purchased a louder one on eBay, which I thought had a built-in transistor to operate (as there is not enough current to power direct from the pin). After testing, it turned out there was no transistor, but once I wired it up with an external one, it provided a good loud beep. >

True, then the `beam1_broken` variable is set to **True**. The rest of the code is within the main loop so that the trigger code is run as quickly as possible. Initially, I had thought I would need to deal with multiple signals as the car passed through the beam. But actually, I haven't had any trouble with false triggers and I just need to know the first instance of the beam being broken.

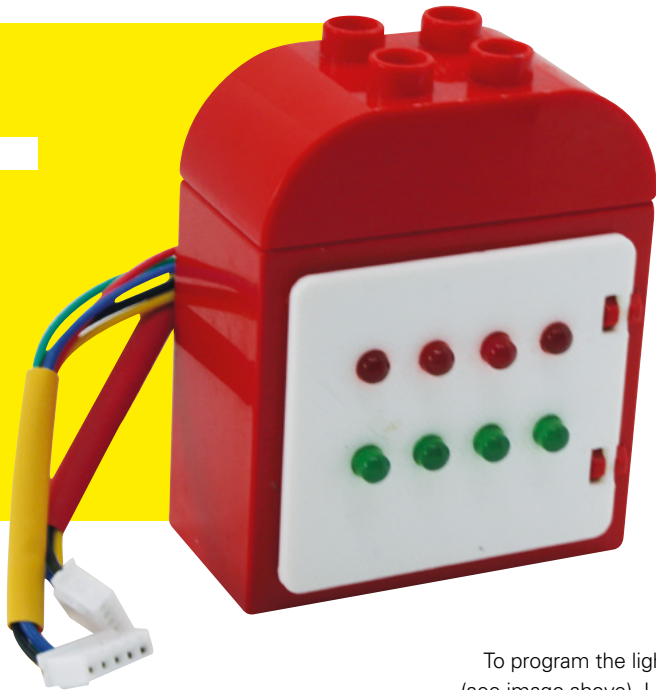
STARTING LIGHTS

Rather than a basic stop/go light, I wanted to create the feel of a race track starting grid with a series of red and green lights similar to those seen on Formula 1. I started designing a brick to house the lights, but realised a DUPLO window brick combined with a 3D-printed front panel for housing the starting lights would considerably cut down the printing time and, of course, a blank was available on Thingiverse [thingiverse.com/thing:2888128](https://www.thingiverse.com/thing/2888128).

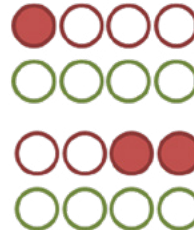


Right  Stripboard ensures correct spacing of LEDs

FEATURE



End of the race



Above ⬆
Red lights alternate to show 1st and 2nd place

Left ⬅
Completed starting lights

connectors make the project easier to take apart, but are not essential. I added some extra connections to facilitate future upgrades. The circuit was covered with a 3D-printed DUPLO brick-size case. A downside of using

DUPLO for the construction is that there is nowhere to route the cables. If repeating this project, I would have built the starting lights and control box into one enclosure to reduce the number of exposed wires.

To program the light and beep sequence (see image above), I created a starting light routine which can be called from the main loop each time a race is started.

The race is started (or reset) by pressing a button mounted in a printed custom DUPLO brick. In the future, I would like to change this to a remote control to reduce the number of wires.

The lights also perform a second function; they are used to show first and second place at the end of the race, with the red lights flashing accordingly to indicate the winner (see End of the race, above).

THE CODE

The main code is a loop within which there is a sub loop that runs while a race is taking place. This sub loop looks to determine if an event has been triggered by a beam being broken. If it is the first occurrence, it is recorded as the winner, and the second occurrence finishes the race and lights display the final race positions.

The setup can be run connected to the computer, but as the intention was to have this as a standalone toy, I saved the code on the Pico as **main.py** so that it would run each time the circuit was powered. I used a USB power adapter, but a battery would make this even more portable.

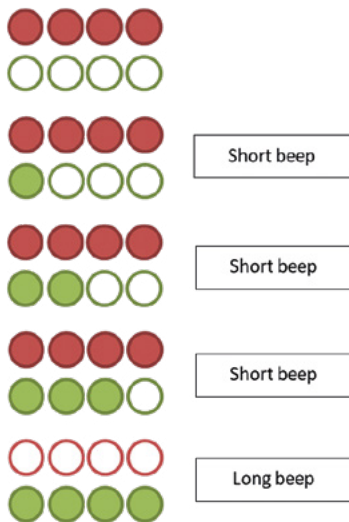
IN ACTION

After some initial testing, the track was put to use at a party. Children quickly got to grips with how to operate and interpret the results. Initially they were just racing cars, but soon they wanted to answer questions like, "Which types of cars were fastest?" and "Does the same car always win?"

FUTURE IDEAS

In terms of construction, there are plenty of potential enhancements, such as ramps and jumps or even a car launcher. These could be 3D-printed or made from cardboard. An upgrade to the electronics would be to display the race finishing times (the full code already stores the times in variables which could be output to a display). Older children might enjoy the challenge of modifying

Starting Sequence



Short beep

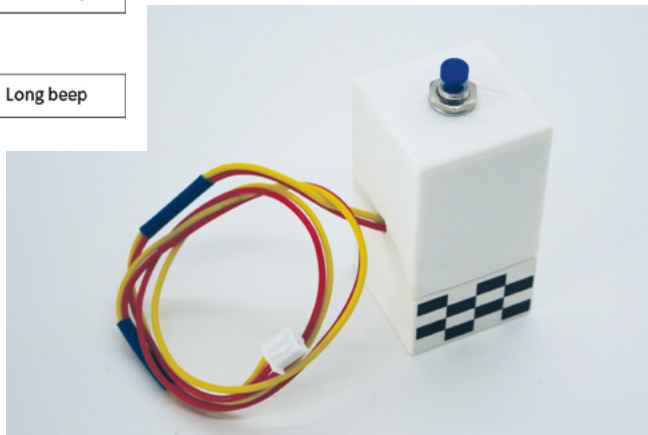
Short beep

Short beep

Long beep

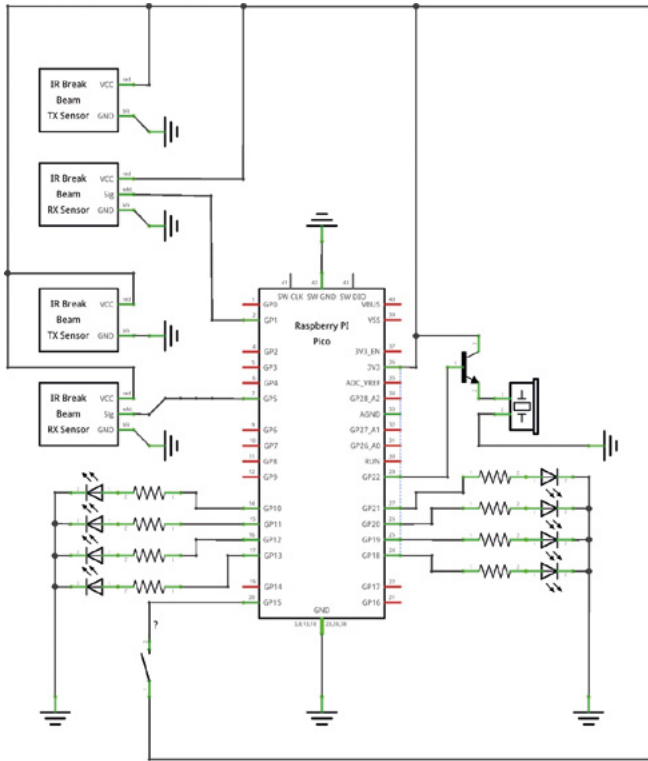
BRINGING ALL THE PARTS TOGETHER

After initially testing the various parts on a breadboard, I decided to make the wiring more permanent using stripboard. The JST



Above ⬆
Starting sequence

Right ➡
Start and stop button printed DUPLO brick



Above ⚡
The schematic for the circuit

Below ⚡
Race track in action

the code further. For example, a relay race where as many cars as possible are sent down the track in two minutes, or a time trial which uses both of the sensors on the same track and measures the time between the two to calculate the speed.

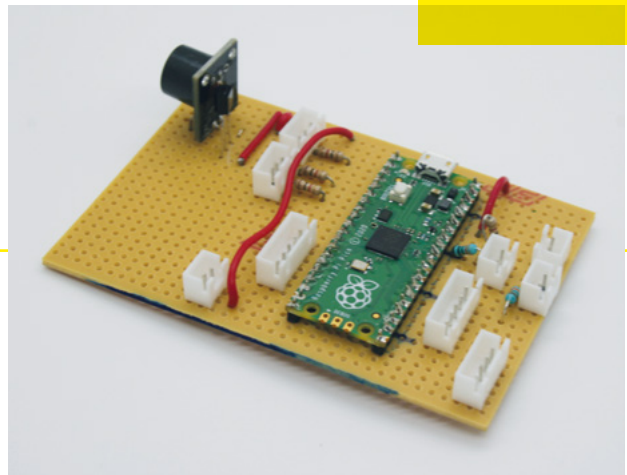
CONCLUSION

The Pico Drag Race Track has provided us with hours of entertainment, and the ease with which it can be packed up, transported, and reassembled has meant it has made regular appearances at family gatherings. This project has developed my 3D printing, electronics, and programming skills and has also been fun to build, especially as I've been able to involve my children with so much of the process. It is an accessible and versatile project that can be adapted in many different ways (if cars aren't your thing, how about modifying this for a marble run?). I've put the code and 3D models on a Facebook group, [hsmag.cc/PicoRace](https://www.facebook.com/hsmag.cc/PicoRace), should you want to create something similar. □

PSEUDOCODE

- When button is pressed to start the race
- Call the starting lights and beeps routine
- Enter the in-race loop
 - First car past the finish line – record the winner
 - Second car past the finish line – the race is finished
- When race has finished, exit the race loop
- Display the winner
- Repeat

Below ↓
JST connectors allows quick assembly with extras for expansion



STRING BAGS



Grocery stores often sell fruit and vegetables in stretchy sacks. Rosie Hattersley looks at how to reuse the discarded plastic mesh



Rosie Hattersley

[@RosieHattersley](#)

Rosie Hattersley writes tech, craft, and life hacks and tweets [@RosieHattersley](#).



One of the really frustrating things about supermarket shopping is all the packaging that needs to be removed before groceries get as far as the fridge or fruit bowl.

Accruing half a bin full of plastic ought not to be the result of a weekly shop. Along with food miles, cutting down on this extraneous, often non-recyclable wrap is surely something supermarkets should be better about. If one of your New Year's resolutions is to throw away less, reusing some of this packaging in sustainable ways is a great goal.

The string bag itself has had something of a revival of late, offering a great way to carry your shopping

Ironically, one of the most common forms of packaging, the fruit net, derives from its more eco-friendly cousin, the string bag. Lemons, limes, and other fruit are routinely packaged in nylon mesh nets for convenience, but their larger versions are generally made from string or silk thread. Japanese shellfish divers and mid-20th century shoppers favoured these highly portable sacks that took up barely any room, while in Soviet Russia, avoska bags were so-called because citizens carried them on the off-chance that something worth having would be on sale.

However, if you find yourself accruing the nylon mesh kind, discarding them just won't do. It's no secret that nylon plastic is not biodegradable and items made from it often end up in landfill or clogging up beaches. Once you've got your fruit and veg home, you could use the mesh bag it came in to hold other items that you need to store together – laundry, PE kits, and so on. You could protect perishable goods from flies by turning the sack into a food cover using bamboo struts to create a taut dome over the top of a plate or tray. Of course, you might want to turn it into something decorative instead, perhaps a no-maintenance bonsai tree: hsmag.cc/PlasticBonsai. Here are some other potential options.

"CUTTING DOWN ON THIS EXTRANEIOUS, OFTEN NON-RECYCLABLE WRAP IS SURELY SOMETHING SUPERMARKETS SHOULD BE BETTER ABOUT"

home from the local supermarket or zero waste shop, as well as tapping into the interest in handicrafts such as crochet and macramé.

BODY SCRUBS

I f you're trying to cut down on single-use items such as toiletry bottles, switching to eco-friendly ways of getting clean seems like a good idea.

Pack the bottom of a fruit net with solid soap, twist the netting several times to secure the contents in place and then fold the remaining net bag over the half you've just filled up. Secure it all by carefully melting a couple of strands together using a lit match, or snip and tie up ends. A bonus here is that

the slightly rough plastic will help slough off dead skin cells – ideal for banishing traces of oil, glue, or general grot embedded in your hands. Meanwhile, the gradually diminishing soap stack can be tied tighter to ensure odds and ends of otherwise lost-down-the-plug-hole soap actually ends up getting used.

Should you find the approach appealing, you can easily apply the same technique to your washing up, foregoing washing up liquid bottles in favour of bars of washing up soap.

Project Maker
RMS ART

Project Link
hsmag.cc/Scrubs

Left ♦
Keep yourself clean and keep the planet clean



BIRD FEEDER

Project Maker
LEARN TO GROW

Project Link
hsmag.cc/BirdFeeder

W intertime can be hard going for our feathered friends, so putting together birdseed cakes held together with fat, and placed in a mesh sack for easy access by peckish garden visitors, is another great option.

The process is exactly as simple as it sounds. You can either make a fat ball or purchase one, then place it in a bag and hang it up. →



CROCHETED RUG OR BASKET

Project Maker
**WALDEMAR
SHA**

Project Link
hsmag.cc/MeshBagRug

Making a string bag into another form opens up the reuse possibilities.

Cutting the plastic into broad loops or strips that you then twist to give them strength, you can turn your plastic string sack into a form of plarn (plastic or fabric-based yarn you use as you would regular knitting or crochet yarn). A demonstration video shows how plarn can be incorporated into a circular rug using string or cotton yarn to secure individual strips of plarn into place. To make the plarn, you can either snip off the bottom

of a sack and cut a continuous strip at a slight angle (see hsmag.cc/MakePlarn), or make shorter hoops by cutting straight across the bag. A further option, shown in a clip in the Instructable, involves twisting and stretching complete potato sack-sized bags to give them tension before following the instructions at hsmag.cc/CrochetRound to crochet the plarn into place.

Once you get the hang of creating taut surfaces, you could extend the plarned string bag upwards to form the sides of a basket or box.



Right ♦
This technique can be used to make a huge number of different items

UPCYCLED GHOST NET BRACELET

Etsy artist Matthew Longley makes all sorts of upcycled items from trash he rescues from the seas off the North Yorkshire coast. Given the issue of microplastics ending up in the water table and seas, his methodical approach to salvaging driftwood and so-called ghost nets (nylon-covered ropes used for fishing) for use in artworks and jewellery is admirable.

Our mesh fruit nets share many qualities with the salvaged plastic from fishing nets and marine debris that systematic sea scouder Matthew collects. A relatively simple reuse idea is to create interlocking bracelet bands secured with a metal clasp. Strips of plarned plastic are plaited into tight strands, then paired with contrasting coloured strips and secured at each end. The unisex dual- or tricolour bands could even be embellished by embedding smooth sea-glass jewels.



Project Maker
CCYCLEBEACHART

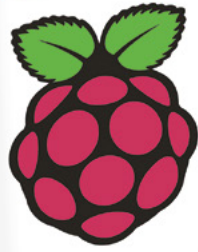
Project Link
hsmag.cc/NetBracelet

ORGANISER BOARD

Get organised for the coming year by creating a pinboard that keeps all your assembled notes and handy stationery items in order. By stretching an oblong of mesh netting over a cork-board, you can hang items off it as well as pinning them to the surface. Because the nylon mesh is pliable and stretchy, you don't need to be too fussy about cutting the fabric to size. You simply need to remove any metal clips that bunch up the ends to

form the fruit sack's shape, or snip along a seam to loosen the edges.

To attach the netting to your pinboard, tap in a couple of nails towards the top left and top right of the frame on the reverse side, then a couple more at the bottom and halfway down each side. Finally, pull the netting into place. If you're after a more rigid mesh grid, spray the sack with epoxy glue and let it dry out while stretched over a frame, then nail it into place as described. □



THE *Official* RASPBERRY PI HANDBOOK 2022



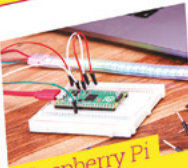
50 Raspberry Pi
Hacks & Hints



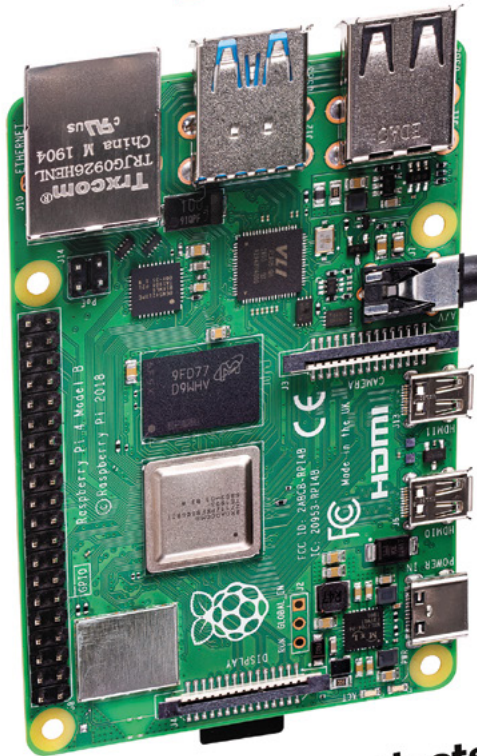
iPod Classic
Spotify Player



BeccaCam
Digital Camera



Raspberry Pi
Pico projects



Build a
Home Assistant



Raspberry Pi
Amiga 600



Walking
Bipedal Robot

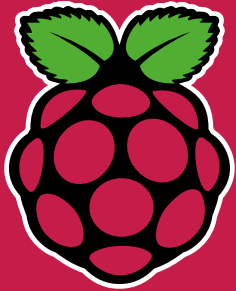


Start incredible projects
with Raspberry Pi

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

THE OFFICIAL RASPBERRY PI HANDBOOK 2022

THE *Official*



RASPBERRY PI HANDBOOK

2022



200 PAGES OF RASPBERRY PI

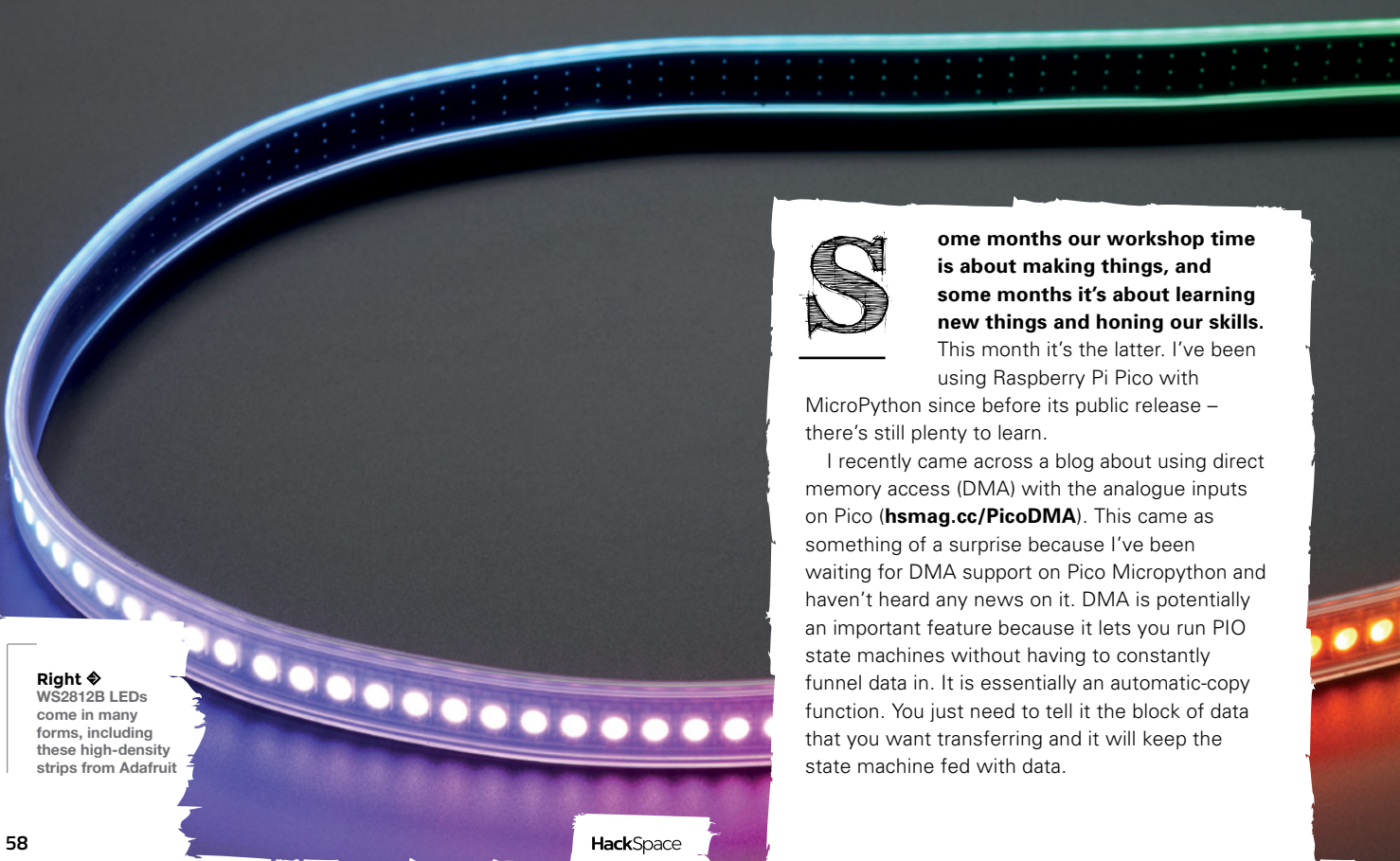
- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects

Buy online: magpi.cc/store

IN THE WORKSHOP: MicroPython direct memory access

By Ben Everard

Getting data out of Raspberry Pi Pico quickly



Right ♦ WS2812B LEDs come in many forms, including these high-density strips from Adafruit

S

ome months our workshop time is about making things, and some months it's about learning new things and honing our skills.

This month it's the latter. I've been using Raspberry Pi Pico with

MicroPython since before its public release – there's still plenty to learn.

I recently came across a blog about using direct memory access (DMA) with the analogue inputs on Pico (hsmag.cc/PicoDMA). This came as something of a surprise because I've been waiting for DMA support on Pico MicroPython and haven't heard any news on it. DMA is potentially an important feature because it lets you run PIO state machines without having to constantly funnel data in. It is essentially an automatic-copy function. You just need to tell it the block of data that you want transferring and it will keep the state machine fed with data.

GitHub repository page for `benevpi/MicroPython_PIO_Music_DMA`. The repository is public and has 1 star and 0 forks. The main content area shows a commit history table with the following entries:

File	Commit Message	Time Ago
<code>DMA_test.py</code>	Create <code>DMA_test.py</code>	8 days ago
<code>LICENSE</code>	Initial commit	8 days ago
<code>PIOBeep.py</code>	Create <code>PIOBeep.py</code>	8 days ago
<code>README.md</code>	Update <code>README.md</code>	8 days ago
<code>rp2040_pio_dma.py</code>	Create <code>rp2040_pio_dma.py</code>	8 days ago

The README.md content includes the title `MicroPython_PIO_Music_DMA`. The right sidebar shows the repository description: "Play music on Raspberry Pi Pico Without CPU involvement".

Left
Our PIO DMA code can be downloaded from hsmag.cc/MicroPythonDMA

Looking in a bit more detail, the blog's author, Jeremy P Bentham, had used MicroPython's `uctypes` module to access and set the underlying registers on RP2040 to enable DMA. In his example, he's used it for collecting analogue samples quickly and at consistent time points. However, the same basic approach should work with other DMA uses, so I set about trying to find out what I could do. The two particular use cases I had in mind were running NeoPixel LEDs and creating smooth animations on HUB75 LED panels. I suppose this time of year has got me thinking about flashing lights!

Fundamentally, a DMA transfer is a simple thing. You need to tell the system where you want the data to come from, where you want it to go, how much you want to go, and how fast you want it to go. That's it. It's a bit fiddly to get this data into the right place, so I've written a wrapper library to make it a bit simpler. You can see the code at hsmag.cc/MPDMA. At the time of writing, it's specifically for DMA-ing data to PIO state machine, not general DMA use, but it might expand over time (pull requests welcome!)

If you load the `rp2040_pio_dma.py` file onto your Pico (running MicroPython) and then run the →

Sound DMA

Sound effects can really lift a project, but can be awkward to implement as they often require a lot of processor attention. Any slight mistimes can have very audible results. It is, then, a perfect candidate for DMA. I've previously created the `PIOBeep` module for MicroPython that lets you play notes (hsmag.cc/PIOBeep). I've started upgrading it to work with DMA that lets you send buffers containing whole sounds that you want to play, and it will then play them without any processor intervention. This is great for things like game sounds where you just want to trigger a sound effect and let it play and not worry too much about it.

It's not perfect (there's no volume control for one thing), but it does work for simple sounds. It uses the DMA's pacing timer running at its slowest setting (about 2000 samples per second) to feed data into the PIO FIFO, so you just need to create an array with the pitches you want to play at each point. This can be the same note for larger blocks of sound, or it could change rapidly for more dynamic sound.

This works, but at the time of writing, I'm still working on the API to make it a bit more user-friendly. You can see the latest version at hsmag.cc/PIOMusicDMA.

actually want to make). If this all sounds a bit convoluted, think of the first DMA channel as the controller. It has an array of data made up of pairs of number of transfers and the read address for the second DMA channel. It can hold as many of these pairs as you like and will fire them one after the other.

