MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | May 2022 | Issue #54

Classic Computers Reborn

Make old computers with new technology

May. 2022
Issue #54 £6

Plus

Geeky Gardening
Tech to improve your plants

Slicing
Optimise your 3D prints

Analogue Output
Build a resistor DAC

LASERS CNC ROBOTS LEGO

Welcome to HackSpace magazine

Modern computers are a wonder of science and engineering. Highly skilled scientists discover ways of squeezing more and more components into the same amount of silicon. Highly skilled engineers find ways of making the components do ever more things. Highly skilled designers package all this up in ways that fit in with our lifestyles. The computers you use today are the result of nearly a hundred years of optimisations on how you can use electrons to compute.

However, like all optimisations, this process has taken something away. As utility, power, and convenience have advanced, understandability, tinkerability, and fixability have declined. There was a time when a determined person could completely understand a computer and the software that was running on it, but that's not possible anymore. The allure of classic computers is the allure of this ability to get to grips with a computer.

This issue, we're looking at ways of rebuilding classic machines from the past, and keeping the dream of understandable computing alive.

BEN EVERARD

Editor ben.everard@raspberrypi.com

PAGE 42

FREE PICO WHEN YOU SUBSCRIBE

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

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

GET IN TOUCH

hackspace@raspberrypi.com

hackspacemag

hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
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, Phil King, Mike Bedford, Ian Dunn

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION

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

SUBSCRIPTIONS

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

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

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

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

ISSN: 2515-5148.

Contents

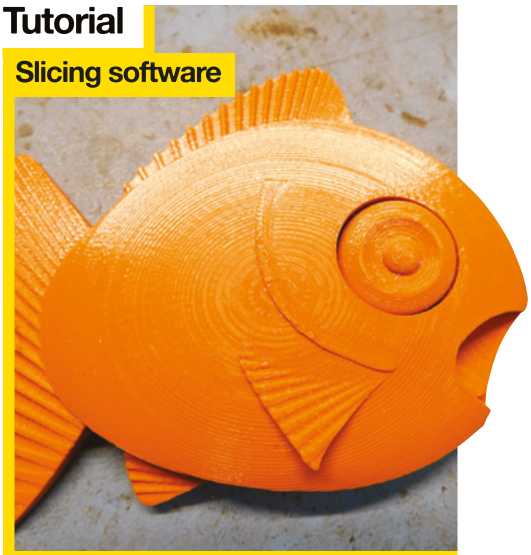
100

Cover Feature

32

Tutorial

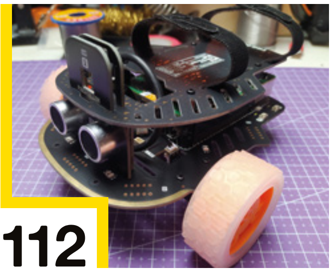
Slicing software



94 Discover the slicer options that can make or break your prints

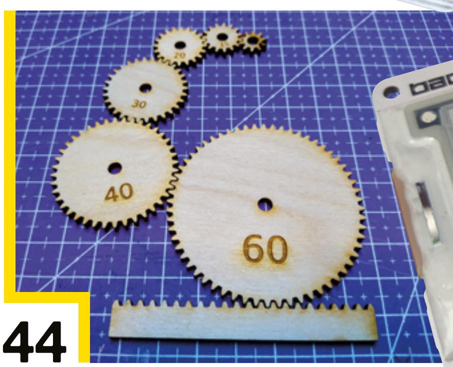
18

62



112

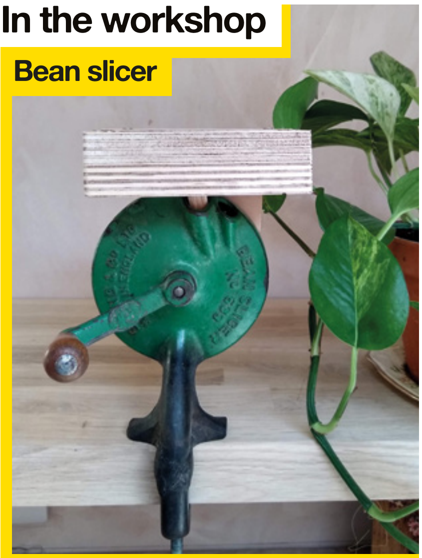
106



44

In the workshop

Bean slicer



Interview

Jude Pullen

110

67 FORGE

99 FIELD TEST

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

Track Link Calendar

By Shiura

hsmag.cc/TrackLinkCalendar

Months are annoyingly irregular. They come in 28-, 30-, 31-, and sometimes even 29-day lengths, which means a typical calendar will always have some surplus days hanging off the edges (open Google Calendar to see what we mean).

Shiura has solved this problem with a mechanical calendar, which he says has several advantages over paper/screen-based alternatives. "It can display all four types of months from 28 to 31 days… When the current month is selected by the top wheel, the day of the week is automatically adjusted. Setting is necessary once a year, on 1 March. Setting is valid from March to February in the next year."

We'll admit that this hadn't seemed like a problem to us before, but now we think about it, it's really annoying, and this project is a brilliant workaround to our flawed system.

Right What are days for? Where can we live but days?


OCT
FEB
NOV
MAR
AUG
7
14
21
28
1
8
15
22
29
2
9
16
23
30
3
10
17
24
4
11
18
25

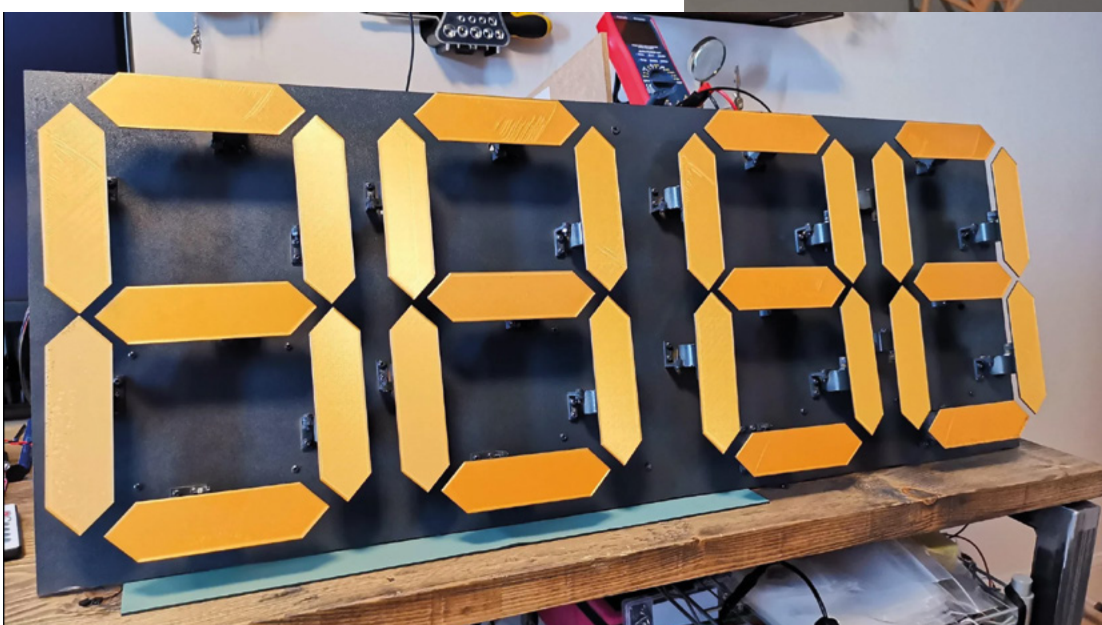
Seven-Segment Display

By David McDaid

hsmag.cc/StepperMotorDisplay

Electromechanical seven-segment displays look amazing, but not everyone is in love with the click-clack sounds they make. David McDaid decided to solve this problem by getting rid of servos and replacing them with stepper motors. In his own words:

"The goal here was two-fold; create something stylish that would look good when hung up on a wall, and ensure that all movements were as quiet as possible! I was inspired by some great existing projects (see link, above), but most had used noisy servos. Using my idea to retrofit silent stepper drivers from 3D printers, I built, tested, and coded my clock completely from scratch over two to three months of my spare time."



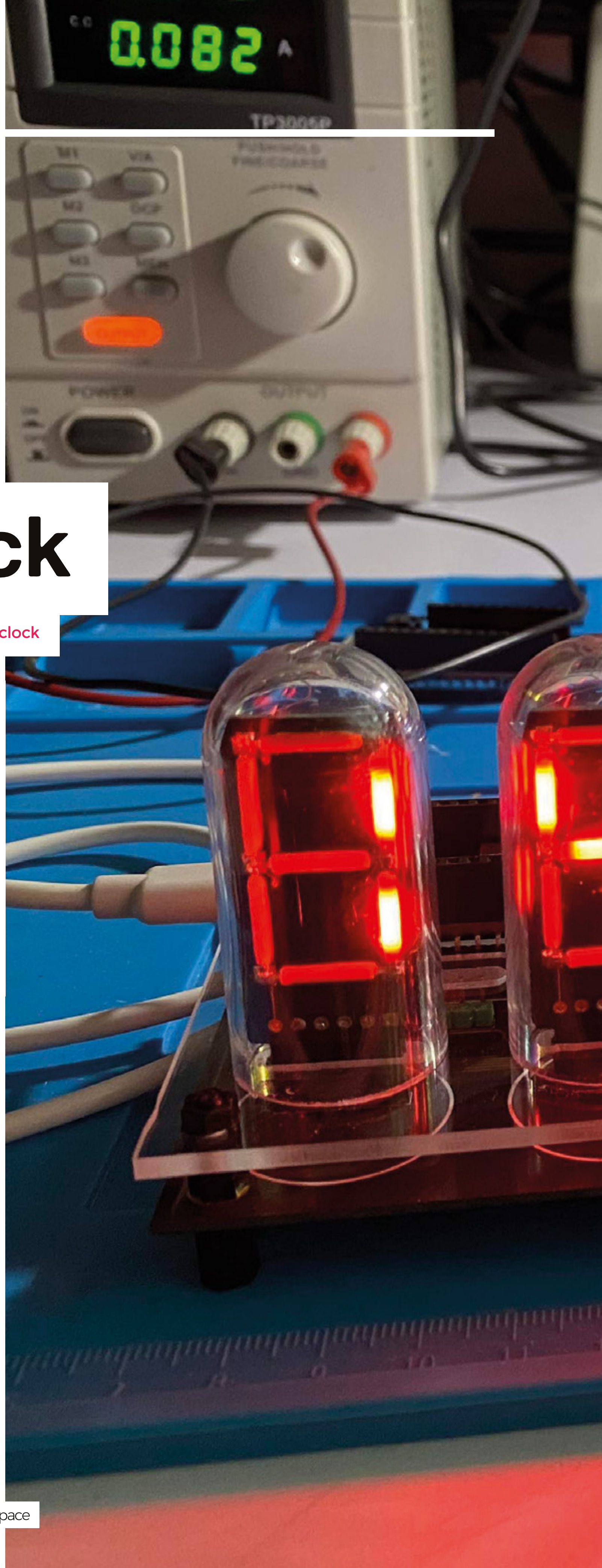
Right
To add a bit of visual interest, David added LEDs to the back panel, so the clock stands out on the wall

Filament clock

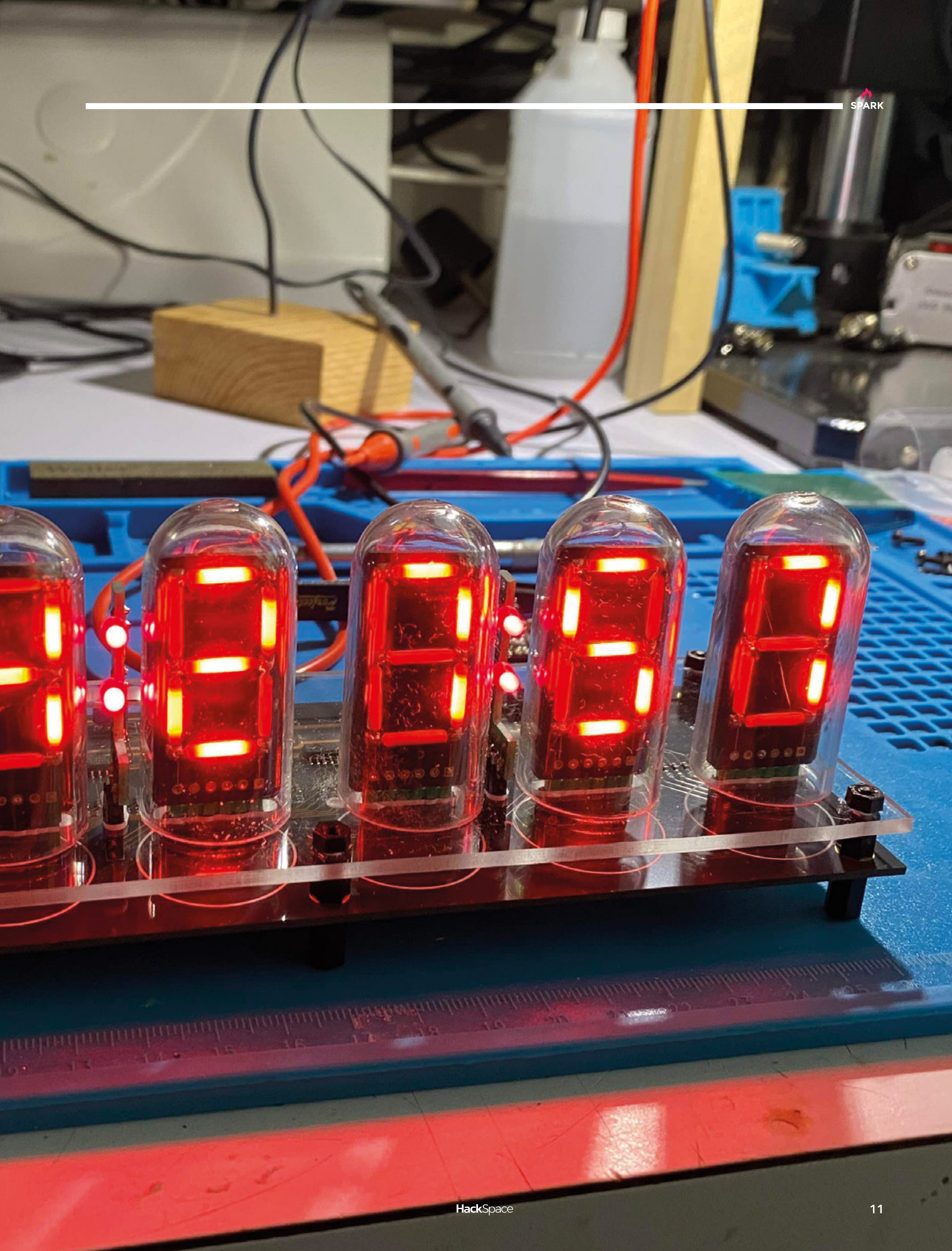
By Jay Hamlin

hsmag.cc/LED-filament-clock

This month, Ben tried and failed to produce a Nixie-esque clock, using layers of LED filament to display ten numerical digits. He should simply have made a seven-segment display, as Jay Hamlin did with this beautiful creation. It uses an Adafruit HUZZAH32 Feather board to send data to four seven-segment displays, with each segment being comprised of a length of LED filament (the same material used in the Edison-style bulbs that are so trendy at the moment). The glass tubes just add to the aesthetic.



Right To find out how *not* to build an LED filament clock, turn to page 62



Infinity Macropad

By **Infinity Workshops** hsmag.cc/InfinityMacropad

This Macropad by Infinity Workshops provides a frankly ridiculous amount of input capability. It uses 20 individual switches, three rotary encoders, and a joystick, held together with 20 switching diodes and a Raspberry Pi Pico. This device is overkill for sending emails and checking Facebook, but for video editing, we can see it saving the right user an awful lot of time.

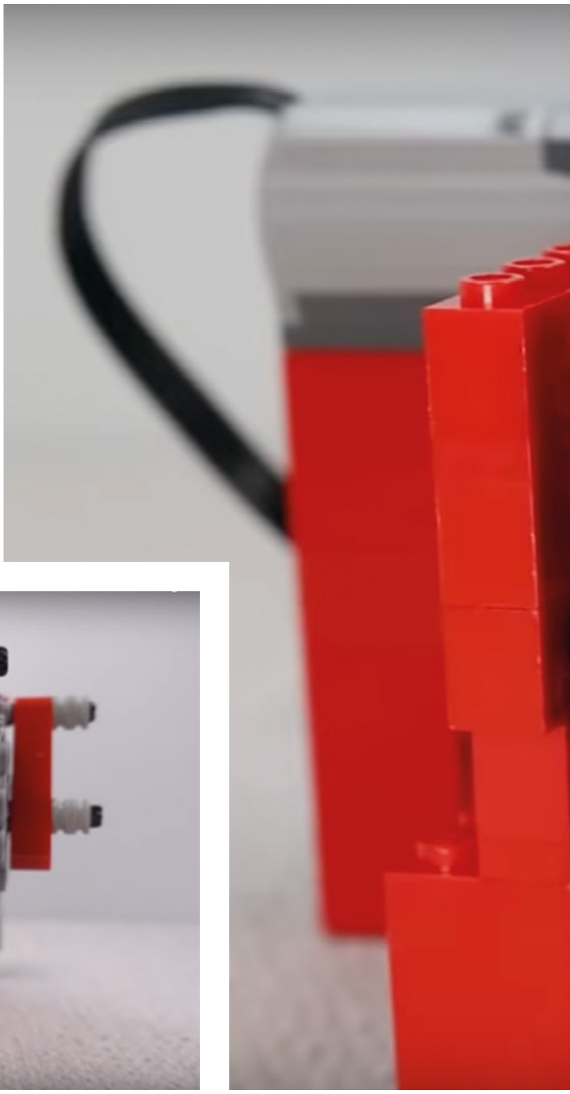
Below Don't like the interface for your favourite software? Build your own!

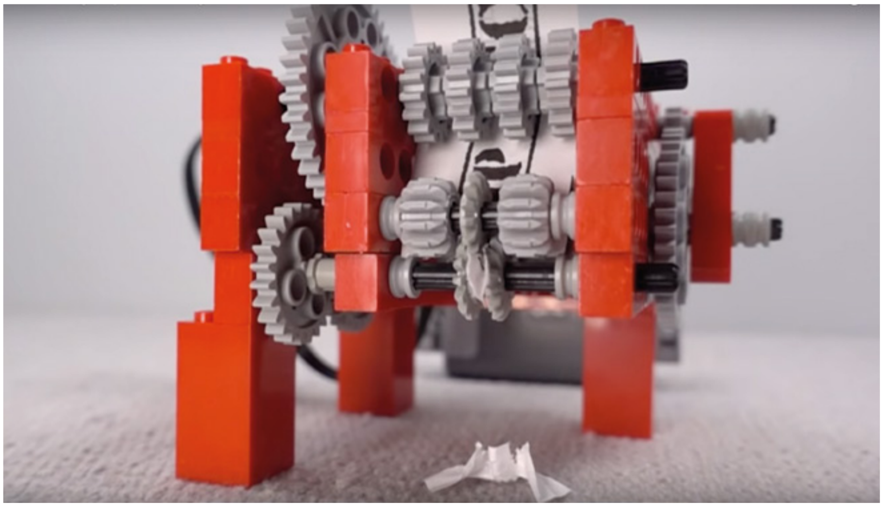
LEGO paper shredder

By **Brick Experiment Channel** hsmag.cc/LegoShredder

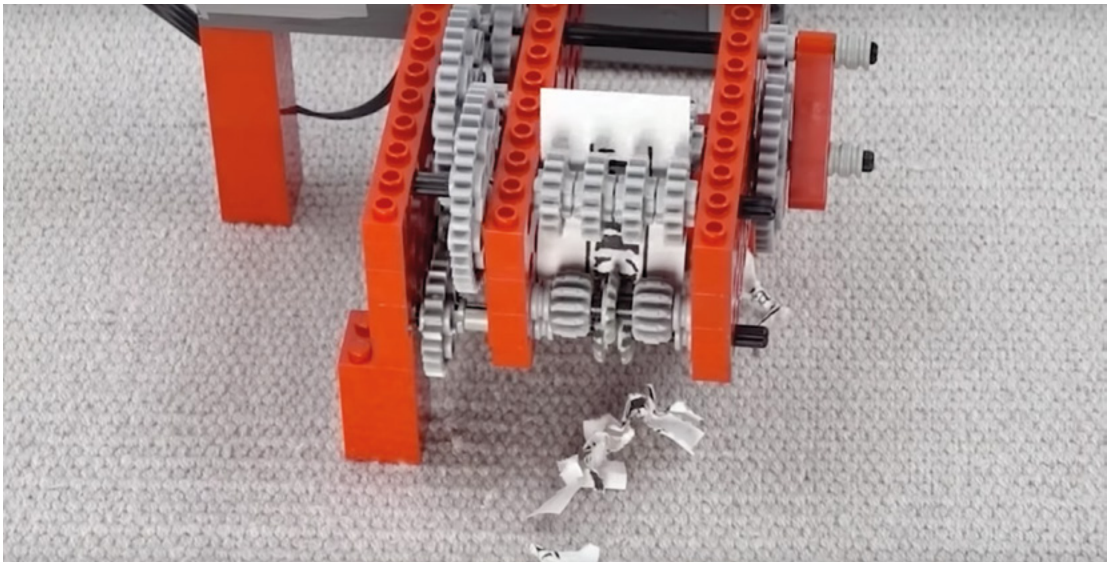
The normal goal when using gears in a build is to transfer power smoothly and efficiently. This build, by the brilliant Brick Experiment Channel, aims to subvert that. By purposefully misaligning the gears in this LEGO-built shredder, the device grabs the paper and shreds it as it's going through the mechanism.

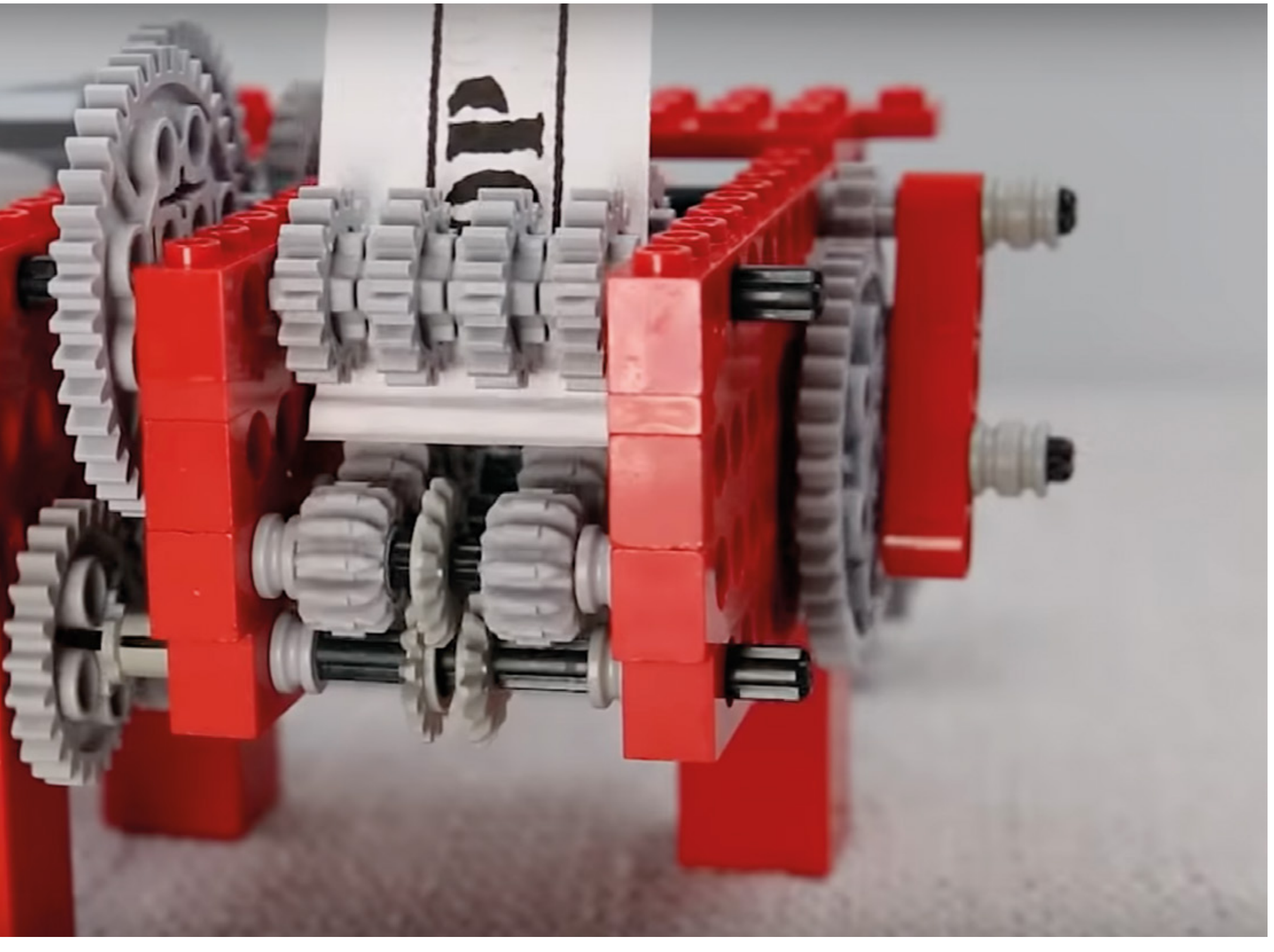
Early versions of this device merely crumpled the paper – watch the video to follow the build process, which concludes with a fully functional device.





Right If you've read the gears article on page 44 and want to get stuck into something semi-practical, try this project on for size





Musical PCB Tesla coil

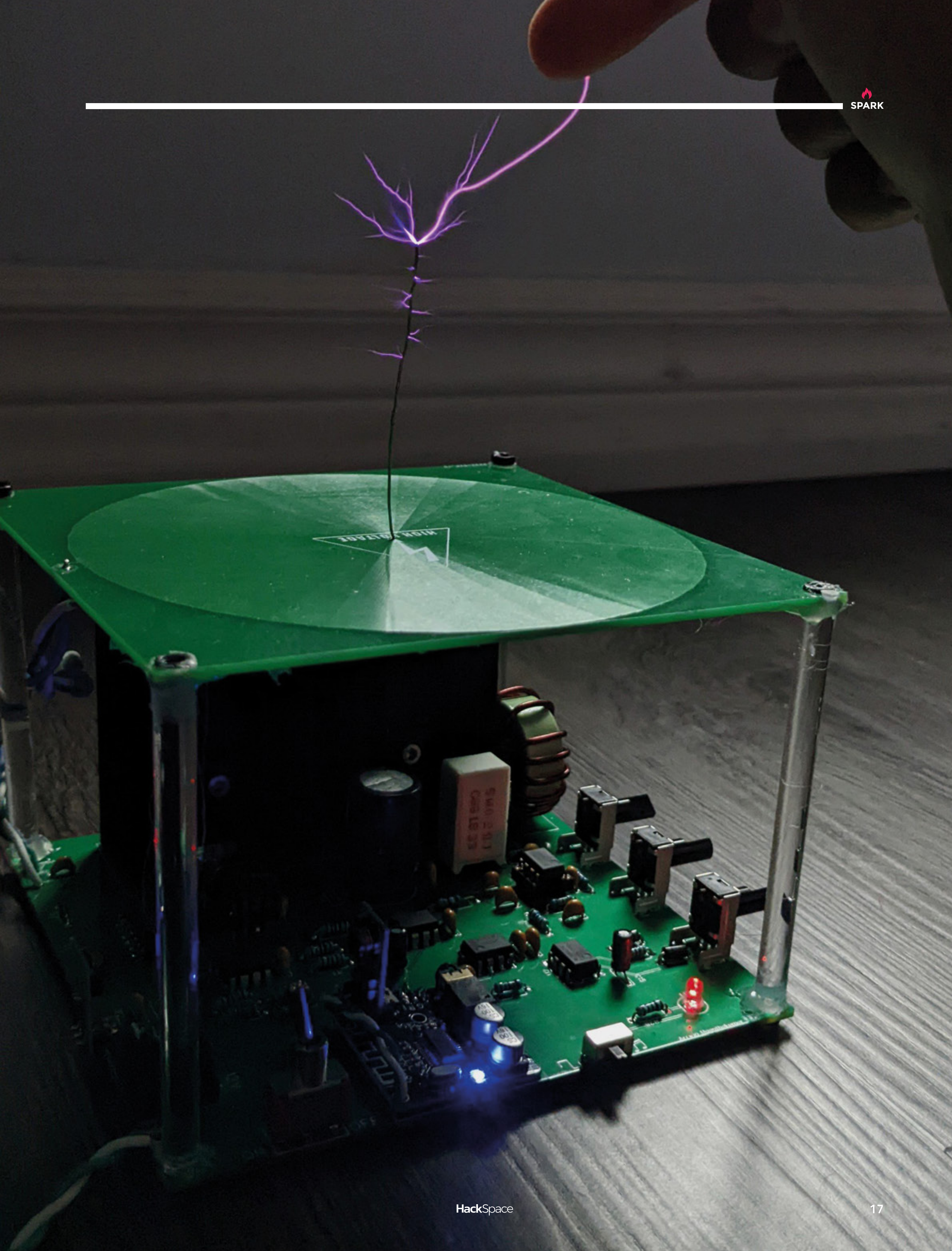
By Edison

hsmag.cc/MusicalPCBTeslaCoil

Tesla coils usually entail masses of copper wire to help create their weird and wonderful electrical effects. Canadian maker Edison has taken that idea and replaced the coils of copper wire with tracks embedded in a PCB, in a similar way to that taken by Carl Bugeja in his PCB motors (as seen in HackSpace, issue 52).