There are four tricks we use to get this working:

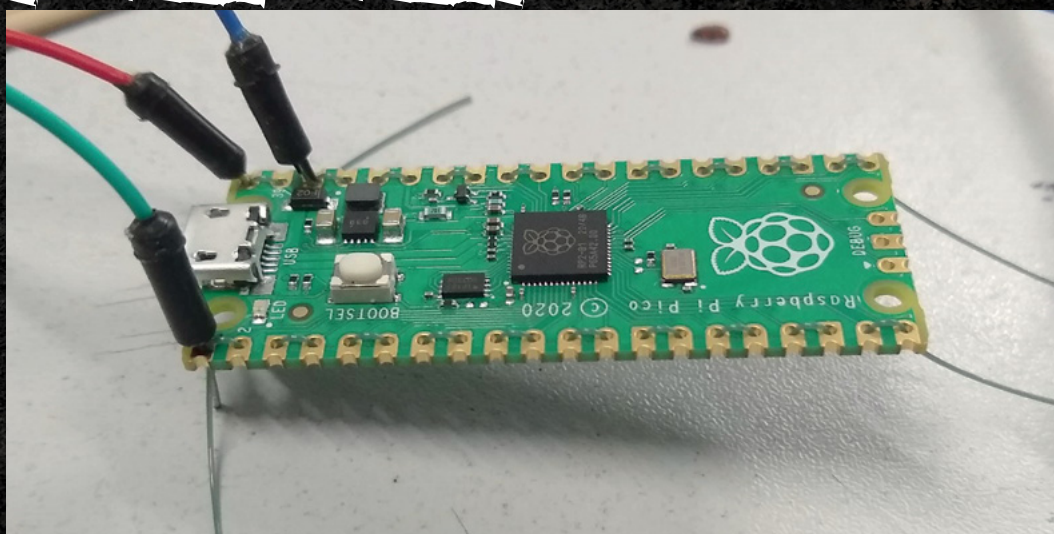
1. There are several 'alternative' register locations for the DMA data in different orders. One is the transfer count followed by the read address. This means that we can DMA the pair into consecutive memory locations.
2. The particular alternative register we're using automatically triggers the transfer after data is written (not all do).
3. The DMA will automatically increment the write address each time, but we can set it to loop back to the beginning after a certain number of bytes. This means that it always starts each pair at the correct location (known as a ring).
4. We can chain DMA requests. That means when one finishes, it automatically triggers another to start. In this case, the first DMA channel writes two blocks of four bytes, which triggers the second DMA channel. Once the second DMA channel has finished, it triggers the first again, which then writes the next to blocks of four bytes, etc.

This lets us run a finite series of DMA transfers one after the other. This can be useful, but what if we want to set something running and just let it go and go? This isn't completely possible, but there is a trick that makes it sort of work. If the control channel has a series of these DMA transfers programmed into it – enough to run for a second or two – we can start that and then at any point before it finishes, reset it. We have to run the reset periodically, but the exact timing isn't critical, so it's easy to keep it going and going.

The code is online and hasn't been fully tested yet, but it's some that we hope to use quite a bit in the future as it speeds up a lot of PIO work. If we properly design it, the complexity should be hidden away, so you may never know you're using it. It's still quite limited, so if you have a need and you're comfortable working with low-level code, hack it apart and see what you can get it to do. Raise an issue on the GitHub repo if you're having problems. ▣

Fishing-line

In non-DMA related news, I learned a tip from Stewart Russell (@scruss) on Twitter. He apparently learned it from Eugene 'thirtytwoeth' Andruszczenko. A bit of fishing-line through the holes on a PCB makes jumper cables stay in easily. You can poke the cables in a bit like you would on a breadboard, and they are held in place. This works really well on boards like Raspberry Pi Pico and Raspberry Pi Zero, which don't come with GPIO headers. Once you're finished, you can just take out the fishing-line, and you've got the same bare board you started with ready for soldering or anything else.





The
MagPi

HackSpace
TECHNOLOGY IN YOUR HANDS

CUSTOMPC

3 ISSUES FOR £10



FREE BOOK



hsmag.cc/hsbook

Subscribe to The MagPi, HackSpace magazine, or Custom PC. Your first three issues for £10, then our great value rolling subscription afterwards. Includes a free voucher for one of five fantastic books at store.rpiexpress.com/collections/latest-bookazines
UK only. Free delivery on everything.

FORGE

HACK | MAKE | BUILD | CREATE

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

PG
70

BACK-LIT PCB

Turn your PCBs into displays

PG
72

INTRUDER ALERT!

Keep your home secure

PG
76

PIPE CAD

Use FreeCAD to model
frames and pipes

PG
82



LED JEWELLERY

Create your own light-up bling

PG
62

SCHOOL OF MAKING

Start your journey to craftsmanship
with these essential skills

64 LED animations

66 Dust sensor

PG
88

POMODORO TIMER

Ditch the distractions



PG
94

UPGRADING A K40

Build your dream laser cutter

LED animations with CircuitPython

Keep your LEDs flashing using classes and inheritance



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.

Is there anything that says maker quite like blinking LEDs? But, what if you could make LEDs do more than blink? What if you want them to fade in and out, or gently blend between different colours? We're going to look at how to do this using CircuitPython and the LED_Animations library.

This should work on any CircuitPython board with either WS2812B (NeoPixel) or APA102 (DotStar) LEDs. However, we've tested it out with Raspberry Pi Pico and WS2812Bs.

Let's dive straight in and get your LEDs blinking. You'll first need to set up your device with CircuitPython, and load the `adafruit_led_animation` and `neopixel` libraries. You can get all of this from circuitpython.org.

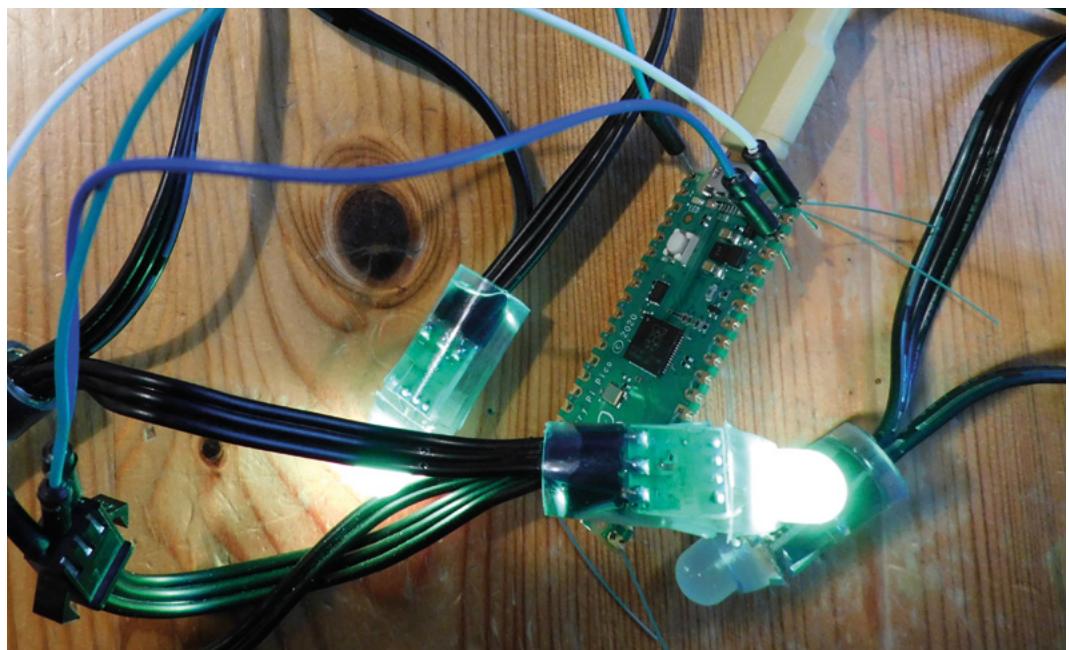
```
import board
import neopixel
from adafruit_led_animation.animation.blink import Blink
from adafruit_led_animation.color import RED

pixel_pin = board.GP0
pixel_num = 5

pixels = neopixel.NeoPixel(pixel_pin, pixel_num,
    brightness=0.5, auto_write=False)

blink = Blink(pixels, speed=0.5, color=RED)

while True:
    blink.animate()
```



Right WS2812B (and WS2811 controllers) come in many shapes and forms. This year, we're using a lot of 'fairly light'-style chains that are viewable from multiple angles, rather than the more common LED strips

This should blink your LEDs on and off, as you've probably seen many times before. The crucial part to notice is that we've created a **blink** object based on the **Blink** class, and we can then call the **animate** method on it. You can call this as often as you like, and it will keep blinking every 0.5 seconds (as set by the **speed**). You don't have to add any sleeps in there, and you can do other work in your main loop other than just handling the timings.

One useful feature of objects in languages like Python is inheritance. This means that one object can be based upon another object and 'inherit' its features. In this case, **Blink** inherits from the **Animation** class which has the **animation** method. This is useful because there are lots of animations that inherit from **Animation** and you can use them almost interchangeably. For example, if, rather than blinking our LEDs, we want to fade them through a range of colours, we can run:

```
import board
import neopixel
from adafruit_led_animation.animation.rainbow import Rainbow

pixel_pin = board.GP0
pixel_num = 5

pixels = neopixel.NeoPixel(pixel_pin, pixel_num,
    brightness=0.5, auto_write=False)

rainbow = Rainbow(pixels, speed=0.1)

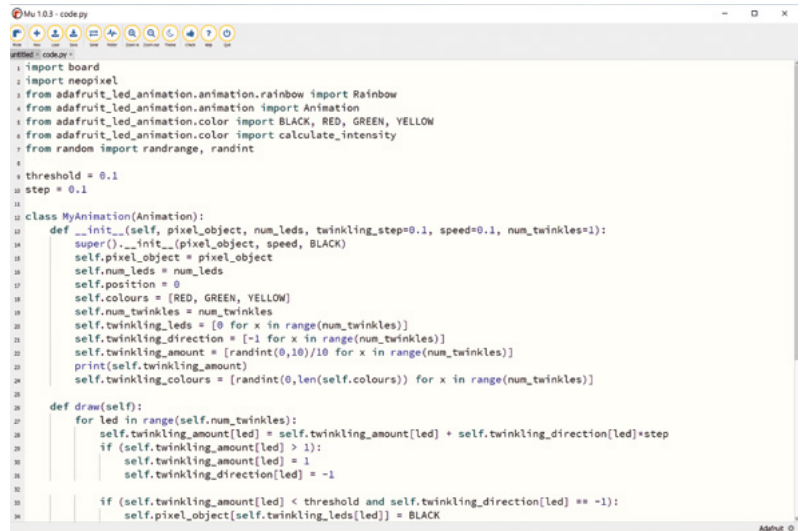
while True:
    rainbow.animate()
```

There are a whole range of animations pre-programmed that you can experiment with. Take a good look at hsmag.cc/LEDAnimation for the full list.

OUR OWN ANIMATION

Let's now take a look at how to make our own animation. You can see the full code at hsmag.cc/ledtwinkle, but here are the key parts. As we said before, we need to create a class that inherits from the **Animation** class.

```
class MyAnimation(Animation):
    def __init__(self, pixel_object, speed=0.1):
        super().__init__(pixel_object, speed,
            BLACK)
        self.pixel_object = pixel_object
        self.position = 0
```



```
self.colours = [RED, GREEN, YELLOW]

def draw(self):
    self.position = self.position + 1
    if self.position == 3:
        self.position = 0
        self.pixel_object.fill(self.colours[self.
position])

pixel_pin = board.GP0
pixel_num = 5

pixels = neopixel.NeoPixel(pixel_pin, pixel_num,
    brightness=0.5, auto_write=False)

animation = MyAnimation(pixels, speed=1)
```

Above
Our code running in the Mu text editor

Here, we've created a new class called **MyAnimation** that inherits from **Animation**. In the initialisation method, we have to call the superclasses **__init__** method to set up the things we are inheriting. In this case, that's the pixel object, speed, and colour (the colour in this case isn't used, but is needed for the superclass).

The other method we need is **draw()**. This is called from the **animate** method periodically (based on the **speed** setting). Note that we don't call the **show()** method of the LEDs as we typically would. The **animate** method takes care of this (and can do things like keep multiple animations in-step with each other).

That's the basics of how to make an animation, now let's make one that's a little more aesthetically pleasing. This one fades the LEDs in and out in a random pattern.

This, of course, is an animation we like. It's ripe for hacking, pulling apart, and changing to suit your tastes. □

Tracking workshop dust

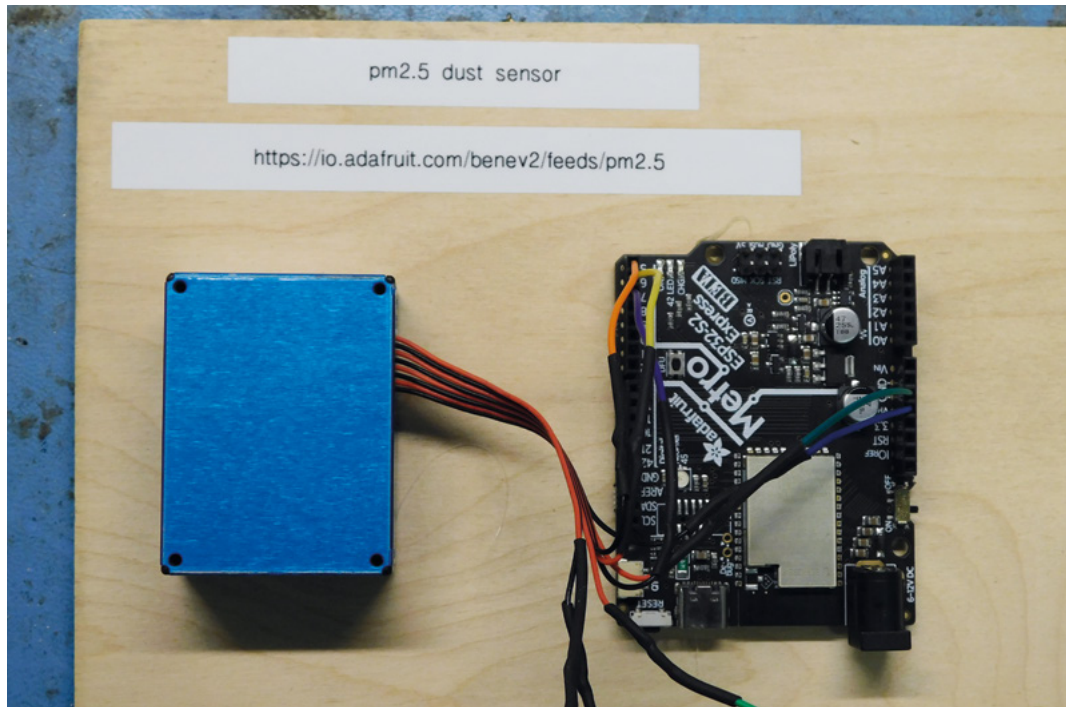
Make sure you're keeping your lungs healthy



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.



Dust is one of the more difficult workshop dangers. We don't just mean little flecks of dead skin that float around and coat old equipment, but tiny fragments of almost anything that float through the air and settle in our lungs. Some dust is well-known to be particularly dangerous – most famously asbestos – but almost all dust is dangerous to some extent. Small particles aren't visible to the naked eye, and if they make their way deep into the lungs, they can be difficult for our bodies to remove. Over years, exposure to even small levels of dust can combine to create some nasty health problems. Many of the tools that we use create these small particles, but some are worse than others. We do have some weapons in our arsenal to fight back

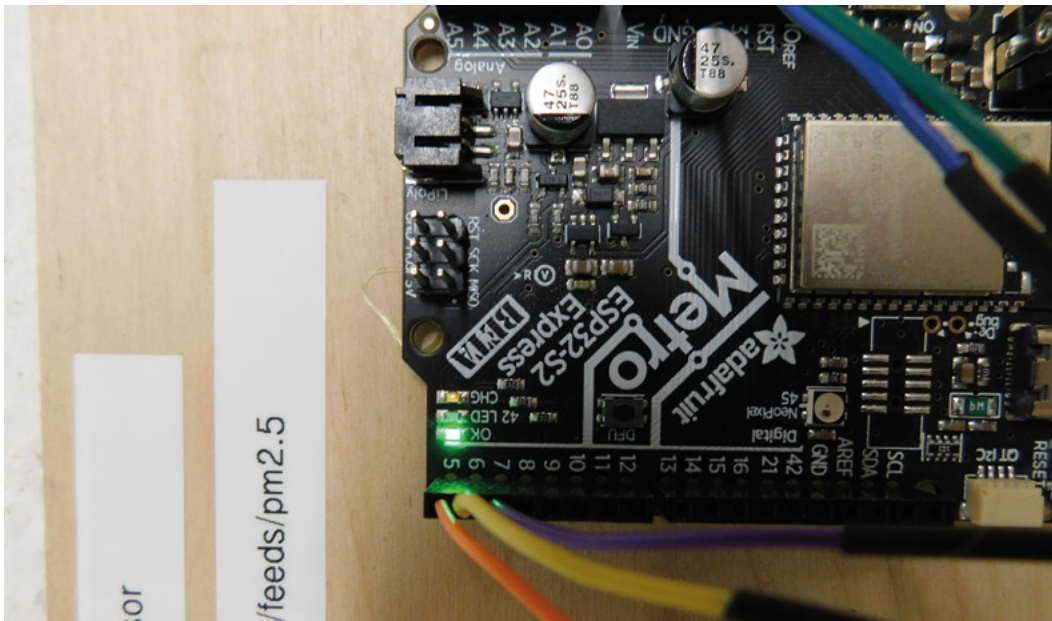
against dust – such as masks. To make sure we're aware of the risks, we built a workshop dust sensor.

Before we go on, we need to point out that this isn't a safety device. The parts we're using – while accurate – aren't medical quality. This should give us an idea of the risk, but shouldn't be relied upon. If you're considering something similar, it shouldn't be used as an excuse not to wear proper protective equipment, but just as an additional bit of information to help you understand your tools and workshop environment.

Dust is usually characterised by the size of the particulates it's made up of. For example, we'll be tracking PM2.5. PM stands for particulate matter and 2.5 refers to the size (in micrometres). These particulates are small enough to get into the lungs and even the bloodstream.

Above

We mounted the parts on some scrap wood and left space for an indicator, should we decide to add one in the future



Left ♦ Pushing wires into socket headers isn't the most secure way of mounting a project. If we find this useful, we'll add a proper shield to make it a bit more secure

We'll be using a PMS5003 sensor which is fairly common. It outputs data using the common serial protocol, so you should be able to get it to work with most microcontrollers, and we're using an Adafruit Metro ESP32-S2 Express, as this works with both WiFi and CircuitPython, making it a great choice for Internet of Things-type devices.

The dust sensor has an eight-wire ribbon cable output. You can get connectors that let you plug this straight into a microcontroller, but we just cut the ribbon cable and soldered on single-core wire which lets us poke the outputs into the socket headers on the Metro. We only actually need five of the eight connectors – power, ground, rx, tx, and reset.

These are (starting with the connection closest to the centre of the PMS5003):

- Power
- Ground
- Enable
- Receive
- Transmit

Power goes to 5V (the signal outputs are still safe for 3V devices at this power level). The other connections are:

- Enable goes to IO7
- Receive goes to IO6
- Transmit goes to IO5

That's all the wiring. Let's now take a look at the software.

There's a CircuitPython library for the PMS5003 sensor; however, we found that it didn't work with CircuitPython 7 on the ESP32-S2. We're not quite sure if this is due to the version of CircuitPython, or some

quirk of the board, but we forked it and got it working. You can get our version of the library using this link: hsmag.cc/PMS5003.

You'll need to copy the folder **pms5003** (which is inside the **library** folder) over to your CircuitPython device.

The **code.py** file (also in this repository) includes the code we've used. There's a bit going on there (which we'll get to shortly), but working with the sensor is very simple. You just have to enable it with:

```
pms5003 = PMS5003(serial=uart, pin_enable=board.I07)
```

You can then read the data with:

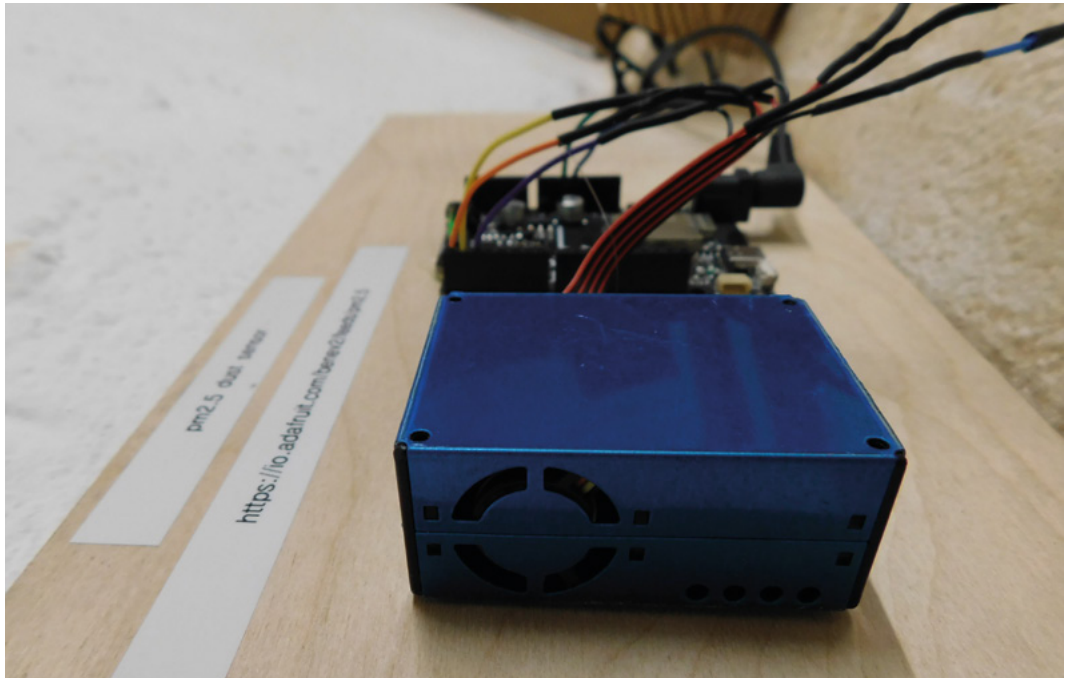
```
data = pms5003.read()
format(data.pm_ug_per_m3(2.5))
```

You can also get the PM10 and PM1 levels from the device, but for simplicity, we're only tracking PM2.5.

Once you've got the data, what do you do with it? This is the question we're always faced with when building sensors. Do you want to make a warning buzzer that goes off when you hit a certain level? An indicator that lets you know the level? Since the danger from PM2.5 is cumulative, we decided to keep a →

NO DATA?

Due to the current wave of Covid-19 sweeping the UK at the time of writing, we're not measuring the dust levels in the workshop (as we're working mostly from home). If you take a look at the URL for data now, you might find that there's nothing there. Once things settle down, we'll plug our sensor back in again.



record of the dust levels by uploading them to Adafruit IO. This is an online service that lets you track data online. We'll be using the free tier, but you can get more storage and data held for a longer time for a fee by signing up for Plus membership.

Before using it, you'll need to sign up for an account and get an API key. Take a look at io.adafruit.com for details of both of these.

Adafruit IO is based around feeds. Each feed basically lets you save how the value of a thing changes over time. We've created a feed called PM25 and we can write data to that as often (well, up to 30 times a minute), or as infrequently as you like. The old values are stored for a month. If you want to, you can combine multiple feeds on a dashboard to display data from many different sensors at once. For our purposes, though, we'll just use a single feed, so we don't need a dashboard.

You can contact Adafruit IO via the MQTT protocol, so we'll use the `adafruit_minimqtt` module

to interface with it. There's a bit of code needed to handle the WiFi and MQTT connections (this is mostly stolen from the example code in the following repository: hsmag.cc/AdafruitIO).

```
import board
import busio
import digitalio
from pms5003 import PMS5003
import time
import ssl
import socketpool
import wifi
import adafruit_minimqtt.adafruit_minimqtt as MQTT

try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py,
    please add them there!")
    raise

aio_username = secrets["aio_username"]
aio_key = secrets["aio_key"]

print("Connecting to %s" % secrets["ssid"])
wifi.radio.connect(secrets["ssid"],
secrets["password"])
print("Connected to %s!" % secrets["ssid"])

pm25_feed = secrets["aio_username"] + "/feeds/pm25"

def connected(client, userdata, flags, rc):
```

OUTSIDE AIR

We're using our sensor to monitor the air inside a workshop where the dust is mostly made by people using tools. However, PM2.5 is also a problem in outside air as it's emitted by (among other things) vehicles and wood-burning stoves. If you want to track PM2.5 in the general atmosphere, the Sensor.Community project is a great place to start. This is a global network of makers who run their own dust sensors (using a slightly different model of dust sensor), and upload the results to the website.

Above The PMS5003 sucks in new air through a fan to keep the sensor fed with the current workshop air

AIR QUALITY

PM2.5 isn't the only thing that you have to be concerned about in the air. There's also nitrous oxide, volatile compounds, and others that can cause problems in an enclosed space. You can get sensors for these, but the accuracy of affordable sensors isn't always great. The best defence against these is an understanding of the tools and substances you're using. If you understand what's likely to emit airborne pollutants, you know what you need to do to prevent you breathing them in.

```

pass

def disconnected(client, userdata, rc):
    print("Disconnected from Adafruit IO!")

def message(client, topic, message):

    print("New message on topic {0}: {1}".
format(topic, message))

pool = socketpool.SocketPool(wifi.radio)

mqtt_client = MQTT.MQTT(
    broker=secrets["broker"],
    port=secrets["port"],
    username=secrets["aio_username"],
    password=secrets["aio_key"],
    socket_pool=pool,
    ssl_context=ssl.create_default_context(),
)

mqtt_client.on_connect = connected
mqtt_client.on_disconnect = disconnected
mqtt_client.on_message = message

print("Connecting to Adafruit IO...")
mqtt_client.connect()

pm25_val = 0

uart = board.UART()

pms5003 = PMS5003(serial=uart, pin_enable=board.
IO7)

while True:
    data = pms5003.read()
    print("PM2.5 ug/m3 : {}".format(data.pm_ug_per_
m3(2.5)))
    print("PM1 ug/m3 : {}".format(data.pm_ug_per_

```

```

m3(1)))
    print("PM10 ug/m3 : {}".format(data.pm_ug_per_
m3(10)))
    mqtt_client.loop()

    pm25_val = data.pm_ug_per_m3(2.5)

    print("Sending pm25 value: %d.." % pm25_val)
    mqtt_client.publish(pm25_feed, pm25_val)
    print("Sent!")
    time.sleep(60)

```

We've left in a few `print` statements to show you how to access other bits of data if you want.

The final thing you'll need is a secrets file. This is a file separate to your code that just contains things you don't want to make public – in other words, secrets. Things like passwords and API keys belong in here. This way, you commit your main code file to a source code repository such as GitHub and not reveal anything you shouldn't reveal. It should look something like this (only with your values in place of the examples):

```

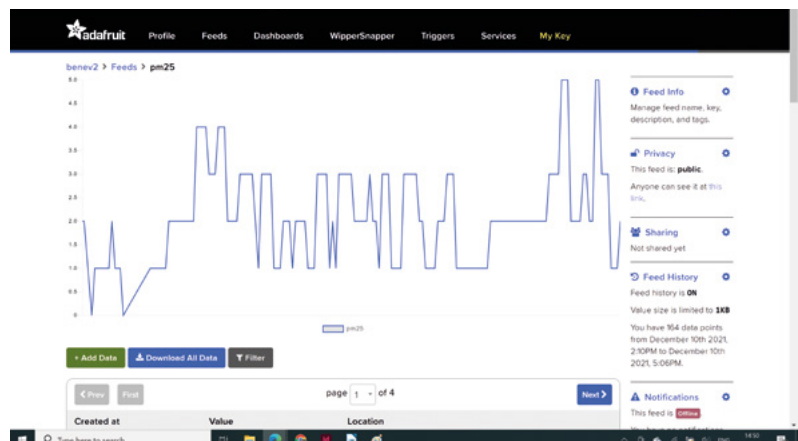
secrets = {
    'ssid' : '_your_wifi_ssid',
    'password' : '_your_wifi_password',
    'aio_username' : '_your_adafruit_io_username',
    'aio_key' : '_your_big_huge_super_long_aio_
key_'
}

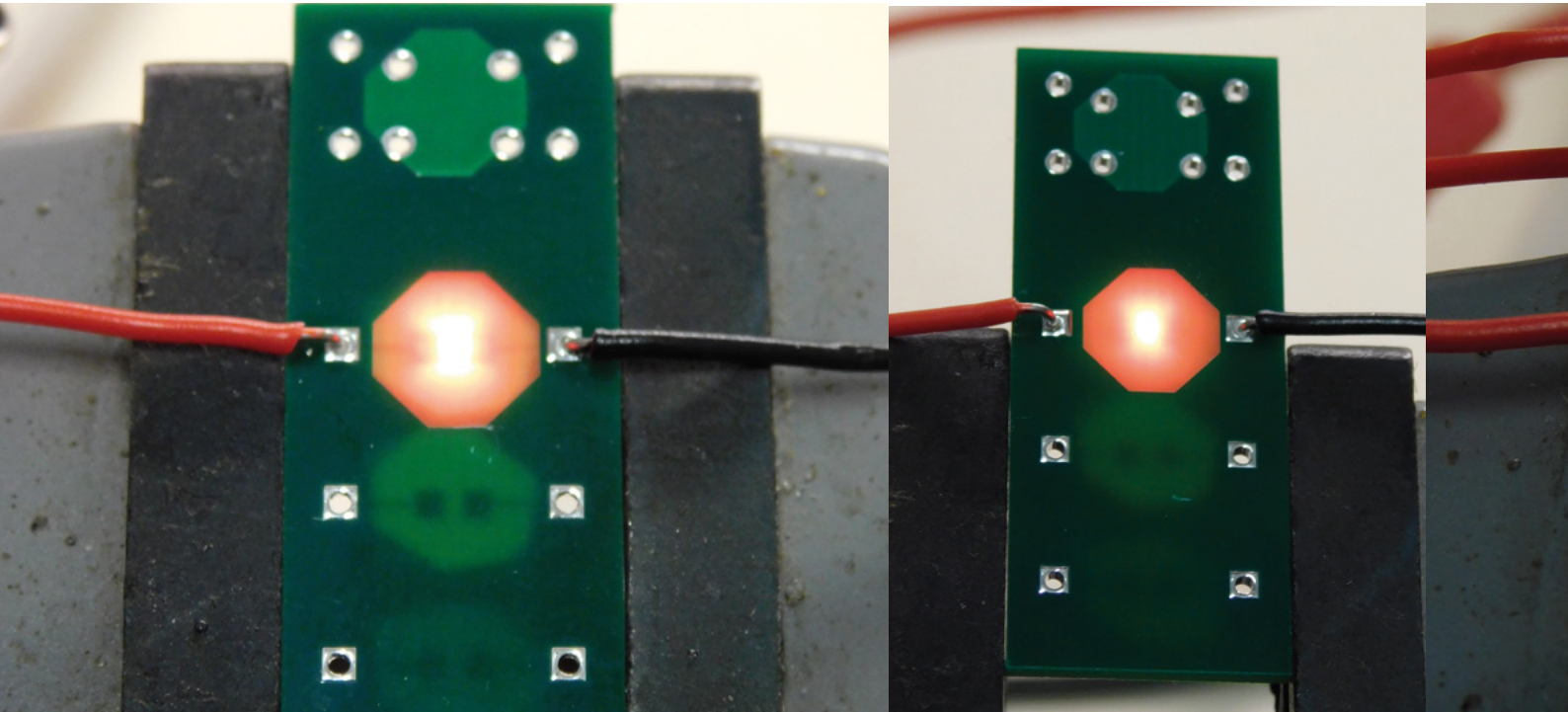
```

It needs to be called `secrets.py` and stored on your CircuitPython device. Obviously, you should never reveal this to anyone.

That's all there is to it. With all that in place, you should be able to power on your sensor and it will keep a record of the PM2.5 levels in your workshop, neighbourhood, or wherever else you want to monitor. ■

Below ◆
The Adafruit IO online service stores data for 30 days and allows you to track changes over that time





Make your circuits glow

Experimenting with back-lit PCBs



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.

PCBs are made of layers of material. The middle bit (in standard two-layer PCBs) is a type of fibreglass called FR-4. This is translucent and will let some light through. It's sandwiched between layers of copper and two different types of paint (solder mask and silkscreen). These are less translucent. With a bit of clever design, you can utilise these material properties to create different effects as you shine light through a PCB.

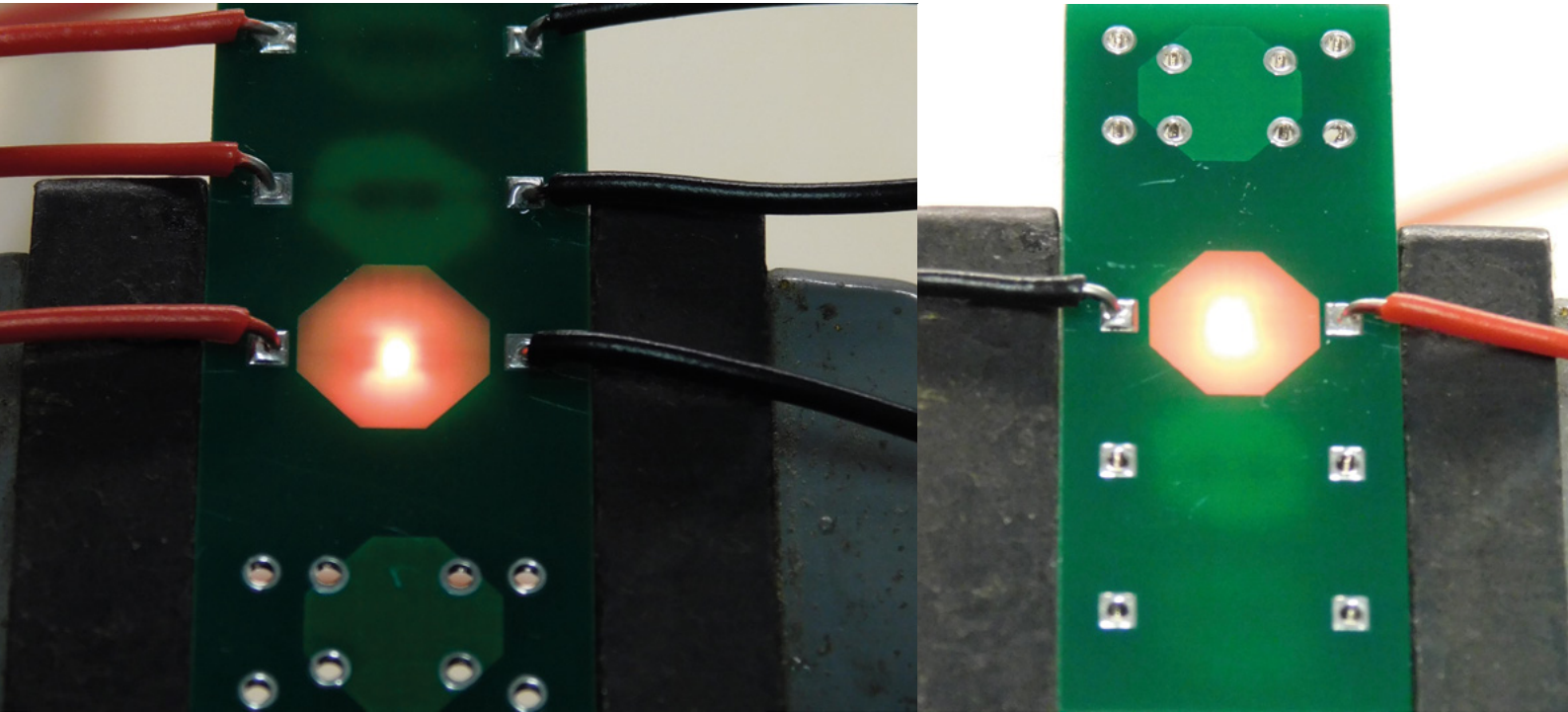
How well does it work? We decided to test it out with a simple experiment, putting different mounting options together with different PCB thicknesses to see what effect we got. We tested no solder mask, solder mask on one side, and solder mask on two sides to see how it looked.

Surface-mount LEDs are designed to shine up from the PCB, not down into it. There are reverse-mount LEDs available that are the other way around, but typically these are designed to fit in a hole cut into the PCB rather than shine through it. Besides, the reverse-mount LEDs we found had a huge lead

time, so weren't practical for our tests. Instead, we simply soldered a regular LED upside down. These LEDs aren't really designed for this, as they only have an electrical contact on the bottom (or what should be the bottom), but with a sizeable blob of solder, we were able to bridge the gap between PCB and contact without any problem.

Firstly, the easy part of the results. Don't bother with solder mask. It looked awful. It's not perfectly even, so had blotches and blocked out too much of the light. Part of the problem could be that we were testing red LEDs and green solder mask, but having played around with this, we don't think it's really going to work. Maybe it's possible, if you wanted to shade part of the light but not totally, but it would be really hard to get the balance right.

The thickness of the PCB is a more subtle point. Obviously, the thinner the PCB, the more light that shines through. The thinnest we tested was 0.6 mm, and this let a good amount of light through. However, you obviously need some traces and pads on the PCB to power the LEDs (we used 0805 parts – larger and smaller LEDs are available). With the



really thin PCBs, the pads were really noticeable, as was the pattern of light out of the PCB (these LEDs don't shine evenly through 360 degrees). As the PCBs got thicker, less light came through, but it was in a more even pattern. With the thickest PCBs on test (1.6mm), we got a far more even glow. The effect is far more pronounced in real life than it is in the photographs.

Another consideration is that, as the PCBs got thicker, the rectangular grid of the fibreglass became

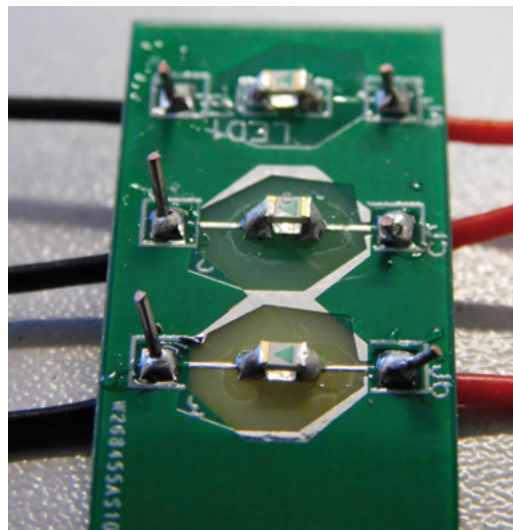
even glow, but there's definitely more to test there. Secondly, we know that some people use edge-lit LEDs. These are designed to be placed on the edge of PCBs and shine outwards. While they don't direct all of their light down, some people claim that enough is shone down to make them work. Thirdly, we'd also like to test RGB LEDs. This poses some mounting problems. The eagle-eyed among you will notice a mounting for an RGB LED on the end of our test PCBs. This particular design didn't work, so we'll try again and see what we can get to work. We'll update you on our experiments in future issues. □

With a sizeable blob of solder, we were able to bridge the gap between PCB and contact without any problem

more evident. Thicker PCBs are available – and indeed quite common – but honestly, we were expecting the thinner PCBs to be better, so didn't test all the way up. To avoid sending too much to landfill, we ended the test at 1.6mm.

NEXT TESTS

This doesn't end here. There's much more we want to test on shining light through PCBs. Firstly, we'd like to test thicker PCBs. Looking at the results we've had, we suspect that 1.6mm is about the sweet spot for letting light through and getting an



Above □ As the PCB got thicker, the light got dimmer, but more even. The effect is more pronounced than shown in these images

Left ◆ We had no problem mounting these LEDs upside down, but obviously it won't work if you're reflowing the PCBs, or using a pick-and-place

Explore the sensory world: Make an intruder alarm



Phil King

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

@philkingeditor

Catch intruders red-handed with this sound- and laser-triggered alarm

In this series, we are exploring some of the most commonly available sensors and their use cases. Following last month's fire and gas safety alarm, this time we'll use sensors to build a different kind of alarm: one to detect intruders.

While there are many possible ways to sense the presence of a person nearby, including a PIR motion sensor or camera, we'll be using a laser beam and sound sensor. When either is triggered, a visual and audible alert will go off.

For this tutorial, we're using the laser sensor from the Waveshare Sensors Pack, available in the UK from The Pi Hut (magpi.cc/wavesensors), and also sold separately, but any similar sensor should work in a similar way.

It continually emits a laser beam, and its receiver only detects a reflected beam of the exact same wavelength (650 nm), so it won't be triggered by other visible light. When it detects the beam, its digital pin outputs 1; when the beam is broken, it's 0.

With the power turned off, connect the laser sensor to Raspberry Pi as in **Figure 1**. We're powering it from Raspberry Pi's 3V3 pin, grounding it with a GND pin (both via the breadboard side rails), and the digital output (marked DOUT on the sensor) is going to GPIO 21.

01 Connect laser sensor

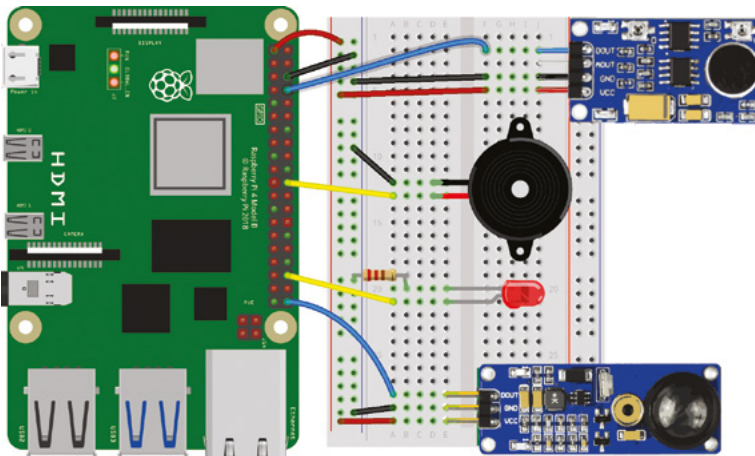
Think of all those movies where a secret agent or thief has to get past some lasers guarding an object: break the beam and the alarm will go off. That's what we'll be doing here.

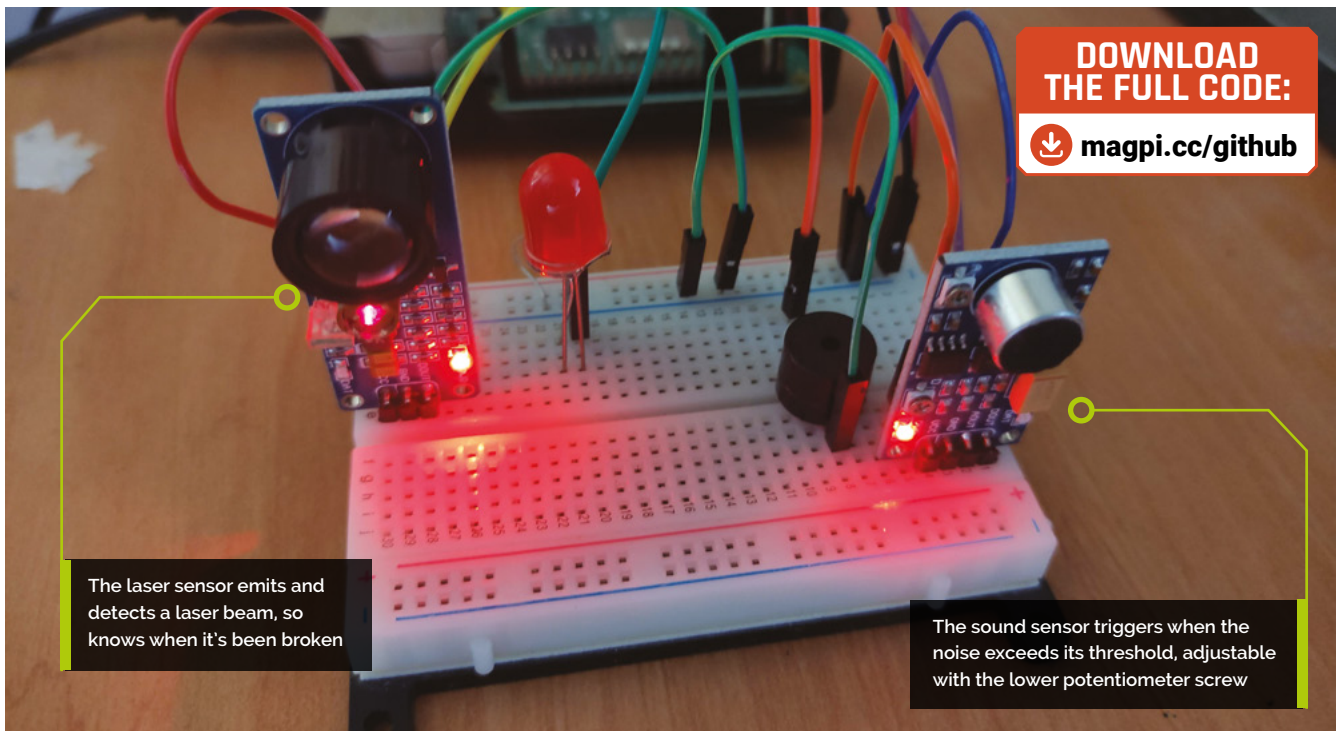
02 Laser positioning

With the laser sensor wired up, turn on Raspberry Pi. You should see the sensor's red power LED (on the right) light up if it's connected correctly. It should also be emitting a laser beam from the metal tube, so be careful never to look straight into it.

Aim the beam at a nearby wall (up to 1.5 m away) and check that its left LED (marked DAT) is lit, confirming that it is detecting the laser beam. You may need to adjust the vertical and horizontal tilt of the sensor, or move it closer to the wall. For the finished alarm, we recommend you place the laser sensor fairly near the floor so that anyone walking through it will break the beam and it won't be anywhere near their eyes.

Figure 1 The wiring diagram for the complete alarm system, including laser and sound sensors, LED, and buzzer





DOWNLOAD
THE FULL CODE:

magpi.cc/github

The laser sensor emits and detects a laser beam, so knows when it's been broken

The sound sensor triggers when the noise exceeds its threshold, adjustable with the lower potentiometer screw

03 Laser test

To begin, we'll create a simple Python program, as in the `laser_test.py` listing, to read the sensor's digital output and print out a message to show when the beam is broken. From the desktop menu, go to Programming and open the Thonny IDE to start coding.

Try breaking the beam with your hand and see if the message changes

As before, we're using the GPIO Zero library; at the top of the code, we import the Button method from it. We'll use this to sense when the digital output from the sensor is high, in effect the equivalent of a push-button being pressed. As it's connected to GPIO 21, we assign the laser object to this with `laser = Button(21)`.

In an infinite `while True:` loop, we check whether the pin is low (`if laser.value == 0`), which means the beam has been broken, and set the message (`msg1` variable) that we'll be printing to the Shell area accordingly. In our print statement, we add the `end = "\r"` parameter so the message is always printed on the same line.

Run the `laser_test.py` code and then try breaking the beam with your hand and see if the message changes to 'Intruder!'. You may find that it works better with your hand more distant from

the sensor. Even if the DAT LED only flickers off momentarily, that should be enough to trigger our alarm later.

04 Add a sound sensor

Now that we have our laser sensor working, let's make our setup even more intruder-proof by adding a sound sensor. We're using a Waveshare sound sensor for this, as featured in the Sensors Pack, but other similar sensors are available, along with USB mics.

Our sensor has pins for analogue and digital outputs, but we only need the digital output for our alarm. With the power turned off, we connect that pin (DOUT) to GPIO 14, and the VCC and GNC pins to 3V3 and GND (shared with the laser sensor via the breadboard side rails), as in **Figure 1**.

Turning Raspberry Pi back on, you'll see the power LED on the left of the sound sensor is lit up. Make a loud noise and you should see the LED on the right light up to show it has been detected.

05 Sound test

Let's create a similar program to test the sensor. In the `sound_test.py` code listing, we assign the `sound` object to GPIO14 with `sound = Button(14)`. Again, we use the Button method to detect when the pin is triggered. →

You'll Need

- Laser sensor magpi.cc/lasersensor
- Sound sensor magpi.cc/soundsensor
- LED
- Active piezo buzzer
- Jumper wires



Danger!
Laser

The laser sensor used here continually emits a laser beam. Be very careful not to point it towards anyone's head as it could potentially damage their eyesight.

magpi.cc/lasersafety

THE MAGPI



This tutorial is from in The MagPi, the official Raspberry Pi magazine. Each issue includes a huge variety of projects, tutorials, tips and tricks to help you get the most out of your Raspberry Pi. Find out more at magpi.cc

This time in our `while True:` loop, we test whether the pin is high (there is a loud enough noise to trigger the sound sensor). As before, this determines which message (in the `msg1` variable) is printed to the Shell area.

06 Make a noise

Now it's time to test our sound sensor to check it's wired up and working correctly. Run the `sound_test.py` Python code and then made a loud noise to make the DAT LED on the right of the sensor light up. You may find that you need to be noisy for a second or so and that there's a short delay before the message changes briefly from 'All clear' to 'Intruder!'.

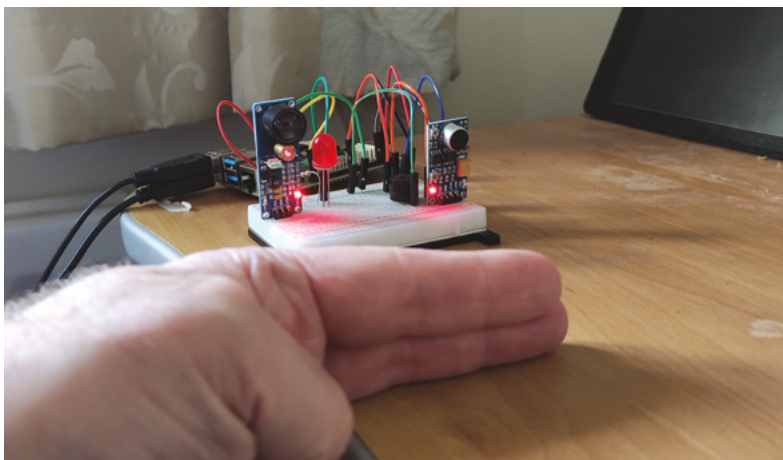
If you're having trouble triggering it, try altering the sensitivity of the sound sensor by adjusting the lower potentiometer screw (marked D for digital) on it: turning it anticlockwise increases the sensitivity, but don't overdo it or the DAT LED will be lit up constantly.

07 Add a visual alert

If your sensors and code are working correctly, it's time to move on to the next part. Printed messages are all very well, but for a proper alarm you need a visual and/or audible alert.

As in last month's guide, we'll add a standard red LED for a visual alert. Ours is 5 mm, but you can use a different size. As always, a resistor is needed to limit the current to the LED to ensure it doesn't receive too much and potentially burn out. With the LED placed in the breadboard, with

▼ Testing the laser sensor by breaking the beam. Be careful not to point the laser near anyone's head



sound_test.py

> Language: Python

```
001. from gpiozero import Button
002.
003. sound = Button(14)
004. msg = ""
005.
006. while True:
007.     if sound.value == 1:
008.         msg = "Intruder!"
009.     else:
010.         msg = "All clear"
011.     print(msg, end = "\r")
```

legs in different unconnected rows, we connect a 330 Ω resistor between the negative (shorter) leg and the ground rail of the breadboard. The positive (bent, longer) leg is connected to GPIO 16 on Raspberry Pi, as in the **Figure 1** wiring diagram.