This musical PCB Tesla coil can play songs over Bluetooth, so you don't need to source MIDI files, but for no longer than ten minutes. Edison also recommends that you don't run it in spark mode for more than 20 minutes, and warns that the arcs it produces will not shock you, but may burn your skin. As always, when working with high voltages, take care. □

Right This PCB Tesla coil can create arcs up to three inches long



Objet 3d'art

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

If you've ever experimented with MIDI, you'll either adore or be infuriated by the way that you can tinker with everything. Sound-waves are really just applied mathematics, so generations of geeks have poured their hearts and souls into the pursuit of the perfect wobbly tone.

This 3D-printed foot pedal for MIDI is relatively simple by comparison with a lot of MIDI projects. It entails just one 100K linear potentiometer, an Adafruit QT Py RP2040, some M3 fixings, and a USB-A to USB-C cable. That, and a beautifully designed and executed foot pedal chassis, designed by Adafruit's Ruiz brothers. As always, the source files are available for anyone with space on their printer bed – in this case you'll need at least 210 × 210 × 100 mm.

hsmag.cc/MidiPedal

Meet The Maker: Ivan Kuleshov

Systems engineer, making his own systems

We're well used to the idea that, nowadays, if you want something that the market doesn't provide, you can build it yourself. Free design software, cheap PCB prototyping services, and online tutorials mean that anyone can design a simple electronics project. But high-performance computers – surely that's something that's best left to the big companies with money to spend on design, testing, and scaling up production?

You'd think that, but Ivan Kuleshov is having a go himself. He's developed (or should we say, 'is developing' – more on his incremental approach to design later) the Compute Blade: a carrier for the Raspberry Pi Compute Module 4 that enables users to easily build their own compute clusters, potentially putting previously out-of-reach amounts of computing power into many, many more hands.

First of all though, what is the Compute Blade?

"It's a carrier board for the Compute Module 4 that I designed for enterprise use. Basically, the main idea is to avoid failure points of a distributed (or cluster) server by scaling its nodes.

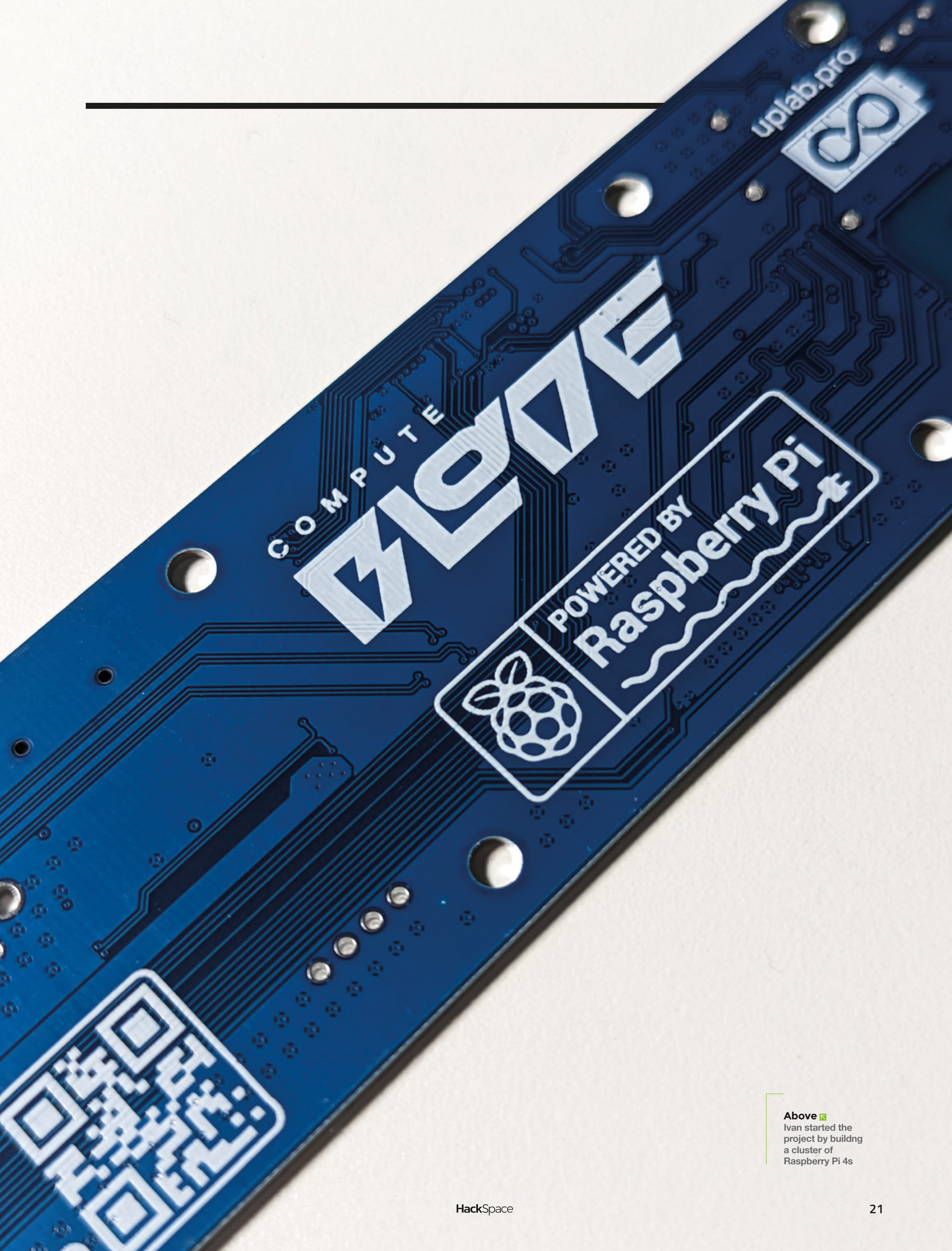
"Compute Blades are individual computers; there's no one point of failure. In normal blade servers there's a big backplane with maybe two power supplies, and it's a big point of failure. Companies that create blade servers, like maybe Supermicro or Hewlett-Packard, they waste a lot of time on research and development.

"Having the whole computer on one board means that you can link them together with cheap PoE switches – I've found them really cheap on eBay.

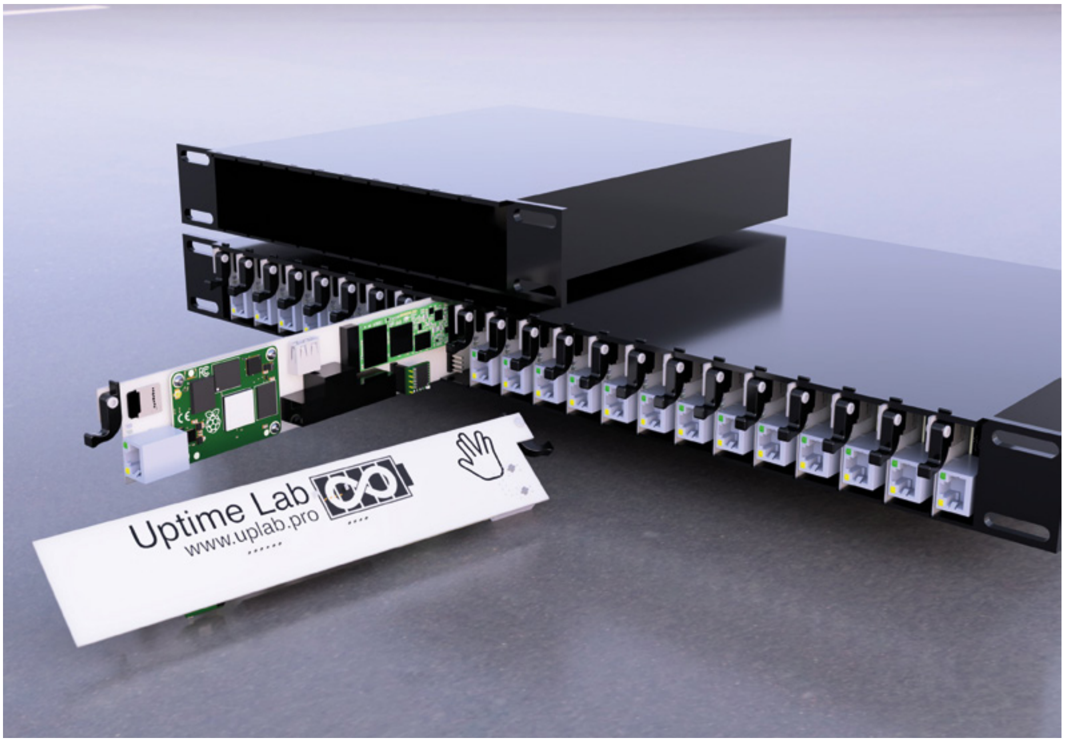
"The project started maybe two years ago when we decided that we needed some Arm devices to develop an application under Arm. After checking the market, we decided that Raspberry Pi would be the cheapest and the easiest option to start with.

"When I tried other platforms that were available on the market, I realised that they have a really small density… you can only use four Raspberry Pis in one unit, or maybe eight in two units. So I designed my own, for twelve Raspberry Pis, and then for 14, with an SSD each.

"And then I thought, 'How can I increase this?' It was just for fun really, because I didn't need to do it. There's enough space in our server room that we can use a cluster of Raspberry Pi 4s, but I decided to try. And then I discovered Compute Module 4. When I got it in my hand, I realised that it's one unit high [the standard height for rack-mounted servers]. Then it was hard to stop. →



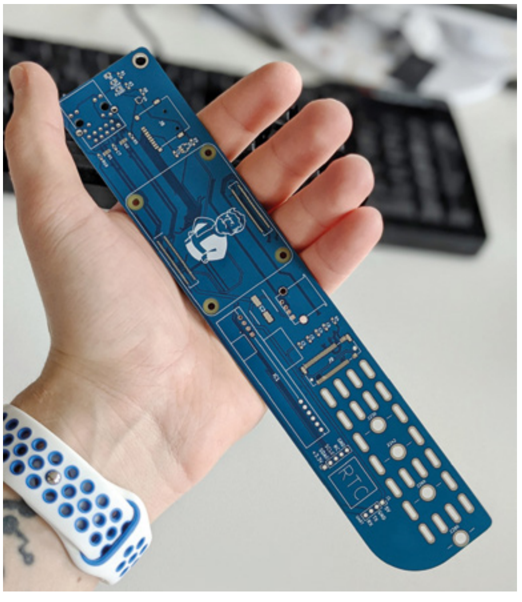
Above Ivan started the project by buildng a cluster of Raspberry Pi 4s



"I checked the data sheets – they are really great; that made it easy to start. Step by step, and now we're here.

"At first it was just for fun, to have a look – what can I do, how can I increase the quantity of devices in one unit only? Just a challenge for myself.

"Of course, we have big X2 Arm servers – really big servers with a lot of cores, a lot of RAM, but these are very expensive, and have limited support, because not many of them were ever produced.



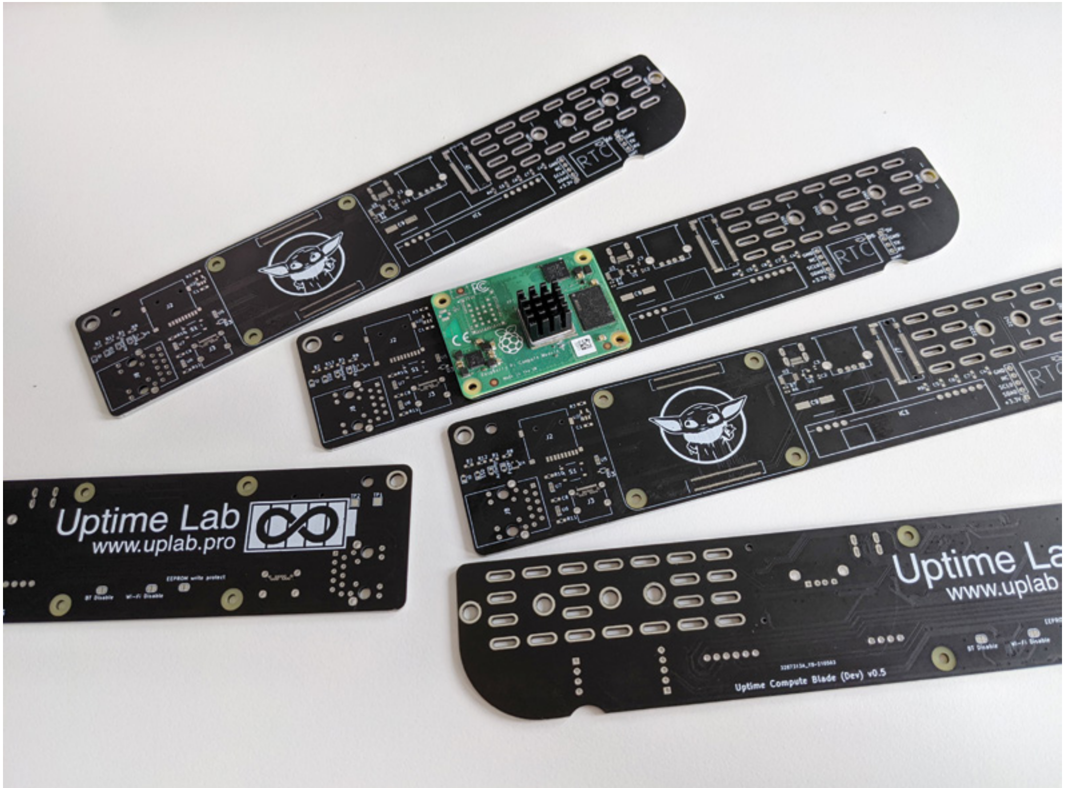
Right Compute Blade is 1 unit high, so it fits in a standard-sized server rack

"Not too many people want to buy it. I don't know why, because it's great for CICD, when you want to build software, but not to test. For testing, Raspberry Pi is great. That's another topic though.

"One reason I've been open about the development process from the start is that I can't patent my ideas at this stage. And I'm sure that someone other will try to implement this idea themselves and try to patent it, so now, if anyone tries to do the same, the idea is out there now – it's established as prior art. That was one of the reasons. The second reason is that it's a great way to make a final product better. I can cancel features that people don't need, and maybe integrate something really useful instead. I've added the digital LEDs on the front panel, with an additional button, a UART port as well… these are ideas that people added in the comments of a Twitter thread. That shows that people need them, so why not add it if I can? I've implemented maybe five new features following suggestions people made to me on Twitter.

"You can, of course, disable the LEDs; you can make them brighter so that the mounted rack of Compute Blades looks like a Christmas tree, or go for something in between, which is better for night-time or in dark rooms.

"Another one is that the smaller HDMI input, like on Raspberry Pi 4, isn't really convenient for me. A lot of the time I can't find the right adaptor for micro HDMI connectors, and a few people have the same problem, so I've changed it.

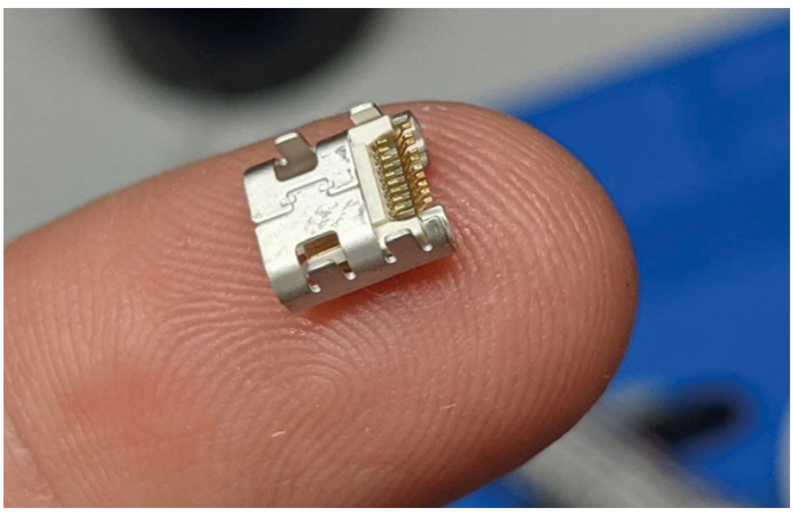


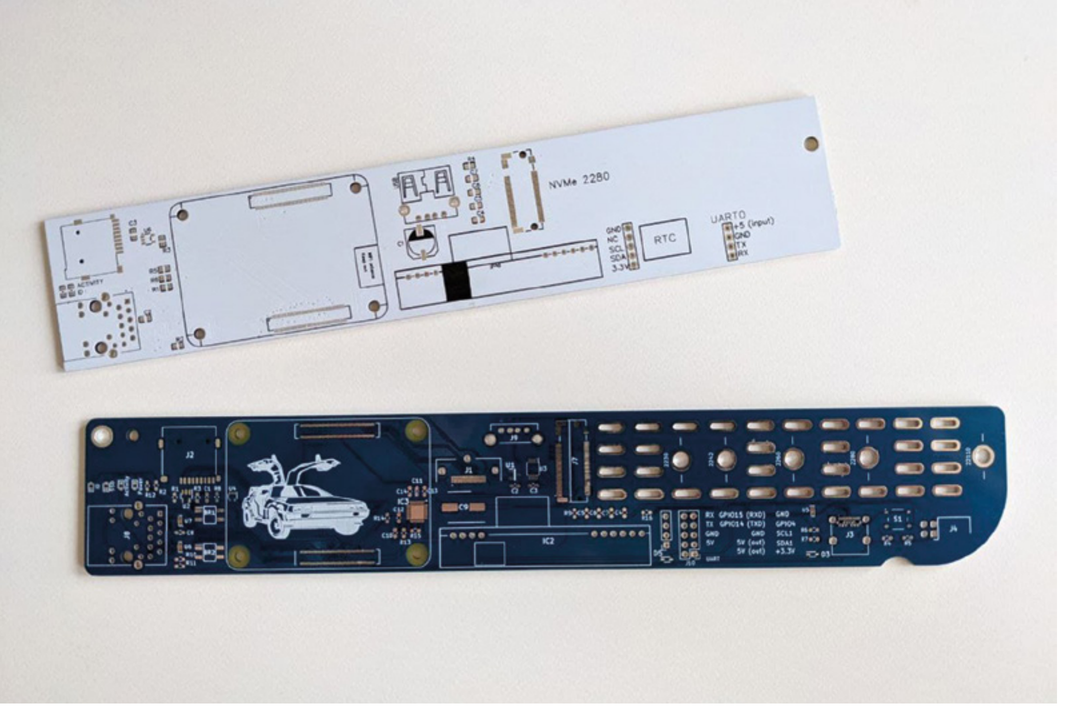
"It's not officially open-source hardware yet. I don't want to open-source the current version, because I'm afraid that if someone repeats it, they'll repeat my mistakes, and then further down the line that will come back to me. I don't see any reason to share at the moment. But later on, I'll probably share the schematics.

"Open-source isn't an end in itself. It's a good way to make products better, but I'm not pursuing it for any ideological reason; it's purely a practical thing. Not only a better product, but it's better for security and reliability. I don't have a lot of experience with implementing schematics, so the more eyeballs are checking the work, the better.

"At the moment I've got 97 pre-orders of the board itself, so we'll see what people think when the reviews come in. I'm having the blades sent to me from the PCB factory, then I'm checking them myself and sending them on to customers around the world. Hopefully I'll not have many more customs problems – I've had trouble where someone has read the theme of the product (Compute Blade) and thought that it must be a weapon of some sort. Maybe I need to rename it to the Compute Not-A-Blade. →

> " At the moment I've got 97 pre-orders of the board itself, **so we'll see what people think when the reviews come in** "



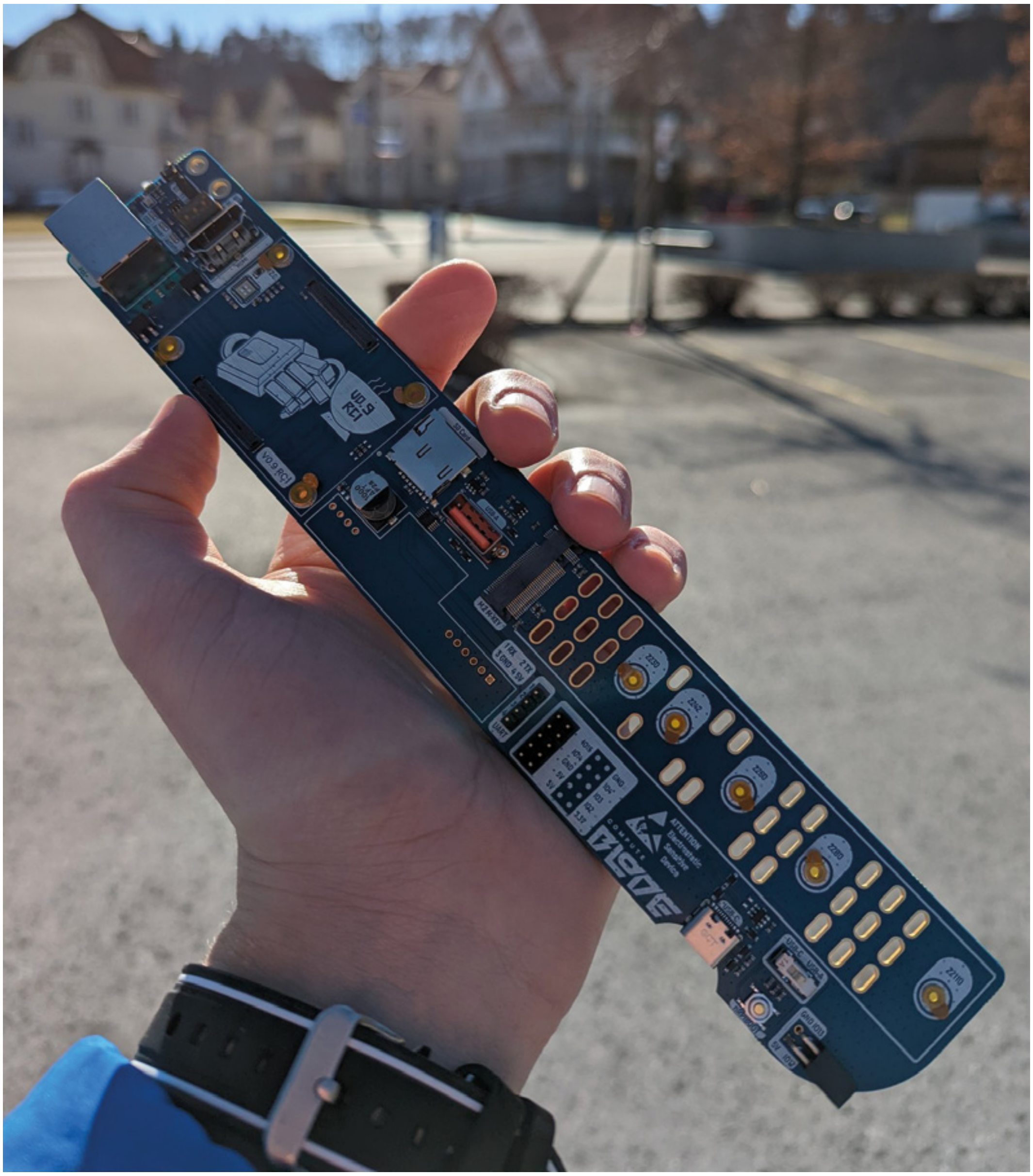


"Before I made this, I'd done one project on an Arduino. Other than that, this is my first hardware project. It's not too hard nowadays, because there's a lot of information on YouTube and spread across the internet. When you have an idea, you have the power to find all the information you need, it's not hard. I have a lot of experience in different tools, from laser engraving, 3D printers, soldering stations,

> The CM4 has great documentation, **so I don't need to solve problems by myself**

big machines, small machines… and the Compute Module 4 has great documentation, so I don't need to solve problems by myself. I just followed the schematics, and it works.

"The first version of the Compute Blade was made in a browser: Easy IDE. It's a CAD system to design PCBs. It's great for building Arduino-based projects, so I decided to use it with the Compute Module 4. Later I switched to KiCad, the open-source design program, and that made things much easier. The second revision, which was made in KiCad, is at work now in a production environment, and it still works. The biggest changes were switching to a



four-layer PCB, which necessitated large-scale changes, and that introduced a few new bugs, of course. But now it's certification-ready.

"It was soldered by myself, because it's faster that way. It's cheaper, yes, but it's also faster, because of the chip crisis, a lot of PCB manufacturers aren't able to get hold of all the parts needed for assembly; I've been able to order different parts from different sources, and then assemble it by myself. I've soldered maybe 20 so far. I should stop now and switch to factory production, because it takes three to four hours of my time to do each one. The first time I did it, I didn't use a microscope, but now that I've treated my eyes to a decent microscope, things have got much better. Some of the connectors on the Compute Module 4 are tiny… I don't know what I thought I was doing. But it's a great way to understand your product – you get a feel for how you might streamline the production process in future.

"I had hoped to get the first boards out to crowdfunding backers in two or three weeks from now, but because of the current situation, it's probably going to be more like six weeks. The company where I work full-time is closing its office in Russia, so there's a lot of extra work falling on the rest of the team.

"As far as I know, there's nothing on the market with the same functionality, at the same price, and the same form factor as the Compute Blade. Even at a higher price, there's nothing to match it."

Above
It's incredible (to us) that the Compute Blade is only Ivan's second hardware project

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**

APPLIED ION SYSTEMS

Thanks for the interview with Michael Bretti from Applied Ion Systems. I still consider myself a dabbler in open-source hardware. The best I've got to so far is printing other people's designs from the Prusa Printers site, then modifying them if needed, and uploading my changes. So in spirit, I'm an open-source hardware maker just like Michael, but in practice there's a whole world of difference! Some of the stuff he says is just mind-blowing. And the idea that one man, on his own, can move things around in orbit around the Earth, when just 50 years ago it took teams and teams of people… it's amazing.

Arthur Gray
Detroit

Ben says: It's part of the beauty of open-source hardware that we're all, in our own way, however big or small, contributing to the greater sum of human knowledge and utility. Whether that's a better 3D-printed object, or a better way of doing something with an Arduino or a Raspberry Pi. Or indeed, if we're building engines that generate a tiny amount of thrust to move things around in near-zero gravity. It's a way of thinking, and it's only because people made that initial leap of faith and shared their designs that we can sit here today and enjoy the fruits of everyone's labour. Every single open-source hardware developer is a flippin' hero.