08 Sound the alarm

For our audible alert, we'll use a small active piezo buzzer to make a beeping noise. You could use something else to sound the alarm.

“ This time we've kept it simple with a single alarm function ”

The buzzer has a longer positive leg and a shorter negative one; their positions may also be marked on its top. Connect the negative pin to the breadboard's ground rail and the positive pin to GPIO 25 (as in **Figure 1**).

09 Alarm code

With everything wired up as in **Figure 1**, you're now ready to program your intruder alarm. In the final code, `intruder_alarm.py`, we add LED and Buzzer to the `gpiozero` imports at the top. We also import `sleep` from the `time` library, to use as a delay.

If you wanted, you could create a separate function with a different message for each alarm (like our fire and gas alarm last issue), but this time we've kept it simple with a single alarm function,

laser_test.py

> Language: Python

```
001. from gpiozero import Button
002.
003. laser = Button(21)
004. msg = ""
005.
006. while True:
007.     if laser.value == 0:
008.         msg = "Intruder!"
009.     else:
010.         msg = "All clear"
011.     print(msg, end = "\r")
```

as we're not bothered how an intruder is detected. When triggered, this executes a **for** loop which toggles the LED and buzzer on and off a set number of times, with a 0.5 sleep delay each time.

In a **while True:** loop, we check the pin values from both sensors and trigger the alarm when the laser beam is broken (**laser.value == 0**) or the sound threshold is exceeded (**sound.value == 1**). If neither is triggered, we show the default message and ensure the LED and buzzer are turned off.

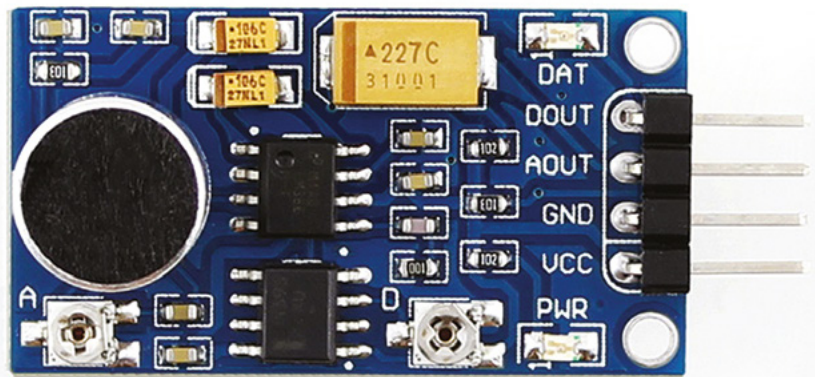
10 Test the alarm

Now to test the alarm system. As before, try breaking the laser beam: the LED should then blink and the buzzer will beep. Do the same for the sound sensor by making a prolonged loud noise; the alarm will trigger again. Each time, the 'Intruder!' message will show in the Shell area.

Taking it further

We now have a simple intruder alarm. To improve it, you could add extra sensors such as a PIR or even a camera to detect movement. You could trigger a larger light and/or play an alert sound or spoken message on a connected speaker. You could also send an email or push notification alert to your phone whenever the alarm is triggered.

Next time we'll create a weather station using temperature, humidity, and ultraviolet light sensors. See you then. ◻



intruder_alarm.py

> Language: Python

```
001. from gpiozero import Button, LED, Buzzer
002. from time import sleep
003.
004. laser = Button(21)
005. sound = Button(14)
006. led = LED(16)
007. buzzer = Buzzer(25)
008.
009. def alarm():
010.     print("Intruder alert!", end = "\r")
011.     for i in range(10):
012.         led.toggle()
013.         buzzer.toggle()
014.         sleep(0.5)
015.
016. while True:
017.     if laser.value == 0 or
018.        sound.value == 1:
019.         alarm()
020.     else:
021.         print("All clear", end = "\r")
022.         led.off()
023.         buzzer.off()
```

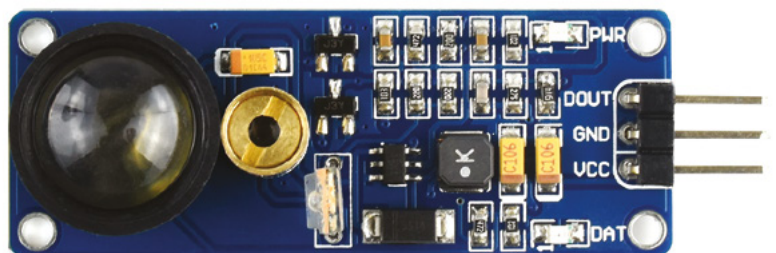
▲ The Waveshare sound sensor. Its sensitivity can be altered by adjusting the A (analogue) or D (digital) potentiometer screw

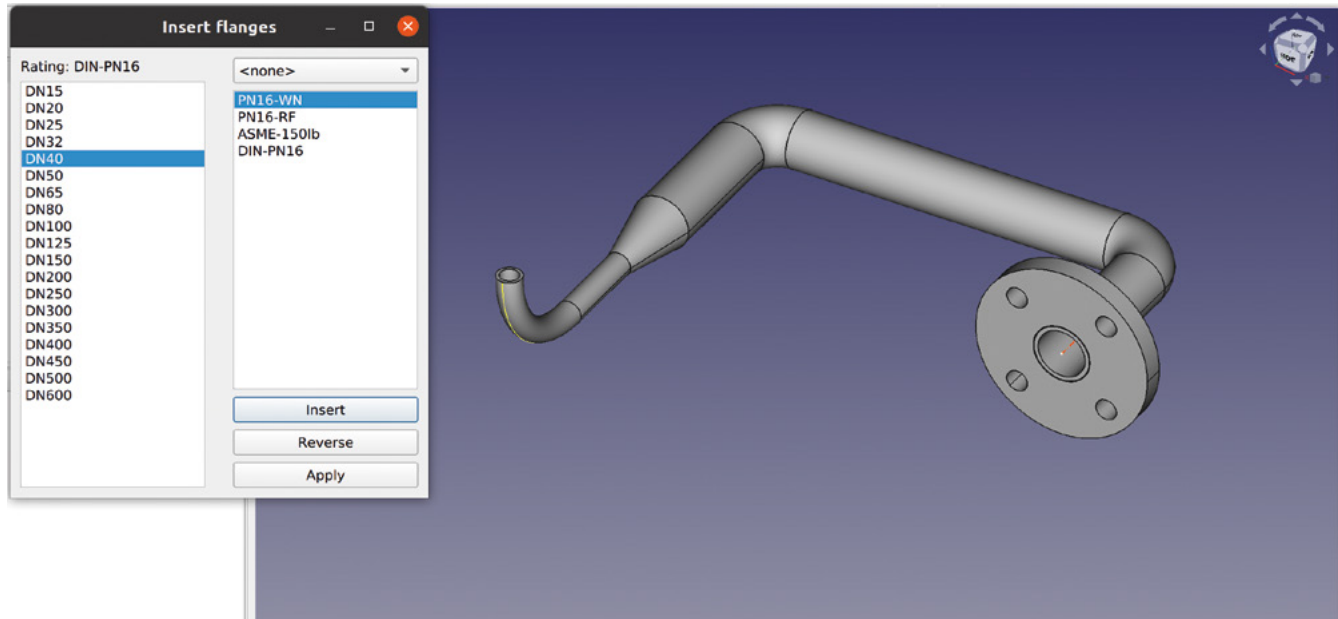
Top Tip

Analogue out

For simplicity, we've used the digital output of the sound sensor to trigger our alarm. To use the analogue output, you'll need to add an ADC chip (e.g. MCP3008) to convert its signal to a digital reading. We'll cover how to use an ADC in the next instalment.

▼ The Waveshare laser sensor. The lens for the receiver is on the left, with the metal laser tube just to the right of it





FreeCAD: frames and pipes

Let's look at a new workbench: the 'Dodo' workbench allows us to create complex frames and pipework from a collection of different profiles

Figure 1 ♦ An example of creating pipework with the Dodo workbench



Jo Hinchliffe

@concreted0g

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

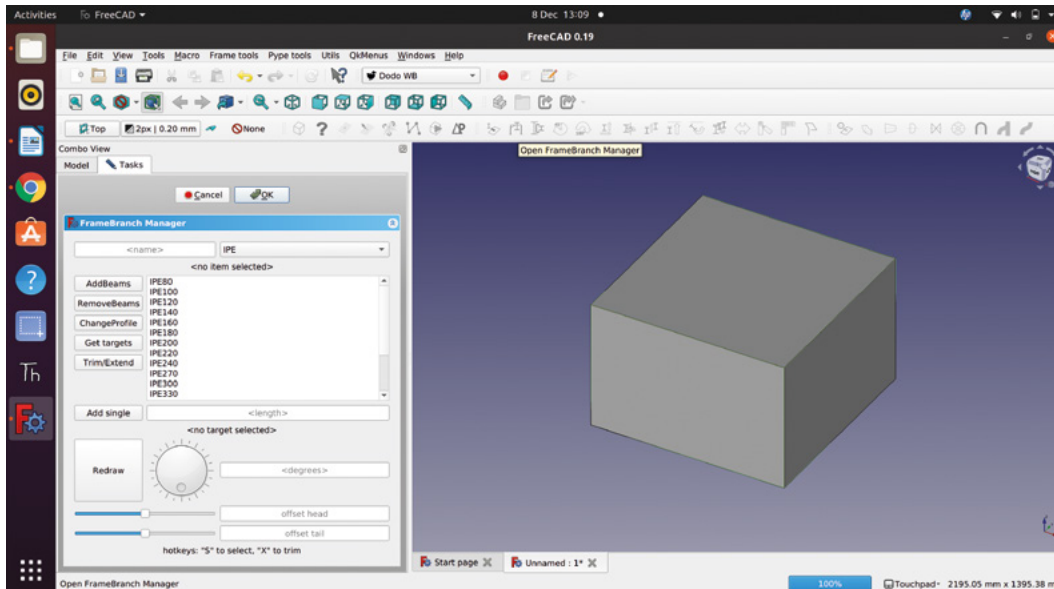
Last month we looked at the Sheet Metal workbench, and there's no denying that, while you could design sheet metal objects without using that workbench, it makes a sheet metal design much quicker to generate and work with. This month we are looking at the 'Dodo' workbench, which treats working with framed objects and designs or pipework in a similar fashion.

To begin, open FreeCAD and click the Tools > Addon manager, then select Dodo on the list. Click install. When installed, you will be prompted to restart FreeCAD. Restart FreeCAD and then create a new project. Let's look at making a framed object first. To begin, move to the Part workbench and use the 'Create a Cube Solid' tool to create a cube that is 750 x 750 x 500 mm.

Next, let's move to the Dodo workbench, which is listed as 'Dodo WB' in the workbench drop-down

menu. Click the 'Open FrameBranch Manager' tool, and you should see a dialog box similar to the one shown in **Figure 2**. To create a frame, we need to select a profile for the frame extrusions. There's a collection of preset profiles included that model real-world standard metal extrusions. Let's select a simple 40mm x 40mm square extrusion with a 5mm wall thickness. Click the drop-down menu that currently reads 'IPE' and select 'RH', and then select the 'RH40x40x5' by highlighting it in the list below. Next, click on any edge of the cube in the preview window and then click 'OK' in the Open FrameBranch Manager dialog box. You should now see that every edge of the cube object has had the 40mm x 40mm extrusion added to create a frame. You can, of course, in the file tree view, make the cube object invisible to get a better view of our frame (**Figure 3**).

You'll notice that our frame has all the required sections but that they all overlap at the corners. We



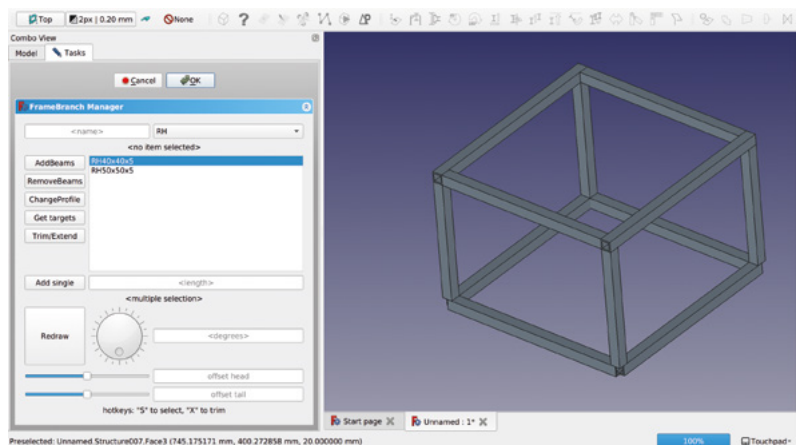
YOU'LL NEED
 A computer with FreeCAD 0.19

Figure 2 The Open FrameBranch Manager dialog, open and ready to create a frame on our cube object

need to extend or trim these frame parts to make a frame that could be fabricated in real life. With the Open FrameBranch Manager still open, we can use the 'Get Targets' and 'Trim/Extend' buttons to help us do this quickly. First, select an inside face of one of the frame sections and then, holding down the **CONTROL** key, select the opposite inside face – with both of those selected, click the Get Targets button. We have now defined the reference as to where we want to trim or extend other frame structure objects. Next, select any face of a frame piece that crosses the target faces that we want to trim to between the target faces. Click the Trim/Extend button and you should see that our cross member now is shortened to directly fit inside the other pieces (**Figure 3**). We don't need to select new targets to also trim other frame pieces to the same length; even the piece below the trimmed piece can be trimmed. We can select faces on the three other pieces and click the Trim/Extend button to bring them all to length. You can also, when required, select the outside edge faces of objects as targets, and the Trim/Extend button will extend pieces to the correct length.

It's worth noting that although we have used a solid object as the basis of our frame, we can apply frame structures directly to sketches. As a quick example, create a new project and move to the Sketcher workbench and create a sketch of a rectangle with an extra line spanning across the rectangle inside – our rectangle was 100 cm x 50 cm.

// We need to extend or trim these frame parts to make a frame that could be fabricated in real life //



Moving to the Dodo workbench again, click to launch the Open FrameBranch Manager. Let's select a different profile – we went with 'IPE80', which is a common steel H-section used in buildings and structure fabrication. Again, click any edge of the sketch in the preview window and then click 'OK'. You should now see a frame appear made from the H-section. You can, of

course, use any face of any of the IPE80 parts to set targets again to trim or extend other sections to. Another useful tool in the Open FrameBranch Manager dialog is the Rotation tool. Select any face of a section of frame and experiment by either moving the dial, or by typing an angle into the input box. →

Figure 3 Our cube has had frame elements added, and the upper section of the cube frame has been trimmed and extended to make the frame members sit correctly

TUTORIAL

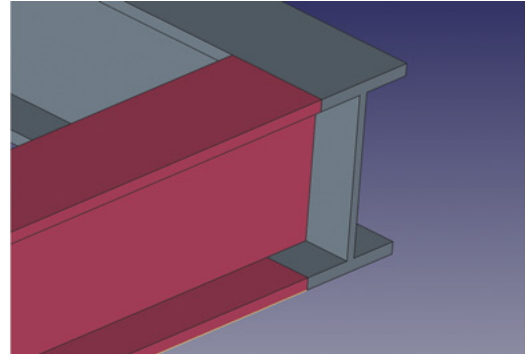
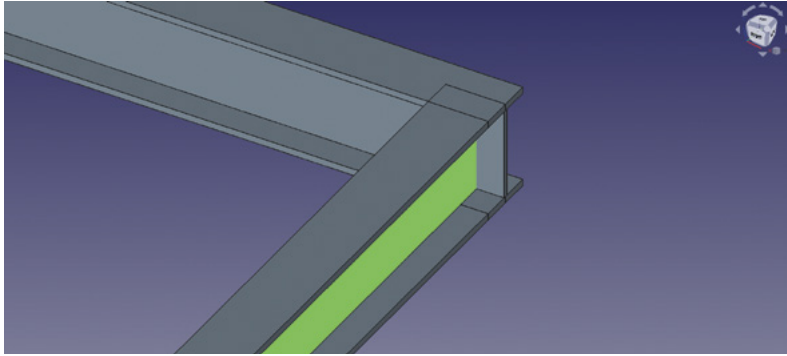
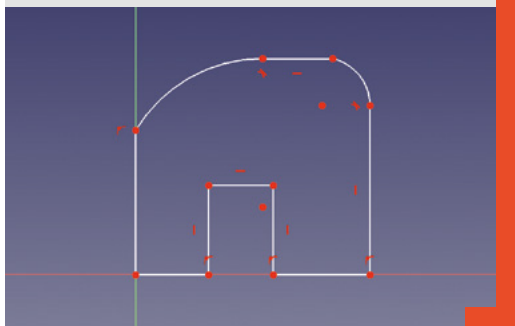


Figure 4 ◆
When H-frame sections overlap, they often create joints that aren't correct for the real world

Figure 5 ■
Applying a workaround solution with the Part workbench 'Slice Apart' tool gives us a more correct H-section interface

EXTRUDE A SKETCH

You might have noticed that within the Open FrameBranch Manager dialog, there is the option to select '<by sketch>' in the drop-down menu where you select the different profile families. This is a really useful feature that allows us to draw a custom profile for extrusion into frame elements as a sketch. However, there is a particular rule that we need to follow when drawing a sketch for it to work correctly. The rule is that you need to draw your profile with the Polyline tool, and you need to draw the polyline in an anticlockwise orientation. In a new project, create a sketch on the XY plane and draw a simple profile shape using the Polyline tool. Remember, you can change the behaviour of the Polyline tool using the **M** key on your keyboard, and it's possible to create profiles with arcs and tangent lines as long as you create the profile moving in an anticlockwise direction in the sketch. For good practice, you should fully constrain the sketch; however, for a simple test sketch profile, it isn't necessary. Once you have a profile sketch created, you can test it by first moving to the Part workbench and creating a large cube to use as a target for a frame and then moving to the Dodo workbench to create the frame. When selecting the profile for the frame, click the drop-down menu and select '<by sketch>'. You should see your sketch 'sketch' listed in the dialog box. Select that, and then select an edge of the cube, and click OK as we have done before. You should see a frame created using your sketch profile. If you've accidentally created a sketch in which the polyline isn't moving in an anticlockwise direction, you will probably still see some frame elements, but they won't follow the edges of the cube and will extrude in random directions.

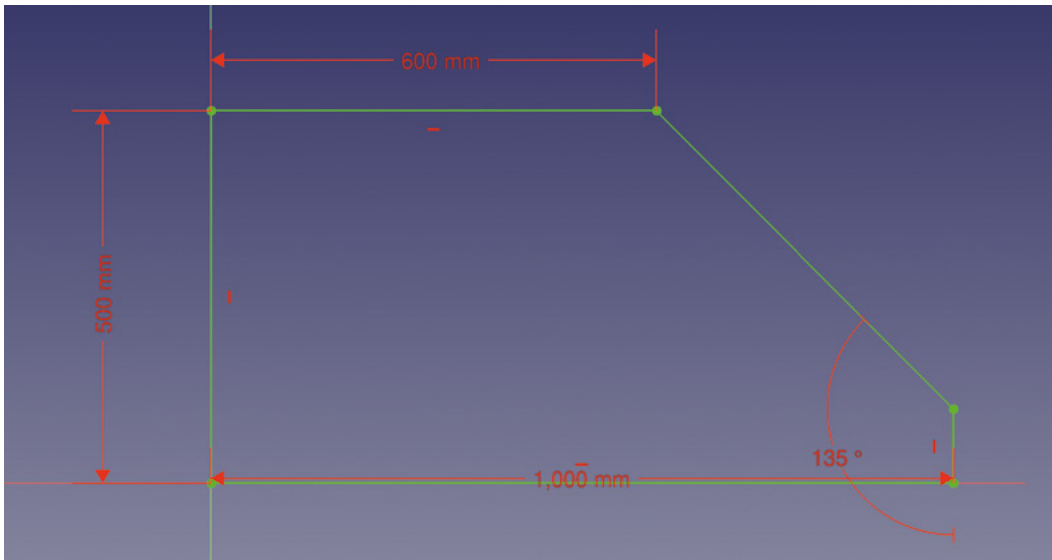


As you move, trim, and extend H-sections, you will probably start to see the limitation of the Dodo workbench. If you position a corner of our frame, similar to **Figure 4**, you will see that although the H-sections are in the correct position, the overlapping portion of the sections aren't trimmed. Of course, this would be impossible to realise in a real-world build. We've created a workaround that works with simple H-section overlaps; this can lead to quite a complex file tree, but it's still useful to look at here. Arrange a corner of the frame so that it looks like **Figure 4**, and then select the two H-section objects that form this corner in the file tree inside the 'Travatura' group – select these in an order so that the part you want to remove the overlap from is selected last. We opted to make the internal part the object we wanted to trim off the overlapping parts, which are two small sections on the top and bottom of the H-section. Next, move to the Part workbench and click the 'Slice Apart' tool, which is in a drop-down menu off a tool icon for the less common Boolean operations. You should now have a new group in the file tree called something like 'Exploded Slice' – inside this group exists a copy of our target intersecting part. In the file tree view, toggle the visibility of the original target part in the Travatura group to invisible, then, in turn, toggle the parts inside the Exploded Slice group to work out which parts you need to remove or make invisible. You should end up with an H-section part within the Exploded Slice group that fits the corner assembly perfectly as if ready for welding in a real-world build (**Figure 5**).

INTERESTING INTERSECTIONS

So far, we have made frames with frame members at 90 degrees to each other. Of course, it's possible to create frame structures around sketches or paths with other angles in them; however, we need to use some workarounds if we want to create accurate and neatly trimmed intersections. As an example, let's create a rectangular object with a large 45-degree plane on one end to create a frame around.

In a new project, open the Sketcher workbench and create a new sketch in the XZ plane. Create a sketch similar to **Figure 6**. In this example, it's probably



useful to fully constrain your sketch so that we know accurately where frame elements will be placed. Next, move briefly to the Part workbench and extrude the sketch into a large frameable object.

Moving to the Dodo workbench, once again open the FrameBranch Manager dialog and create a frame on our object in the usual way – we opted again to use the RH 40 × 40 × 5 square box profile. When you initially create the frame, you'll notice that the angled sections need to be extended to fully meet the vertical and horizontal sections. If you then select targets and Trim/Extend, you'll see that the angled section now extends through the other sections with some excess material. You can see this in **Figure 7**, where the front angled section hasn't been extended and the rearmost section has.

REMOVING OVERLAPS

Continuing to Trim/Extend, we can get the horizontal and vertical frame members to adjoin correctly, as seen in **Figure 7**. Now we are ready for our workaround solution for the angled parts. Basically, the idea is to create parts that cover all the material we want to remove and then, using the Part workbench, we can use Boolean cut operations to remove the unwanted overlap sections. There are, as ever in FreeCAD, numerous ways we could achieve this. We could, for example, create simple solid cuboid objects on the Part workbench, move them into place, and then perform the cut operations.

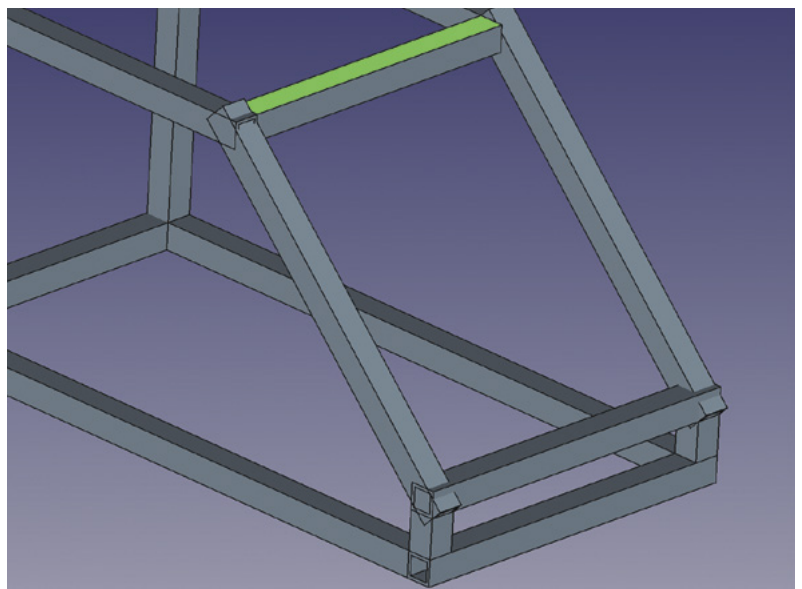
Another, perhaps cleaner, approach is to map a sketch onto the frame to create an extrusion that matches the material we need to remove and then perform the cut operation. Select the outside face of one of the short upright frame sections, then, on the Sketcher workbench, create a sketch. When prompted, choose 'FlatFace' as the sketch

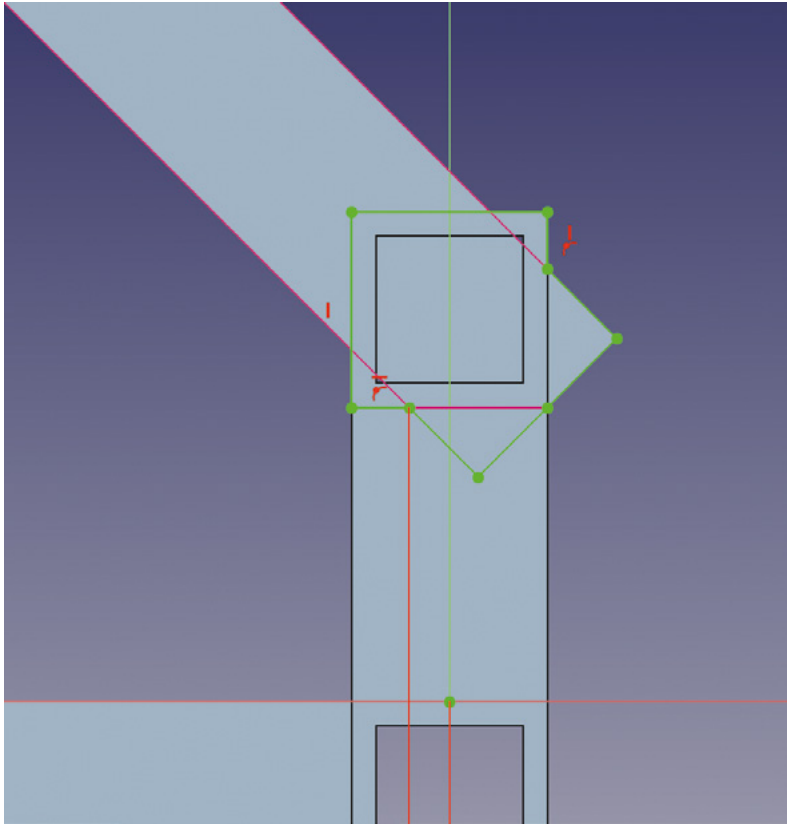
attachment method. In the sketch, use the 'Create an edge linked to external geometry' tool to import any edges and vertices that will help you create your sketch. Then, use the Polyline tool to create a sketch that will remove all the material you need to remove from your target frame section. You can see our sketch in **Figure 8**.

Next, move to the Part workbench and extrude the sketch you just created. Extrude it in such a way that it moves through the frame element you wish to cut material from. You can only perform a cut operation using two parts, so the next step is to select one of the angled frame members from inside the Travatura folder in the file tree and then select the extrusion we just made. Click the 'Make a cut of two shapes' tool; this should perform the correct cut on one of the →

Figure 6  Our base sketch in the XZ plane to create our wedge-shaped frame object

Figure 7  When working with angled frame sections, it's hard to make frame members adjoin correctly





// You can add pipe objects and other peripherals by simply attaching them to an existing pipe

//

angled frame members. Next, select the other frame member you wish to cut excess material from and then select our extrusion object, which now can be found inside the 'Cut' object folder that was just created in the file tree. In **Figure 9**, you can see that this has worked perfectly and has given us the frame member arrangement that we wanted to achieve.

The Dodo workbench doesn't just have excellent tools for creating framed designs, it also has a collection of tools for making pipework designs easier to realise. Like many aspects of FreeCAD, it's incredibly capable, but we can start with some simple examples that you can build on as you explore the tools. In a new project, let's start by making a sketch in the XY plane. We created a simple square U shape using the Polyline tool. We made it a reasonable size because, if it's too small, when we add pipes following the sketch, it might fail if the pipe corners can't fit in the dimensions of the sketch. It's worth noting that pipe and frame objects on the Dodo workbench can be applied to not just sketches but also wires created using Draft tools, or you can simply add pipe objects onto other existing pipe objects.

With a sketch created, move to the Dodo workbench and click the 'Open PypeLine Manager' tool icon.

In the PypeLine Manager dialog box, there's a range of standard pipe types to select from – we chose the 'DN40' pipe from the 'SCH-STD' group. This is a common pipe dimension with a 40 mm internal diameter. Next, click any part of the sketch in the preview window and then click the 'Insert' button in the PypeLine Manager dialog. This should

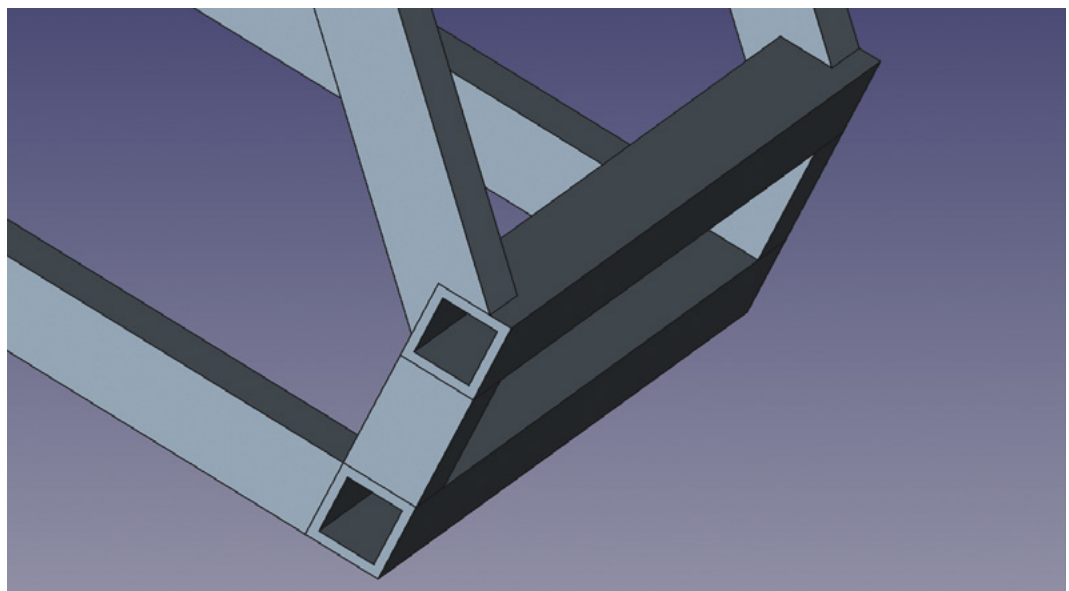
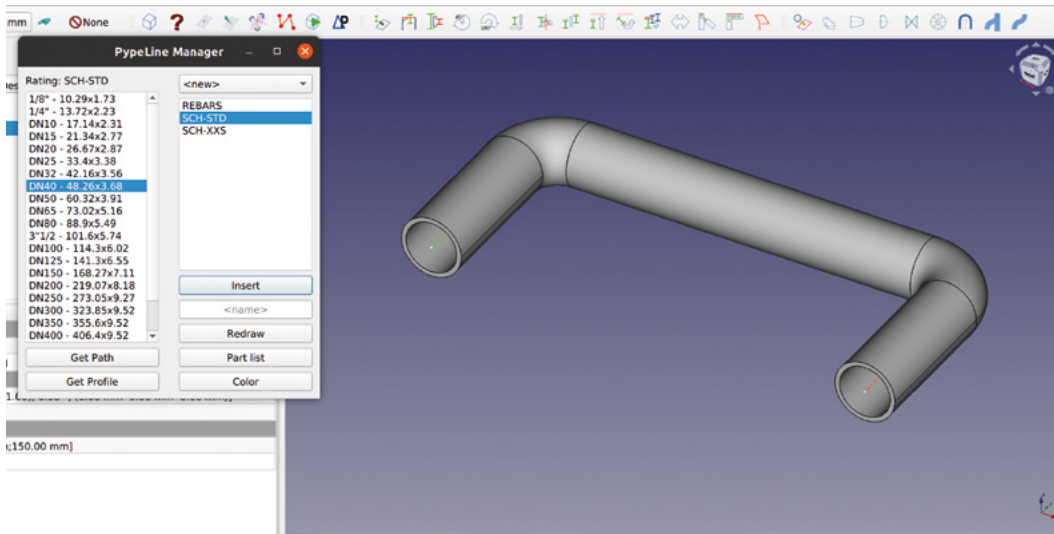


Figure 8 ♦ A sketch set up for extrusion to create an object that will cut away and trim the frame members correctly

Figure 9 ♦ Using our workaround, we now have the ends of the angled frame members correctly trimmed



automatically create a pipe that follows the complete sketch path with the 90-degree curved sections added (**Figure 10**).

We mentioned earlier that you can add pipe objects and other peripherals by simply attaching them to an existing pipe. There are numerous tools that help us do this. Selecting the outer edge of one end of our U-bend pipe construction, click the 'Insert a Reduction' tool. The larger end of the reduction is automatically defined by the diameter of the selected adjoining pipe, and so we simply use the 'Insert Reductions' dialog to select the smaller end of the reduction. In **Figure 11**, you can see we opted for the 'DN20' size, and then clicked the Insert button to add the reduction.

It's as straightforward to add a simple length of straight pipe to an existing design. Again, select the edge of the thin end of the reduction we just added and click the 'Insert a Tube' tool. In the dialog box, select the tube type that matches the diameter of the reduction (DN20) and then type a dimension into the length input box to set the length of the tube you require. Finally, click the Insert button to insert the tube. Similarly, we can add a curved section or 'elbow' into the design using the 'Insert a Curve' tool. When inserting a curve, you select a target edge as before and then select a matching tube profile – again, we chose DN20. You then need to add a bend angle and a bend radius into the input boxes. Starting with values of 90 for each gives you a good result – you can experiment with other values. A small issue with this is that it's easy to input values that can't actually exist and therefore won't work! Once you have your values and options selected, click the Insert button to insert your curve. Finally, with the 'Insert Elbows' dialog, you have a dial wheel that you can adjust to change the angle of the curved elbow you have just added. You can see the reduction, manually added pipe, and the angled curved elbow we have just added in **Figure 1**.

Often pipework uses flanges as joining mechanisms between pipe sections, and as such, the Dodo workbench has a Flange tool. At the other end of our pipe experiment, we selected the outer edge of the pipe end and clicked the 'Insert a Flange' tool. There are various flange standards in the dialog box – we selected a DN40 type to fit our existing pipe edge. We then clicked Insert to add the flange (**Figure 1**).

We're sure by now that you have the general idea, but it's worth exploring the other tools available to manipulate and create pipe systems. There are tools for adding valves of differing types, U bolts to secure pipes to other structures, tools to cap the end of pipes, and tools to create flowing branches in pipes. There really isn't much you can imagine that you couldn't design using these tools. Finally, it's also worth exploring how you could add to designs using the workbenches we have looked at previously. When adding flanges, for example, it would be quite straightforward to use the 'Fasteners' workbench to create nuts and bolts to join flanges together. We looked at the Fasteners workbench as part of the tutorial working with assemblies in issue 43. □



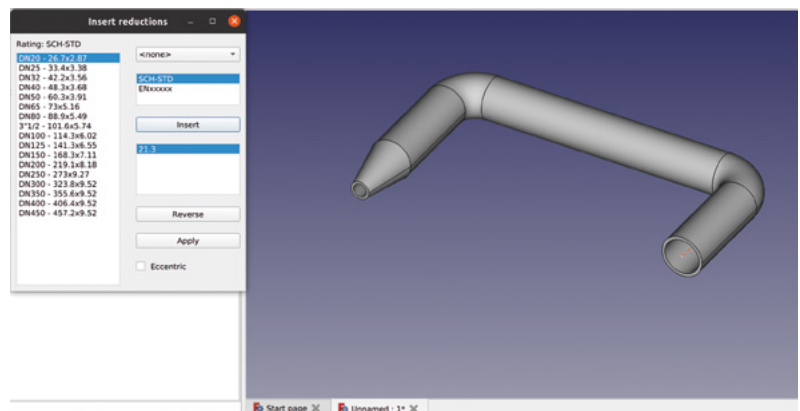
Figure 10  Creating pipework that follows a sketch is simple using the PypeLine Manager tool

Figure 11  Adding a reduction part to our simple pipe example



Handmade earrings – a beginner's guide

Making some basic earrings is way easier than you might think.
Nicola King has her pliers ready to bend some wire



Nicola King

 @holtonhandmade


Nicola King is a freelance writer and sub-editor who is a jack of many crafts, master of... maybe one or two...possibly?



Making earrings, or making any jewellery for that matter, might seem a step beyond for many people, but don't be deterred by thinking that it's a difficult skill to acquire. Making very simple pieces of jewellery, using easily accessible findings and a few beads, is actually not that hard and can be a very quick process, once you get the hang of it. This author was frankly astounded at the speed that she could produce basic pairs of earrings when she first picked up her jewellery pliers, and has since produced many pairs as gifts for friends and family. So, in this tutorial, we are going to take a look at the basics of

making earrings: what you need in order to make them, some quick techniques, and some inspiration to take it further – including using that old favourite, some LEDs, in your makes.

The advantages of making jewellery yourself are numerous. Firstly, it can be extremely quick to do – in fact, a simple pair of earrings can take a few minutes to put together. If you buy some ready-made findings and charms online, you can prepare a gift remarkably quickly – think about that in the run-up to seasonal gift-giving, or when you need to produce a last-minute present for someone. Secondly, handmade gifts mean so much more than something bought off the shelf, as taking the time to make something yourself,

Above 
Our finished ultra-quick earrings; a speedy last-minute gift

specifically tailored to an individual, is something special. You can personalise gifts using the recipient's favourite colours or styles; most of all, you can be creative, making something completely unique. The final advantage is monetary – we'll be using some inexpensive earrings hooks and components in this tutorial, but you might also be pleasantly surprised at the cost of sterling silver earring hooks and other components – they are very affordable from jewellery-making supplies stores, and so making some sterling silver earrings as a gift can be cheaper than purchasing from a high-street jeweller, plus you also have the kudos of having made the gift yourself.

THE BASICS

So let's first talk about terminology, specifically jewellery 'findings' – what are they? 'Findings' is the umbrella term for the various components that are central to making a piece of jewellery. There are many such components, but for the purposes of this tutorial, they include:

Jump rings (open) – circles of wire that are not welded together, so you can open and close them. Here, we'll be using them to link some of our earring pieces together.

Head pins – pieces of straight wire with a 'head', or stopper, at one end that ensures beads don't fall off. You can thread beads, pearls, and so on onto them, and they provide the 'skeleton', if you like, that your beads can sit on.

Earring wires – pieces of wire that have been shaped to fasten an earring to a pierced ear. Proficient jewellery makers can make their own findings, including earring wires, but in this tutorial we'll be using pre-made components bought online.



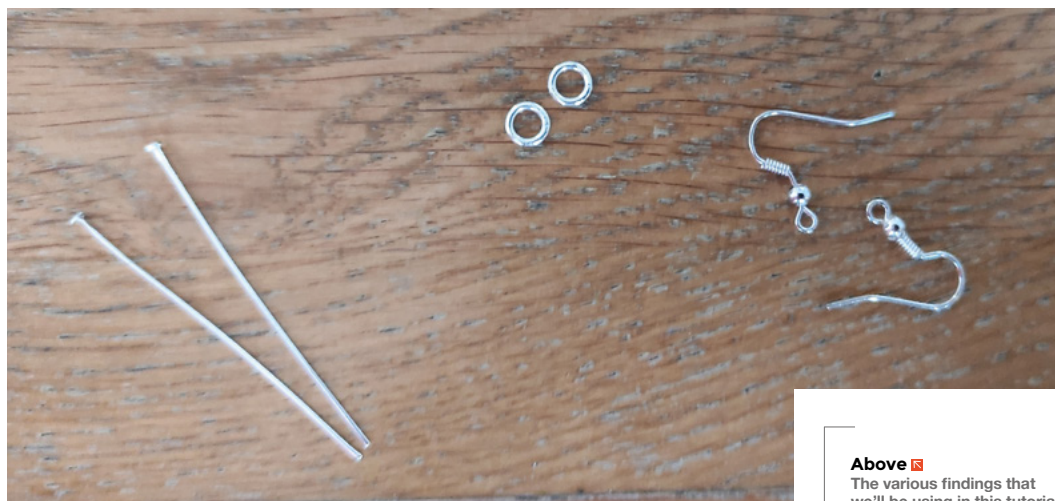
Above Yes, you too can have a pair of baguettes, doughnuts, or wine glasses dangling from your ears. Pre-made charms can be cheap, fun, and rather unique!

A SIMPLE MAKE

So, armed with the fundamentals, let's first make a very uncomplicated pair of earrings. Here, we're going to use some inexpensive pre-made charms, purchased from Amazon. This kit actually came with some earring wires and jump rings included (hsmag.cc/SimpleEarrings), so these literally couldn't be any easier to make, as everything you need is in one small kit.

STEP 1

Make sure that you have two jump rings, two earring wires, and two charms in front of you, along with a pair of round nose and chain nose pliers. →



Above The various findings that we'll be using in this tutorial

YOU'LL NEED

For basic earrings

- ◆ **Jewellery making toolkit** (including several types of pliers)
- ◆ **Earring findings:** including earring wires, jump rings, and head pins
- ◆ **Ready-made charms**
- ◆ **A selection of colourful beads**
- ◆ **Eye protection** (advisable when cutting wire)

For the LED earrings

- ◆ **Superglue and craft glue**
- ◆ **Craft knife**
- ◆ **Pin or needle, and reaming tool**
- ◆ **2 × jump rings**
- ◆ **2 × earring wires**
- ◆ **2 × CR1225 lithium batteries**
- ◆ **2 × 15 mm googly eyes**
- ◆ **2 × 3 mm white LEDs**
- ◆ **A kit containing small glass baubles or bottles, and decoration** (e.g. hsmag.cc/EarringBaubles)

TUTORIAL

QUICK TIP

Ensure that you have good lighting in your workspace so that you can see what you are doing clearly. When you are making jewellery you are, more often than not, handling some extremely small components.

Right

The jump ring has been opened, ready to add the earring wire

Below

Pliers aplenty – round nose and chain nose pliers, plus side cutters



STEP 2

Take your round nose pliers and pick up one of the jump rings. Try and find the point in the jump ring where the two ends of it meet, i.e. where it is cut, and ensure that that is at the top. Take your chain nose pliers in your other hand and place them opposite to your round nose pliers, and gently twist open the jump ring, just as you would open a gate. Twist your non-dominant hand towards you and your dominant hand away from you, and this will cause the jump ring to open. What’s important to remember is that you should never just pull the jump ring apart sideways (into an oval shape) as it will lose its round shape and you will never, no matter how hard you try, be able to return it to its original round shape. Opening jump rings incorrectly also weakens the metal of the ring.

STEP 3

Now that your jump ring is open, still holding it with your round nose pliers, thread the charm onto the jump ring with the loop that it should have at the top of it, and then thread the bottom loop of the earring wire onto the ring too. All that remains is to close the jump ring. So, pick up your chain nose pliers once again in your other hand and gently twist the jump ring closed, just twisting it back until the two ends meet neatly. Repeat the process for the other earring.

There you have it: one completed pair of earrings. Yes, it really is that quick and easy (but don’t tell the recipient they only took you five minutes to put together – just bask in their admiration and appreciation!)

Below

The wires bent to an angle, ready for the next step



BASIC TOOL-KIT

The key tools that you need for jewellery making need not be expensive. There are many options, but for the purposes of making some fairly simple earrings, an absolute essential is a set of jewellery making pliers. This is a useful first step into the world of jewellery making paraphernalia, and such kits can easily be found online (hsmag.cc/Pliers). Sadly, an old pair of pliers you may have lying around in the shed will probably not be suitable here, so make a small investment and it will reap dividends if you plan on making a lot of pieces of jewellery. Integral to a jewellery making kit are chain nose pliers (great for tight areas, precision work, and opening jump rings), round nose pliers (perfect for bending wire and creating loops), and side cutters (needed for cutting wire) – these should be all you need to get going. If you progress, you can widen your tool-kit to include specific tools tailored to the numerous realms within jewellery making, such as working with wire, setting stones, and so on.





QUICK TIP

Think about how you are going to present your earrings as a gift to the wearer; make a small card on which to mount them, and put them in an organza bag for a more luxurious touch.

TAKING IT FURTHER

Now, let's make our own charms to hang from our earrings.

STEP 1

Ensure that you have a small selection of beads, two head pins, and two earring wires in front of you, along with your pliers, including a pair of side cutters this time.

STEP 2

Take a head pin and thread or stack some beads onto it in whatever order you like – we're using some Swarovski beads and some small faux pearls as a contrast here. Then, take your chain nose pliers and bend the wire at the top of the first bead to a 90-degree angle.

STEP 3

Take your side (wire) cutters and trim off any excess wire so that you have about 1.5cm left. N.B. Be very careful to hold the end of the wire as you cut it – if you don't, it could fly off and hit someone in the eye or bury itself in a carpet, so do exercise some caution here.

STEP 4

Next, take your round nose pliers and place them on the end of the wire, and roll the wire back towards the bend in the wire until you have a small loop, →



You should never just pull the jump ring apart sideways (into an oval shape)



ACE OF BASE METAL

In this tutorial, we have used surgical steel earring hooks for two reasons: firstly, surgical steel is a good base to use if the wearer of the earrings has an allergy to silver or some plated findings, and secondly, it is an inexpensive option compared to other metals used for making earrings. An internet search will throw up plenty of sources where you can purchase findings, but here's where we purchased ours: cooksongold.com.

You can also find lots of other cost-effective findings which are silver-plated, gold-plated, bronze... the choice is yours. Sterling or 925 silver (so named as it is an alloy made from 92.5% silver and 7.5% other metals), gold, and other precious metals are obviously the most sought-after and popular materials to use for jewellery making, but these do come with a slightly higher price tag. As a beginner project, it's probably not advisable to spend too much money on your supplies; if you decide to take jewellery making further, you can invest in more extravagant materials. That said, sterling silver and gold are great options for earring wearers who have a sensitivity to other metals, and may be the only thing they can wear without having a reaction.

Above

Our finished earrings, with bespoke, handmade charms

TUTORIAL



Above ♦ The jump ring and earring wire are now attached to the battery casing

Below ▣ These were made using some dies, a die-cutting machine, some faux leather, and fabric glue. Bling it up!

with the cut end meeting that bend. (If it looks a little crooked, straighten it out with your chain nose pliers.)

STEP 5

Finally, take your chain nose pliers and open the loop that we just made, lifting the end of the wire up so that we can attach our earring wire. With that added, close the loop again neatly, pulling the end of the wire back down so that it meets the other side of the loop and is secure. Repeat the process once again, then sit back and admire your handiwork.

DIY LED EARRINGS

Given that so many HackSpace projects include LEDs, it seems only appropriate to incorporate them here too, into an earring make. So, we've made these LED earrings based on this great tutorial by IdunnGoddess (hsmag.cc/IdunnGoddess), which we've slightly adapted, but the concept is the same. LEDs in jewellery certainly add a cool and unique twist.

CUTTING YOUR OWN DESIGNS

You can fashion all sorts of things to dangle from ear wires, and a very simple idea is to cut a design from faux leather using some earring dies and a die-cutting machine. We used some cheap dies bought online (hsmag.cc/EarringDies), and, along with a manual die-cutting machine and some various colours of faux leather, we made some vibrant earring trinkets very speedily. These are simple, lightweight, and you can make them in any colour you want!

Alternatively, if you have access to a digital cutting machine, you can also use that, so no dies needed. You'll need a machine that can cut leather, or the equivalent, so one of the larger types of digital machine such as a Cricut Maker. A search on YouTube will throw up lots of inspiration and tutorials on this, including this web page – hsmag.cc/CricutEarrings. All you need to do is hook on an earring wire with a jump ring and job done!



QUICK TIP

Buying pre-made charms in multipacks means you can easily make more than one pair if people comment on your fantastic make, and ask for a pair for themselves!

STEP 1

Take a googly eye and make a thin slit on one side (remove the black part of the eye too). The slit should be large enough so you can slide your CR1225 battery inside. We'll now rename the googly eye, and call it the casing. Then make a small slit on the opposite side so that the legs of the LED can poke through it. Repeat for the other eye. N.B. Take great care with the craft knife here, as this is a fiddly process.

STEP 2

Take a pin or a sewing needle and make a small hole on the white base of the casing, on the opposite side of the small LED slit. Make this hole a little larger with a reamer tool if you need to. Then, thread through a jump ring, with earring wire attached. Again, repeat for the other casing.

STEP 3

Decorate the top side of the casing with glitter or whatever you have – our kit came with some decorative stars, so we craft-glued those on.

Below ♦ Light up your life with these homemade LED sparklers!



WARNING

It's a good idea, when cutting wire, to wear some form of eye protection, as small flying pieces of wire can be dangerous.

Also, coin cell batteries should be kept well away from young children, as they can be extremely dangerous if swallowed.

STEP 4

Take your glass baubles, or bottles, and fill with the same glitter or decorations. N.B. This can be a little messy, so have a dustpan and brush to hand!

STEP 5

In each earring, glue the LED into the neck of the bauble with superglue and leave to set. When set, assemble your earring by inserting the LED legs into the casing and supergluing them to the small slit to



You could also bake some earring charms by rolling out and moulding some modelling clay



keep them in place. Add the batteries into the top of the casings – the top side, marked '+', should contact the longer LED leg – and your earrings should light up! Hey presto: LED luminosity hanging from your ears!

Feel free to modify and customise this to meet your requirements. Our LEDs' legs were a little long and we left the part between the bauble and battery holder casing exposed – we could have cut them shorter and glued the bauble to the bottom of the casing. You could also consider using a proper battery holder (e.g. hsmag.cc/BatteryHolder) in your make instead of googly eyes, which could easily be disguised with some faux leather, for example.

JEWELLERY MAKING INSPIRATION

Some useful websites that include tutorials, supplies, and plenty of suggestions include:

cooksongold.com

jewellerymaker.com/en-gb

knowcraft.com/jewellery-making-tips

hsmag.cc/PinterestJewel

beadsdirect.co.uk

Amazon and AliExpress also have a plethora of options that pop up if you search for 'charms', 'jewellery making', 'earring making', etc.



FINAL THOUGHTS

The world is your oyster when it comes to making earrings; think about 3D-printing some earring charms (hsmag.cc/3DPrintEar), or how about making some out of resin (hsmag.cc/ResinEarrings)? You could also bake some earring charms by rolling out and moulding some modelling clay (hsmag.cc/ClayKit) – a fun way of making something unique. There are also a multitude of jewellery making techniques to potentially master, including wire-wrapping, stone setting, and silversmithing, to name just a few – we've only skimmed the surface here. The possibilities are virtually limitless, so have a go, surprise your loved ones with handmade gifts, and give your creativity a fresh outlet. □

Above ♦ All you need is a small rolling pin – this Fimo kit even includes cutters, and you can bake your own earrings!