UPCYCLING

Upcycling is one of those things that's never really entered my brain, until now. If I've ever heard the word, I've dismissed it as something a bit twee, like painting some flowers on a wooden table and pretending that you've added to its value. And I've seen projects where there's some new electronics that have gone into an old enclosure, like a cigar-box for example; I just never put two and two together and realised that that's upcycling as well. Painting flowers on an old hardwood table is what the world likes to see as upcycling; sticking Nixie tubes on an old Geiger counter and calling it a clock is more the maker way!

Ben
Manchester

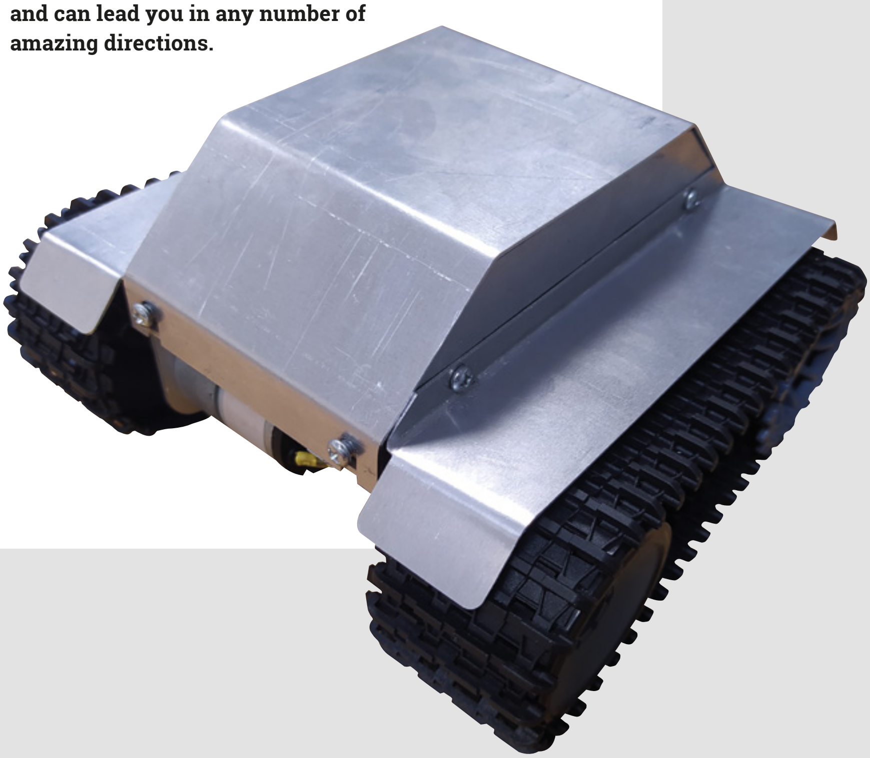
Ben says: It's the adding value part that's the key, as far as I'm concerned: that's where the 'up' in 'upcycling' comes from. Reusing an existing object is obviously excellent for sustainability, but turning it into something more useful is what takes skill. If you have access to junk, upcycling is a lovely, low-risk way to speed up your builds, as you don't have to make everything from scratch – cigar-box electronics projects being an excellent example of this.

SHEET METAL

I've been following your FreeCAD tutorial series since the beginning, and it's made me a much better, more confident user of design software – all design software, because, as I know that I can do it in FreeCAD, I know that I can have a go in whatever design package, knowing that I'll be able to get it right when I come 'home' to FreeCAD. It's been a real eye-opener, with all the different workbenches, but I get the sense that there's more to it than that, and all you've given us is a springboard. I'm still having fun exploring all the workbenches that FreeCAD has, and I can only encourage others to do likewise. It's brilliant!

Sarah
Coventry

Ben says: FreeCAD is indeed brilliant, which is why we're pulling all the tutorials we've published on it during the past 16 issues into one place. We're going to make it available for free on the internet (though, of course, if you want to toss a coin in our hat for it, you'll be welcome to). Most of all, we hope it inspires new makers to get started with this tool, which, as you quite rightly say, is fantastic, and can lead you in any number of amazing directions.



CROWDFUNDING NOW

Are you talking to me?

Voice-controlled AI for your printer bed

From £250 | **kickstarter.com** | **Delivery:** May 2022

Just to be completely clear – we have not tested the Wizmaker P1. In the crowdfunding section, we look at products that have caught our eye for one reason or another, and actual hardware reviews are in the final section of the magazine. Sometimes they catch our eye for good reasons, sometimes for … well, other reasons.

The Wizmaker P1 has one killer feature – voice control. If you find it hard to think of a situation where this would be useful, then the video has the perfect example: a lady is baking and has messy hands, yet would also like some decorations for her biscuits. Fortunately, she can just shout 'auto-levelling' and 'print the first file' to send her printer into action. If this is a situation you regularly find yourself in, then perhaps this printer is for you.

Wizmaker claims that this is the printer for Domestic, Entertainment, Architecture, Business, Medical, and DIY. This is a little surprising, since these six areas have quite different requirements for a 3D printer, and Wizmaker chose not to elaborate on exactly what makes this printer suitable for such a diverse range of applications.

Despite all this, there does seem to be one genuinely interesting feature of the printer – the Halo zoned print bed. This means that, for small prints, you only have to heat up a small area of the print bed, which saves energy. This isn't the first 3D printer to feature zoned print beds but, historically, they have been limited to larger format printers. Not content with simple zoned heating, though, apparently Wizmaker does this with artificial intelligence. We honestly are at a loss as to understand what intelligence you need to heat up a part of a print bed, but apparently there is some.

At the time of writing, this campaign is fully funded, so Wizmaker will get the chance to put its ideas to the test. Perhaps we are being cynical, and come the delivery date of May 2022, shouting at your 3D printer will be the latest trend and we'll be sadly lagging behind. Time will tell.

BUYER BEWARE

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

Left New for 2022 – voice-controlled 3D printers


NPI?
FREE SHIPPING
ON ORDERS OVER £33 OR $50 USD*
DIGIKEY.CO.UK
START with DIGI-KEY!
Over 400,000 NPI products stocked!
0800 587 0991
DIGIKEY.CO.UK
Digi-Key® ELECTRONICS
10.8 MILLION+ PRODUCTS ONLINE | 2,300+ INDUSTRY-LEADING SUPPLIERS | 100% FRANCHISED DISTRIBUTOR
*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

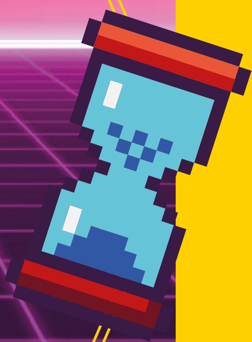
LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

RECREATING CLASSIC COMPUTERS

Early computers are fascinating machines, but often expensive to buy and difficult to maintain. One solution is to build a working replica, as Phil King explores



Want to learn more about the roots and evolution of computing? Fed up with software bloat on modern PCs? Longing for a golden age when machines were simpler? Welcome to the world of classic computers.

Large machines with blinking LEDs and toggle switches for programming and debugging are a far cry from the polished, user-friendly interfaces of today's computers, but they have their own charm. "A big part of the attraction is the aesthetics of a blinkenlight front panel, and how close that brings you to the lowest-level hardware of a computer," explains replica kit designer Oscar Vermeulen. "That is the part where I think there is educational value too.

"The design of a super-early computer like the LGP-30 is at the same time so easy to understand and so mind-blowingly smart," he adds. Yet the roots of computing are still very recognisable in these old machines. "You can even run an early version of UNIX on a PDP-11 from the 1970s."

The bulk and high cost of many early computers makes them impractical for most homes, however, which is where working replicas come in. In this feature, we'll take a look at some notable re-creations of early computers, from the 1950s to 1970s – as well as popular home micros from the 1980s – and share some expert advice on how to build a replica.

Whether you elect to buy a kit or embark on the ambitious project of recreating a classic computer from scratch, there's a community of like-minded makers who can help you. "I believe we all have a goal of bringing these wonderful creations to the attention of a new audience in addition to reminding the older crowd about a part of their past," says replica builder Michael Gardi. →

REPLICA PROJECTS & HITS

Explore some of the most famous and influential computers from the past, and their modern reproductions

LGP-30

Original Launch: 1956; $47,000

hsmag.cc/LGP30Replica

The desk-sized (not desktop) Librascope LGP-30 was a 31/32-bit word machine with a 4096-word drum memory. It was used by computer pioneer Margaret Hamilton (of Apollo 11 fame) while doing research on weather forecasting at MIT.

Jürgen Müller has created a palm-sized replica of this historic machine using a Numato Mimas FPGA (field-programmable gate array) dev board to model the LGP-30's architecture, along with a custom 'LittleGP-30' PCB. While dispensing with the original's 113 vacuum tubes, it does have the obligatory blinkenlights and adds an LCD screen – it can also be connected to an HDMI monitor.

Project files and full build instructions can be found on Jürgen's website. He has also created miniature working replicas of the RPC-4000 (LGP-30's 'big brother') and LGP-21.

PDP-8/I

Original Launch:
1968; $12,800

hsmag.cc/PiDP8

The successor to the original PDP-8 of 1965, the PDP-8/I was the first in DEC's family of machines to be built with integrated circuits. Available in pedestal and cabinet configurations, it could calculate using 12-bit numbers with a clock speed of 333kHz and 4096-word memory. It was used in a wide range of applications, from controlling tollbooths to launching rockets.

Oscar Vermeulen has created a modern 2:3 scale replica. Dubbed the PiDP-8/I, and available as a kit, it's based on a Raspberry Pi running a modified version of the SimH emulator (**simh.trailing-edge.com**). As well as flicking the panel switches (to debug or program) and admiring the blinkenlights, you can log in remotely using a laptop, or use a vintage serial terminal. You can even play Spacewar! on it, using a simulated vector display on a monitor. Meanwhile, the Raspberry Pi is still free to perform other functions, such as acting as a media server.

KENBAK-1

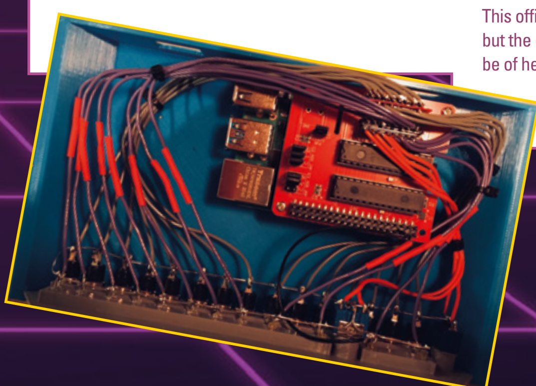
Original Launch:
1971; $750

hsmag.cc/Kenbak25

The KENBAK-1 was invented by John Blankenbaker in his garage. It is considered to be the first commercially available personal computer, despite its failure in the marketplace (it sold fewer than 50 units). Since it predated the invention of the microprocessor, it was based on small-scale TTL chips. The clock speed was 1MHz and it had a mere 256 bytes of memory. Front-panel buttons were used to program it manually in pure machine code.

Michael Gardi's 2:5 scale working replica, called KENBAK-2/5, makes it easier by allowing the user to enter a program by typing in assembly language instructions in an IDE. As he explains, "It allows one to have a 'classic' KENBAK-1 experience, then enhances that experience by allowing the user to use more modern tools like an assembler and a debugger."

He 3D-printed all the enclosure parts, which just snap together. The Raspberry Pi 4 is connected to the front panel via a 32-channel expansion HAT and also runs the IDE, which can be accessed on a remote computer via SSH or VNC. →



OTHER MAKES

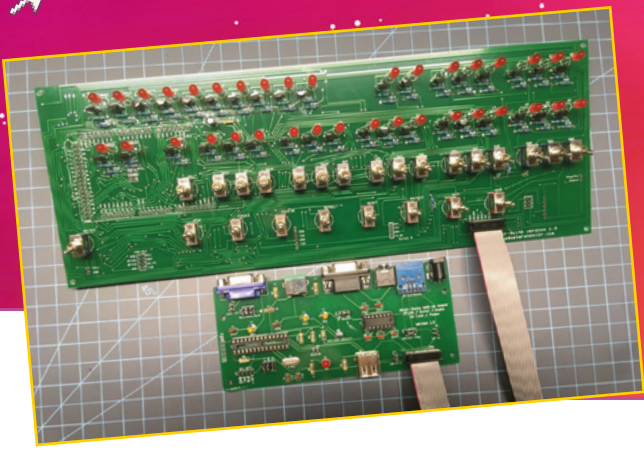
µKENBAK-1 (*adwaterandstir.com/kenbak*)
This ATmega328-powered 1:2 scale replica kit has an impressive steel case. There are also pocket- and full-sized kit versions.

KenbakKit (*kenbakkit.com*)
This official reproduction kit is no longer available, but the extensive manuals and documentation may be of help.

Altair 8800

Original Launch:
1975; From $395

adwaterandstir.com/altair



The MITS Altair 8800 is considered by many to be the first personal computer. It was very successful with hobbyists due to its affordability, although the basic kit version had just 256 bytes of internal memory and lacked a keyboard and printer.

Its Intel 8080 CPU (clock speed 2MHz) and RAM came on circuit cards that plugged into slots on the S-100 bus, which could be used to add a paper tape reader, extra RAM, and an RS-232 interface. Famously, Microsoft founders Bill Gates and Paul Allen wrote a BASIC compiler for the Altair 8800.

As such, it's a popular machine for replica makers. Adwater & Stir have created the Altair-Duino – yes, you can emulate an Altair 8800 on an Arduino! The standard $180 kit comprises a PCB and Arduino Due, and features a USB serial port, WiFi Telnet connectivity, and an RS232 DB9 serial port. The $260 Pro version adds an acrylic case and I/O expansion. Build instructions and detailed documentation can be found on the website.

OTHER MAKES

Altair 8800 Raspberry Pi case (*hsmag.cc/Altair8800RPi*)
Download the STL files and panel face stickers for this 3D printable case for a Raspberry Pi 3 or 4 on which you can emulate the classic computer.

Altair 8800 Clone (*altairclone.com*)
A full-size, fully functional replica (for $621) powered by a Microchip PIC24FJ128 microcontroller and support chips on a single circuit board.

AltairKit (*altairkit.com*)
Grant Stockly painstakingly recreated every board and component of an original Altair for this ambitious build.

SCELBI-8H

Original Launch:
1974; From $440

hsmag.cc/Scelbi8HReplica

Based on Intel's first 8-bit microprocessor, the 8008, the SCELBI-8H had from 1kB to 16kB of RAM, a cassette interface, plus Teletype and oscilloscope interfaces. Mike Willegal has produced reproduction PCBs and panels to build an 8H or 8B model.

PDP-11/70

Original Launch:
1975; $80,000

hsmag.cc/PiDP11

This was the largest model in DEC's influential and phenomenally successful PDP-11 minicomputer series, which ran from 1970 to 1997 and sold over 600,000 units. While UNIX was initially developed on the PDP-7, the first officially named version was run on the PDP-11/20 in 1970.

The PDP-11/70 was the last in the series to feature a 'proper' front panel with toggle switches and LEDs. It featured a 2kB memory cache and the ability to support up to 4MB of RAM via a new main memory bus.

Designed on a more living room-compatible 6:10 scale, Oscar Vermeulen's PiDP-11 is an amazingly accurate replica kit, even down to the original's key and lock. As the name suggests, it's based around a Raspberry Pi, but could also be powered by an FPGA. Like on the PiDP-8, the Raspberry Pi runs the SimH multi-system emulator and accurately handles input and output from the panel. You can also connect a monitor, or use a terminal via SSH or RS-232.

Altair 680

Original Launch:
1975; From $293

altair680kit.com

Arriving about a year after the Altair 8800, the 680 was smaller, cheaper, and based on the Motorola 6800 CPU. Again, Grant Stockly created a superb working reproduction of this classic computer.

IMSAI 8080

Original Launch:
1976; From $439

hsmag.cc/IMSAI8080Replica

Inspired by the success of the Altair 8800, the IMSAI 8080 featured a similar design, and used the same Intel 8080 CPU, but sought to improve on some of the internal components. The kit version was also easier to assemble.

The IMSAI ran a modified version of the CP/M operating system. Like the Altair, there was no keyboard, so you needed to program it using the front panel switches or on a video terminal connected to the serial port. Its S-100 bus featured 22 slots to add various expansion circuit cards.

The High Nibble has created a full-size replica of the front panel, complete with red and blue toggle switches and LED lights. The emulation software runs on an ESP32 board configured with 64kB of RAM. Along with RS-232 driver and DE-9M sockets for serial UART, there's a simulated GUI accessible via WiFi. The replica is available as a kit, but there's also a bill of materials and documentation on the website. →

Sol-20

Original Launch:
1976; From $995

hsmag.cc/sol20replica

Billed as a 'terminal computer', the Sol-20 is considered by many to be the first all-in-one personal computer, as it featured an integrated keyboard and a video output for a monitor or TV. Available as a kit or pre-assembled, it was based around the Intel 8080 CPU with a 2.0MHz clock speed and boasted 1kB of SRAM. Extra memory and add-on boards could be slotted into its standard S-100 expansion bus. Only 12,000 machines were ever produced before the Sol-20 was discontinued in 1979.

Some 40 years after first spotting a Sol-20 in a small Toronto computer store, Michael Gardi was inspired to create a full-size replica, based around a Raspberry Pi 4 running an emulator programmed in Python. Rather than using sheet metal for the case, he opted to 3D-print the blue panels so the build would be accessible to more people wanting to make one, but he did add wooden side panels.

Another Sol-20 fan had already recreated a Sol-20 keyboard replacement using modern parts, so Michael used that design. While a CRT display would have been ideal, he couldn't find a suitable TV or monitor so ended up designing one to look like a CRT but with an LCD display.

KIM-1

Original Launch:
1976; From $245

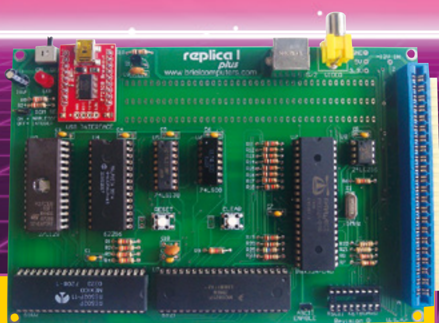
hsmag.cc/KimUno

Looking more like a calculator on a PCB, the KIM-1 was the most affordable computer of its day, and thus very popular with hobbyists. Oscar Vermeulen has created a KIM UNO functional replica kit that adds Apple I compatibility, 6502 calculator mode, and 'software gems' collection.

Apple I

Original Launch:
1976; $666.66

hsmag.cc/replica1



Designed by Steve Wozniak, Apple's first-ever computer was sold 'naked', lacking a case, power supply, keyboard, tape drive, and monitor. Still, it had ports for a keyboard, monitor, cassette, and expansion, and was easier to use than an Altair 8800. The machine was based around a MOS 6502 CPU running at 1MHz, and had 8kB of RAM. Only around 200 units were ever produced before the fledgling firm moved on to producing the Apple II. No wonder it's now a collector's item – one even sold recently for $400,000.

To get the Apple I experience without obtaining a second mortgage, you might want to consider the Replica 1, created by Briel Computers. The most recent version is the Replica 1 Plus, now available as a kit from **reactivemicro.com**. While it doesn't look at all like the original Apple I PCB, as it's based on modern hardware, it is a fully functional replica.

Anything you can do on an actual Apple I, you can do on the Replica 1 Plus. You can connect an ASCII keyboard to it and enter code and commands. A composite video output can be used to connect it to an Apple II monitor, or your TV using an RF modulator – alternatively, use an HDMI adapter to connect a modern TV/monitor.

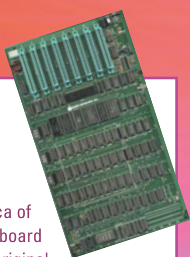
OTHER MAKES

Mimeo 1 (*hsmag.cc/mimeo1*)
Created by Mike Willegal, this is a hyper-accurate reproduction of the Apple I board. The kit is out of stock at the time of writing, but should be available again this summer.

CLASSIC HOME MICROS

Relive the eighties home computer boom with these replica projects and upgrades

The late 1970s and 1980s were a golden age for home computers (as covered in the book, *The Computers That Made Britain*, hsmag.cc/ComputersBritain), which are now prime material for replicas, revamps, and upcycling. Here are just a few of the many fascinating projects… →



Apple II (1977)

The first home computer to attain mass success, it sold around 5 million units in total. Mike Willegal created a replica of the original Revision 0 motherboard to replace a failed one in his original Apple II (**hsmag.cc/AppleIIReplica**).

Sinclair ZX80 (1980)

This ultra-low-cost computer featured an NEC 780C-1 CPU, 1kB of RAM, membrane keyboard, and monochrome TV output. Cees Meijer's superb replica is made from 3D-printed parts and based around a Raspberry Pi Zero (**hsmag.cc/ZX80Replica**).

BBC Micro Model B (1981)

Anyone who attended a UK school in the 1980s will remember Acorn's bulky microcomputer. If you've got an original model, you can enhance its capabilities with PiTubeDirect on a Raspberry Pi (**hsmag.cc/PiTubeDirect**). Or you can even emulate a BBC Micro on a Raspberry Pi Pico (**hsmag.cc/BBCMicroPico**)!

Jupiter ACE (1982)

Rather than BASIC, this small, low-cost computer could be programmed in Forth, which was faster. Cees Meijer has built another excellent replica with a 3D-printed case and keys, plus a custom PCB (**hsmag.cc/JupiterAceReplica**).

Sinclair ZX Spectrum (1982)

Low-cost with colour graphics, this best-selling British computer helped to spawn a homegrown games industry. Why not upcycle an old, non-working Spectrum with a keyboard adapter and emulation on Raspberry Pi (**magpi.cc/88**)?

Commodore 64 (1982)

The Spectrum's great rival featured superior graphics and sound. Some 30 million units were sold. Why not resurrect an old machine with this revamp project using a Raspberry Pi and LEGO (**hsmag.cc/C64Revamp**)?

Commodore Amiga (1985)

Along with the Atari ST, this powerful 16-bit series of machines revolutionised home computing and gaming in the late 1980s. Billy Nesteroulis built an Amiga 600 replica with a Raspberry Pi 4 running the AmigaOS emulator (**hsmag.cc/AmigaPi600**).

REPLICA MAKERS

We talked to a couple of prolific replica makers to get an insight into the hobby

Michael Gardi

Michael has built scale replicas of the KENBAK-1, Sol-20, and VT100 Terminal, as detailed on his website, mikesmakes.ca. In addition, he has made two Turing machine demonstrators and a tribute to the Digi-Comp I mechanical computer from the 1960s.

"I'm sure that a lot of it is nostalgia," he says, as to why so many people have a fascination with early computers. "Part of this is that classic computers were simple enough that you could completely understand everything that was going on with the machine. "Another factor is that fabrication technologies like 3D printing, CNC, and PCB manufacturing, plus the availability of cheap computing power like Arduinos and Raspberry Pis, make it possible for motivated individuals like myself to make these reproductions."

He admits that he wasn't much of a maker when he started work on the projects, but as he had just retired, "I had lots of time to dedicate to learning the skills I needed like 3D modelling [and] printing, PCB layout, woodworking, CNC, etc." He also joined a local makerspace (**kwartzlab.ca**) and not only got access to the tools needed, but also some expertise.

As for whether replicas of such early computers can have a practical purpose today, "Certainly I would consider education to be a practical use for these replicas. I'm hoping to show some of my reproductions at Maker Faires when they start up again. Not everyone will be interested in the history, but certainly my 1970s reproductions show us where the personal computers we know and love today came from."

Oscar Vermeulen

As well as creating homebrew versions of classic computers, Oscar makes kits for other makers to build, including the PiDP-8 and PiDP-11. All of his designs are open-source. Find out more on his site, Obsolescence Guaranteed: **hsmag.cc/Obsolescence**.

While he's been collecting vintage computers for 35 years, he says many early machines are impractical due to their bulk, as well as the difficulty in keeping them running. "So making a replica seemed to make sense. The PDP-8 and 11 specifically, because together, they cover the developments of the period 1965–1975 so well. Both were massively popular in their time, so there is actually loads of software to play with."

The PDP-8 series mirrored the evolution of computing – something Oscar wanted to replicate in his PiDP-8 tribute. "So you get to feel the evolution from paper tape and tiny machine code programs entered on the front panel, all the way to something that feels like an MS-DOS personal computer."

Regarding the PDP-11, "UNIX grew up on the 11, the characteristics of the C language are what they are because it fits the PDP-11 CPU – and the machine lived for a long time. You can still run a website on a PDP-11 today!"

As for the practical uses for classic computer replicas, Oscar says, "It's about archaeology, and finding a really nice programming object. And discover how computers work. Quite a few universities use PiDP-8s in class, because it is the simplest computer architecture yet it pretty much works like a modern computer. Understand the CPU of the [PDP-8 and] you understand how a CPU works."

BUILDING A REPLICA

Want to create your own replica of a classic computer? Here's some expert advice

Electronic hardware

One of the main choices you'll need to make for your replica build is what electronic hardware to use. Here, the main options are whether to emulate the system on a single-board computer (e.g. Raspberry Pi) or microcontroller, or to model its architecture with an FPGA.

"Each project is unique when it comes to the selection of hardware," advises Michael Gardi. "In general, reproducing the motherboards from the originals and populating them with vintage parts is usually prohibitively expensive."

The alternative is to emulate the original device functions using a modern microprocessor: "Picking what processor is dependent on what you are trying to accomplish," says Michael. "For instance, the KENBAK-1 has been successfully replicated using an Arduino, but I chose to use a Raspberry Pi 4 for my KENBAK-1 build because I also wanted to incorporate a modern development environment."

"For a while I felt FPGA was more 'real', but I started to doubt that," says Oscar Vermeulen. "As I started to figure out that FPGAs are not really creating hardware, they're a different way of programming logic, I became technology agnostic. As long as a replica passes the original DEC hardware diagnostic tests, it's a real replica to me."

He continues: "I like the software emulation better as it is cheaper, more flexible, and more hackable. For instance, we've added new CPU instructions to the PDP-11 to use the SPI hardware interface on the emulating Pi."

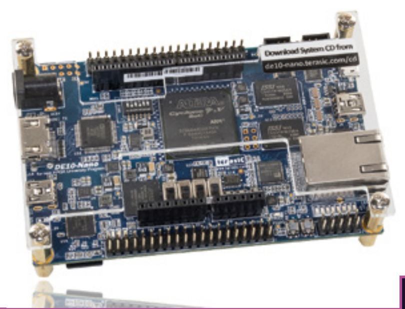
Making a case

While many classic computers had sheet metal or even wooden panels, it's usually more practical to recreate them with 3D-printed parts.

"Most of my reproductions were modelled from photos alone," says Michael. "One trick that I learned is that you can import the image of the part being created into the modelling software (I use Fusion 360 and the feature is called Canvas) and that 'background' photo can be calibrated to be the exact size of the original part. Then the part can then be modelled by 'tracing' the outline… Similarly for a case, say a front panel with switches, LEDs, and buttons, you can use this technique to position all the part cutouts precisely."

Many of the front panels of his reproductions feature labels and logos. "I was able to reproduce these by 'extruding' them up from the surface of the panel when modelling, then printing them in a different colour by pausing the print and switching filaments."