Make a talking Pomodoro timer with Pico

Time your tasks and pace your work with a talking timer powered by a Pico



Rob Miles

[@rob miles](#)

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 rob miles.com.

We're going to make a talking 'Pomodoro' timer that you can use to time a task. Because, as reality TV shows demonstrate, nothing motivates people like having someone shout 'Five minutes remaining!' towards the end of a tricky task. The Pico Talking Pomodoro timer can announce the time left at fixed time intervals (five or ten minutes) or random intervals for that proper TV show feel. It can count elapsed time or time remaining and has an option for 'silent running' with no display or sound output until the time expires. It is powered by a Raspberry Pi Pico

driving a tiny MP3 player containing sound files for all the announcements.

WHAT IS A POMODORO TIMER?

The Pomodoro Technique aims to improve productivity by performing work as a series of tasks. It was described in the book *The Pomodoro Technique* by Francesco Cirillo. The name 'Pomodoro' comes from the tomato shape of the kitchen timer he used when developing the technique. The idea is that you work on a given task for a fixed time (25 minutes). Then you break for 5–10 minutes and repeat. After your fourth repeat, you take a longer break before starting again. The original advice was

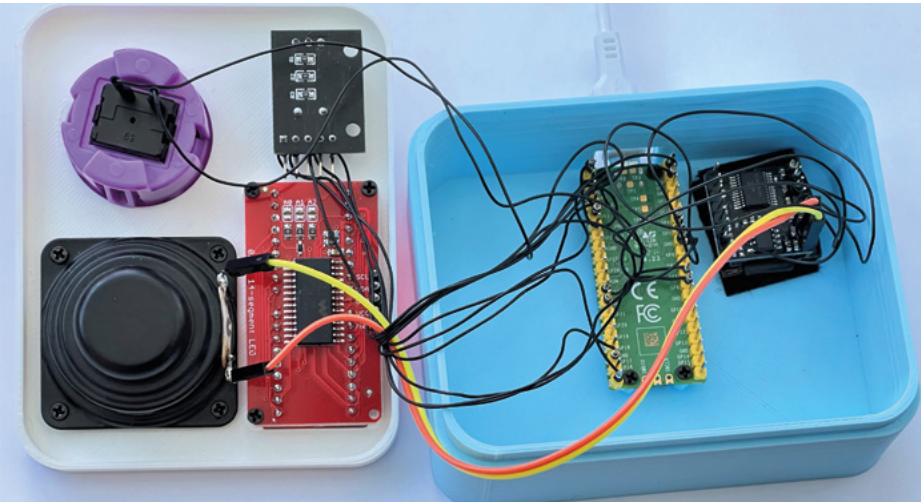


Figure 1 ♦

On the top left of the timer is a four-character alphanumeric display for messages. To the right of that is a rotary encoder. Underneath, you find holes for the speaker and the announcement button

Figure 2 ♦

On the white lid of the device, on the left, you can see (from top left clockwise) the purple push-button, rotary encoder, alphanumeric display, and speaker. Inside the box, on the right of the photo, you can see (from left to right) the Pico and the DFPlayer Micro device. The DFPlayer MP3 player can drive two speakers for stereo output. It also has line-level stereo outputs for an amplifier or headphones

to use a mechanical timer to measure the length of each task, because the act of winding up a clockwork mechanism was part of the preparation for the task. We are not going to use a spring to power our timer, but we are going to use a rotary encoder to set the time duration, which should suffice.

THE PICO POMODORO TIMER

Figure 1 shows the finished product, along with the original clockwork version. The Pico Talking Pomodoro timer uses a rotary encoder to allow the user to enter the duration of the task and select timer options. Tasks can last up to eight hours. The rotary encoder is also used to select whether the timer counts up (telling you how much time has elapsed) or down (telling you how much time is left). The four-character alphanumeric display shows the time, and is also used to select the timer options. There are five different options:

- **None** – no time is displayed and the timer just plays a sound at the end of the task
- **Disp** – the time is displayed with no announcements
- **Five** – the time is displayed and the time left (or elapsed) is announced every five minutes

- **Ten** – the time is displayed and the time left (or elapsed) is announced every ten minutes

- **Rand** – the time is displayed and the time left (or elapsed) is announced at random intervals

- You can request a time announcement by

pressing the large button. You can cancel the timer by pressing down the rotary encoder and then selecting Stop.

INSIDE THE BOX

Figure 2 shows the contents of the timer. Most of the connections are made using wire wrap. The speaker was connected using slightly thicker wire to better handle the audio signal from the amplifier. There are no mounting holes on the DFPlayer device, so it was stuck to the bottom of the box using double-sided tape. Note that the DFPlayer is oriented so that we can remove and replace the microSD card that is fitted into it if we want to change the sound files.

Figure 3 (overleaf) shows the circuit for the timer. The four-character alphanumeric display on the left needs two power inputs: a connection to the 5-volt supply to power the LEDs, and a connection to the 3.3-volt supply to power the logic interface to the Pico, along with I2C connections to the Pico. The DFPlayer device on the far right is connected to a serial port and receives commands from the Pico to select the sound files to be played. The rotary encoder in the middle needs a power supply to be connected, as well as the two square wave outputs and the push-button signal.

IMPORT CONTROLS

```
import board
import busio
import time
import random
import rotaryio
from digitalio import DigitalInOut, Direction, Pull
from adafruit_debouncer import Debouncer
from DFPlayer import DFPlayer
from adafruit_ht16k33 import segments
```

YOU'LL NEED

- ♦ **A Raspberry Pi Pico**
- ♦ **A DFPlayer Mini MP3 player** (search for 'DFPlayer Mini')
- ♦ **A microSD card.** This is perhaps the most expensive component. However, you don't need a very large one. The author used an 8-gigabyte card that he received with a tiny camera many years ago
- ♦ **A large push-button** (preferably one with a nice colour)
- ♦ **A rotary encoder** (search for 'KY-040 encoder')
- ♦ **A small speaker** (You can get a very nice one from Pimoroni: shop.pimoroni.com/products/mini-speaker-4-3w)
- ♦ **A micro USB cable to link the Pico to the host**
- ♦ **Connecting wire** (search for '30 AWG wire wrap') which needs a wire wrap tool (search for 'wire wrap tool')
- ♦ **A box.** There is a 3D-printable design, or you can put your timer in any box you fancy
- ♦ **Screws.** You'll need some screws sized M2 4 mm in length to fix things to the case (search for 'laptop screws')
- ♦ **Sticky fixer.** The MP3 player doesn't have any fixing holes, so you must stick it inside the case using double-sided tape or a sticky fixer pad

Make a talking Pomodoro timer with Pico

TUTORIAL

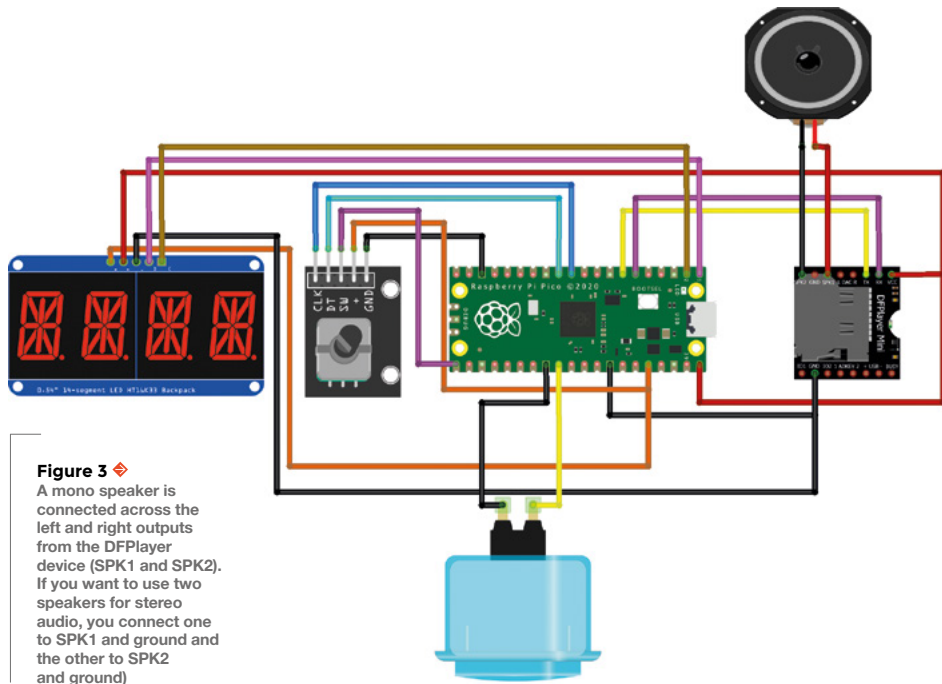


Figure 3 ◆
A mono speaker is connected across the left and right outputs from the DFPlayer device (SPK1 and SPK2). If you want to use two speakers for stereo audio, you connect one to SPK1 and ground and the other to SPK2 and ground)

At the start of the Pomodoro code, we import all the libraries that are needed by the program. You can see all the `import` statements that are used. These specify the CircuitPython libraries that contain the device drivers. Some of these libraries are supplied with CircuitPython 7.0; others need to be added to the `lib` folder on your device.

Figure 4 shows the libraries that you will need to install. You can find a copy of this library folder on the GitHub site for this project using the following link: hsmag.cc/TalkingPomodoro.

A CLASS ACT

The devices are controlled by the CircuitPython program running inside the Pico that controls the device. The program contains a `PomodoroTimer` class that provides all the timer behaviours:

```
timer = PomodoroTimer(i2c_sda=board.GP0,
                      i2c_scl=board.GP1, play_tx=board.GP4,
                      play_rx=board.GP5, vol=60,
                      encoder_a=board.GP8,
                      encoder_b=board.GP9,
                      encoder_button=board.GP16,
                      time_button=board.GP22
                      )
```

The code above creates an instance of the `PomodoroTimer` class. The constructor is given parameters which are pin instances to be connected to the underlying hardware. As an example, the alphanumeric display is controlled over an I2C (inter-integrated circuit) connection. The serial data (`sda`) signal for the display is linked to pin `board.GP0`. The serial clock (`scl`) for the I2C is linked to `board.GP1`. This means that you should connect the SDA signal

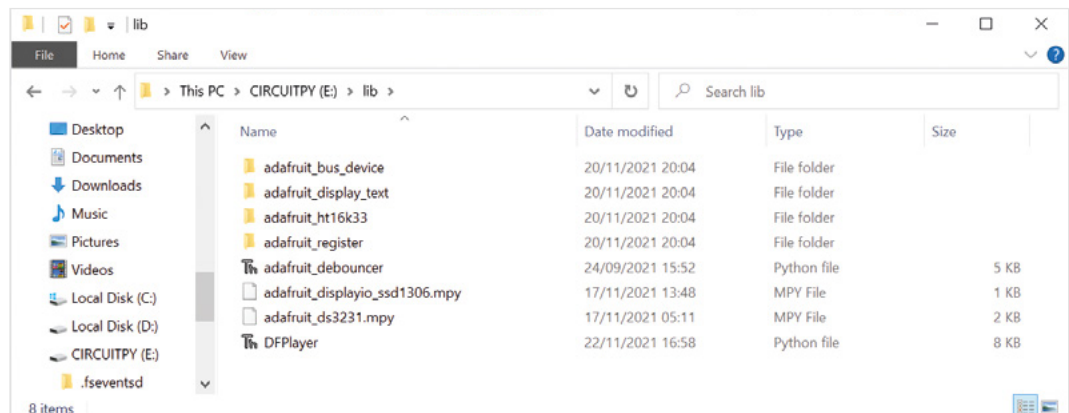


Figure 4 ◆
A CircuitPython device exposes a USB storage device when connected to a computer. This holds program code, data, and library files. You add libraries to the device by copying them into this folder on the device

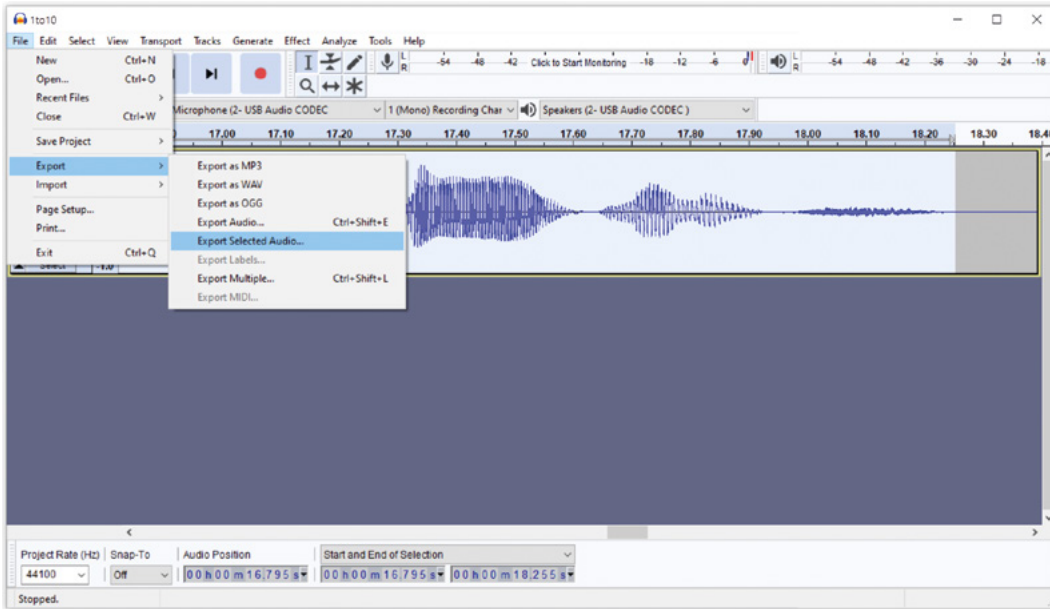


Figure 5 This is what the phrase 'Fifty-four minutes' looks like in the Audacity editor

Figure 6 Set the file format to MP3

from the alphanumeric display to pin GP0 on the Pico and the SCL signal to GP1. If you work through all the constructor parameters, you can see how all the components are connected. If you want to change the pin assignments, you can modify these values – although you need to be careful because not all the pins on a Pico are interchangeable. The constructor also sets the initial volume of the announcement playback. A volume level of 60 seems to work well with the speaker that we are using.

```
def __init__(self, i2c_sda,i2c_scl,
             play_tx,play_rx,vol,
             encoder_a,
             encoder_b,encoder_button,
```

```

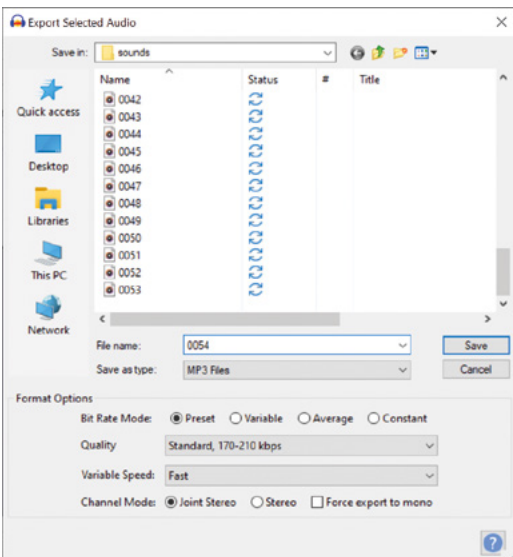
        time_button
    ):
        # Make i2c connection
        self.i2c = busio.I2C(i2c_scl, i2c_sda)
        # Make display
        self.display = segments.Seg14x4(self.i2c)
        self.display.fill(0)
        self.display.print('Hi ')
        # Make mp3 player
        self.uart = busio.UART(tx=play_tx, rx=play_
rx)
        self.dfplayer = DFPlayer(uart=self.
uart,volume=vol)
        # Make encoder
        self.encoder= rotaryio.
IncrementalEncoder(encoder_a, encoder_b)
        # Make encoder button
        tmp_pin = DigitalInOut(encoder_button)
        tmp_pin.pull = Pull.UP
        self.encoder_button = Debouncer(tmp_
pin,interval=0.01)
        # Make time button
        tmp_pin = DigitalInOut(time_button)
        tmp_pin.pull = Pull.UP
        self.time_button = Debouncer(tmp_
pin,interval=0.01)
```

The code above is the `__init__` method in the `PomodoroTimer` class. It is called the constructor method for the class. It uses the setting parameters that it receives to create driver objects connected to the hardware devices.

BUILDING IN TEST

When you build a new device, it is a very good idea to also write some tests that you can use to prove that →

QUICK TIP
 This construction technique is sometimes called 'dead cockroach' because of the way that devices are arranged on their backs with their legs in the air.



Make a talking Pomodoro timer with Pico

TUTORIAL

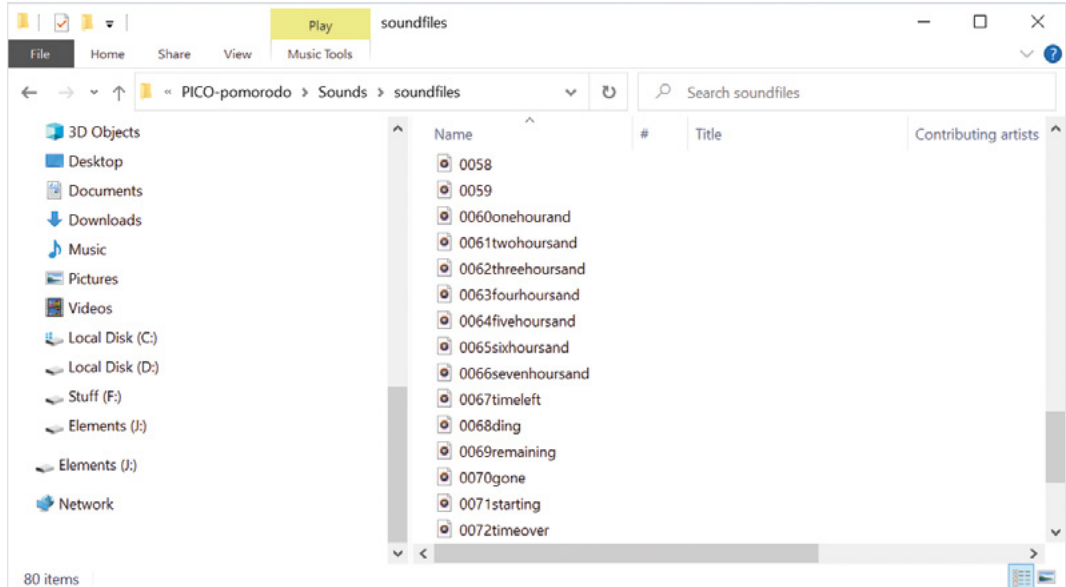


Figure 7 The files that just have numbers for their names are the minutes units

all the components work correctly. Otherwise, you might end up having to fix the hardware while you are trying to write the code to control it.

```
def test(self):
    print("Pomodoro Self Test",Version)
    # record the current encoder position
    old_enc_pos=self.encoder.position
    # say hello
    self.display.print("Test")
    # loop forever
    while True:
        # update the button on the encoder
        self.encoder_button.update()
        # has the encoder button been pressed?
        if self.encoder_button.fell:
            # print a message if the button was
            pressed
            print("Encoder pressed")
            # update the time button
            self.time_button.update()
            # has the time button been pressed
            if self.time_button.fell:
                # print a message
                print("Time pressed")
                # play some messages
                self.blocking_sound_play(67)
                self.blocking_sound_play(61)
                self.blocking_sound_play(25)
            # get the encoder position
            enc_pos=self.encoder.position
            # has the encoder been turned?
```

```
if enc_pos != old_enc_pos:
    # display the encoder position
    self.display.fill(0)
    self.display.print(enc_pos)
    # record the old encoder position
    old_enc_pos = enc_pos
```

The code above is the test method for the Pomodoro. It tests all the different devices and should give you a good idea of how they are controlled from CircuitPython. This method is part of the **PomodoroTimer** class and can be called to make sure that everything works.

TIMEKEEPING

The timer program contains a loop that repeatedly compares the elapsed time with the end time which was set when the timer was started. If the value of the minute changes, the timer updates the display and checks to see if it should make an announcement. Each time round the timing loop, the time button is checked to see if has been pressed, and a time announcement played if it has been.

MAKING SOUNDS

The time announcements are assembled from sound files stored on the microSD card which are played in sequence. For example, the phrase 'One hour and fifty-four minutes remaining' is made up of three samples - 'One hour and', 'fifty-four minutes' and 'remaining'. Each of these is stored in a different sound file on a microSD card loaded into the DFPlayer Mini. The program plays each in turn.

QUICK TIP

When recording sound files that are going to be stitched together, be careful to add gaps that make the resulting output sound realistic. You also need to be careful about the inflection at the end of each word so that they will combine in a natural way.

The Audacity program was used to capture and edit the sound samples (**Figure 5** and **6**), which were then stored in MP3 files for use by the DFPlayer. You can download Audacity from here: audacityteam.org.

SOUND PLAYBACK

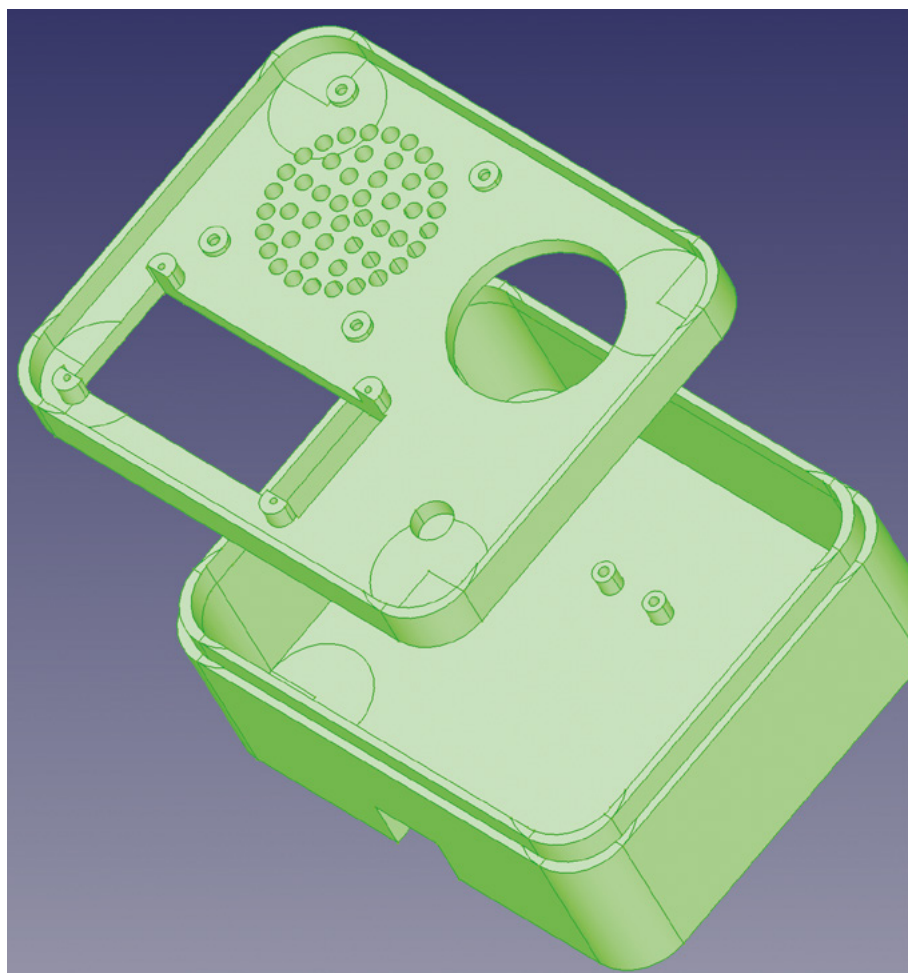
The DFPlayer Mini is an amazing device. You can use it to play music, or you can use it to give a voice to any device that you make. If you want a thermometer that says ‘Too hot’ at the appropriate time, just record the audio and then add a DFPlayer to the thermometer. The DFPlayer can be controlled by using buttons connected to it directly, but for more complex control, you use a serial connection to interact with the device. You can find details of how to use the device here: hsmag.cc/DFPlayer. The Pomodoro timer uses a library from GitHub: hsmag.cc/GHDFPlayer, which has been slightly modified. The library implements commands to select files for playback and control the volume and equalisation settings of the player. The library is part of the package stored on the GitHub site for this project.

Unfortunately, the way that files are located by the DFPlayer is a bit confusing. The files are stored on the microSD card in the usual way, with each being given a different name on your computer. However, as far as the DFPlayer Mini is concerned, the name of the file is irrelevant. Files are located by their position in the directory on the storage card. In other words, the first file on the card is number 1, the second 2, and so on.

```
self.blocking_sound_play(60) # one hour and
self.blocking_sound_play(54) # fifty four minutes
self.blocking_sound_play(69) # remaining
```

The statements above would announce ‘One hour and fifty-four minutes remaining’ by playing the appropriate samples. The method `blocking_sound_play` plays a sound file and doesn’t return until it has finished playing. Note that the numbers passed into the method are not file names, they are an index into the device of the file to be stored. It is a very good idea to put a number at the start of each file name so that you can map the numbers onto the files you want to use.

When you put the sound files onto the microSD card, you need to be careful to copy all the files in the correct order. If you just drag a collection of files from one folder to another, the operating system will copy lots of files at the same time, and so the order in which they are stored in the folder will change.



You can solve this by copying one file at a time. The GitHub site for this project contains a tiny Python program that can be used to copy the files from your PC to a storage device. It pauses for a second between each copy so that the files are stored in the correct order (as in **Figure 7**).

BOXING TIME

You can put your timer into any kind of box that you like. However, if you have a 3D printer, there are printable designs available, as shown in **Figure 8**.

FURTHER DEVELOPMENT

You can use the techniques in this project to create all kinds of noisy devices. You could make a burglar alarm that whispers, “Is that you, Joe?” when it detects someone in your room. You could make a soundboard covered in buttons that plays a different sound effect or music track when each button is pressed – this would be very useful if you record podcasts. You could also make a music box that just contains your favourite tracks. And, with a bit of luck, you might even find new clarity and purpose in all your work efforts by using it to time your tasks. □

Figure 8 ♦ The holes for the speaker were created using a tiny piece of Python that draws four successive rings of holes

QUICK TIP

You can vastly improve the performance of any speaker (especially the small ones suggested for this project) by fitting it into a case. The sound quality produced by the DFPlayer and tiny speaker is quite impressive.



Moving beyond the K40

Building a better laser cutter

Above ♦ Most images of a K40-type laser look similar, but not the same. That's because most users of the K40 laser modify the machine to suit their own needs. This machine has been modified to add air and smoke assist, laser power monitoring, and has a bolt-on controller added above the laser tube enclosure



Dr Andrew Lewis

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

If you own a K40-style laser cutter, then you probably know that it's a machine with several quirks and limitations. In this series, you'll see how to build a more capable relative of the K40 from scratch. In this instalment, you'll look at the general anatomy of a laser cutter, with a specific focus on the outer case.

Laser cutters are not very complicated machines, but they rely on technologies that can be very dangerous if they're used incorrectly. Because they have the potential to cause such harm, the idea of building one from scratch can (quite rightly) be intimidating. For that reason, many people start off with a K40-style imported laser as their first experience with laser cutting. Unfortunately, the typical K40 laser is a death-trap when it's first

delivered. Missing earth wires, poorly fitting case components leaving gaps in the chamber, and a complete lack of safety interlocks are all unusual discoveries in a newly acquired K40. It's expected that the end-user will correct all of these problems before they begin using the machine, and it's expected that the end-user will know what needs doing and how it should be done.

SIZE DOES MATTER

Once the potential problems are taken care of, the K40 can be a very capable machine. So why should you consider building your own? The most obvious answer is that the K40 bed is about the size of a sheet of A4 paper, and lots of flat things are bigger than a sheet of A4 paper. You should also

think about the quality of the components you're working with. The K40 is cheap for a reason, and pretty much every component inside it is the bare minimum needed to get the job done. Building your own machine gives you the opportunity to choose the components that will last, improve overall performance, and give you the cutting experience that best suits your needs. It will also give you a familiarity with the machine that can only be gained by self-building.

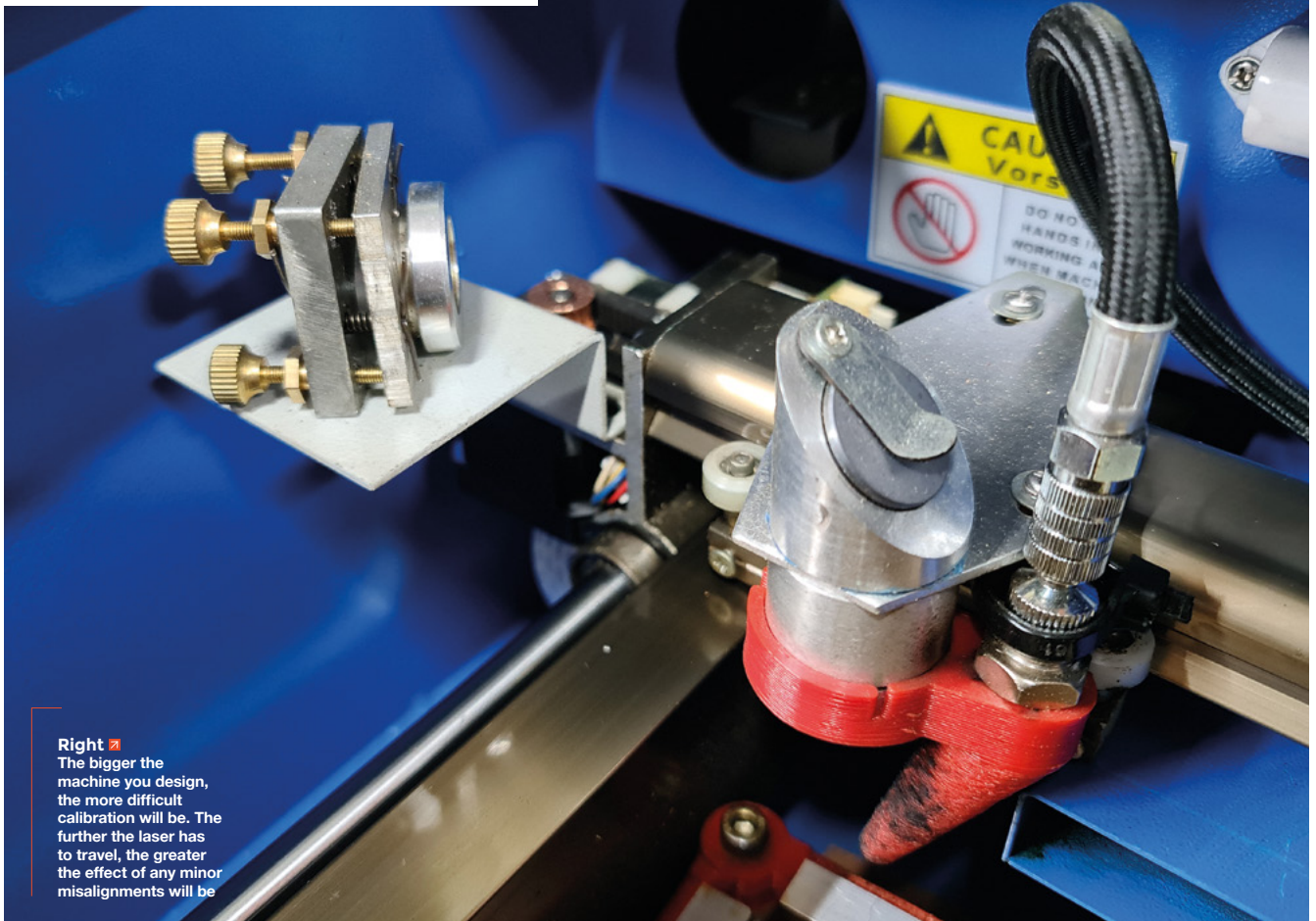
BIG HEAVY CABINET


The first thing you'll build is the enclosure. It sounds boring, but the chassis of the laser is important for a number of reasons. The laser cutter produces smoke and fumes when cutting, as well as invisible laser radiation. You can't just cobble together a case out of MDF and hot glue, and hope it'll turn out OK. You need to get your basic dimensions figured out and draw a plan. Let's assume that you want to build a →

STAYING SAFE



A laser cutter is probably one of the most dangerous tools you'll ever use. Invisible laser radiation can disfigure or blind you permanently in a fraction of a second. On a fully working machine with safety interlocks in place, the dangers are minimised, but not entirely eliminated. On a partially constructed machine, or a machine that's undergoing testing, the dangers can be significant. Even a diffuse, reflected beam can cause permanent blindness. Treat a laser with the same respect you would afford a loaded weapon. Always wear laser safety goggles when working with the laser, and be aware of the risks from high voltages and reflective surfaces. Don't undertake a laser cutter upgrade if you're not confident in your abilities to do so safely



Right  The bigger the machine you design, the more difficult calibration will be. The further the laser has to travel, the greater the effect of any minor misalignments will be

ANATOMY OF A LASER CUTTER

Laser cutters work by focussing a powerful laser beam onto a material to vaporise it. The power of the laser determines whether the beam will cut through or engrave into the material. The laser beam is generated by a large glass tube that is fixed in place inside the housing for the laser cutter. The laser tube uses very high voltages and needs liquid cooling to generate the laser beam. It's impractical to move the laser tube around like the hot-end of a 3D printer, so the cutter uses mirrors and lenses to direct the beam into position. The mirrors are mounted on an X-Y gantry, similar to an old-style paper plotter. The beam is focussed onto the material being cut using either a fixed or adjustable lens. If the lens is fixed, then the machine usually has an adjustable cutting bed that can be positioned to set the material at the optimum distance from the lens. Some more expensive machines have focusable heads and fixed position beds.

The X-Y gantry motors, bed motors, and laser power supply are all connected to a control board that normally uses either G-Code or DSP signals to control the laser. The price of control boards can vary between £15 and £500, depending on which version you choose. The power supply for the laser usually has connections for safety interlocks, which will disable the laser when a fault is detected. Faults can include a lack of water flow in the cooling system, the door to the cutting chamber being open, or excessive heat in the cutting chamber.

The laser will usually have an extraction system to remove smoke, and sometimes an air assist and smoke assist system as well. All of these systems will be covered in detail in the future, but for now, you are going to focus on the most fundamental of the components – the chassis itself.

laser cutter that can work with sheets of stock around 600 mm x 900 mm in size. That means you'll need to build an enclosure large enough to fit that size stock inside, and also have room for the motors, mirrors, extractors, air tubes, laser tube, and all of the other gubbins that live inside a laser cutter. The power of the laser will also affect the size of the machine, because laser tubes are long. If you want a higher power laser, you're going to need a longer machine. Taking this into account, you'll probably end up needing an enclosure around 750 mm x 1200 mm x 300 mm to house the laser and gantry, each of which will be housed in their own chamber. The main chamber will be the cutting chamber, where the material sits and the lens zips about on its gantry. Behind that will be a separate area where the laser tube itself sits. These chambers house the most dangerous parts of the machine, and should →

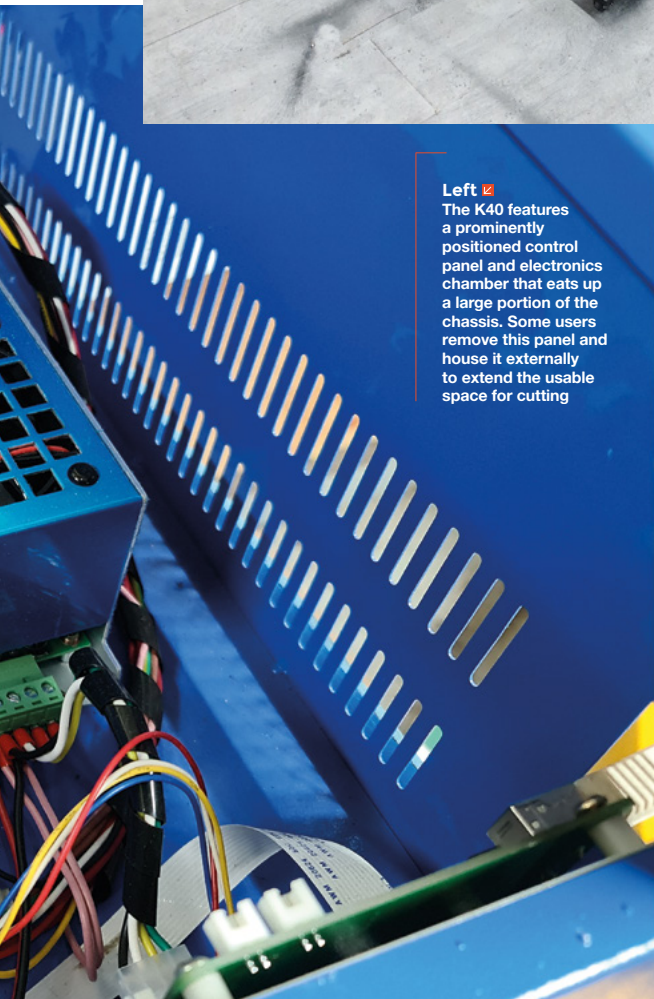


Above Note that the K40 control board uses a different type of M2 nano-controller that isn't compatible with either G-code or DSP standards





Left ♦ Big metal enclosures are heavy. Fit locking wheels to the bottom of the chassis as soon as possible, and take care while you're manoeuvring them around



Left ♦ The K40 features a prominently positioned control panel and electronics chamber that eats up a large portion of the chassis. Some users remove this panel and house it externally to extend the usable space for cutting

MATERIAL CHOICE

To make the enclosure as safe as possible, you really only have a few options when it comes to material choices.

Steel is the most common option. It's low-cost, resilient, and fireproof. It can be easily worked if you have the right tools. Steel is heavy, which can be viewed as an advantage or a disadvantage. If you crave portability, the weight of the machine will matter. A steel enclosure for a laser cutter can weigh around 60kg before parts are even installed, which is quite a weight to push around or try to lift. If you're not too concerned about portability, the extra weight of steel will give the machine some inertia, which might help soak up vibrations from the head moving at higher speeds.

Aluminium is another possibility, although it's expensive and less resilient than steel. Aluminium is easy to work with, can be shaped and bent very easily, and is lightweight – roughly 2.5 times lighter than steel. This might be an advantage if you want a particularly portable machine, but the cost difference and increased ductility could be prohibitive. In the event of a fire, aluminium would fare less well than steel.

Aluminium Composite Sheeting (often referred to by the trade name DIBOND® or simply ACS) is a lightweight sandwich of two thin aluminium sheets, with a plastic core. ACS is rigid and easy to work with, but most don't offer great protection in the event of a fire. Special types of ACS, such as DIBOND® FR (the FR stands for Fire Rated) have a better fire rating. Although ACS is relatively expensive, it's often used for printing signs and site hoardings, and it can be found cheaply second hand if you don't mind some minor dings or a printed surface. If you do decide to use ACS, the fire-rated version is strongly recommended for safety reasons.

QUICK TIP

If you're building big, remember to measure the doorway. Don't end up welding yourself into a corner. Or a shed.

POP POP

Rivets, more specifically pop rivets or 'blind' rivets are a type of permanent fixing that gets frequently overlooked. Pop rivets are extremely useful for joining multiple layers of dissimilar materials, and are particularly effective for joining thin sheets of metal. Unlike nuts and bolts, a rivet can't be removed once it's been fitted, unless you're willing to drill it out. While it doesn't hold as firmly as a welded joint, rivets are a cheap, effective way of joining things together. They are a good option for joining the thin sheet metal of an enclosure to the thicker metal chassis. All you need to get started is a drill, some rivets, and a rivet gun.

Rivets consist of two parts: the body and the mandrel. The body of the rivet is the part that gets squashed into place when the rivet is pulled into a hole. The mandrel is a metal shaft that pulls the body inwards towards the gun, and then snaps off at the base once the body is fully expanded. The only thing a rivet gun does is to grip the mandrel and pull it until it snaps.

Choosing the right rivet for the job is a matter of selecting the right width and length of rivet body, shaft size, and material. Generally speaking, the thicker the rivet you choose, the stronger the joint will be. To get the length of the rivet you need, you multiply the total thickness of the materials you are joining together by 1.5 times the diameter of the rivet body you are using. Choosing a rivet that's too long will probably work, but the resulting joint won't be very strong.

When it comes to the choice of material, rivets are available in a wide range of metals. It's wise to match the metal to the materials you're joining, because mismatching materials can corrode over time, resulting in premature failure. If you're joining a very soft material like leather or fabric, you'll need to add a washer to the back of the rivet, and possibly use a rivet with an extra-large flange.

If you want a watertight or airtight joint, you can get special closed rivets that have a fully sealed body. To get a fully waterproof joint, you'll need to exploit the sphere of influence (SOI) of the rivet. The SOI is the distance around a rivet that the jointed materials are affected by that rivet. The diameter of the SOI is generally four times the diameter of the body of the rivet. So, if you have a 4 mm rivet, the SOI will be 16 mm diameter. If you position your rivets so that the SOI intersect each other and you are using watertight rivets, then your joint should be watertight. For the best results, a double row of rivets with offset holes will provide a more reliable seal. If you're feeling disconcerted by how many rivets that would need, you can use a combination of rivets and silicone rubber in less critical applications.

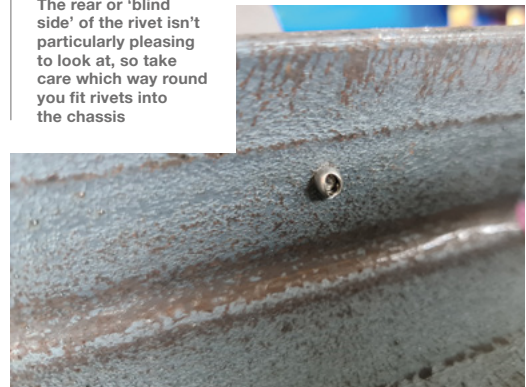
Right

A manual rivet gun will work fine, but it takes a fair amount of strength to pull the trigger. On most manual guns, you'll need to pull the trigger multiple times for each rivet. You will get tired arms after fitting a few rivets. You can pick up an air riveter for a very reasonable price, and they are worth every penny if you have access to a compressor. If you have access to a battery drill or impact driver, you can get riveting attachments for under £20



Above & Below

The front and rear of a rivet have a very different appearance. The rear or 'blind side' of the rivet isn't particularly pleasing to look at, so take care which way round you fit rivets into the chassis



be constructed with safety in mind. Outside of these chambers, you can create other areas to store the electronics and cooling systems for the laser. Most conventional laser cutters use the space beneath the main enclosure for this.

You need to pay particular attention to the techniques you use to join the parts of your enclosure together. The enclosure you're building needs to be close to airtight so that you can control the flow of air using fans and extract smoke without it leaking out in unexpected places. You also want to make sure that the enclosure is rigid and won't flex out of shape if something heavy is dropped inside.

A good strong construction technique is to use 3 mm thick 50 mm x 50 mm mild steel angle to build a rigid chassis, and then use thinner 1 mm steel panels to fill in the sides of the enclosure. Using different

LOOKING IN

A noticeable feature of the K40 laser is that it has a big window in the hatch so that you can see the machine in operation. From a safety perspective, this window is a weak point. The plastic used in the K40 is laser resistant, but a fire in the chamber would quickly melt the plastic sheet, and a sustained reflected beam could still burn through the plastic. It's actually much safer to have a windowless enclosure, and monitor the chamber using sensors and a camera.



WELDING

Welding is a huge topic to deal with, and even a series of articles wouldn't be enough to explain how to weld properly. For this project, you don't really need to be a good welder. So, if you're working with some 3mm steel angle, you might want to take the opportunity to prime yourself with a few YouTube videos and get some practice in. Be aware that welding is toxic, messy, and flammable, so take suitable precautions and work outside if you can. You should be able to tackle 3mm steel joints quite easily with a fluxed MIG welder or a stick welder, both of which are reasonably cheap to buy, along with a helmet and some good gloves. A MIG welder, in particular, is regarded as 'the hot glue gun of metal working' because it's easy to get the hang of, and while you might not make the best welds when you start out, you can usually fix problems by grinding away the evidence and slapping some paint over the top of it.

Welding 1mm sheet metal is a difficult proposition, and if you're not a good welder with a decent machine, it's not really worth trying. Thin metal is notoriously difficult to weld, and will distort very easily while welding. It's also very easy to blow through the metal and create a hole if you accidentally apply heat for even a second too long.

Above ♦ If you've got a gap to fill with a welder, copper sheet comes in very useful. Weld doesn't stick to the copper very well, so you can use the copper sheet as a sort of scaffold behind the area you need to fill. The copper can also reduce the possibility of blowing through with the welder on thin material

“ You need to pay particular attention to the techniques you use to join the parts of your enclosure together ”

material thicknesses like this reduces the overall weight of the box, but doesn't really compromise the overall strength of the enclosure. The 3mm steel sections are easy enough to weld together, even if you're relatively new to welding. However, the thinner 1mm panels can pose a problem to even an experienced welder, depending on the tools that they have available to them. For thinner metal panels, rivets might be a better solution.

Clean and paint your chassis with a machine enamel, to give it some protection from corrosion, and then get ready to start moving more exciting parts of the mechanism inside. □

DON'T MISS THE **BRAND NEW** ISSUE!



SUBSCRIBE FROM JUST £5

- **FREE!** 3 issues for the price of one
- **FREE!** Delivery to your door
- **NO OBLIGATION!** Leave any time



**WORTH
£20**

**FREE
PI ZERO 2 W***

With your 12-month subscription to the print magazine

magpi.cc/12months

* While stocks last

Buy online: store.rpipress.cc

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG
108



SMÔL

A dev board for when you're short on space

PG
112



TROIKA

A multi-tool that also writes

PG
102



BEST OF BREED

The finest electronic conference badges



ONLY THE BEST

The world of conference badges

Electronic, wearable, hackable, and sometimes a bit scary

By Marc de Vinck

@devinck

When asked to cover the topic of electronic badges for this month's Best of Breed, I thought, 'Sure that's easy! I've been to countless conventions, cons, and

festivals, and know the topic well. If I remember correctly, the first experience I had with an electronic badge was in 2008 at The Last HOPE, hosted by the Hackers On Planet Earth, and sponsored by 2600: The Hacker Quarterly.

The badge was part of the AMD project, or Attendee Meta-Data project. It had an active RFID system that allowed attendees to visualise their locations and interactions at the conference, even the late night, more like all-night, Club-Mate sipping social gatherings. It was interesting to see your friends mapped throughout the event in the Hotel Pennsylvania, but it was also a bit scary to see where things were going as far as privacy. Remember, this is the time of the iPhone 3G, and worrying about your location privacy wasn't really a 'thing'; at least it wasn't for me.

OK, back to the reviews.

Reality quickly hit me. After I started researching some of my favourite badges, I realised that they are typically only available at the events, and in limited quantities. If you're lucky, you can sometimes purchase them afterwards if they happen to have extra stock. However, they are also typically open-source, which means the files should be available if you want to make one yourself. But that's not something I can do for these reviews, as it would take months to secure all the parts, and that's if there wasn't a massive supply chain issue right now.

So now what? Well, the answer is, I'll just review a collection of badges that are either available online as kits, assembled, or have the files available for you to source the parts and build one for yourself. But that plan also ended up being challenging. Most of the badges that I found online did appear to have links to the schematics and code, but after looking a bit deeper, many were 404 or outdated. I can't blame the makers of the badges, as they were designed for a specific event, and maintaining the code and schematics in a database or website can be challenging. Thankfully GitHub has made it a lot easier! So, let's look at some of my favourite electronic badges.



MakeltHackin DEF CON 29 Electronic Badge vs Retro Computing badge

MAKEITHACKIN ◆ \$50.00 [tindie.com](https://www.tindie.com)

HACKADAY ◆ hackaday.io

When most people think of conference badges, they think of those plain laminated tags, and nothing special. But if you are asking someone in the DIY electronics community, they will most likely think of DEF CON, and all the different DEF CON badges. It must be one of the largest conferences, and you can always find lots of custom badges.

The MakeltHackin DEF CON 29 Electronic Badge is interesting and easily programmable. It features an SAO port, ATtiny85, OLED, NeoPixels, and a light sensor. In addition to the standard electronics, it also has both an 8.2MHz and 58kHz security tag. You can easily program the ATtiny to display custom graphics, and interact with the light sensor and NeoPixels. It's

not too advanced, and it's simple to program. And best of all, the creator has included a full list of parts, schematics, and code, so you can build your own.



Left ◆
A badge or a useful dev board?

Below ▣
PRINT "HELLO WORLD"

The Retro Computing badge must win the prize for most buttons on a badge. I've seen some badges with more parts, like ones that have large LED matrix displays, but I never saw one that had a fully integrated QWERTY keyboard and more!

The badge has a full 55-key keyboard, an RGB TFT LCD screen with resolution of 320 × 240, and a PIC32MX370F512H microcontroller with 512/128kB of internal program/data memory. Yes, it's like wearing a small computer around your neck. All attendees at the Hackaday Conference in Belgrade received one, and I'm a little jealous.

The badge runs a BASIC interpreter and emulates a Z80 computer with the CP/M operating system. It's



ready to be hacked, or at least programmed and enjoyed. The creators have done an excellent job archiving the code, schematics, and everything you need to build one for yourself. →

VERDICT

DEF CON 29
Electronic Badge

Well-
documented and
easily hackable.

9/10

Retro Computing
badge

A personal
computer badge!

10/10

eChallengeCoin

BRADÁN LANE STUDIO ♦ \$40 | tindie.com

Not a conference badge per se, but it has a lot in common with hackable conference badges, so I thought I'd include it in this roundup. I think there are a lot of interesting features that might inspire your next conference badge build.

The eChallengeCoin Redux by Bradán Lane STUDIO is an electronic puzzle, similar to some

conference badges I've come across online. To learn how to use it or, in this case, how to solve the embedded puzzles, you connect to it via UART. Now you can read the story and explore the various challenges and how to solve them. The creator hasn't released the full code, as they feel it would ruin the challenge, but they have made some parts available under the MIT license. I'm looking forward to the release of new ones each year.



Left ♦
Put yourself
to the test

VERDICT

eChallengeCoin
An interesting
challenge awaits.

9/10

BalCCon 2016 Badge

HACKADAY  hackaday.io

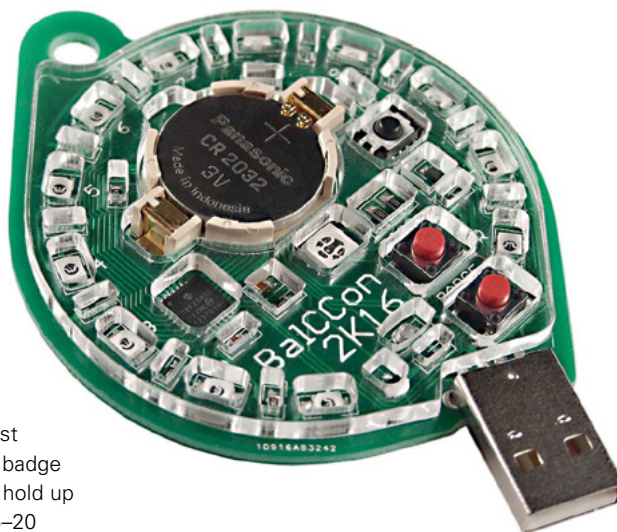
The BalCCon 2016 Badge is one of my favourite badges for one simple reason; it has some practical uses that I need! The badge was made for the Balkan Computer Congress


hacker conference organised by LUGoNS, a Linux Users Group of Novi Sad and Wau Holland Foundation from Hamburg and Berlin.