For making kits in quantity, however, Oscar opted to use injection moulding. "As a one-off, 3D printing is viable," he says. "A lot of parts for the DEC machines are available as 3D models for 3D printing, including all the parts (case, switch caps) that I made." □



MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
CLASSIC COMPUTERS REBORN
Make old computers with new technology
Plus
Geeky Gardening
Slicing
Analogue Output
LASERS CNC ROBOTS

SAVE 44%
MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc | May 2022 | Issue #54
CLASSIC COMPUTERS REBORN
Make old computers with new technology
Plus
Geeky Gardening
Tech to improve your plants
Slicing
Optimise your 3D prints
Analogue Output
Build a resistor DAC
LASERS CNC ROBOTS LEGO
DIY ROBOT
Make a rover from folded metal
Transform
APPLIED ION SYSTEMS
Open-source thrusters... IN SPACE
LEDS ANALOGUE
hsmag.cc | March 2022 | Issue #52
PCB DESIGN
Create custom footprints for form and function
LASER CUTTING
zap zap, cut, cut
the most out of
ENSORS FILAMENT

Getting going with gears

Let's delve into how to make and use these mechanical wonders

Jo Hinchliffe

@concreted0g

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

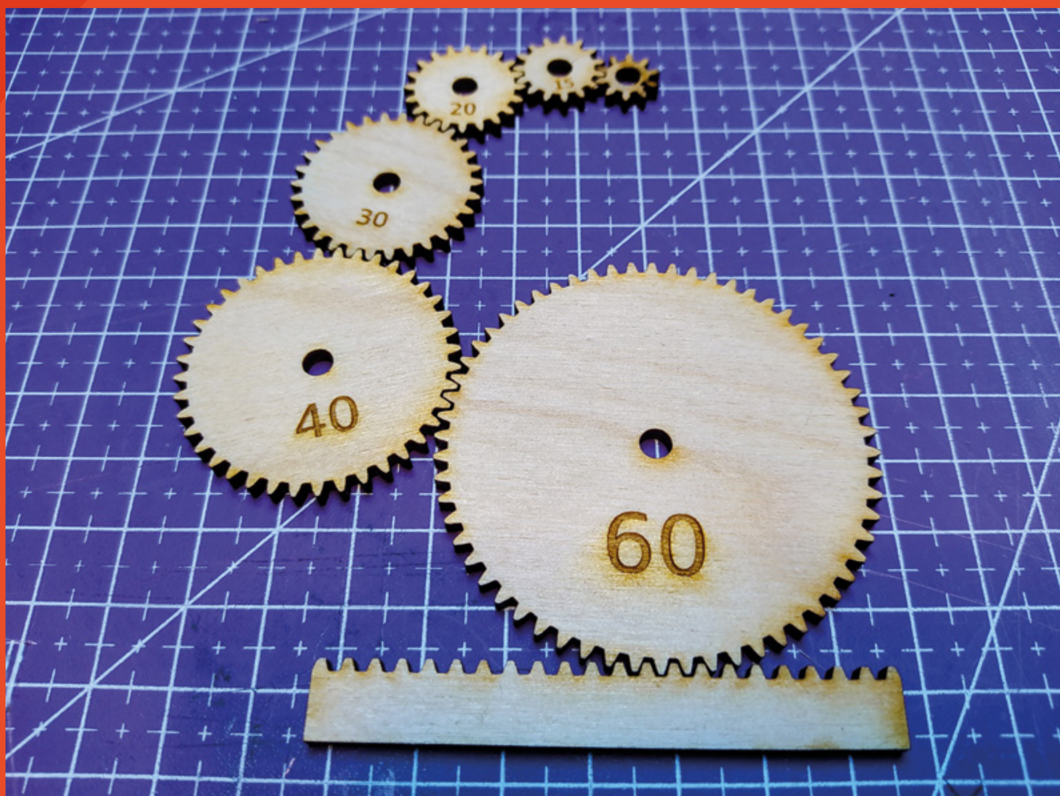


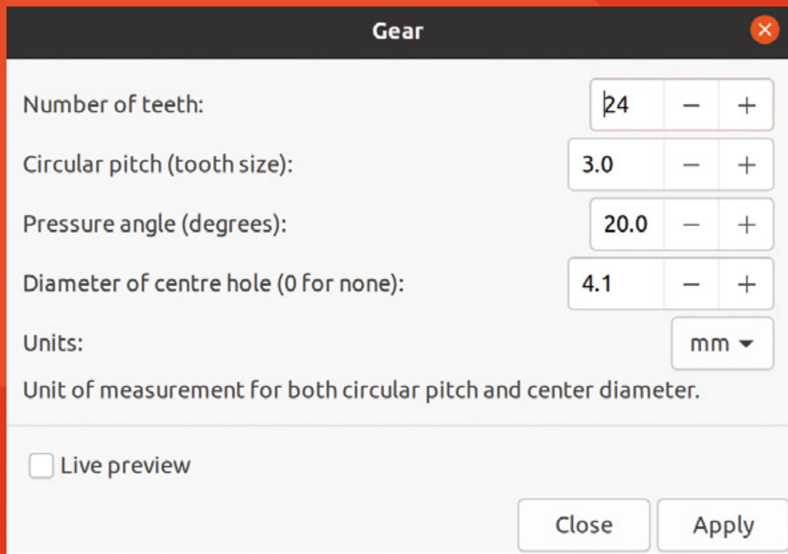
Figure 1 A collection of laser-cut gears and backplate generated in Inkscape

YOU'LL NEED

- Computer with Inkscape and FreeCAD
- Access to a laser cutter or 3D printer

Gears are all around us. These clever interlinking drive systems underpin many mechanical wonders, but they have always been quite difficult to make. With modern maker tools like 3D printing and laser cutting, they are now commonly available and customisable for you to explore and use in your projects. Gears come in many different sizes and flavours, but they are often used for the same small number of tasks. These uses include increasing or decreasing the speed of a system, increasing or decreasing the torque or power of a system, or changing the direction of rotation or direction of thrust. It's fair to say that you can deep-dive gear geometry and theory and find there is no end of stuff to learn. In this tutorial, we'll skim the surface so that you will know enough to get started.

Inkscape is a powerful free and open-source vector graphics application, and it's packed with excellent tools to aid all manner of making. Bundled into Inkscape are a collection of extensions that are incredibly useful add-on features – included in the bundle is a gear generator. Clicking Extensions > Render > Gear > Gear… you'll see a pop-up window into which you can input data to create a gear



(**Figure 2**). First, check that the Units option is set to 'mm', then you can experiment with 'Number of teeth' and the size of the tool using 'Circular pitch (tooth size)'. The pop-up also neatly has an option to include a hole in the centre of the gear, as you'll often want to mount it on some kind of shaft. We've talked about the 'pressure' or 'contact' angle elsewhere in the article, but for now it's a good idea to leave it set to the default 20 degrees.

Once you have some settings in the gear generator pop-up, you can click Apply and it should draw a gear perfectly on your canvas. The gear generator pop-up doesn't close after applying and creating a new gear – this means you can repeatedly make more gears with different values. In **Figure 3**, you can see that we have made a collection of gears, all of which have the same tooth size so should be compatible.

You can also see in **Figure 3** that we have made a 'rack gear' section. Rack gears are where a small traditional round gear (pinion) engages with a long flat-toothed gear (rack). These rack-and-pinion gear systems can be used to create horizontal movement, with the pinion gear remaining in position and the assembly attached to the rack moving side to side in a linear fashion. If you click Extensions > Render > Gear > Rack gear... you'll see a pop-up dialog to create a section of rack gear teeth. In our installed version of Inkscape, we found that we needed to set the tool spacing to double the amount of our tooth height when creating a standard gear to create a matching rack section. So, for the gears that we generated with 3 mm teeth, we needed to set the tooth spacing to '6'. Again, leave the 'contact angle' set at 20 degrees and set the length of rack you want to generate. Click Apply and you should see the length of rack gear teeth appear. It only generates the teeth, so you can use the regular drawing tools to turn the rack line into an object for your rack gear project.

The great thing about using a laser cutter to cut gears generated in Inkscape is that you can also add details onto the gears that can be engraved. In our set of gears, we have engraved the number of teeth to each gear so that we can easily identify which gear is which. Finally, to complete our gear set, we created a backplate that had two 4 mm wide slots which would allow us to bolt our gear wheels to the board and slide them into various positions (**Figure 1**).

A useful concept to understand is gear ratios. When talking about gear ratios, it's common to consider which gear is 'driven' and which is the 'driver'. The driver gear is the gear connected to something that makes it turn, for instance, a motor, engine, rubber band, hand crank, windmill – you get the idea! Sometimes, instead of 'driver' and 'driven', you might hear the terms 'input shaft' and 'output shaft' – again, with the input shaft being the mechanism connecting the driver gear to a rotational source. The gear ratio →

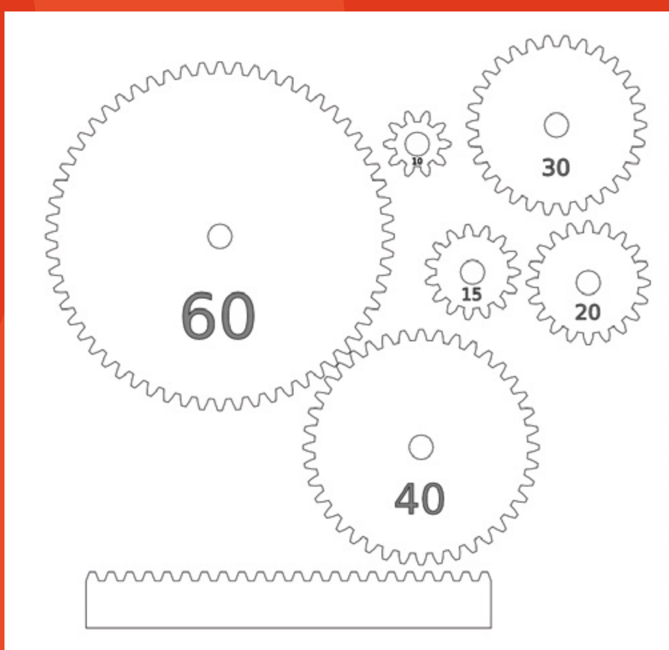


Figure 2 The small dialog box used to input key information to generate an involute gear drawing

Figure 3 A useful collection of gears for experiments, ready to be laser-cut

QUICK TIP

If the first gear you render is quite small, it might appear behind the gear generator pop-up!

Figure 4
A 10-tooth gear driving a 40-tooth gear

when we consider two gears, as in **Figure 4**, is defined as the number of teeth on the driver gear connected to the input shaft divided by the number of teeth on the driven gear connected to the output shaft. So, in **Figure 4**, we can see that the driver gear is a smaller 10-tooth gear, and the driven output gear is larger with 40 teeth. This creates a ratio of 4:1. Therefore, the smaller gear has to turn four times for the larger gear to make a single revolution.

With gears connected similarly to our example, there is also a change in revolution frequency between the input and the output gear, and so the larger gear will rotate more slowly than the smaller gear. Again, this is proportional to the gear ratio, so if, for example, the smaller input gear is driven at 80 revolutions per minute (rpm), the output gear will rotate at 20 rpm.

There is also a change in torque when gears are combined. You may understand this instinctively if you have ridden a multigear bicycle – changing the front gear to the small cog and the rear-wheel gear to the large cog enables you to ascend a steep hill

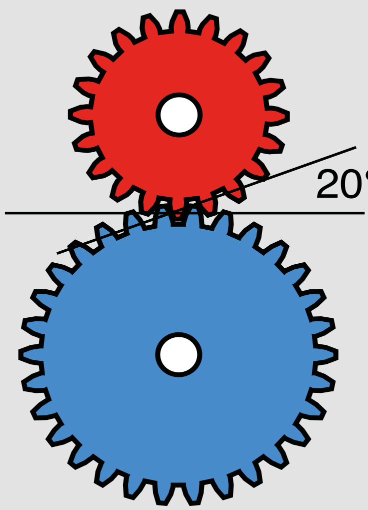
> "The smaller gear has to turn four times for the **larger gear to make a single revolution**"

more easily. Using our example of the 40-tooth and 10-tooth gear, if we imagine that we have some kind of crank or motor attached to the smaller gear that is inputting a torque of 1 newton, then using the above equation we can see that the output torque would be 4 newtons.

You've probably noticed a small difference between our directly coupled gears and our bicycle discussion. In our example, the direction of rotation is opposite between the gears, so if the input gear is rotating clockwise, then the output gear is anti-clockwise. If we need gears to be directly coupled and want to maintain the direction of rotation at both the input and the output, we need to add an extra gear to the 'gear train'. Using our laser-cut gears, we can achieve this by adding an extra shaft and placing a middle gear. Another reason to add an extra gear to a gear train can be that you can reduce the size of the system when you require a large gear ratio. For example, if we want a ratio of 120:10, which can be simplified to 12:1, we could use a 10-tooth

PRESSURE ANGLES

We've mentioned the 'pressure angle' or the 'contact angle' when talking about gears a couple of times, but what does it actually mean? In the image below, you can see two gears that are meshed together, and we can imagine one is driving the other. The forces exerted by the input gear are transmitted through the gear teeth along an angled line defined by the geometry of the gear. If we draw an imaginary horizontal line that is tangent to the gear circles, then the pressure angle is the angle between this line and the line displaying the force. Without delving deep into the complex maths and geometry underlying this, pressure angles are commonly 20 degrees, as this offers a good balance of efficiency in force transmission, minimised pressure on bearings or shafts, and generally produces least wear on a gear.



QUICK TIP

Don't have access to a laser cutter? You could make larger gears by printing Inkscape gears onto paper, gluing them to wood, and using a bandsaw or fretsaw to cut them out.

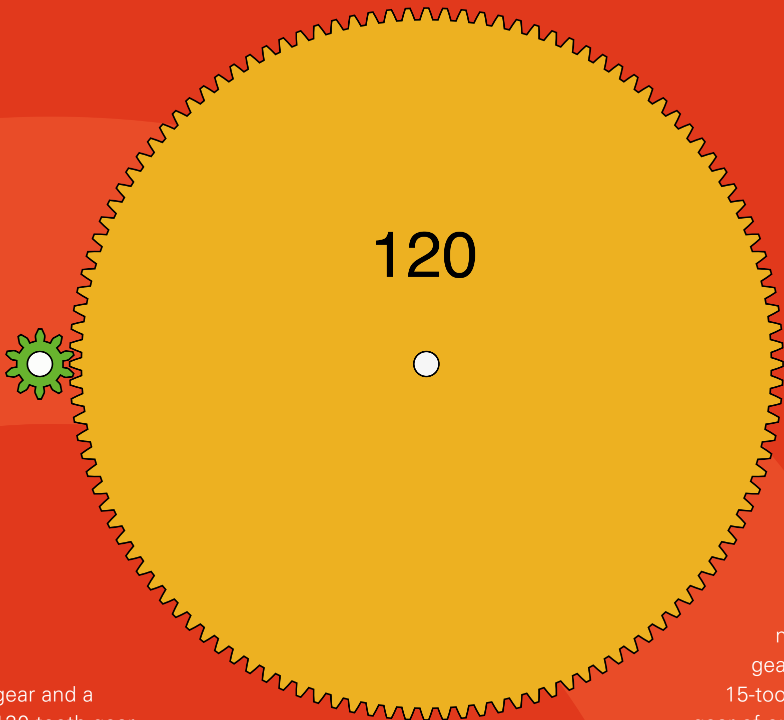


Figure 5
A 120:10 gear ratio: the 120-tooth gear would be around 120 mm tall – too big for our backplate

gear and a 120-tooth gear (**Figure 5**). However, that 120-tooth gear will be physically quite large, and we might want to reduce the size. We could consider a 60:5 ratio, but, as a rule of thumb, gears with less than ten teeth begin to run not quite as well as their larger counterparts.

Crucially, adding an extra gear doesn't change the gear ratio overall, and while you can check this simply by looking at and rotating gears, you can also see why in the maths. If we stick with a 10-tooth input gear and a 120-tooth output gear but place a 30-tooth intermediary gear, we can calculate the overall gear ratio by multiplying the two ratios made by the three gears. First we have the 10-tooth gear and the 30-tooth gear, making a 30:10 ratio. Then we have the 30-tooth gear and the 120-tooth gear, making a 120:30 ratio. To check the overall ratio, we need to multiply these two ratios together which looks like this (30/10)×(120/30). This returns a value of 12, which could be written 12:1, the same as our original 120:10 ratio.

While we have created a system where the input and output gear now rotate in the same direction (**Figure 6**), we have added a gear to our already large gear train, making it even bigger. We wanted to reduce the size of the mechanism. We can achieve this by using a compound gear system, where the middle gear is actually two gear wheels joined together. Continuing with our 120:10 example, when we added the 30-tooth gear, we had a 120:30 ratio on the output side. We can simplify this ratio or fraction and identify smaller gear combinations that will give the same ratio. For example, the 120:30 ratio could simplify to 60:15 or 40:10. Using our first simplification means we could create a gear train where we attach a 15-tooth gear onto the 30-tooth gear of our 30:10 ratio pair, forming a compound gear, and then swap the 120-tooth gear for a 60-tooth gear. Again, if we check the maths, we have (30/10)×(60/15) which returns our expected result of 12. As you can see in **Figure 7** (overleaf), a 30:10 to 60:15 compound gear train is much more compact than our original idea of a 120:10 would be.

When you start to create gear trains, or you begin to consider using gears that aren't traditional flat involute gear wheels, you may consider 3D printing as a method to produce gears. Similar to Inkscape, there are free and open-source pieces of software to help you generate 3D models of gears. We ran →

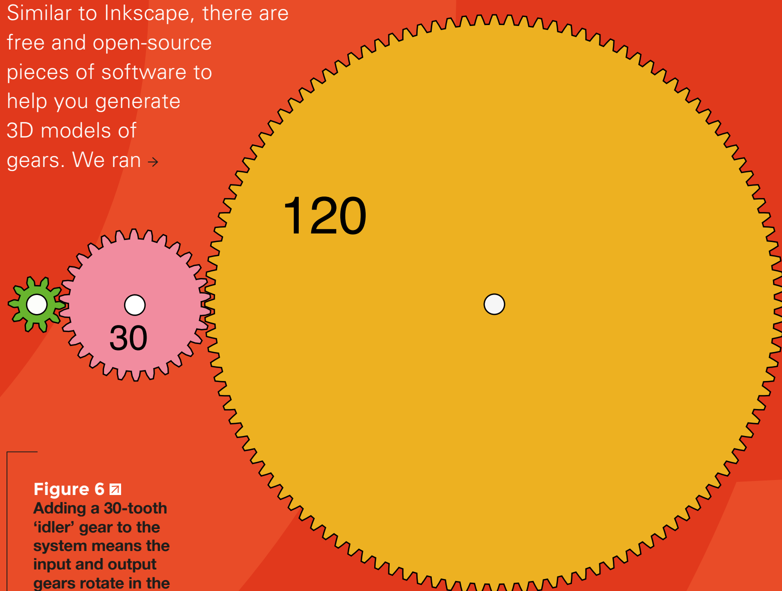


Figure 6
Adding a 30-tooth 'idler' gear to the system means the input and output gears rotate in the same direction

a long-running series of tutorials using FreeCAD – these are soon to be compiled into a free-to-download book **hsmag.cc/FreeCAD**. Brilliantly, FreeCAD also has an add-on workbench that makes designing gears really easy. Go to the Addon manager in the Tools drop-down and then search the list for 'FCGear' – install this workbench and restart FreeCAD. In a new project, move to the Gear workbench and you will see a collection of nine yellow icons that represent different types of gears you can create.

> Brilliantly, FreeCAD also has an add-on workbench **that makes designing gears really easy**

First, let's quickly make an involute gear wheel similar to the ones we created in Inkscape. It's incredibly simple to get started – simply click the 'Create an involute gear' option and a gear will appear in the live preview and as an item in the file tree (**Figure 8**). In the lower window of the combo view, you can see and edit the gear properties. Gear properties are numerous and can get complex quite quickly – many books have been written on the subjects of gears! To get an idea of the working parameters, it's worth looking through the FreeCAD documentation on the Gear workbench as it provides a good overview. A page specific to the involute gear properties can be found here: **hsmag.cc/FCGear**.

To create a variety of involute gear wheels, we can apply some changes. Scrolling down the properties, we can change the number of teeth using the 'teeth' option. Changing the number of teeth automatically calculates the new diameters and dimensions of the gear. We can change the height of the gear using the 'height' option to make a thicker or thinner gear. Lastly, we have found that it can be useful when 3D-printing gears to increase the 'clearance' property from the default 0.25. We found increasing the clearance to 0.4 made our 3D-printed gears mesh a little more easily.

In **Figure 8**, we have made a 25-tooth gear that is 7 mm tall. Notice that we have created this inside an active body that we set up on the Part Design workbench. This means we can add to our gear using the Part Design workbench tools. In simple ways, we can add a hole for a shaft through the gear, but 3D design and printing allow us to experiment with attaching gears to other parts of designs; for example, in **Figure 9**, we have connected our 25-tooth gear directly to a simple wheel hub design. Printing this is

Figure 7
Using a compound gear system to reduce the size of a gear train

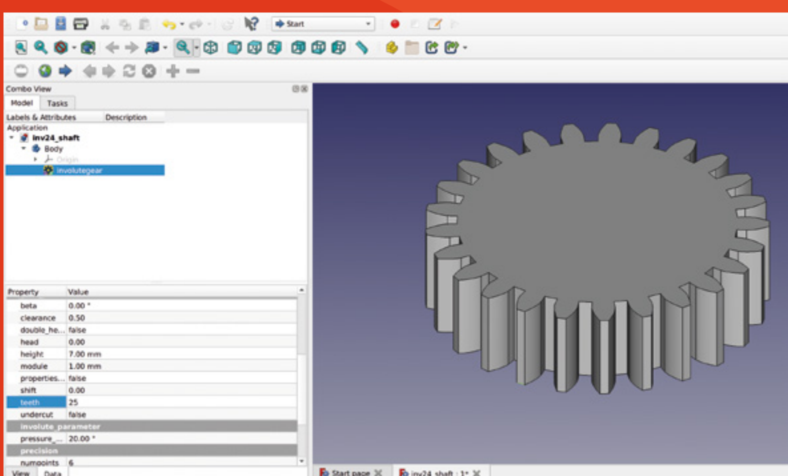
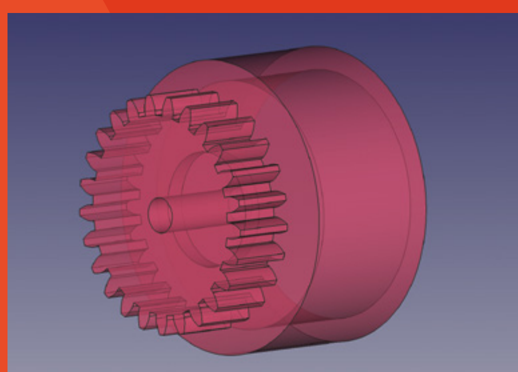


Figure 8
The FCGear add-on workbench in FreeCAD makes it really easy to make 3D gear models for printing

Figure 9
Making gears in FreeCAD means that you can connect gears into other parts, such as this wheel hub with an attached gear



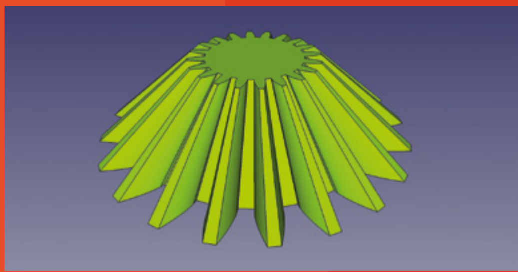
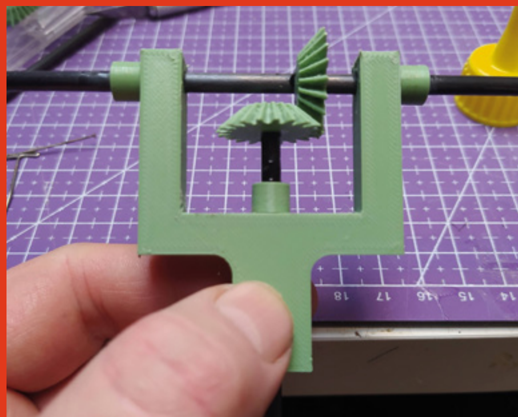


Figure 10
Making a bevel gear is simple in the Gear workbench

relatively straightforward, especially if we compare the process to creating this part in metal using lathes and milling machines and indexing tools.

The Gear workbench allows us, with only basic gear theory, to explore other families of gears. The bevel gear, or a pair of bevel gears, is a great example of a gear system that can create a change in the direction of work or thrust. It's easy to create. Again, in a new project, simply click the 'Create a bevel gear' tool. Similar to the involute gear, we can make changes to the design using the object properties. In **Figure 10**, we have created a 5 mm tall, 20-tooth bevel gear with a clearance of 0.4. We again created this inside a body so that we could add a sketch to pocket a hole through the centre of the gear to receive a shaft. Printing these small bevel gears is quite the torture test for your 3D printer. We printed them in PETG, and as the gear narrows to the upper surface, it's tricky to tune the print to not get small amounts of stringing in between the teeth; however, with a little light work with a needle file after the print finishes, it is possible to get a well-meshing pair. As an example of using two bevel gears to change the direction of thrust, we made a quick rubber-band-powered vehicle using some lightweight carbon fibre rods and some further 3D-printed housings (**Figure 11**).

Worm gears are another interesting example of changing the direction of thrust when driving another gear. A more correct pairing for a worm gear is a 'worm wheel', which appears like a regular involute gear albeit with a curve cut into the gear teeth. A worm wheel is a tricky gear to generate, but you can create reasonable worm gear and wheel examples using a regular involute gear wheel. Worm gears are a type of helical gear where each tooth is a helix wrapping around the length of the gear. If we click the 'Create a worm gear' icon, we will create the default worm gear. The default worm gear has three teeth, or, as it's hard when looking at a worm gear to discern the separate teeth, they get referred to as 'starts'. This makes sense when you look at the flat end of the worm gear and you can see that there are three lobes which are the start of each tooth helix. In terms



of gear ratio, they work in a similar way to involute gear wheels. A three-start worm gear coupled with a 24-tooth involute gear will create a 24:3 or 8:1 ratio. Another interesting aspect of worm gears is that they are often referred to as 'self-locking'. If you set up a worm gear driving an involute gear wheel, you will find that if you try and turn the involute gear, it will lock and not move the worm gear; compared to other systems where you might need to leave a system powered or apply a mechanical brake, this can be a useful feature (**Figure 12**).

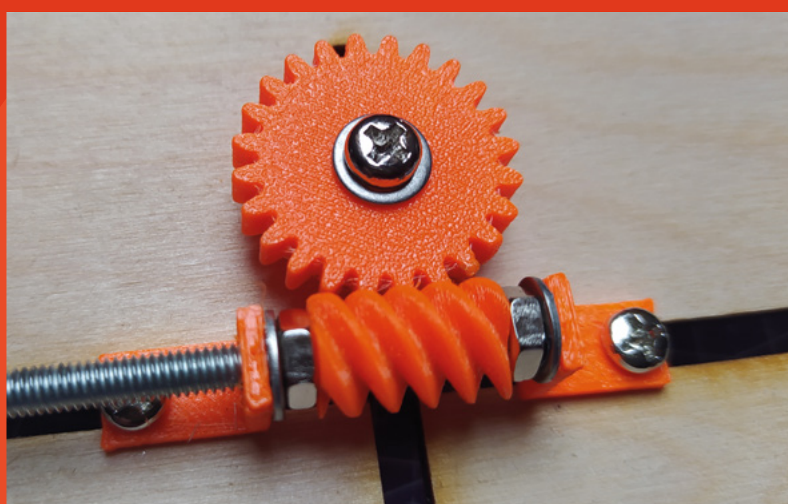
As ever, we have merely scratched the surface of both the FCGear workbench and gears generally. FCGear is capable of creating many variants of gear, and also timing pulleys, the type of which are commonplace on belt-driven 3D printers and other CNC machines. It's certainly worth clicking each of the 'Create a gear' icons on the Gear workbench and having a play with some of the parameters. There is a wealth of information online about gear geometry, construction, and theory of use, and while it's a fascinating subject, it can at times be a little overwhelming. We hope that this tutorial gets you playing with gears in your next project! □

QUICK TIP

Input torque × gear ratio = output torque. Sometimes referred to as 'torque at the wheel'.

Figure 11
Using a pair of 3D-printed bevel gears and some other simple printed parts to make a rubber-band-driven vehicle

Figure 12
A 3D-printed worm gear and an involute gear generated in FreeCAD



HackSpace magazine meets...

Jude Pullen

Come for the nifty Internet of Things device, stay for the ethical conundrums

Jude Pullen is a product designer. We got in touch when we saw one of his recent products, the Good Air Canary. Instead of just sensing particulate matter or $C0_2$ in the air and displaying numbers, the Canary speaks to the user, asking you to open a window, complains that it's getting stuffy, or, if the air quality is really bad, it dies, just like the original canaries that were taken down mines because it was cheaper to replace a canary than a man.

We talked about the Canary, and then, all of a sudden, it was two hours later and we were talking about disinformation on Facebook, Tony Blair, *Men in Black*, using feminism to sell cigarettes to women (apparently they were marketed as 'Freedom Torches'), and what really goes into sausages. What follows is an attempt to distil some of the more design-led ideas into words. →

Right The Good Air Canary was a collaboration between Jude and DesignSpark

HS One of the things that's refreshing about the Good Air Canary is that you've thrown the challenge out to a bunch of other makers around the world, and they've come back with some genuinely different approaches. It's easy to talk about sharing ideas in the abstract, but this is an example where you've done it, and it's worked.

JP Well, these first ten people I hand-picked; I'm aware of their work and I know they would each come up with some different ideas. We're opening it out to submissions from another 50 makers. Again, there has to be a selection process, otherwise you might get a few duplicates. Having said that, even if the project itself is the same, the data profiles you'll get in, say, India and Australia are going to be very different.

With the first ten, I wanted them to be different geographically, I wanted them to be different from a data perspective. And I wanted it to be different from an inclusivity perspective as well. It's easy to get a superficial kind of diversity, but more important than that, I wanted to see creative and cognitive diversity. I really wanted people to think differently.

Yes, of course I want to have representation. But I didn't want it to be a tick-box exercise; I wanted to choose people first and foremost through absolutely loving their process, and their attitude towards design. And I'm proud that there's a good group, with lots of different backgrounds.

HS Was this a DesignSpark-led project, or did you pitch it to them?

JP I would say a bit of both. What I love about some of my clients is that working with them is almost like going on a date. It's not a sales pitch; it's a date. It's saying, you tell me a little bit about what you're interested in doing. I'll tell you a little bit about what I think is interesting and happening outside of your bubble. And then we sort of cook up what is the metaphorical baby or product of those two interactions.

In my earlier career, I sometimes thought of things and pitched them to big companies, telling them 'you should totally do this.' And, of course, they just laughed you out the building, because they weren't warmed up and it wasn't a date. If you're not part of the company, you've not signed any NDAs with them, they can't control you, so in their way of thinking, it's massively dangerous to engage with you.

And so, it's much, much better to start at a date metaphor of saying, let's not worry about getting married here; let's just talk, let's just see where the energy is. And so, I feel that way is much more useful to both parties. It sort of removes tyre kicking. But it also means that, when we arrive at something that's on point, a good brief feels like a no-brainer.

This project feels like a no-brainer, because we can see the business imperative, we can see the social impact. Designers shouldn't be made out to be these gurus that come down from the mountain. I think designers are these embedded, multilingual people who connect lots of different things. And also, we're connected to the world outside of the company, so we're not drinking the Kool-Aid. So we come in, and it's not that we're being bolshie – it's just that in order to get a feel for the brief, you have to ask the obvious questions: 'why do you do it like this?'

By virtue of doing that, by saying 'I don't understand why you do it like this', you sometimes do them a huge favour. This is the emperor's clothes moment. And what's great is that, as you're an outsider, you don't burn as much stuff with the politics, because you can come in and be like a court jester in Shakespeare: you can come in and you can say the unsayable without having your head chopped off.

I love the fact that you can go in and you can broach some really tough stuff with companies; that's the moment when I can feel this magnetism, this energy in the room.

I've been doing all this remotely as well; I think there's a myth that you can't innovate remotely. I have never been busier in my life. And yet, I've never been less stressed.

HS That's a winning combination. There's a lot to be said for working for yourself, but a low-stress environment is not normally one of the benefits that people mention.

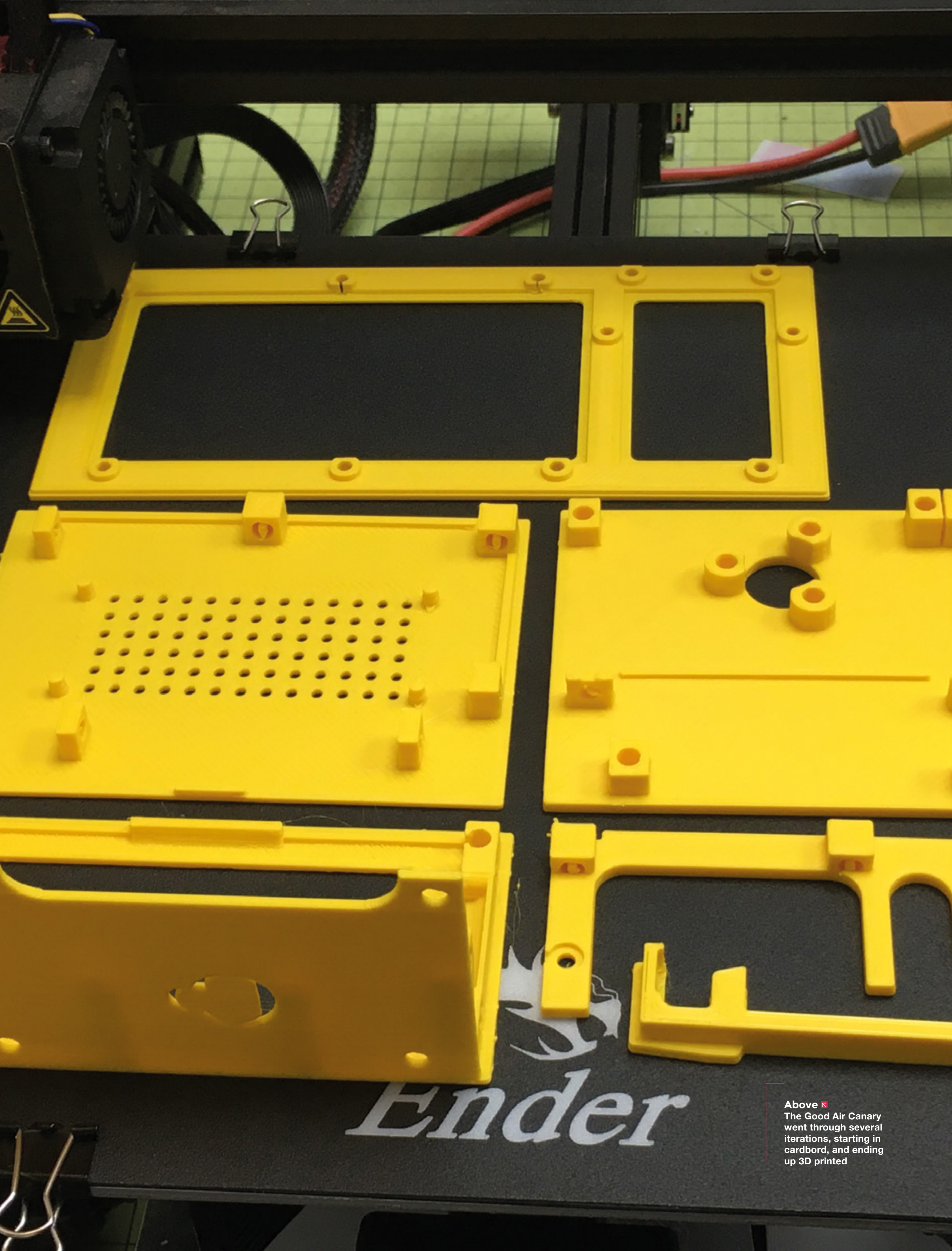
JP Yeah. It's funny, someone was asking me this just yesterday – how I divide my time. And the best advice I got from someone was, take what you need to live and multiply it by three, because a third of the time you'll actually be earning that money. A third of the time you'll be hustling. And a third of the time it'll either fall through and you don't get the gig, or you're just researching the next thing that you think you need to invest your time in. I spend easily a third of my time just reading, researching, going places, and thinking.

I think what I love about design, as a profession, is that I feel like I have more in common with some journalists and the way you learn to sniff out a story. But, unlike a journalist who writes a story on paper, I take that story and I knock on the door of a company and go, 'this story's gonna widen your eyes, I guarantee it. So what are we going to do about it?'

But the difference is a journalist doesn't help you solve the problem. The journalist just says, 'oh, sustainability, how do we solve it?' Whereas my job (taking sustainability as an example) is to look at what this means for your bottom line. And let's look at what this means for your product roadmap, and your innovation strategy. And let's look at legislation. Let's look at what your competitors are doing.

Let's look at what happens if you do nothing – what happens then? How many years will that work? And I think if you spell that out to companies, there's still a lot of people who will be just say: 'you're sure I can't just run this into the ground for three more years?'

And actually, it's been great to feel that there's a wind of change in your sails to do the right thing, and still help companies pay the bills and be →



Above
The Good Air Canary went through several iterations, starting in cardbord, and ending up 3D printed

profitable. I studied sustainable design in Norway in 2007-2008. But I can't think of any companies I could have engaged who would have taken it seriously back then. Whereas now, the question is not 'should we take it seriously?', it's 'how are we going to implement this?' It's completely urgent in most good companies.

At Dyson it was a revelation to realise that we could shave off X number of grams of plastic from a part and save energy that way. But it was completely eclipsed by a 1% change in the energy efficiency of the motor. We realised that that was far, far more important.

It doesn't mean that you did either-or: you tried to do both. But you didn't kid yourself that shaving off a gram of plastic was as important as making the motor 1% more efficient. I think engineers, scientists, makers, have an ability to see things holistically. You know, instead of worrying about the paper cup element in the takeaway coffee, I should be worrying about the production and the shipping and the making and the drinking of the coffee in the first place, and the boiling of the water? All of that stuff probably eclipses the value of switching from plastic to paper cups.

HS What was your role at Dyson?

JP Dyson was my first graduate job; I really hit the ground running. I had two years cutting my teeth in engineering, doing new product development [NPD], and two years in new product ideas [NPI]. In NPD, I was doing things like the Dyson Stick Vacuum. I did some of the early testing on that product, which was the first product of its kind. I was basically helping to calibrate the brush bars. It's not wildly prestigious, but you've got to learn your chops. I did well on that, or so I like to think. And then I worked on the DC37, which became by far the quietest Dyson machine, with no compromise on suction. That entailed a huge amount of aeroacoustic engineering, which I had absolutely no background in. And I love the fact that, in six months, I could hold my own with any of the aeroacoustic team. Dyson instilled in me a sort of can-do attitude, that within a few months you can teach yourself pretty much anything. Even if it feels like rocket science, there's just an attitude of 'jump in; you'll figure it out'. There were about 400 engineers there when I started; by the time I left, it was close to 800. So you can imagine that the pace was extraordinary.

I felt that it equipped me to be a manager and to run my own team at Sugru – when I was there, I took it from basically two people to twelve in a few years, and completely overhauled the formulation so it could be automated and scaled up. I loved being able to walk the walk at Sugru, because it was a smaller company and you could literally just say 'this is now in effect', and it was. It's harder to make those sorts of changes in a big place such as Dyson, or LEGO; it's a bigger machine to adjust.

HS I've heard you use a beautiful phrase in the past: purposeful aimlessness. Can you tell us what that means?

JP Thomas Heatherwick coined this phrase, either in a TED talk or in his book, *Making*. I adore it as a phrase because whenever you say you're going to build something, be it a bridge or a highway or whatever, there's always a way to do it. There are assumptions, things that you've got to verify, and success criteria that prove whether it's going to work or not.

I think purposeful aimlessness is a nice counterpoint to that notion. It's acknowledging that you need to feel your way into a brief, and you're going to react to the insights as they come up. I think you move design closer to art and away from science by thinking that way, because you live and die by what you discover.

With the Canary project, I was very gung ho at first; I still have a fancy weather thing on top of my shed, because at the beginning of the project I was convinced that outside air quality was going to be more relevant than indoor air quality. That's where the pollution is, right?

Six months later, it's all about indoor air quality. By speaking to experts, reading white papers, doing the tests, and building prototypes, we've realised that indoor air quality is the hot topic that's emerging.

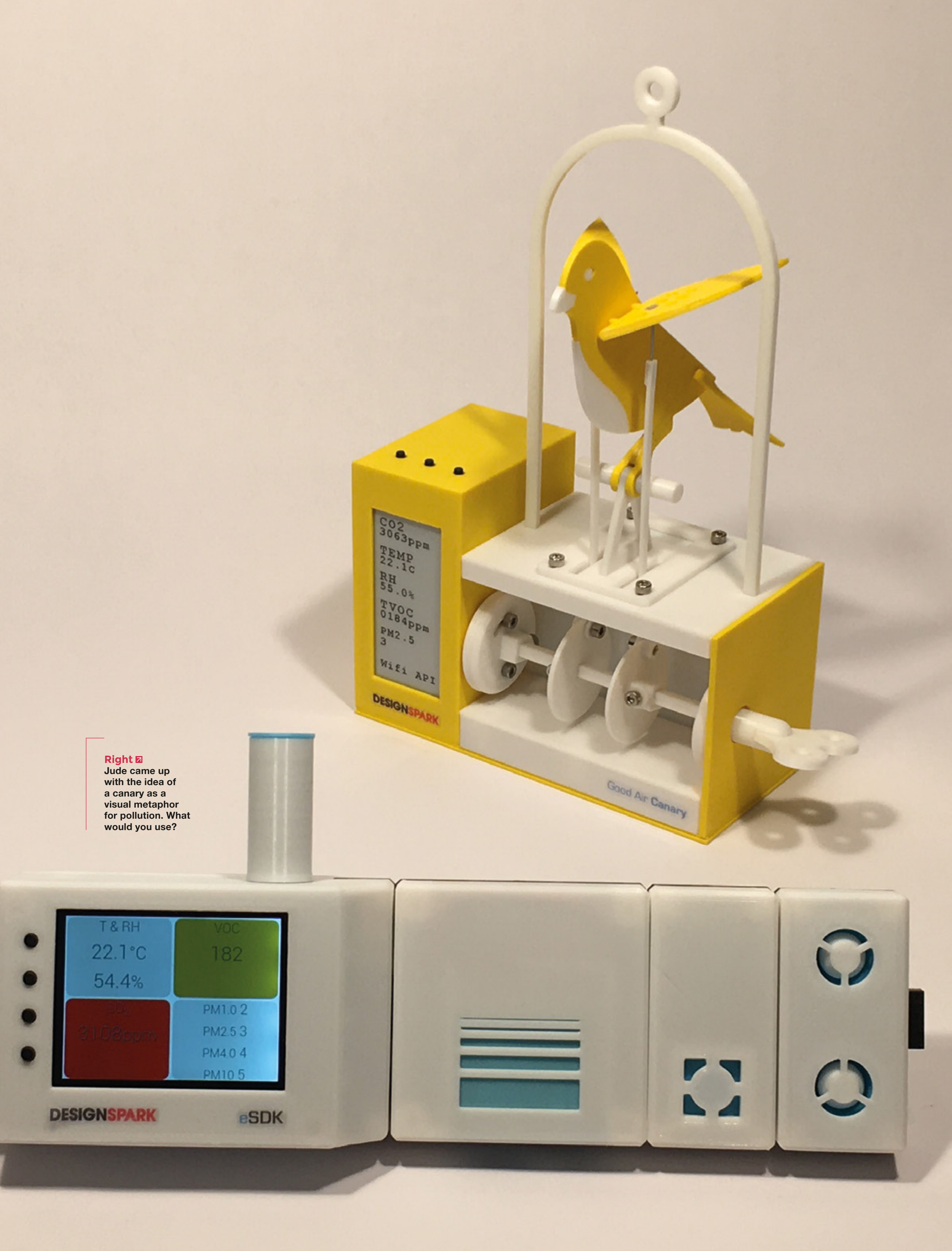
I'd pitched the idea of a project built around air monitoring, four and a half years ago, but it just didn't feel right. It got a lot of interest, but timing is everything, and it just didn't fit. Sometimes you revisit an idea, and when I realised that it would be about indoor air pollution, that's when it started to make sense. Four and a half years ago we weren't talking about Internet of Things and 5G the same way. The product evolved in response to the zeitgeist around it. That's what I love about purposeful aimlessness: if I'd dusted off the old brief, it would have flopped again.

It is much more feasible to adjust your indoor air quality for the better than it is to try and worry about a colossal wind coming in from China – you can study that, but you cannot do anything about it. Whereas, inside your house you can buy this itty piece of equipment for a couple of hundred dollars, and you can build it, get some data, and learn which of your rooms need a different response. It's tangible.

HS So you don't have any time for supposed carbon neutral airports, or these single-use coffee capsules that claim they're green because they're made from a material that you can recycle?

JP There's a lot of greenwashing out there. We fundamentally know that consuming less is better. Consuming coffee in little single-shot packets is never, never going to be as good as buying a 2 kg bag of coffee grounds and making it yourself. It just never is, and no amount of tinkering around the edges with packaging will change that.

There's a real risk that if you sell people a lie, you poison the well; you create cynicism. This is the bit that I think designers have to work really hard for, is that some people really want an excuse not to do the right thing. So if you give →



Right
Jude came up with the idea of a canary as a visual metaphor for pollution. What would you use?

them a scandal that says 'this is all a farce, it's all chicanery, and politics and trickery and numbers', then they go 'well, I always wanted to go to Tenerife. Let's burn some jet fuel.'

HS Let's get back to the design process. When I'm making something, I start with what I've got, which is usually a cut-off bit of plywood, and I start from there. How does a professional do it?

JP I think a lot of jobs, and a lot of schooling, trains you to think that everything is all right with the world as long as you know what's happening tomorrow. And I think design teaches you to get much more comfortable with the idea that you are going to have weeks, possibly months of not knowing what's happening next.

There's something in your human DNA that your caveman or woman doesn't really want to wake up and not know what's happening tomorrow. And so it's scary. I genuinely think you have to unlearn your nature to get excited about the fact that you don't know what's happening tomorrow, because it doesn't cost you your life. Not knowing what the product is is the emotional equivalent of not knowing where food or water is coming from, and that's scary.

HS Sustainability is one thing – we have a duty to not make the world any worse – but do you think you can make the world better through design?

JP I am very drawn to ethics. I realised that the bit in my design school that I didn't really get into was ethics. You got taught to make stuff, but you didn't get taught to question the ethics deeply. And that's not a slight on my degree; you can only fit so much into a course. But I think that, just as a doctor is taught ethics, I think designers should be taught ethics.

Victor Papanek, in his book *Design for the Real World*, says: 'there are professions more harmful than industrial design, but only a few'. His premise is saying a good designer creates something that's desired. And that is inherently dangerous to the planet. Because if it's desirable and harmful, then people are going to ask for more. And it's going to create a problem. And so, if they create something that's desirable and is good for the planet, that's very powerful.

When you try and design things from scratch, as a startup, you have to have some ethical training in place. And I think that's why I'm coming back to education. I think educators should teach ethics, especially to coders. Coders are in a wild west at the minute. There's a glorification of 'move fast and break stuff' out there. And I think some of the breaking stuff is laudable because they are changing the world, challenging dogma; they're doing great things.

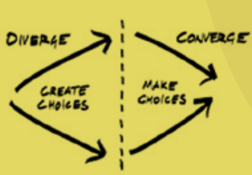
I love working with coders and developers, I just want to put that on record. But I think the critical thinking to know how to implement ethics in the work is sparse.

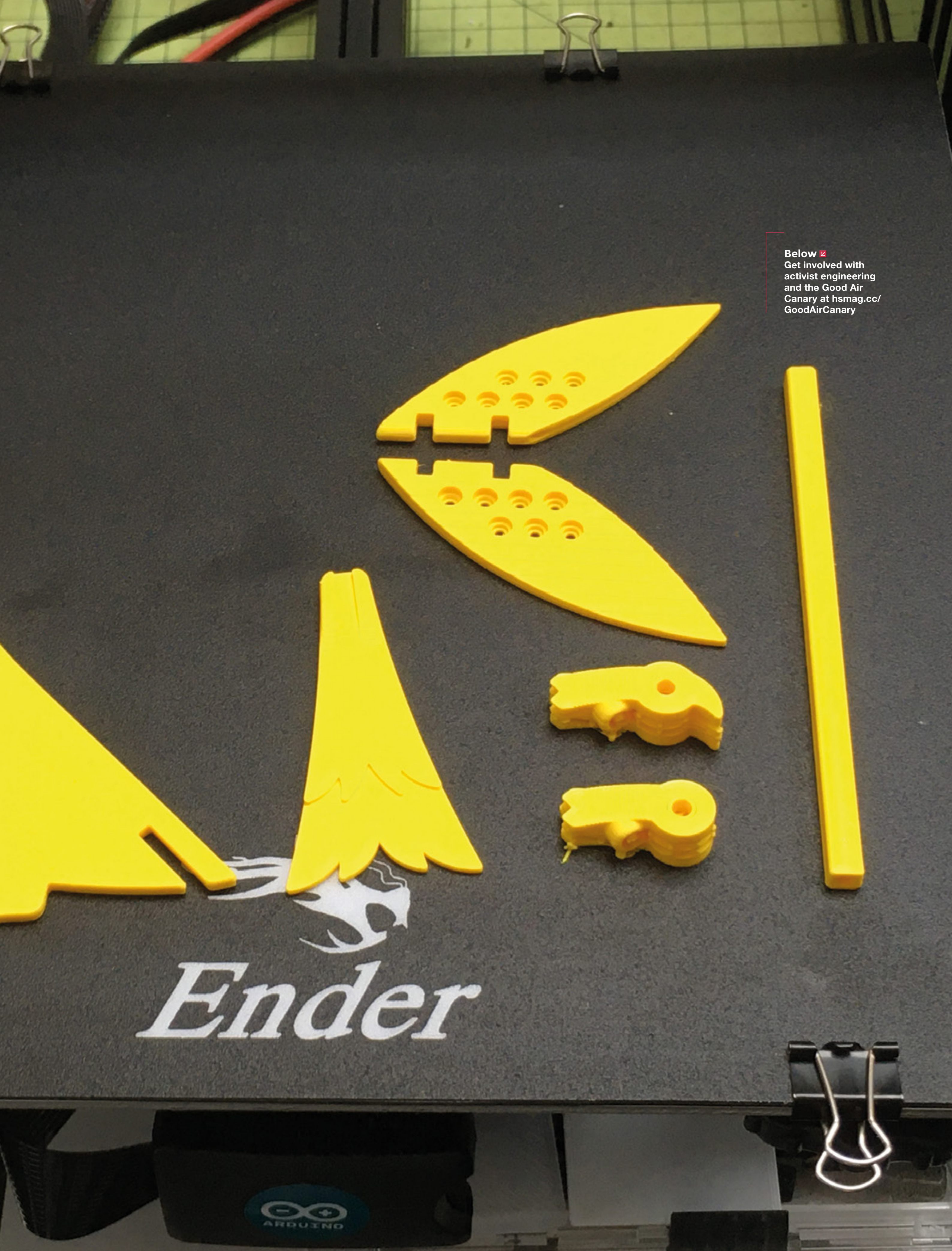
The ship has sailed to try and complain about Facebook and Twitter. I think the only way – I don't think you can correct those companies – is to create new companies that offer a better alternative. □

THE DOUBLE DIAMOND

A Double Diamond is basically two diamonds of expansion and contraction: the first one is kind of discovery and exploration. And the second one is about execution. The idea is that you broaden your thinking, understand the lay of the land, and then come back to what you think is the right decision.

When you start your career, you either get too scared of doing a bigger diamond, because you don't know how to pull it back in, or you have absolutely no self-restraint, and you just stay out and you never come back in properly. And so, you stay out in the ether. I think the mastery of a good designer is that you're able to go exceptionally wide, but you still know how to bring it all in on time, on budget, and on spec.





Below
Get involved with activist engineering and the Good Air Canary at hsmag.cc/GoodAirCanary

IMPROVISER'S TOOLBOX

BAMBOO

Lightweight and incredibly fast-growing, bamboo is an excellent material for all sorts of sustainable projects

Bamboo turns up in all sorts of unexpected places, from T-shirts to transport. Its speedy growth and biodegradability make bamboo attractive to anyone keen to make planet-friendly products – furniture, flooring, musical instruments, and even sunglasses.

It's a popular choice in the consumer accessories world, where the eco credentials of bamboo speakers, tablet cases, and earphones provide a notable contrast to the rare minerals used to make smartphones. If you like the idea of making your own speaker, bamboo-focused maker site Elegant Experiments has a straightforward walkthrough: **hsmag.cc/BambooSpeaker.**

Bamboo scaffolding, fencing, and hedges make use of bamboo's hardiness and reliably straight hollow shoots, as well as its low cost. Its use in gardens as privacy-providing sustainable fencing is not without its issues, due to some bamboo varieties' fast-spreading rhizomes that also make it hard to eradicate. Growing bamboo requires far less resources than nutrient-sapping, water-intensive cotton, however. Sustainable fashion often includes items made from bamboo fabric, or bamboo cotton yarn, with breathability giving the material an advantage for sportswear and summer clothing. Another great DIY project by Elegant Experiments details how to make your own bamboo clothes airer: **hsmag.cc/BambooClothesAirer**.

Bamboo has been cultivated for at least 7000 years, and grows on almost every continent, having evolved from prehistoric grasses. It splits into reliable strips, making it ideal for binding things together, and for construction projects such as these beehive houses in Ethiopia: **hsmag.cc/BambooHouse**.

Vehicle manufacturers, such as Ford, also see potential in bamboo as an eco-friendly construction option. This supercar débuted back in 2011: **hsmag.cc/BambooCar.** Although largely a proof of design concept, Ford and others see potential in developing vehicle exteriors from easy-to-replace bamboo. GOYTI has experts from the world of racing cars seeking to develop, not only formula-grade performance cars, but bamboo-clad electric vehicles that could provide low-impact transport in developing countries: **hsmag.cc/BambooComposites**. There's future mainstream potential too and, enough other designs have cropped up in the intervening decade since Ford's concept model, that there's now a Top 10 bamboo cars chart.

BAMBOO FOLDING CHAIR

Bamboo is an eminently practical choice for making furniture. The ideas used in this fairly straightforward bamboo and cord chair could equally be used for the base of a bamboo bed or sofa. Because it's naturally hollow, bamboo is easy to drill into and add hooks or rope to, or slot in thinner pieces of wood. The legs of this folding chair are pushed into place once circular holes have been gouged out of the thicker bamboo sections – no glue or nails are needed. Bamboo pegs are used instead, with excess length sawn off once they've been gently knocked into place with a hammer. After this, equal-sized spaces are measured out on the bamboo pieces that form either side of the seat, holes drilled through each side, and rope threaded through and tied off. A neat final touch is a rope loop added at one end so the foldaway chair can be hung up when not in use. →

Project Maker
TLC Channel

Project Link
hsmag.cc/BambooFoldingChair

Below Bamboo: the original tubular construction material

BAMBOO CASCADING FOUNTAIN

Project Maker
GARTH WEBB

Project Link
hsmag.cc/BambooFountain

Garth Webb got the inspiration for his lovely cascading bamboo fountain from a visit to a hardware store in Chicago, where he chanced upon a water pump that was going for a song. "I had no use for a water pump", he relates, "but American Science & Surplus makes you want to find uses for everything" – a very relatable comment! When he then noticed a stack of tall, sturdy bamboo canes at another store, he immediately realised the pump's potential. He split each 10 ft bamboo pole into half lengthways and then across the middle, forming channels for water to travel along. Shorter, narrower bamboo pieces were used to suspend either end of each bamboo trough, and nylon garden ties were used to secure them to Garth's garden fence. A water hose strung lower down the fence line delivers water to the top bamboo trough; at the other end, it empties into a small pond Garth filled with plants, but not before the family cat has had a chance to slurp a few tasty drops.

> "SHORTER, NARROWER BAMBOO PIECES WERE USED TO SUSPEND EITHER END OF EACH BAMBOO TROUGH"

Right
Bamboo: split it lengthwise and it's natural guttering

BAMBOO BASKETS

Weavers the world over seem to have found creative and practical uses for bamboo. This example from India shows how broad, flat strips of bamboo are splayed out in a circle and thinner pieces of plain and dyed bamboo are then threaded through the spokes to form a pliable basket-shaped structure. By leaning on a raised metal canister and moving the bamboo basket as she works, the weaver can get at successive spokes far more efficiently than if she had to work her away around the basket. New pieces of bamboo are added in by poking one end through the existing work to secure it. Once the design is complete, an edging pattern is worked around the top of the basket, excess bamboo strips tucked back on themselves, and any extra lopped off.

Project Maker
Indian Village Art

Project Link
hsmag.cc/BambooBaskets

Left The stiffest tree is most easily cracked, while the bamboo or willow survives by bending with the wind

BAMBOO DEATH STAR

Few of us would look at a piece of plywood and think, that seems like the perfect material with which to recreate an iconic spaceship. Frank Howarth considered bamboo's exceptionally light weight and decided the force was strong with this one. An experienced creator of small-scale architectural builds, with an enviably well-equipped workshop, Frank's spacecraft consists of two segmented halves. Each half has nine overlapping rings assembled from 13 rectangular bamboo segments – a figure chosen for its suitable evil connotations. The segments were fitted into aluminium formers before the layers were glued together and the edges were sanded down to create the Death Star's smooth, but imposing, exterior. An extra ring holds the two halves together where they meet. Frank used a CNC router to create the profile of the dish and the hole in the Death Star, figuring he had the most control over accuracy with this approach. Frank completed his build which, naturally, he revealed on May 4th. □

Project Maker
Frank Howarth

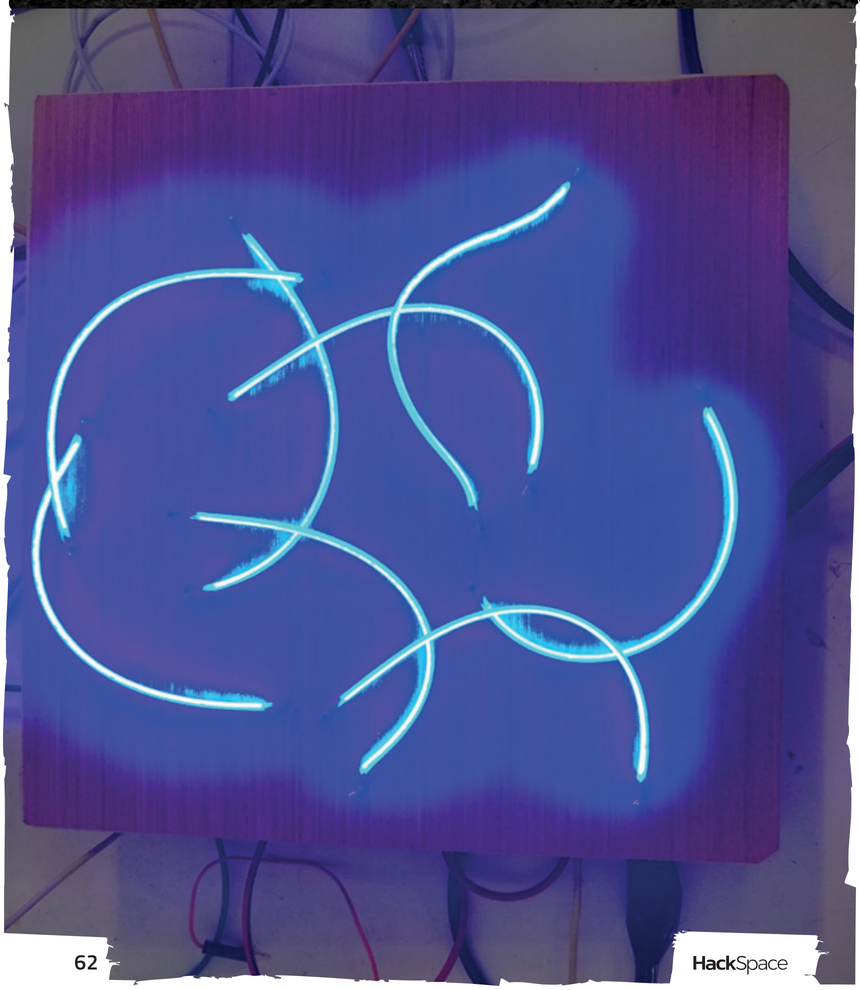
Project Link
hsmag.cc/BambooDeathStar

FEATURE

IN THE WORKSHOP: A fail and a success

By **Ben Everard**

Sometimes things don't go to plan. Sometimes they do



Left We're calling this style the Spaghetti Lamp

We've been playing with flexible LED filament for a few weeks. It's basically exactly what it says it is – strips of LEDs that are encased in flexible silicone, and wired up in parallel so you can run them at 3V. They're designed for making fancy lamps, as you can bend the filament into all sorts of shapes. We decided to try something a little more ambitious – a clock. By bending a filament into the shape of each number, and stacking the digits 0 to 9 on top of each other, we thought we could make something approximating a Nixie tube but with a bit more of a neon glow. A test with the digits 0,1, and 2 looked great and worked well (we only had three spare filaments when we did this test).

Time to crack on with the clock. There are probably makers out there who would have found a way of planning this. I don't know if there's a CAD program out there that lets you design with bendy things, but there probably is, and you could probably use it to find out just how hard this build would be.

The risk of ending up with spaghetti was obvious from the start, but we ploughed on in the naive optimism that I'd figure it out as I went. Well, spoiler alert – I didn't.

This build has unfortunately ended in failure. I got pretty close. If there were perhaps

Left I did my best reader, but it just didn't want to work

Below Key storage and bean slicing in one useful project

eight digits to stack on top of each other, rather than ten, it would might have worked, but I'm not quite ready for an octal clock yet.

So, where next for this build? I've got 40 blue, 13 cm flexible LED strips, and it'd be a shame to waste them. Since the build ended up looking like spaghetti, I decided to go with this and make a spaghetti light.

I wired them up in sets of four in parallel

Essentially, I just placed the filaments on a square sheet in a random-ish pattern, and wired them up. As with the previous lamp (see issue 52), I wired them up in sets of four in parallel. This meant I could drive it off a 12 V power supply, and I have loads of these lying around from old appliances.

The end result is interesting. As an object, I'm pleased with it. However, I'm disappointed the clock didn't work. I do feel like there's more that can be →

FEATURE

Most bean slicers follow the same basic design

done with these filaments than I've been able to do. They're a really interesting component that's a bit unlike anything I've worked with before. I really hope some other makers can experiment with them and find some tricks that have eluded me.

Fortunately, it hasn't all been failure this month. This month's experiment with upcycling had a bit more success.

SLICED BEANS

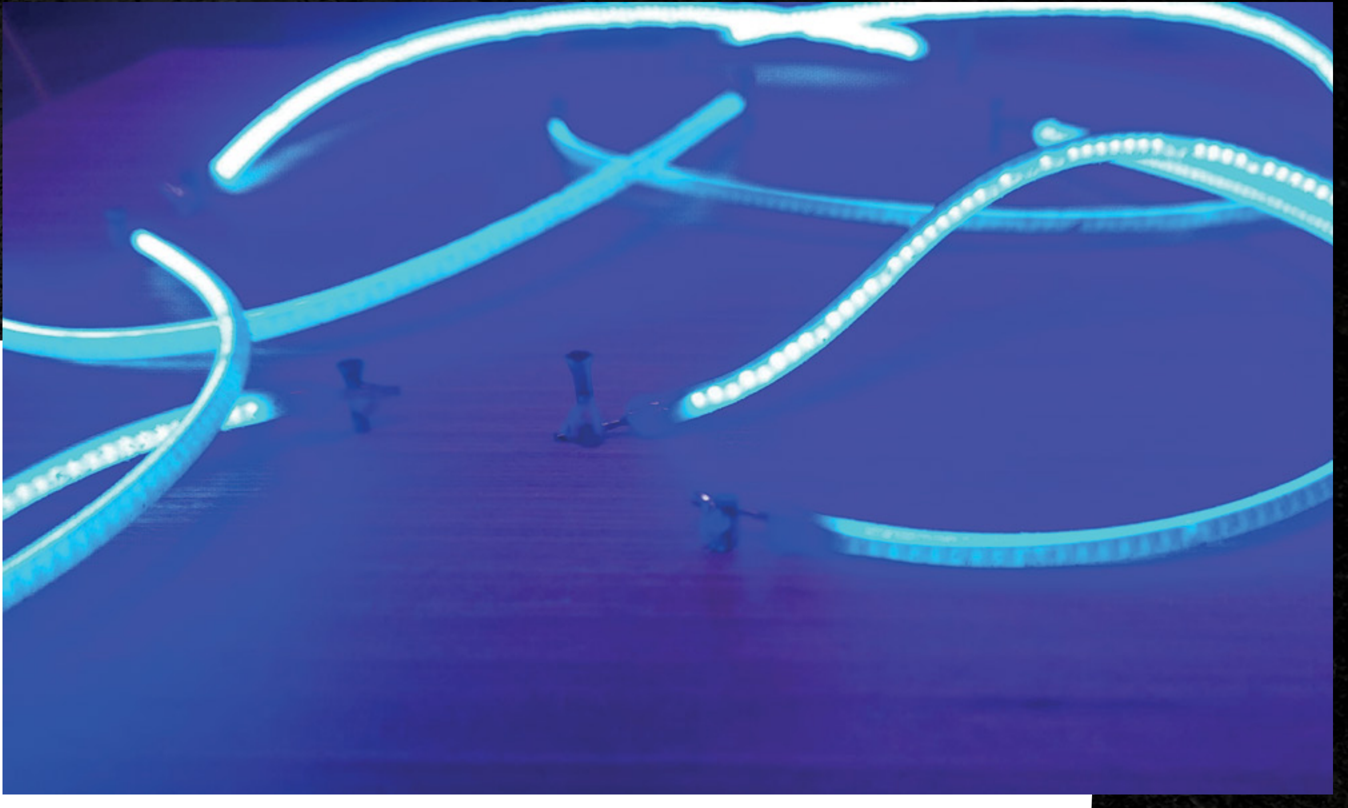
Runner beans were once part of the staple diet in the UK. They're easy to grow and can provide plenty of flavour and nutrients. They were so popular that bean slicers – tools to speed up the preparation – were once popular. These days, few people eat enough of these vegetables to keep a bit of kitchen equipment especially for their preparation, so these bean slicers are often for sale at charity shops and car boot sales.

Most bean slicers follow the same basic design – a cast iron body that clamps onto a work surface and has a handle to turn the three rotating blades. I really like the look of the cast iron housing, and it's something like this that I'd like to display. However, for it to find a permanent place in my house, it needs to have a function as well as a form.

It could certainly be turned into one of the upcycling classics – a lamp or a clock – but there's enough of these around the house (especially with the added Spaghetti Lamp). Also, I wanted to keep access to the bean slicer's original functionality. While I don't eat a lot of runner beans, I do have a few vegetable beds in the garden, and runner beans are one of the easy crops that the slugs don't destroy, so we usually have a few plants each year.

After mulling the options over, we decided on a small knick-knack holder. The place you can chuck your keys or small change when you get in so that

Right Is engineered wood the modern equivalent of cast iron?



you don't lose them. This can be mounted on a shelf or sideboard in a convenient location.

Keeping with the theme of upcycling, we made this from offcuts of plywood, rather than starting with new material.

We needed a way of adding and removing the tray. There were a few options for this. There is a bolt that goes through and holds the handle in place, so we could have attached something through this. We could have clamped something to the handle and somehow jammed it from spinning. However, after playing around with the options for a while, we decided to go with the simplest solution, which is to use a plug shaped to fit in the hole for the beans. We also braced it on the back of the slicer to make it more stable.

The first task was to make something the right size for the hole. It's a bit of an odd shape, so rather than try to model it, we just cut a bit of wood to approximately the right size, then brought it down to the final size on a sander by repeated trial and error. It's not perfect, but it's close enough to have a tight fit.

The pot at the top was made by cutting out a base and then some sides on a table saw. This could have been done using a lot of different techniques, but I've been watching too many woodwork videos on YouTube, and they seem to use table saws for everything and I haven't done a table saw project in a while. A few screws joined everything together, and I've got my finished pot.

There are lots of reasons to upcycle. This project, though, is all about aesthetic of the Industrial Revolution. Obviously aesthetics are a very personal preference, but many people enjoy the iconic looks of other time periods, and upcycling is a great way to enjoy these other aesthetics in the most authentic way possible. □

We decided to go with the simplest solution

Above If you look closely, you can see the individual LEDs

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

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

Build a digital-to-analogue converter

Output a range of voltages using just a handful of resistors

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.

Microcontrollers are great for understanding and producing digital signals. These electrical equivalents of 1s and 0s power the vast majority of our electrical devices. However, the world is not digital, so sometimes, if we're interacting with the world around us, we need our digital devices to use analogue signals. These, rather than being on or off, can range over a certain set of values.

Most microcontrollers have one or more analogue inputs which let you sample a voltage and see where in a range it falls. However, relatively few have the ability to output an analogue value. Fortunately, this is quite an easy feature to add. Let's take a look how.

If we have, say, seven digital outputs, we can use these to create an analogue output so that each digital value represents a 'bit' in final voltage.

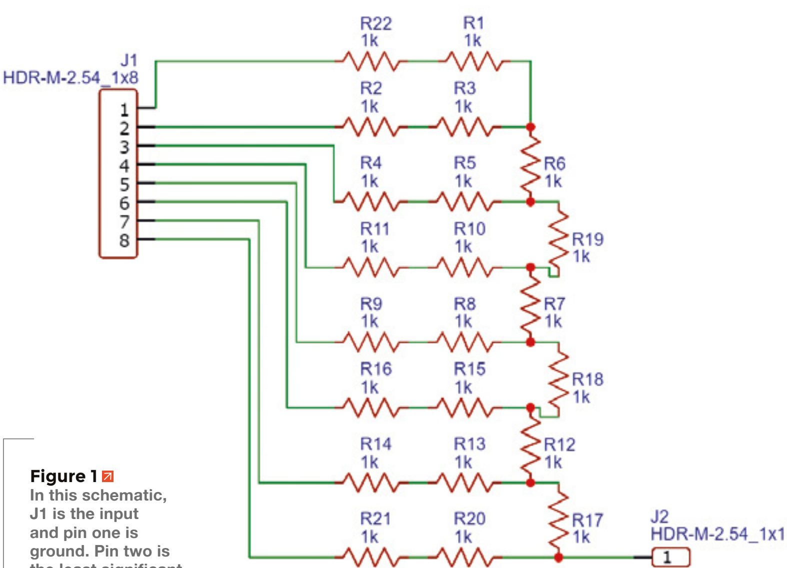


Figure 1 In this schematic, J1 is the input and pin one is ground. Pin two is the least significant bit, and it goes up from there

For example, if we wanted to output the lowest possible voltage (other than 0), we'd just switch the first digital output on. If we wanted to output the highest possible voltage, we'd switch them all on. This gives us 127 possible voltages we can output (representing every combination of digital outputs turned on).

If you're familiar with binary numbers, this should all make sense. Essentially, each output is 1 bit in the binary number that corresponds to the level of the outputs. For this to work, we need each output to contribute double the voltage of the proceeding output to the final voltage total, but how can you get a digital output to contribute a certain amount to a final output?

We won't dwell upon the theory, but the circuit is a simple network of resistors (see **Figure 1**) – this is called an R-2R ladder because it can be made with two values of resistor, and it doesn't matter what they are as long as one is twice the value of the other. As you can see in **Figure 2**, we've wired it up with just a single value of resistor and placed two of them one after the other to get twice the resistance where needed. There are no hard and fast rules for what resistor values to use. Lower-value resistors will give you higher current, whereas higher-value resistors will consume less power. Bear in mind, though, that you're never going to get much current out of an R-2R ladder, so you'll need an amplifier to boost the current if you want to drive anything that requires more than a few electrons.

The electronics, then, are pretty simple. Let's look at the code.

Basically, you just need to write your output level to the GPIOs as a 7-bit binary number. Some microcontroller programming languages have the ability to set GPIOs using a mask that would let you do this in a single command. However, MicroPython does not, so we have to do it bit by bit. Fortunately, it's not too hard:

```
import machine
import time

pins = []
for i in range(7):
    pins.append(machine.Pin(i, machine.Pin.OUT))

def output(number, pins):
    for i in range(7):
        if 1<<i & number:
            pins[i].on()
        else:
            pins[i].off()

while True:
    for i in range(127):
        output(i, pins)
        time.sleep(0.01)
```

This code lets us set a voltage, and we can cycle through the different voltages easily enough. However, what if you want to output a particular waveform?

With low-frequency waves, you can use **`time.sleep()`** or **`time.monotonic()`** to slow down a loop to the speed you want. However, this won't be particularly accurate because Python doesn't always have predictable speeds. There's some memory management that takes place behind the scenes that can cause occasional pauses. Also, Python's not the fastest language around, so you'll be quite limited by the speed you can get out with this.

Fortunately, RP2040 has a little trick up its sleeve: Programmed input/output (PIO). This lets you attach a state machine (basically a very simple processor) to some I/O pins and do some simple processing. Crucially for us, they have very accurate timings. We've looked at PIO a few times before, so we won't cover it in detail, but the program we load into the state machine is:

```
@asm_pio(out_init=(PIO.OUT_LOW, PIO.OUT_LOW,PIO.
OUT_LOW,PIO.OUT_LOW,PIO.OUT_LOW,PIO.OUT_LOW,PIO.
OUT_LOW,), out_shiftdir=PIO.SHIFT_RIGHT
def seven_bit_dac():
    label("loop")
    pull()
    out(pins, 7) [2]
    out(pins, 7) [2]
    out(pins, 7) [2]
    out(pins, 7)
    jmp("loop")
```

We use the instruction **`out(pins, 7)`** to send 7 bits of data from the output shift register to the output pins. Each time we do this, it shuffles the data in the OSR along 7 bits. The OSR is 32 bits long, so we can only do four **`out`** commands before we must refill it using **`pull`** (which gets data from the transmit FIFO). We use the **`jmp`** command to turn this into an infinite loop. If you're familiar with PIO, →

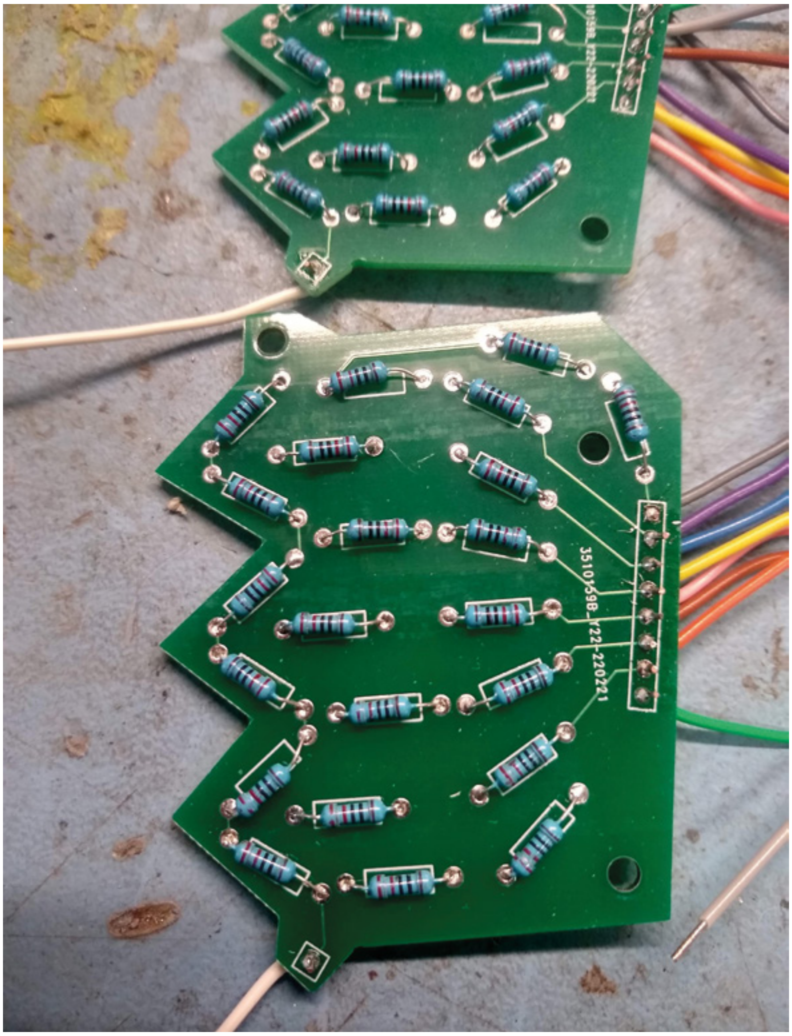
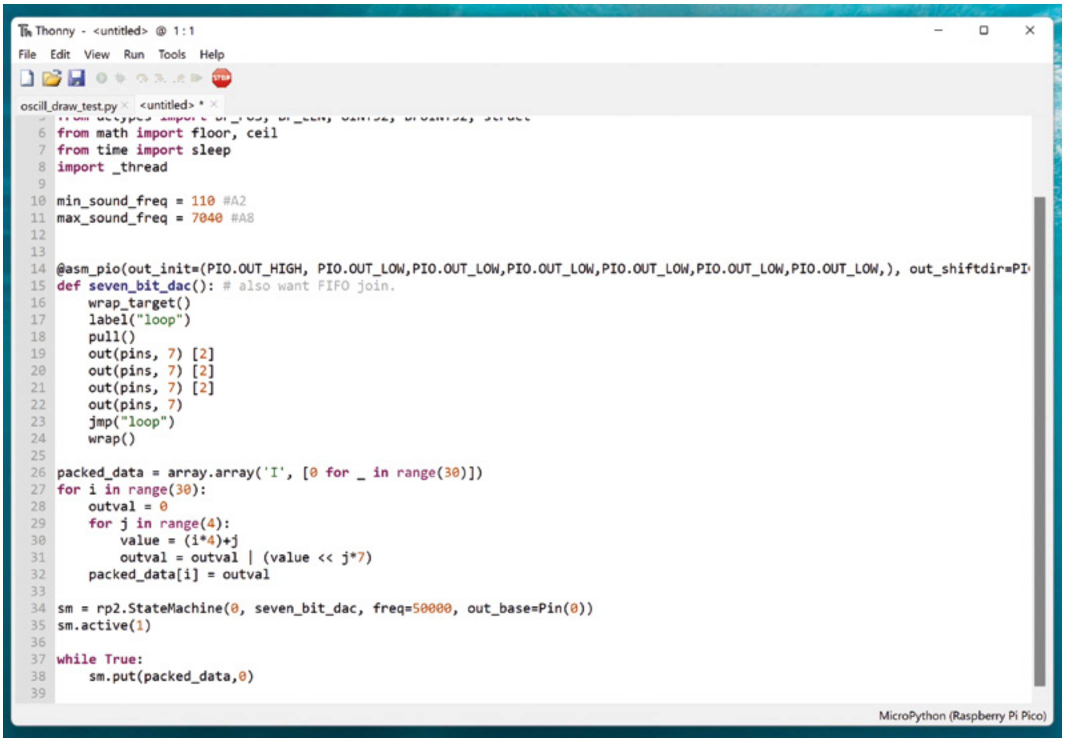


Figure 2
Just add lots and lots of the same value resistor

WHY 7 BITS?

We've built a 7-bit digital-to-analogue converter, but why this length? It's quite short and not even a standard number of bits.

Basically, it comes down to resistor accuracy. We're using a lot of resistors in this circuit, and resistors aren't perfect. In fact, they come with ratings that say what percentage of their final value they should be in. Seven bits is about the maximum accuracy you can get from 2% resistors. You can get more accurate resistors, but they're expensive, and if you need more than 7 bits, you'll probably have an easier time getting a self-contained DAC that you can control over I2C or SPI.



Right If you've not used MicroPython before, you can get started at hsmag.cc/WhatIsMicroPython

you might notice that we could speed this up using **`autopull`** and **`wrap`**, but actually, speed isn't our main concern. We want to output audio frequencies, and there's a limit to how much we can slow down the state machine, so a bit of slowness is a good thing here.

Every single instruction in a PIO program takes exactly one clock cycle to run. You can also add optional pauses as we have done here with **`[2]`** – these pause for two clock cycles. We've added these pauses so that the final instruction also has two additional instructions to run (**`jump`** and **`pull`**) before it gets back to the first **`out`**.

Our PIO program takes data in a slightly unusual format. We need to load the OSR with 32 bits of data which consist of four 7-bit numbers (plus four spare bits). We can create an array for this with the following:

```
packed_data = array.array('I', [0 for _ in
range(30)])
for i in range(30):
    outval = 0
    for j in range(4):
        value = (i*4)+j
        outval = outval | (value << j*7)
    packed_data[i] = outval
```

The standard Python notation for arrays are really called lists. They work well as they let you store all sorts of data and retrieve it easily. However, in our case, we need something more analogous to an array in C. That is one that has a very specific binary form. For this, we can use the array module. The array is created with a type code. 'I' in this case means 'unsigned integer' (you can see the full range of options at **hsmag.cc/Arrays**).

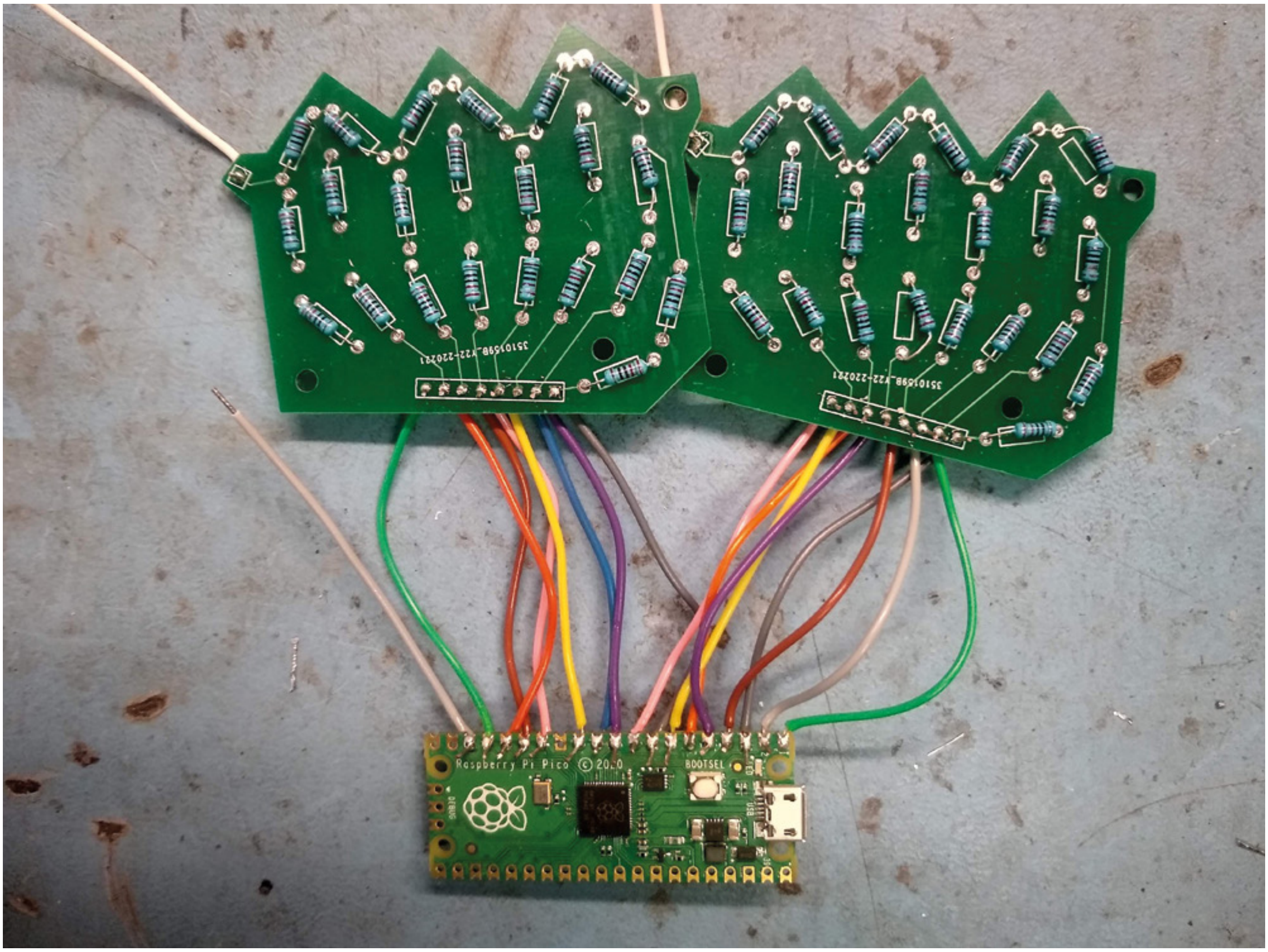
The above code creates a sawtooth wave that ramps up from 0 to 120 and then drops straight to 0. Each item in the array **`packed_data`** contains four 7-bit

MAKE IT PERMANENT

Resistor DACs are great when you need an analogue output and don't have many components to hand, but they can also be a more permanent solution. After all, you won't find many components cheaper than a handful of resistors.

An R-2R DAC is a great project to help you learn or practice your PCB design. There's only one component (well, two if you include the headers), and the routing is quite straightforward. This author has a slightly reckless disregard for the standard that PCBs should be rectangular and components placed at right angles to each other, so the one he's created for this article (and an upcoming project) can loosely be described as punk-jellyfish.

If you need a DAC, you can design your own or use (or tweak) this author's design, which is available at: **hsmag.cc/R2RDAC**.



values. We pack each of these four values with the two lines:

```
        value = (i*4)+j
         outval = outval | (value << j*7)
```

The `<<` operator shifts a value by a set number of binary digits. So, 1 << 2 would be 100 (in binary). The `|` operator is bitwise OR. This means that it ORs each bit in turn. We start with `outval` being 0, and anything ORd with 0 is itself. This lets us set each 7-bit block of `outval` independently. We use this to place each of the four 7-bit numbers at the right place in the 32-bit binary stored in `outval`.

We now have our data and a way of outputting our data. We just need to join the two together. This is done with the main loop that simply used the state machine's `put` method to output data as fast as possible:

```
sm = rp2.StateMachine(0, seven_bit_dac,
freq=50000, out_base=Pin(0))
sm.active(1)
```

```
while True:
    sm.put(packed_data,0)
```

The `put` method will stall if there isn't space in the FIFO for the data, so this will just feed the values into the state machine as fast as the state machine wants them.

We can set the frequency of the waveform by setting the speed of the state machine. This is done when we create it.

```
sm = rp2.StateMachine(0, seven_bit_dac,
freq=frequency, out_base=Pin(0))
sm.active(1)
```

That's all there is to it. You can see the full code at **hsmag.cc/DAC_PIO**.

In this example, we're just outputting sawtooth waves at a specific frequency, but you can use this code to generate all sorts of complex waveforms at a wide range of frequencies. The state machines can run at up to 125MHz, but the maximum frequency you can output will depend on the length of your wave in data points.

Above
You can connect more than one to a microcontroller at the same time. In this case, we've connected them to adjacent pins so that we can set both values via the same PIO program, but this time, setting 14 pins rather than 7 pins

Make an AI LEGO® face

Your robot face will be able to recognise and react to a range of objects

MAKER

James Robinson

Senior Learning Manager, Pedagogy and Training at Raspberry Pi. Ex-Teacher, spaceenthusiast.

raspberrypi.org

MAKER

Mark Scott

Marc is a former computer science teacher, and manages informal learning content for the Raspberry Pi Foundation.

raspberrypi.org

Over the course of this tutorial, you will construct a robot face from a combination of LEGO and electronic components. You'll then use an existing machine learning model to help your face recognise different objects and react to them.

You should build your robot face using whatever LEGO and other materials that you have available. We have used the LEGO SPIKE™ Prime kit (**magpi.cc/spikeprime**). This tutorial can also found online on the Raspberry Pi Foundation's website (**magpi.cc/legorobotface**).

01 Get started

Before you begin, you will need to have set up your Raspberry Pi computer and attached your Build HAT. Follow the instructions for Make a Build HAT Game Controller in *The MagPi* magazine, issue #113 (**magpi.cc/buildhatgamecontroller**).

As you are connecting the Build HAT to Raspberry Pi, don't forget to attach the Camera cable and feed the cable through the slots on both your Raspberry Pi and the Build HAT.

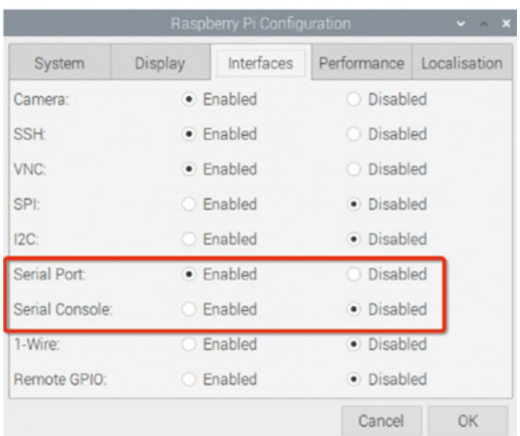


Figure 1 The Raspberry Pi OS Configuration settings used in this project

Once your Raspberry Pi has booted with the Camera Module attached, open Raspberry Pi's Configuration tool by clicking on the menu button and then selecting 'Preferences' and then 'Raspberry Pi Configuration'.

Click on the 'Interfaces' tab and set Camera, SSH, VNC, Serial Port to Enabled, and Serial Console to Disabled (as shown in **Figure 1** – Configuration Settings).

You can find instructions on how to set up the camera guide here (**magpi.cc/getstartedcam**). If using Raspberry Pi OS 'bullseye' enable Legacy Camera mode (**magpi.cc/legacy**).

02 Test the machine learning model

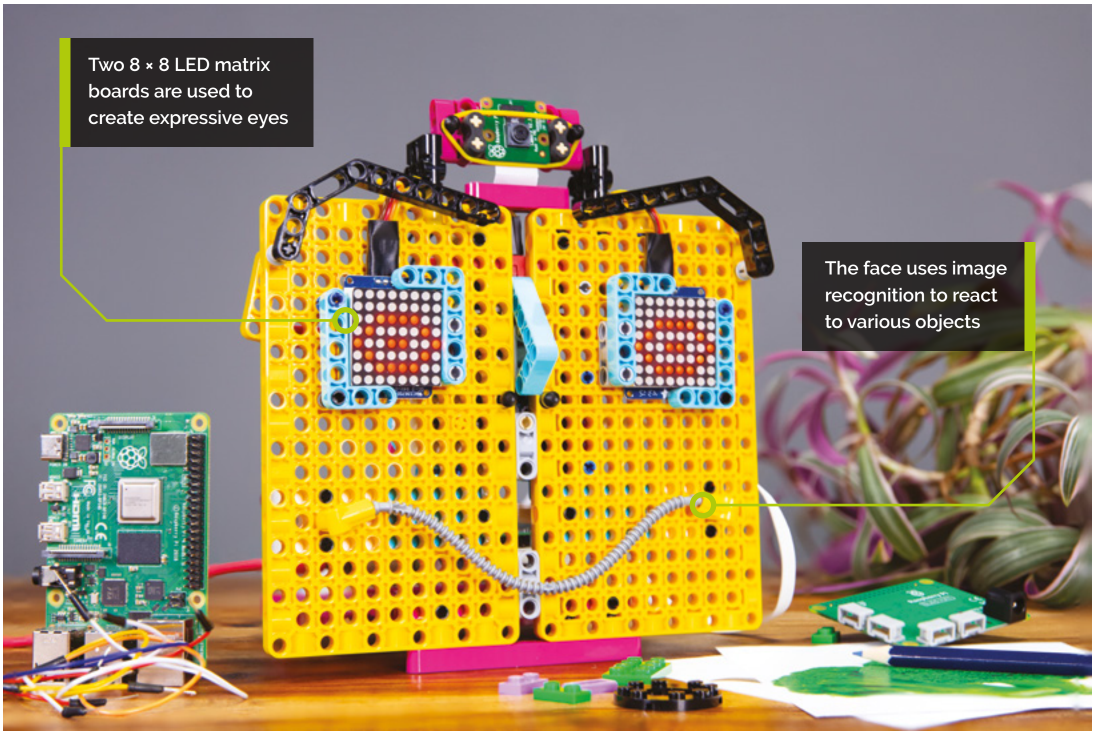
Your first step is to understand and test how you can use a machine learning model to recognise objects. For this project, you won't be creating and training your own model, but using an example model that can recognise a range of objects.

Computers do not have a natural ability to learn. Most things that computers do have been directly programmed by a human. This makes them great for tasks that have a few clearly defined rules, but they struggle with more human-like tasks such as recognising different objects. Using machine learning, a computer can be shown thousands and thousands of images, each of which has been labelled. Gradually, the program can learn the characteristics of a group of images and then give them the correct label. The end result of this process is called a model. Once trained, models can be used in the real world to perform tasks.

To get started, download the resources for this project to your Raspberry Pi from GitHub (**magpi.cc/legorobotfaceresources**).

You'll find a range of files that will be useful for the project, but for this step, you're going to use:

- **model.tflite** – The machine learning model file



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**

- **labels.txt** – Labels for each object the model can recognise
- **classifer.py** – A Python program to test the model

Open Thonny, which is under the Programming category in your Raspberry Pi applications menu.

Open and Run the **classifier.py** program. Your Raspberry Pi will display what the camera is 'seeing' and the name of the main object in view that it recognises. Try presenting the camera with different objects, and investigate which ones it can recognise with confidence.

Find at least four objects (or images) that your camera can recognise reliably – you'll need them for your machine learning model.

> "Once trained, models can be used in the real world to perform tasks"

03 Use emoji for your robot face

The goal is to build a robotic face that can respond to objects that it recognises. If you break that down into smaller steps, you might say that your robot face will:

1. Use the Raspberry Pi camera to look for objects
2. If an object is detected, use that object to change the face
3. Match the object detected to a reaction or emotion
4. Change the look of the face to represent a reaction
5. Return to step 1 to look for the next object to react to

For the project to work, it's going to need a selection of reactions that it can display using simple facial expressions. Emojis are a great example of this.

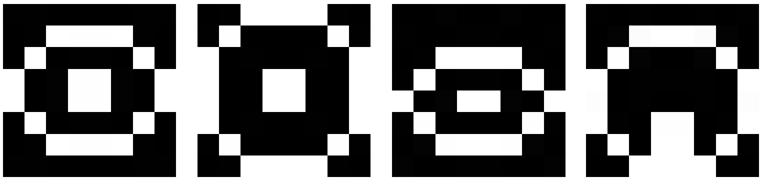
An emoji is an example of an abstraction, a simplified representation of a real face. All of the complexity has been removed and limited to the simple key parts of the face. In this project, you can use four emojis to represent eyes: Neutral, Wide, Angry, Look Down (see **Figure 2**, overleaf).

04 Connect objects to the expressions

From your experiments in the previous step, you will have identified at least four objects that your camera and machine learning model can reliably detect. →

You'll Need

- A Raspberry Pi computer
- A Raspberry Pi Build HAT **magpi.cc/buildhat**
- A Raspberry Pi Camera Module
- 2× Small LEGO Technic motors
- 1 × Large LEGO Technic motor
- 2 × Adafruit 8×8 LED matrices **magpi.cc/88ledmatrix**
- Assortment of LEGO (we used a selection from the LEGO SPIKE Prime kit)
- Jumper leads, mini breadboard, and soldering kit



▲ **Figure 2** The four emojis used to represent eyes

Choose which objects will trigger which reactions in your robot. Each expression should have a reaction associated with it. For our example, we used the following:

Object	Reaction
Broccoli	Neutral
Teapot	Wide
Snake	Angry
Hotdog	Look Down

05 Making a face

Now that you know which objects will trigger which responses, you need to build a robotic face that can be animated to show the different expressions.

To build this robot face, we used parts from the SPIKE Prime and SPIKE Expansion sets. However, most of the parts can be found in many existing LEGO sets. To construct this model, or inspire your own, download Raspberry Pi's handy build guide (**magpi.cc/robotfacebuildguide**). This Sketchfab model is what your face will look like (**magpi.cc/sketchfabface**).

06 Add the eyes

The LED matrices used in the example face are connected to Raspberry Pi's I2C interface. Devices using I2C are connected using a specific number called an address. As you are using two matrices, each will need its own address.

Before you connect them up, you need to follow the relevant assembly instructions (**magpi.cc/matrixassembly**). Assembly of the LED arrays requires some soldering, so get permission from an adult before you use any tools. You can follow our soldering guide at **magpi.cc/soldering**.

The matrices used in this project all come with the same address, meaning that for two to work together, one of them needs a new address. For this, some more soldering is needed.

Using your soldering kit, close the A0 connection of only one of your matrices (see **Figure 3**).

Place the eyes into the square sockets on your robot face; use elastic bands to secure them and make sure the pins are at the top (see **Figure 4**).

Now that the basic construction of the robot face is complete, you need to add your Raspberry Pi and connect your components to it.

> "Most of the parts can be found in many existing LEGO sets"

07 Add Raspberry Pi

For this project you'll ideally want to use the Maker Plate element to mount your Raspberry Pi and Build HAT. To see a visual guide with photos of this step take a look at the build instructions online (**magpi.cc/robotface5**).

Mount your Raspberry Pi onto the Maker Plate using M2 bolts and nuts, making sure Raspberry Pi is on the flat side.

Before adding the Build HAT, you'll first need to attach the camera ribbon cable to the Raspberry Pi and thread it through the hole in the Build HAT. If you haven't already connected the camera board to your Raspberry Pi, you can do so by following the instructions at 'Getting started with the Camera

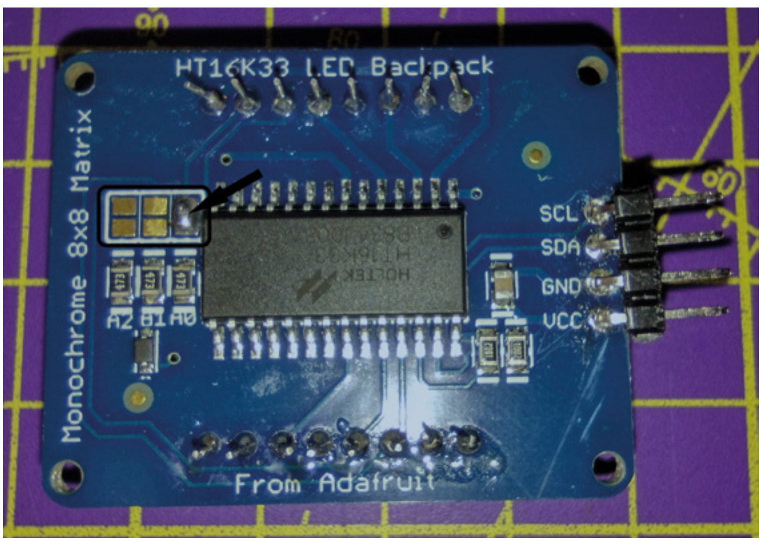
Top Tip

Install Build HAT

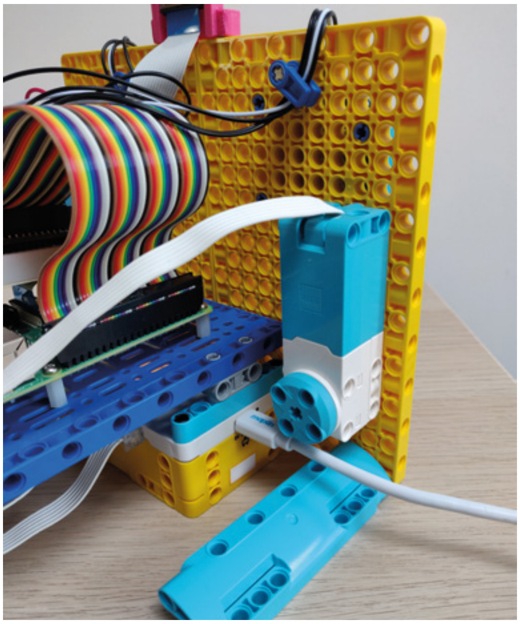
Open a Terminal window on your Raspberry Pi by pressing **CTRL+ALT+T**. At the prompt, type:

```
sudo pip3
install buildhat
```

Press **ENTER** and wait for the 'installation completed' message.



▼ **Figure 3** Soldering the Adafruit LED Backpacks



▲ LEGO Technic motors are used to adjust the eyes and mouth

Module' (**magpi.cc/getstartedcam**). Leave the camera ribbon connected to the Raspberry Pi, but remove the camera board from the loose end of the ribbon by pushing up the small black clip and sliding the ribbon out. Poke the ribbon through the underside of the Build HAT and out through the top, making sure the ribbon isn't twisted.

Connect the Maker Plate to the back of your robot face using some black studs. Next, connect up your small LEGO Technic motors to ports A and B, ready to control the mouth.

Connect up your large LEGO Technic motor to port C, ready to control the eyebrows.

To connect the pair of eyes to Raspberry Pi's GPIO, they first need to be connected together using a breadboard, and then to the GPIO pins from the breadboard.

Use eight male-female jumper wires to connect the four pins from each eye together on the breadboard. Make sure that both VCC pins are in the same row of the breadboard, both GND pins are in the same row, and so on. Then connect to the 3V3, GND, SDA, and SCL pins on the Raspberry Pi, as shown in **Figure 5**.

Your robot face is now built, connected, and ready to be programmed!

08 Motorise the mouth

Open Thonny from the Programming menu and save a new file called **robot_face.py**, in the same directory as **classifier.py**, **labels.txt**, and the 8×8 pixel art images. Now enter the Python code from the **robot_face.py** listing. →

robot_face.py

DOWNLOAD THE FULL CODE: magpi.cc/robotfacepy

Language: **Python 3**

```
from buildhat import Motor
import board
from adafruit_ht16k33.matrix import Matrix8x8
from PIL import Image
from classifier import Classifier
from time import sleep

## Set up the motors
mouth_r = Motor('A')
mouth_l = Motor('B')
eyebrows = Motor('C')

## Move the motors to 0 position
mouth_r.run_to_position(0)
mouth_l.run_to_position(0)
eyebrows.run_to_position(0)

## Set up the eyes
i2c = board.I2C()
left_eye = Matrix8x8(i2c, address=0x70)
right_eye = Matrix8x8(i2c, address=0x71)

## Link names of expressions to images in the Resources directory for the eyes to display
neutral = Image.open("neutral.png").rotate(90)
wide = Image.open("wide.png").rotate(90)
angry = Image.open("angry.png").rotate(90)
look_down = Image.open("look_down.png").rotate(90)

## Link names of expressions to motor movement and to eye display in a dictionary
faces = {
    "neutral":{"mouth":0, "right_eye":neutral, "left_eye":neutral, "eyebrows":0},
    "happy":{"mouth":45, "right_eye":wide, "left_eye":wide, "eyebrows":-150},
    "angry":{"mouth":-20, "right_eye":angry, "left_eye":angry, "eyebrows":150},
    "sad":{"mouth":-45, "right_eye":look_down, "left_eye":look_down, "eyebrows":-40}
    }

## Use the classifier.py to recognise different images (file is in resources directory)
seen_items = Classifier(label_file="labels.txt", model_file="model.tflite",threshold=0.5)

```

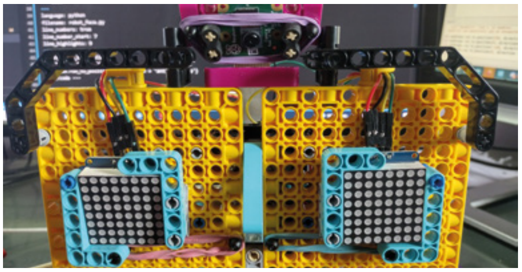


Figure 4 Rubber bands are used to hold the eyes in place

It starts with the import you will need to control the LEGO Technic motors (**robot_face.py** line 1).

Create two new objects for the left and right motors. In this example, the right motor is connected to port A and the left to port B (line 9 and 10).

Both motors should start in the 0 position when the program starts. The `move_mouth` function (line 43) will move the mouth motors. They need to turn in opposite directions, so the left motor will turn to a negative value and the right motor to a positive value. Adding `blocking=False` will make both motors turn at the same time.

> "The third motor is used to move the face's eyebrows"

09 Program the eyebrows

The third motor is used to move the face's eyebrows. The `eyebrows.run_to_position(0)` object on line 16 is used for the eyebrows' motor.

Make sure that your large motor is positioned so that the lollipop and the circle are aligned, and that your face's eyebrows are set horizontally. If they are not, you may need to adjust your build a little.

You set the motor to turn to the 0 position when the program starts.

There are three eyebrow positions shown here, but you can create more.

- 0 will make the eyebrows appear horizontal
- 150 will lower the eyebrows
- −150 will raise the eyebrows

The `move_eyebrows` function (line 49) gets the current eyebrow position, and if the position it is supposed to move to is less than the current one, it will move anticlockwise; otherwise it will move clockwise.

robot_face.py (cont.)

Language: **Python 3**

```
040. ## Set reactions for different objects that are recognised
041. reactions = {"broccoli":"neutral", "teapot":"happy", "Indian cobra":"angry", "hotdog":"happy"}
042.
043. def move_mouth (position):
044.     '''Move the mouth to value of position parameter'''
045.     mouth_l.run_to_position(position * -1, blocking=False)
046.     mouth_r.run_to_position(position, blocking=False)
047.
048.
049. def move_eyebrows (position):
050.     '''Move the eyebrows to value of position parameter'''
051.     current_position = eyebrows.get_aposition()
052.     if position < current_position:
053.         rotation = 'anticlockwise'
054.     else:
055.         rotation = 'clockwise'
056.     eyebrows.run_to_position(position, direction = rotation)
057.
058. def change_eyes(left, right):
059.     '''display the PIL objects on the left and right eye'''
060.     left_eye.image(left)
061.     right_eye.image(right)
062.
063. def set_face (face):
064.     '''call all functions that change the expression, according to the face from the faces dictionary'''
065.     change_eyes(face["right_eye"],face["left_eye"])
066.     move_mouth(face["mouth"])
067.     move_eyebrows(face["eyebrows"])
068.
069. ## Loop forever and check the list of seen items and set the correct face if the object has been seen
070. while True:
071.     sleep(1)
072.     if seen_items.item != seen_items.last_item:
073.         item = seen_items.item
074.         if item in reactions.keys():
075.             set_face(faces[reactions[item]])
076.     sleep(1)
```

Top Tip

Raspberry Picamera

The Python Picamera module is currently not, by default, compatible with the latest version of Raspberry Pi OS (called Bullseye). To use the Picamera module, you will need to enable legacy support for the camera Open a Terminal window and type the following command:

```
sudo raspi-config
```

Set Legacy Camera to Enable under Interface Options. See **magpi.cc/legacy** for more information.

10 Program the eyes

The LED matrices can show 8×8 pixel images on their displays. These can be used to display different motions of the eyes.

The three libraries on lines 2, 3, and 4 enable you to display images on the LED displays.

```
import board
from adafruit_ht16k33.matrix import Matrix8x8
from PIL import Image
```

The `i2c`, `left_eye`, and `right_eye` objects (lines 19, 20, and 21) control the eyes. For now, the images on each eye will be the same, but you can adjust your code later, if you want to use different images on the different displays, depending on which one you soldered the A0 pads on.

Using the PIL library, the images are opened and stored (starting line 23).

```
neutral = Image.open("neutral.png").rotate(90)
wide = Image.open("wide.png").rotate(90)
angry = Image.open("angry.png").rotate(90)
look_down = Image.open("look_down.png").
rotate(90)
```

The `change_eyes(life, right)` function changes the eyes that are displayed on the LEDs.

11 Changing faces

Now it's time to bring all your different functions together to change the whole face. The `faces` dictionary is used to store the different facial expressions you want to use. This will give values for the mouth motors, the eyebrow motor, and the eyes (lines 30 to 35).

And the `set_face(face)` function (line 63) is used to set the mouth, eyebrows, and eyes.

The classifier.py program detecting broccoli

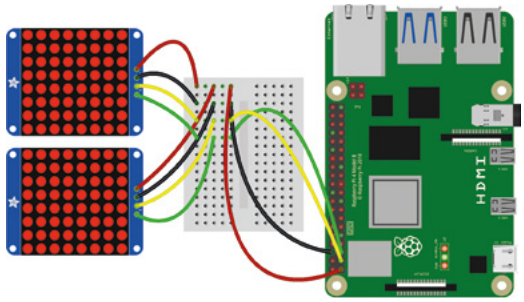


Figure 5 Wiring up Raspberry Pi

12 Emotional responses to objects

To finish off the project, you can make the robot face display different expressions depending on what the camera can see.

Start by importing the `Classifier` class from the file that you used earlier, along with the `sleep` function (lines 5 and 6).

Then create a `seen_items` list of objects that the classifier can recognise (line 38). You can adjust the threshold later to make the classifier more or less accurate with its recognition.

The `reactions` dictionary (line 41) links objects to different emotions. You can choose your own objects and emotions depending on what you have around you, or images that you have printed out.

Lastly, the `while` loop (line 70) checks what is in the `seen_items` list every two seconds, and then display the facial expression according to your `reactions` dictionary.

> "Add more expressions by altering the rotation of the motors"

13 What next?

There are lots of ways to extend your robot face project. You can add more objects that the face can recognise, and add more expressions by altering the rotation of the motors and the images displayed on the LED matrices.

There is also the possibility of adding sound to your project, using Pygame to play laughs, screams, and sighs when the face recognises different objects.

Or, how about taking all you have learned in this path and creating something original and new!

If you've completed all the LEGO Spark projects and want to have more fun exploring Python, then you could try out any of these projects.

Communication by ground conduction

Mike Bedford unveils an unusual form of communication, and shows you how to try it out yourself

Mike Bedford

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

'Father of radio' – many of us have heard the phrase, and it refers to the Italian electrical engineer Guglielmo Marconi, right? Not according to the residents of the small city of Murray, Kentucky. They claim the title for one of their own sons, melon farmer and inventor Nathan B Stubblefield. His inventions didn't really involve radio, as we now understand the word, but he did demonstrate wireless communication, in the broadest sense of communication, with no wires between the transmitter and receiver.

Right Nathan B Stubblefield's invention might not have truly involved radio, but it did permit communication without wires

Stubblefield experimented with a wireless technology called inductive communication, although our subject here is another of his inventions. It's been given various names – including ground conduction and earth current communication – and it involves injecting a signal into the ground through a pair of earth electrodes at the transmitter, and receiving it some distance away using another pair of electrodes. Nathan Stubblefield demonstrated it to the press on his farm in 1902, and soon after in Washington DC to prospective financial backers. He was able to provide speech communication over a range of a mile with reasonably portable equipment. A similar method was later used in the First World War, because it was less susceptible to damage than telephone cable on the battlefield, and some radio amateurs experimented with it during the Second World War when their transmitting licences were revoked. And it's no less interesting today, as we'll soon discover.

A BIT OF THEORY

Since the ground conducts electricity, especially if the soil's wet, it's not hard to imagine how a telephone system could use a wire as one of the two conductors between the transmitter and receiver and the ground as the other. That wouldn't be a wireless system, though, and if we try to use the ground in place of both conductors, things get more complicated.

An electrical current flows through the ground between the two transmitter electrodes in an infinite number of paths, and these aren't just on one side of the electrodes, as shown in **Figure 1**, but they're also on the other side and they flow deep into the ground too. At the receiver, the electrodes intersect with the flow of an earth current and, as a result, a voltage appears across them and is amplified in the receiver.

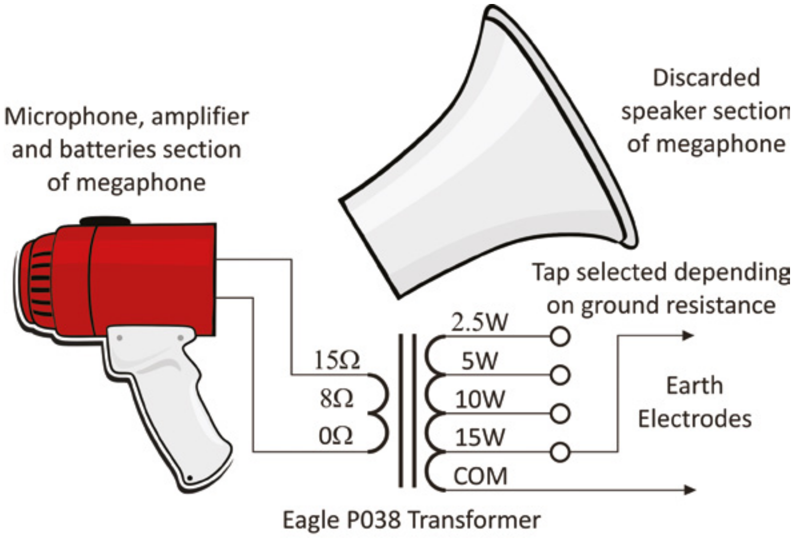
As the receiver is moved further from the transmitter, the signal gets weaker. This is exactly what we'd expect because it's just like radio or pretty much any type of communication. Conversely, as the two electrodes at the transmitter are moved further apart – and the same applies at the receiver – the signal gets stronger. So, although you'll just have to get used to the signal getting weaker as you increase the range, you can improve things by moving the two electrodes at the transmitter or receiver further apart. You're going to have to experiment to find the ideal setup, and it'll vary from place to place, but ten metres or more would be a good starting point for the separation between the transmitter electrodes and between the receiver electrodes.

> **You can improve things by moving the two electrodes** at the transmitter or receiver further apart

BUILDING THE KIT

If you fancy getting some hands-on experience of ground conduction communication, we've got good news – it's not difficult. The transmitter arrangement is microphone > amplifier > earth electrodes, and the receiver arrangement is earth electrodes > amplifier > speaker. You can buy audio amplifiers as cheap modules – but you'll need both a pre-amplifier and a power amplifier, and check that they're compatible with each other and with the microphone or speaker – so we've come up with an easier solution. For the transmitter, we bought and modified a cheap megaphone which cost less than £10 and, for the receiver, we used a laptop with an external USB sound card. Here's the lowdown.

Our megaphone was a Pulse MP20 which we bought from CPC, although we assume most products at this sort of price will be similar. Having tried it out while it's still intact, so you know how to use it, you need to dismantle the megaphone to the extent necessary to remove the speaker. Having done that, what you'll →



Above The transmitter is made from a dismantled megaphone and the matching transformer

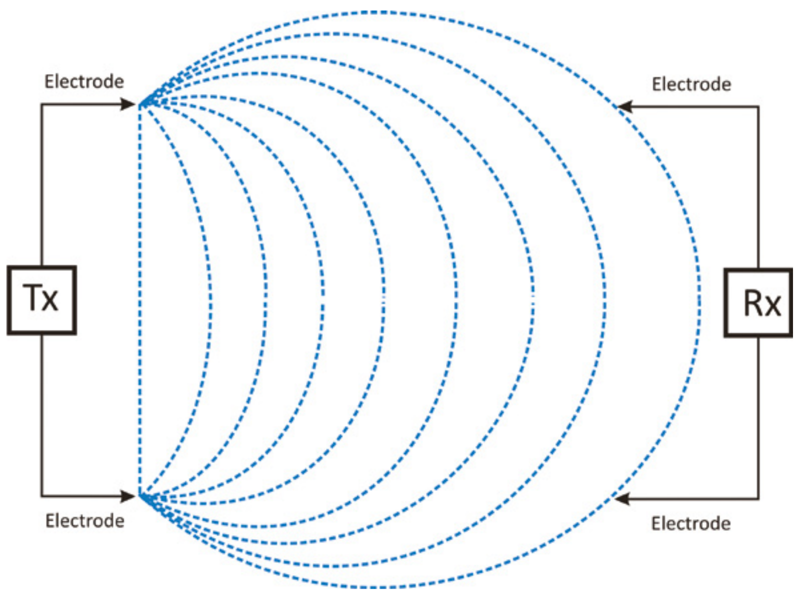
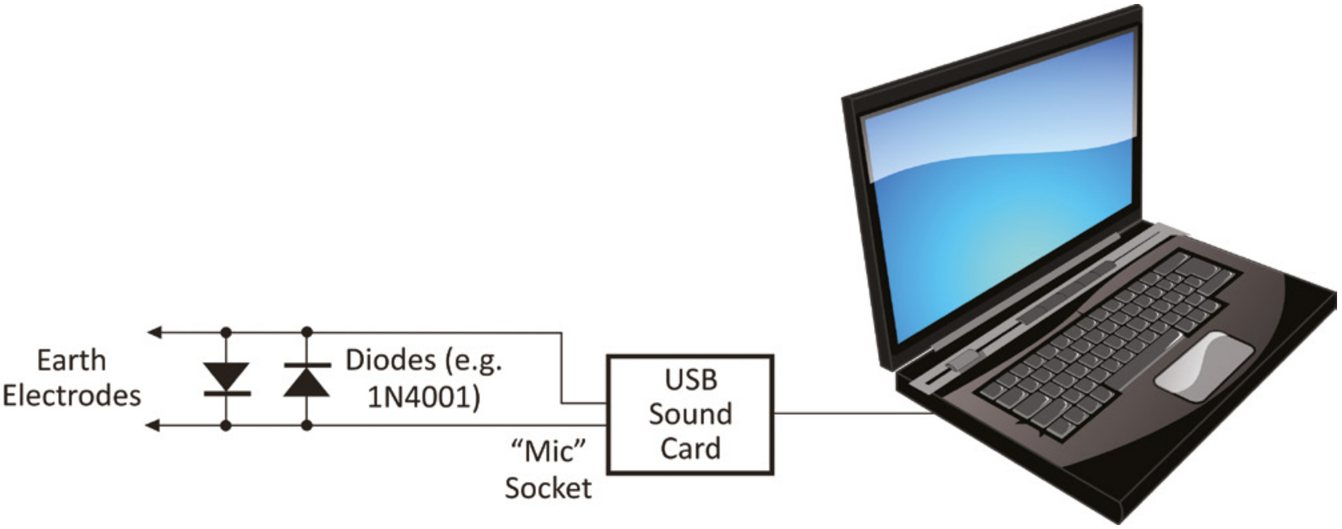


Figure 1 Current flows between the transmitter electrodes, and spreads out, so a voltage is detected by the receiver electrodes

USE A LOOP ANTENNA

Something else that you might like to try is to use a loop antenna instead of the earth electrodes at the receiver. Performance will be better with a large diameter loop and lots of turns, but just to see that you can receive with a loop at short range, we suggest you use a one-metre diameter loop, with 20 turns or so. The loop could easily be made from four metres of ribbon cable, made into a square, and with each conductor connected to the next one where the two ends meet. It would be useful to use a couple of rods, of some sort, to hold it in shape.

If you can receive a signal on a loop antenna that isn't connected to the ground, it might seem that a ground conduction transmitter generates a radio signal. However, it doesn't. Without getting into the details, let's just say that the signal that the loop antenna receives is mostly a magnetic field, rather than a proper radio signal.



Above The receiver only needs a laptop but, as a precaution, we suggest you use an external USB sound card

have left is the remains of the megaphone which will have two unconnected wires that had previously been connected to the speaker. These are the wires that you'll connect to the electrodes, but not directly. That's because the megaphone's speaker will probably have an impedance of 4 or 8 Ω (ours was 4 Ω), but the resistance between the transmitter electrodes could be 100 Ω or more. If it were 100 Ω, our 10 W megaphone would transmit less than half a watt into the earth, so the range wouldn't be very good. You might like to try out a direct connection to start, at short range, but to improve the range you need to do something different. The solution is to use a transformer and, in particular, we suggest a 100 V line audio transformer, for example, the Eagle P038. If you connect the transformer's 15 Ω tap to the megaphone, the 15, 10, 5, and 2.5 W taps will match the megaphone's 4 Ω output to 178, 267, 533, and 1067 Ω respectively, as we learned by experimenting.

Turning to the receiver, all you need is a laptop, so long as it has a microphone input. Ours didn't, so we used an external USB sound card, and that's still a

Right The first job in building the transmitter is to dismantle the megaphone so that you can disconnect the speaker

good idea even if your laptop does have a microphone input, since any high voltages that might be picked up by the ground electrodes could damage your laptop's internal circuitry. So, as an extra precaution, we suggest wiring two diodes across the microphone inputs, i.e. between the electrodes, one polarised oppositely to the other – pretty much any diode will work. This method should protect the USB input on your computer; however, we recommend you start with a low-price computer like a Raspberry Pi, rather than a laptop costing several thousand pounds. The electrodes and associated diodes should be connected to a 3.5 mm jack so it can plug into the microphone socket on the sound card, and you need to configure the laptop so whatever signal is present at the microphone input drives the speaker or, so you can easily hear outside if it's windy, it could drive headphones. Under Windows, do that by selecting 'Listen to device' at Control Panel > Sounds > Recording > double-click on USB microphone > Listen. Alternatively, if you're experimenting alone, so nobody can listen to the laptop while you're speaking into the megaphone, you can use your laptop's audio recording facility.

TRYING IT OUT

With the equipment ready to go, all you need are the ground electrodes. These are metal rods of some type that you can drive into the ground, and two lengths of wires to connect the electrodes to the transmitter or the receiver. The wires are easy: they just need to be long enough to allow the electrodes to be sufficiently far apart, and terminated at one end with a crocodile clip for easy connection to an electrode. However, the electrodes need a bit more thought.

The job of an electrode is to make a connection to the ground, but it's important to minimise the resistance between the electrode and the ground. →

Above
Cave rescue teams use ground conduction systems for through-rock communication, but they operate with an LF carrier
Photo: Bartek Biela

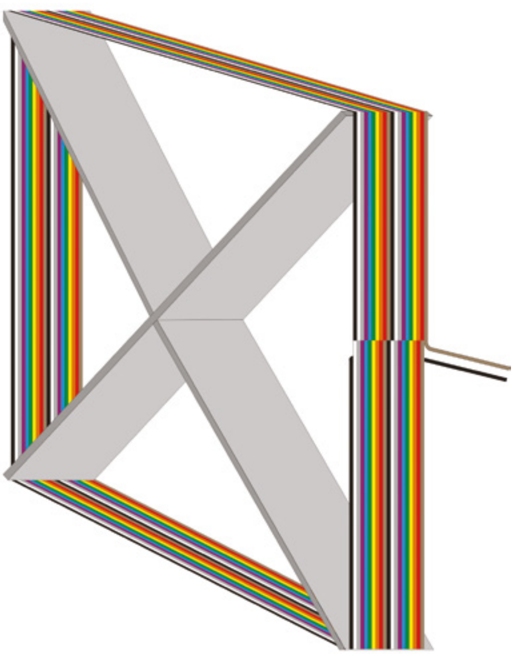
USING A HIGHER FREQUENCY

If you build a ground conduction setup like the one we've described, you'll find there's lots of hum. This is due to interference from mains electricity, and one way of eliminating that is to move away from using a pure audio signal. If you modulate the audio onto a higher-frequency carrier, you'll get rid of the hum, but you'll need much more complicated equipment, essentially a radio transmitter and receiver. Not only that but, although ground electrodes aren't the most efficient antennas imaginable, as the frequency increases, you'll start to generate a radio signal which, unless you have an amateur radio licence, would be illegal.

You might be interested to know that cave rescue teams use this technique so underground rescue personnel attending to a casualty can remain in contact with the surface controller, and perhaps ask for a doctor to be present. This is achieved using low frequencies which can transmit through the rock. At a typical frequency of 87kHz, though, because the wires to the earth electrodes are so short compared to the 3.5 km wavelength, very little radio signal is transmitted, so there are no legal concerns.

Above
There are lots of options for the electrodes, but these L-section tent-pegs work quite well

Right
As an alternative to earth electrodes, a loop antenna could be used at the receiver



The resistance is reduced by increasing the electrode's surface area, and one way to do that is to use a long electrode. Earthing rods used for electrical installations can be over a metre long, but we don't recommend them. The snag is that you might not find it easy to drive them all the way into the ground and, even if you do, you'll really struggle to pull it out when you're done. Instead, we suggest you use much shorter electrodes. We used steel tent-pegs, and the type with an L-shaped cross-section have a greater surface area than those made from small-diameter steel rod. They're not very long, but there are a couple of ways you can improve their performance. First, you could use several tent-pegs as each electrode, all connected together and separated by at least the length of the pegs. And if the ground is dry, you could pour water around them. Better still, use a solution of a salt – we suggest Epsom salt (magnesium sulphate), as most other salts would harm any vegetation.

Once you're ready to try everything out, there are still decisions to be made, including the electrode separation and which tap to use on the transformer.

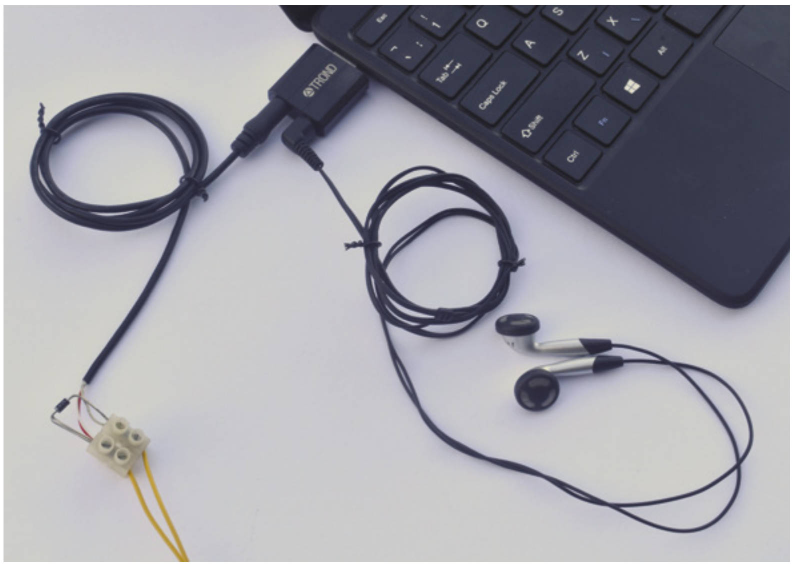
We suggest a trial and error approach, starting with a short range of just a couple of metres, and working up, but we ought to say a few words on the transformer tap. The transformer is used to match the output of the megaphone to the resistance between the electrodes, and we've already listed the taps that correspond to various resistances. However, ground resistance isn't easy to measure: you can't just use a test meter. So, we recommend a cautious approach so you don't risk overloading the megaphone's amplifier. After all, unless you dismantle the megaphone further, you won't know whether the circuitry protects against an overload or whether it'll burn out. So, start with the tap corresponding to the smallest resistance, and only move up if you're confident that you're not close to overloading the megaphone. To do that, measure the current in one of the electrode wires and monitor the current while whistling or 'aah-ing' into the microphone. Don't allow the current to exceed that which flows with the original speaker in place.

We don't expect you'll be abandoning your phone for a ground conduction anytime soon. After all, you'll probably only manage a few hundred metres at the most, and that might need a more ambitious transmitter and receiver than we've described here, but we trust you'll enjoy this foray into an unusual form of communication.

"**We recommend a cautious approach so you don't** risk overloading the megaphone's amplifier"

Left
You thought the transmitter was simple? Well, the receiver is even simpler

Below
Here are the main components of our ground conduction transmitter laid out so you can see it easily, but it would be good to physically attach the transformer to the the megaphone



Moving beyond the K40 part 4

Controls and safety interlocks

Dr Andrew Lewis

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

Without adequate control methods and safety systems, a laser cutter is nothing more than an astonishingly dangerous high-voltage cigarette lighter. In this article, you'll take a look at the control board for a laser cutter, the essential safety features that prevent the laser from firing accidentally, and how to monitor the laser while it's running.

The first thing most people do when they buy a K40 laser is what several websites call 'essential upgrades'. This is a polite way of saying that the stock K40 laser is built to the bare minimum functional standard, and is a potential death-trap in its default state. That isn't a slight against the K40; it's a statement of fact. For any cheap machinery, you're going to have to do some work to get things working properly; tram in the table, tension the belts, and check earth connections. The advantage of building your own machine from scratch is that you get to put those features in from the start. You get to choose what control panel features you want, and which safety interlocks are most important.

Unlike a milling machine or a lathe, a laser cutter is designed to work exclusively under computer

Above
The control panel of the K40 laser isn't complicated. Most of the buttons are for adjusting the laser power because the default control board doesn't support software control. The important controls are machine power and emergency stop, and laser power. The computer at the rear of the machine adds buttons for homing the gantry, and starting various types of cutting or engraving actions. There's no need for these buttons to be mounted on the actual machine, nor is there any need for them to be physical buttons. The general control of the machine can be dealt with much more effectively by software than with hardware controls

control. You don't need to move anything manually, and so the onboard controls for a laser cutter are relatively simple to implement. The important controls are the power switch, the emergency stop, and the laser power switch. The laser power switch should isolate the connection between the laser enable pin on the power supply and the control board. In some cases, it might be appropriate to fit a key switch for this purpose so that the laser won't be able to fire unless the key is inserted and switched to the 'on' position. This will mean that no unauthorised person can activate the laser without the key, but they will still be able to activate the machine and home the axes from the control software. Pretty much everything else is dealt with via G-code sent to the control board, and can be controlled in software from a connected computer.

When it comes to monitoring the laser, things get a bit more complicated. The laser consumes power and generates heat. Monitoring those two things is a sensible precaution. Fitting an ammeter to your control panel is a great idea and has been discussed in a previous article in this series. Monitoring the temperature of the coolant is also a sensible precaution, since overheating the laser can reduce its lifespan significantly. A simple thermistor-based probe and display is sufficient for monitoring the temperature, and the probe can be inserted into the coolant pipes using a modified pipe fitting. For safety reasons, it is strongly advised that a laser should have a clearly visible indicator light to show when the laser is active, and when the system has power. It should not be possible for the laser to have power while the indicator light is out.

> "The laser consumes power and generates heat. **Monitoring those two things is a sensible precaution**"

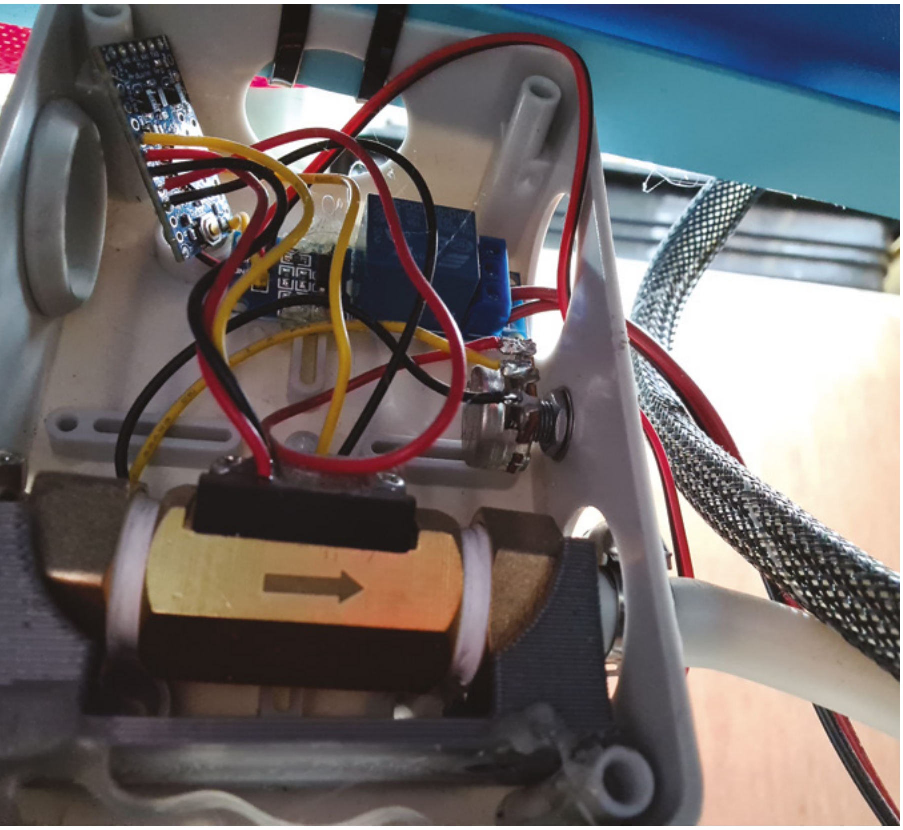
Passive monitoring is only part of the solution for building a safer laser cutter. Knowing the power consumption and temperature of the coolant will give you an idea of what's happening as long as you're looking at the panel, but what happens when you're not? You'll want to employ some active monitoring to make sure nothing dangerous happens while your back is turned. This is where safety interlocks really become useful. Arguably the biggest fault with the design of the K40 laser is the lack of an interlock on the door of →

Above This fresh-from-the-factory K40 has missing earth connections, a broken temperature gauge, no laser power meter, and no safety interlocks. This is normal for a K40, and it's perfectly safe as long as you leave it inside the box and never plug it in. If you actually want to use the machine, you'll need to check everything and take care of the problems before you plug it in

STAYING SAFE

A laser cutter is a dangerous machine. Invisible laser radiation can disfigure or blind you or someone else permanently. When operating a laser, your responsibility for safety does not end with yourself, it extends to the entire area around the laser and everyone who might be affected by an accident. Never operate the laser in an unsafe situation, and never let the machine run out of your control. Never rely on a safety system to deactivate the laser automatically. Always wear laser safety goggles when working with the laser, and be aware of the risks from mechanical parts, high voltages, and reflective surfaces. Never leave the laser in a state where it could be activated accidentally by someone who does not understand the risks. Never leave the laser active and unattended.

Left You can insert a thermistor into the coolant by waterproofing it and sealing it into a suitable T-fitting with silicone rubber



Above This hastily thrown together flow switch has a couple of problems. Firstly, the microcontroller isn't screwed down into position, and the actual flow sensor is made of brass. Brass will react with distilled water, eventually causing a mechanical failure. It will also affect the conductivity of the water, potentially shorting the high-voltage laser power to ground

QUICK TIP

The K40 laser is unusual because the board doesn't directly control the laser's output power, so the K40 panel also has laser power controls.

the laser chamber. It's possible to lift the lid on the chamber while the machine is running and the laser is active, and the machine doesn't have any idea that there's an issue. Some people don't have a problem with this, but most normal humans without the desire to be permanently blinded by an easily corrected oversight do have a problem with it, and they wire in a switch to detect when the door is opened. The switch wires into the interlock loop on the laser tube's power supply, so that the power to the laser is cut if the interlock loop is broken. The interlock loop can have any number of switches or relays wired into it to stop the laser in the event of a fault and, with a little bit of ingenuity, you can use an Arduino or a Raspberry Pi Pico to trigger a relay when some threshold condition is met on a sensor. You can also wire multiple sensors into one microcontroller to trigger a relay and break the interlock loop and disable the laser.

A water flow sensor is probably the most common sensor added to the interlock loop, and it is used to monitor the flow of water around the cooling system. If the cooling pump stops pumping for any reason, you want to stop the laser pretty quickly to prevent an overheat damaging the laser tube. For monitoring, the exact flow-rate of the coolant doesn't really matter too much, as long as you've got some appreciable movement of fluid around the system.

UNINTENDED ANTENNA

You need to take care when positioning and wiring the ammeter into the panel, because it has a high-voltage wire from the laser passing through it. The interference from the high-voltage wires can cause havoc in the low-voltage side of the laser cutter, so do your best to route them sensibly.

Above The plastic flow-meter is a better choice for distilled water, and it's cheaper, too. Most flow-meters use a Hall probe and magnet to measure the flow-rate, although knowing the precise flow-rate isn't important in this application

The water flow sensor is usually a small impeller enclosed in a tube that uses a magnet and a Hall probe to detect rotation of the impeller as fluid flows through the tube. A simple Arduino app is all that's needed to measure the number of pulses from the Hall probe and close the relay if they rise above a certain number of pulses per second.

You might also use an airflow sensor to monitor the extractor fans in the laser chamber. Failure of the cooling fan can lead to high levels of smoke and gas, potentially causing a fire or breathing hazard. Detecting airflow is a little bit more difficult than detecting water flow, as there aren't really dedicated sensors for this. However, it's possible to place a regular computer fan in-line with the exhaust of the air blower and use this fan to generate a current that can be detected by the analogue input of a microcontroller board like the Arduino. You'll need to modify the fan so that the AC outputs from the fan's

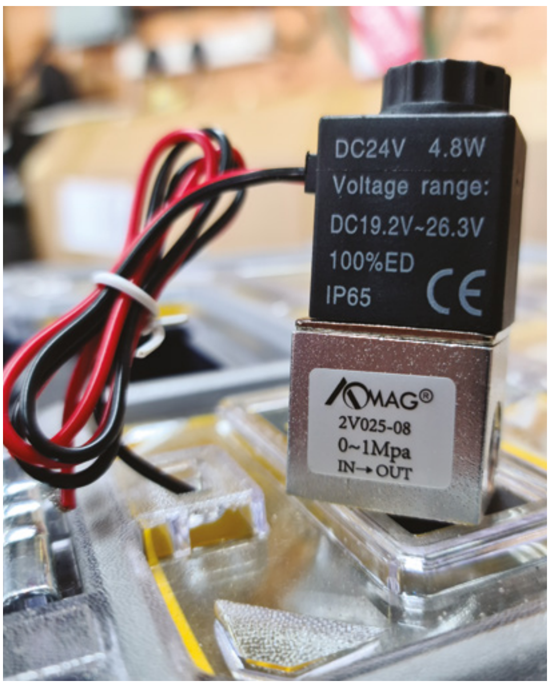
Left
The coils of a computer fan can be adapted to make the fan work as an AC generator. Connecting a microcontroller to the output from the fan means that you can detect airflow quite easily. Just be sure to rectify and smooth the generator output, and limit the output voltage to prevent damage to the microcontroller

coils are converted into a smoothed DC output using a bridge rectifier and a capacitor, but the modification is not difficult – you can find lots of information online by searching for the term 'PC fan generator'. Note that the output of the fan might exceed 5V when it is running at full speed, so a potential divider circuit can be used to drop the voltage down into the range that an Arduino or Pico can handle safely.

An interesting note about the interlock system is that it operates directly from the laser power supply, running independently of the laser cutter's control board. That means it will still operate if the control board crashes or is programmed incorrectly. You cannot override the interlocks from a terminal and do something stupid. The whole point of the interlocks is that you'll need to exercise sustained, proactive, and wilful stupidity to make the machine do something dangerous. To this end, some people suggest additional battle-hardening of the interlock system. Tips like using magnetic switches on the chamber door to prevent someone intentionally wedging the switch down with tape or cable ties can be useful in certain environments, but it's up to the machine owner to decide where the line of user trust lies. If you need to worry about deliberate misuse by machine users, then you probably shouldn't have those people anywhere near something as obviously dangerous as a laser cutter.

> "You cannot override the interlocks from **a terminal and do something stupid**"

Choosing and installing a control board for your laser is a big subject, but thanks to the popularity of 3D printers, the number of available controller options is huge. If you can afford it, there are dedicated, high-powered controllers specifically designed with laser control in mind, with touchscreen controls that cost several hundred pounds. However, there's also a slew of more general-purpose or 3D printer-oriented boards available that can do the same job with a less fancy interface, and a slightly more complicated configuration process. The default board in the K40 laser is usually the M2 Nano. As you might expect from the K40, this is a bare minimum board that can just about control the cutter without catching fire. It's a simple board that runs a non-standard form of G-code, doesn't handle laser power control, and has no easing or power control on corners, which can lead to crispy-fried and smoked corners on cut objects. The M2 Nano also doesn't have a controller for the Z axis of the machine, which →