The badge has a few functions. First, it has twelve LEDs in various colours and one RGB LED. It displays predefined patterns of light and, when two badges 'see' each other, they can synchronise to the same pattern. I've seen a few badges that act similarly, and they are always fun to use.

The badge also features a built-in universal remote control for turning off TVs. Handy, but not the reason I need one. The last feature is what I need. You can also use this badge as a password manager. It can generate and hold up to ten independent passwords, each with 15–20

characters. And since it acts as a USB HID device, it's as simple as plug and play. I could do with a few of these badges! →



Left  Synchronise your lights and your passwords


VERDICT

BalCCon 2016 Badge

A useful, multifunctional badge.

9/10

ADAFRUIT CIRCUIT PLAYGROUND LANYARD

ADAFRUIT  \$1.95 | adafruit.com

If you are looking to download the files from one of the mentioned badges, you'll most likely want a lanyard to go along with it. Adafruit has a fun, inexpensive, and high-quality lanyard that's perfect for a high-tech badge. It features the Circuit Playground friends and special characters designed by artist Bruce Yan. Pick one up for your next con.



Open Hardware Summit badge

HACKADAY hackaday.io



SH Park, along with Screaming Circuits, produced this beautifully purple and functional electronic conference badge for the 2018 Open Hardware Summit. The

board, designed by Alex Camilo, is based on a concept from ESP trINKet by Mike Rankin, and features an ESP32 microcontroller with built-in WiFi, and ePaper display.

An interesting feature is its ability to update the display via your phone, tablet, or laptop. That makes it a lot easier and fun. I would assume it also promotes more changes and interactivity throughout the conference. It took some digging around, but I did eventually find a link to the project page:

hsmag.cc/OSHWABadge, and it does, in fact, have links to GitHub, so you can still make your own, or, at the very least, learn more about the design. □

Below ♦
A wifi-connected name tag

VERDICT

Open Hardware Summit badge

An easy to use digital display

10/10



SKULLY

NERFHAMMER ♦ \$40 | tindie.com

Maybe not totally on theme with this roundup, as it doesn't have the typical hackable accessories and ports, but I just couldn't skip the Skully by nerfhammer. It's a beautiful, skull-themed PCB that features 18 backlit amber LEDs around the perimeter that display a simulated flame pattern. Grab one while you can!



THE OFFICIAL Raspberry Pi Beginner's Guide

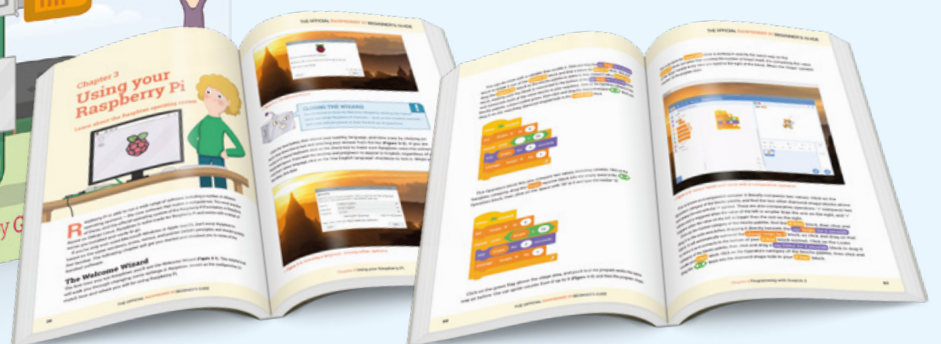
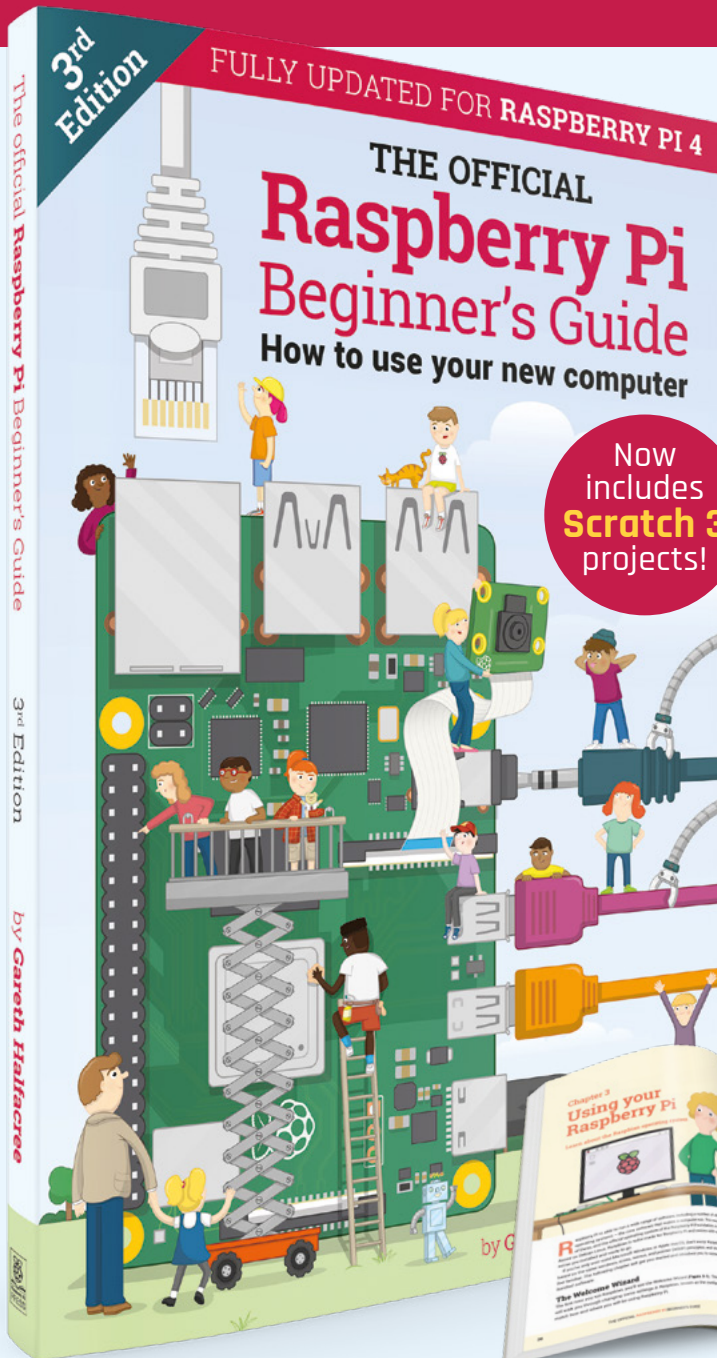
The only guide you
need to get started
with Raspberry Pi

Inside:

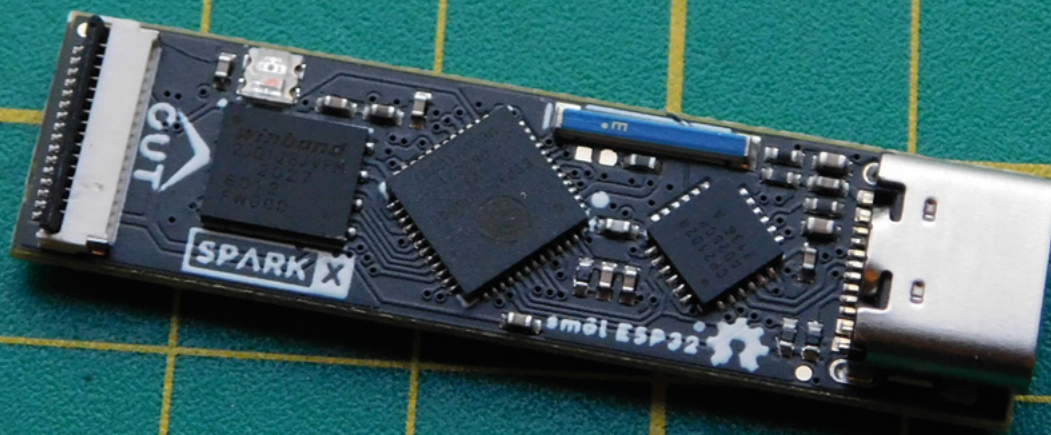
- Learn how to set up your Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch 3 and Python languages
- Create amazing projects by connecting electronic components to Raspberry Pi's GPIO pins

Plus much, much more!

£10 with **FREE**
worldwide delivery



Buy online: magpi.cc/BGbook



SparkFun smôl

Can good things come in small packages?

SPARKFUN ♦ from \$17.95 | sparkfun.com

By Ben Everard

[@ben_everard](https://twitter.com/ben_everard)

You've probably guessed from the photos and the title what the headline feature of this board is. It's small. Very small. Actually, it would probably be more accurate to describe it as narrow. There are plenty of boards that are this short (the smôl is 40.6 mm long) or shorter, but we've not come across another development board that comes in under 11 mm wide. However, it would be a bit of a mistake to consider this a one-trick pony.

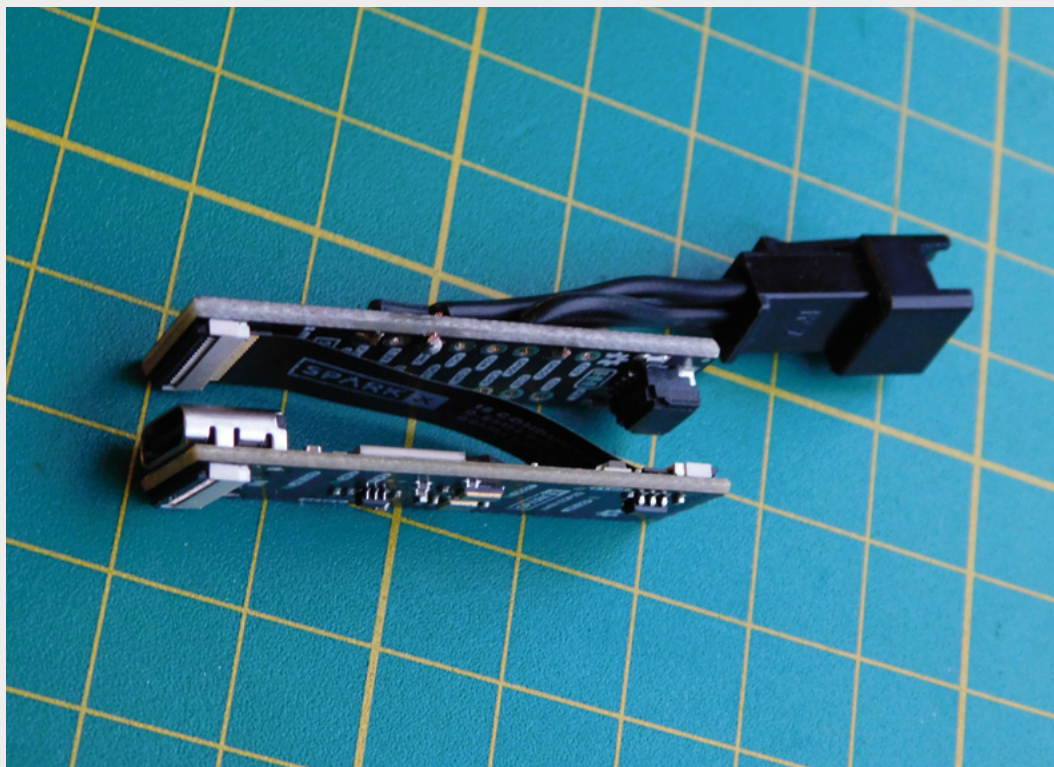
At the moment, there's only one processor board in the smôl ecosystem – the ESP32. This does use quite a bit of power when running, but it has an efficient deep sleep which lets you get down to about 0.5mA. The current draw is critical because, if you're running off battery, it doesn't matter how small your PCBs are if you need to have a whopping great battery to keep everything running. 0.5mA is a decent sleep current, but if you want to run for a

long time, SparkFun has a smôl Power Board LiPo which lets you use and charge LiPo batteries – but it also has battery management which will turn your main processor off and on as and when you need it, and this can drop your current down to around 6 microamps. That's 0.006mA. At this level, depending on what you're doing and how much you need to turn the processor on, you can potentially run for months off a small battery.

That, at least, is the idea of the board. How well does it work in practice?

The processor board packs quite a bit of computing power in such a small space – a dual-core ESP32 running at up to 240MHz with WiFi, 520kB of RAM, and 16MB of storage. However, it has very limited I/O connections. There's just a ribbon cable with 16 channels – this has connections for an SPI bus (with up to three chip selects), an I2C bus, and two GPIO pins. That's enough to connect with quite a wide range of hardware, but it's hard to access on the

Above ♦
Is this as narrow as a development board can go?



Left ♦ Our tiny LED controller is barely wider than the USB port and can be wrapped in heat-shrink to make an almost invisible controller

board itself. Fortunately, the ribbon cable connects to other boards in the smøl range, letting you zig-zag boards on top of each other, creating stacks that are narrow but may be quite tall. For example, the smøl header board breaks out all the connections, and the previously mentioned Power Board LiPo adds battery and power management. In addition to the ones we've spoken about, there are GPS and satellite communications boards.

The flex-connectors are a little fiddly to get in, but nothing too taxing.

Connecting them concertina-style means there's no bend on the cable once it's inserted, so we wouldn't expect it to wear out over time.

SOFTWARE STACK

Aside from the unusual form factor, the board is fairly normal to use. It's Arduino-compatible and works – from a software perspective – just as most other development boards do.

Just make sure that you have the board definition files loaded (see the Hookup Guide at [hsmag.cc/HookUpGuide](https://www.hackmag.cc/HookUpGuide) for details), and you can transfer your sketches. Additionally, there are Arduino libraries for the other smøl boards to give you easy access to the features.

The smøl ecosystem is probably not an ecosystem that you use because you want to – it's more expensive than the equivalent boards in other sizes and a bit fiddly to put together. It's an ecosystem that you use when you need to. There simply isn't another setup like it that lets you jam processing

power into such a small space.

The project we've built to test it out is making a USB LED controller. The ESP32 board, together with the header board, make a small, solid combination that we

can heat-shrink over to be a small, solid package with a USB-C port on one end and cables coming out the other. The end result is more compact and professional-looking than other controllers we've made (but works exactly the same).

Smøl could be useful for wearables, as the narrow width would fit into a bracelet or necklace far more inconspicuously than larger boards. As well as being small, they're lightweight, which makes them a good choice for wildlife trackers. These boards could also be useful for hacking hardware where there's only a small amount of space for additional electronics. These are just a few of our ideas, but if you come across a project where you only have 11 mm of space to play with, then this could be the board for you. □

“ There simply isn't another setup like it that lets you jam processing power into such a small space

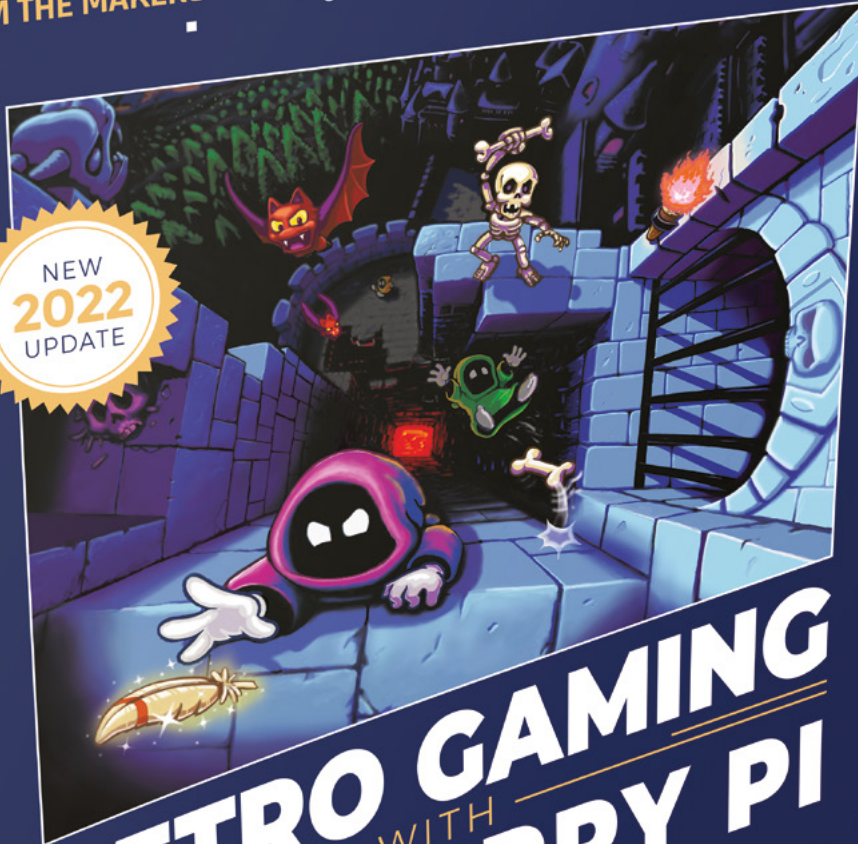
VERDICT

When you need a tiny controller, this is a great option.

9/10

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

NEW
2022
UPDATE

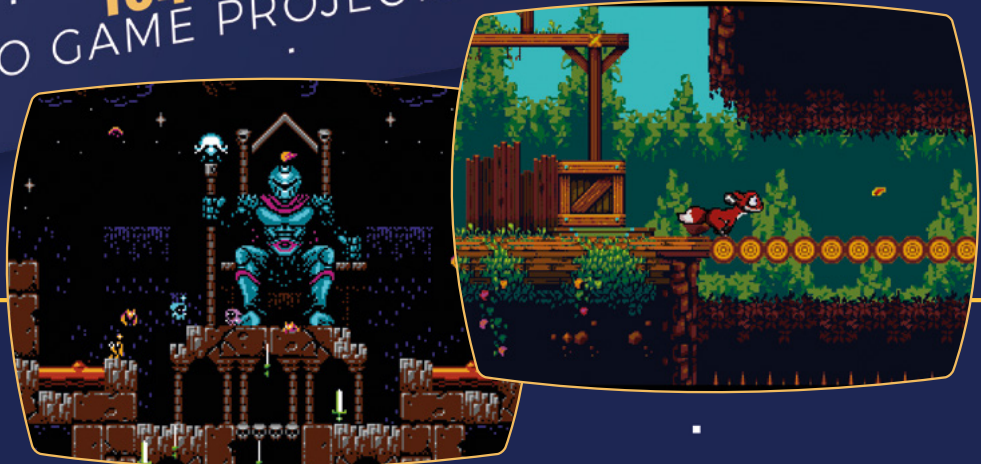


PLAY
& CODE
GAMES!

RETRO GAMING WITH RASPBERRY PI

2ND EDITION

164 PAGES OF
VIDEO GAME PROJECTS



RETRO GAMING

WITH

RASPBERRY PI

2ND EDITION

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

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



BUY ONLINE: magpi.cc/retrogaming



Troika multi-tool pens: Construction and Liliput

A toolkit that also writes

TROIKA ♦ £22.80 | pimoroni.com

By Jo Hinchliffe

🐦 @concreted0g



ften as makers and hardware hackers, we need to scribble the odd note, sketch an idea, or get organised with a 'to-do' list for our making time. While any old pen or pencil might suffice for this,

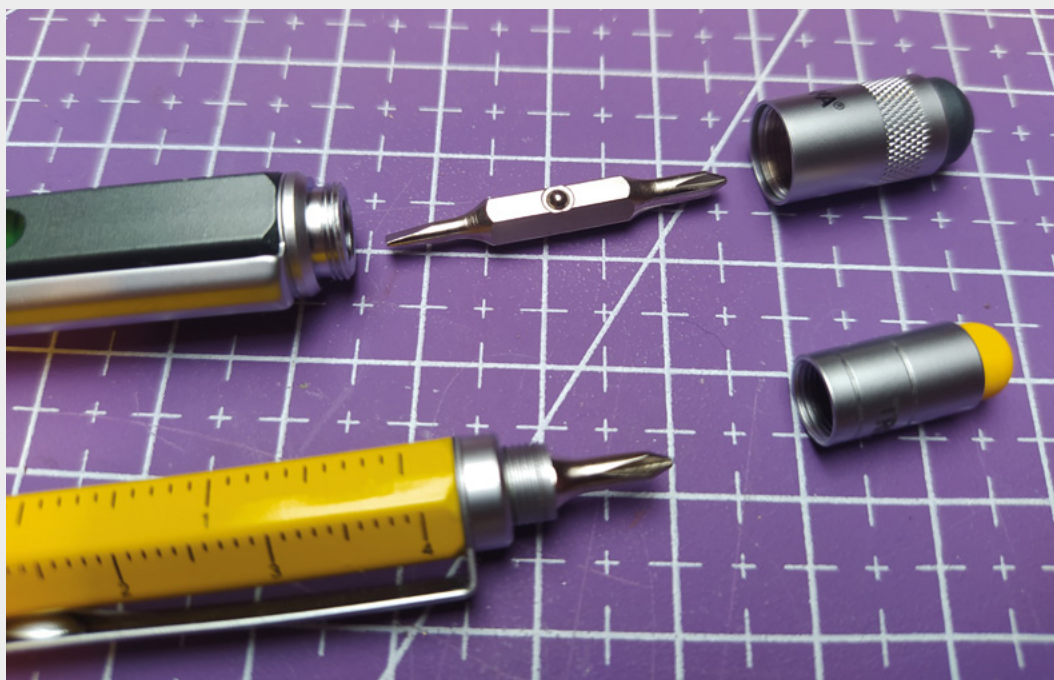
we decided to take a look at an increasingly popular range of pens that are designed to be used in projects and builds. Troika items are widely sold online from many stationery suppliers, but increasingly, maker suppliers are starting to stock them. We bought ours from Pimoroni, where it's far too easy to slip one of these pens into your shopping basket while you're shopping for other electronics supplies.

The two models we're looking at are from the Construction range: the larger Construction multi-tool and the Construction Liliput. There are some similarities between them both, but before we dive

into their features, we want to talk about their build quality. It's hard to convey how nice these pens feel. Both pens require a twist of the tip to deploy the ballpoint refill, and you can instantly feel the quality and precision in these pens. Both pens feel well-weighted in hand; it's a luxurious, quality feeling. For us, the actual writing experience comes above the other 'multi-tool' aspects of these pens, and we are pleased to report that they both write very nicely with an even flow of ink. Illustrators will be pleased to hear that it takes a reasonable number of strokes with either pen before there is a small build-up of ink at the ballpoint, common with any ball-point pen.

Both pens have hexagonal body tubes, which we like for multiple reasons. They are easy to grip and use, and they don't move when placed on a desk or other sloped surface – it's helpful in the workshop or lab that they remain wherever you place them. They

Above ♦
These pens look great and are built to a high quality; both feel good in hand



Left ♦
The Construction and Liliput pens have a capacitive touch tip and built-in screwdriver, with the larger pen also having a spirit level

Below ♦
Both Troika Construction pens came well-packaged; the Liliput also arrived with some refills

VERDICT

Both pens are excellent! We deducted one mark as the Liliput might not be ideal for a long writing session, and the Construction multi-tool model should possibly include one refill at its price point. However, these minor points don't detract from what are very nice writing tools.

9/10

both feature graduated rulers in metric and imperial on the body, which is handy for quick measurements. The Construction multi-tool also has some scale ratios, which is useful when working with scaled plans. Whilst we are talking scale, the Liliput is, obviously, small. It's not uncomfortably small to write with; however, it's possibly fair to say that it's not the one we would reach for if we had more than a few lines to write.

At the non-writing end of the pens, both feature a cap with a capacitive tip for use on touchscreens. These work well across a range of screens we tried them on, although they are quite a large stylus to use on smaller mobile phone screens. However, as an input device for a tablet or touchscreen-enabled laptop, they are great. Beneath this threaded cap, both pens have a reversible screwdriver bit with a flat-head bit and a Phillips bit. They again work well – the Phillips screwdriver bit is perfect for adjusting spectacle screws, including those found on many of our PPE glasses, so this is perhaps our commonest use case for them. The screwdriver bit has the familiar 4 mm hexagonal shank, so, if required, you could fit other compatible screwdriver bits from a different set.

Both pens feature a sturdy pocket clip that holds the pen securely in place. The size of the Liliput, combined with the clip, makes this really easy to carry in a shirt pocket – not an uncommon task – and also easy to carry in a trouser pocket without feeling uncomfortable. The final feature for the larger Construction multi-tool is that it has a spirit level incorporated into the body of the pen. The spirit level

is decent quality, and we've been surprised by how many times we've used it, from checking a camera rig is level to checking CNC machine beds, 3D printers, and more.

The Liliput costs around £10 and comes with three refills. The Construction multi-tool costs around £22 and doesn't come with refills. The Construction multi-tool, however, uses a common 'D1' refill, which is a standard refill and is available from a range of companies – this means you can explore different inks, tip sizes, and more. It's also worth checking out the range of colours and finishes available for these pens, but be warned, they all look very nice – we were tempted by all of them! □





issue

#52

ON SALE
17 FEBRUARY

RASPBERRY PI PICO
BUILDS

ALSO

- CIRCUITPYTHON
- 3D PRINTING
- PCB DESIGN
- LEDS
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe



Open-Source Leg

You can download all code, design files, and instructions for this leg from hsmag.cc/OSLeg. While it's not cheap (it'll cost approximately £22,000, though over half of that is machining custom parts), it gives you a limb that is infinitely customisable. The leg is programmable in Python from a Raspberry Pi 4, meaning you can get it to behave exactly as you want. While this version is primarily designed for researchers, its open design may filter down into consumer units in the future.

100,000+ NEW PRODUCTS ADDED IN PAST 90 DAYS

DIGIKEY.CO.UK

ENABLING THE WORLD'S IDEAS®

**FREE
SHIPPING**

ON ORDERS OVER
£33 OR \$50 USD*



Digi-Key®
ELECTRONICS

*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of \$18.00 USD will be billed on all orders of less than \$50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel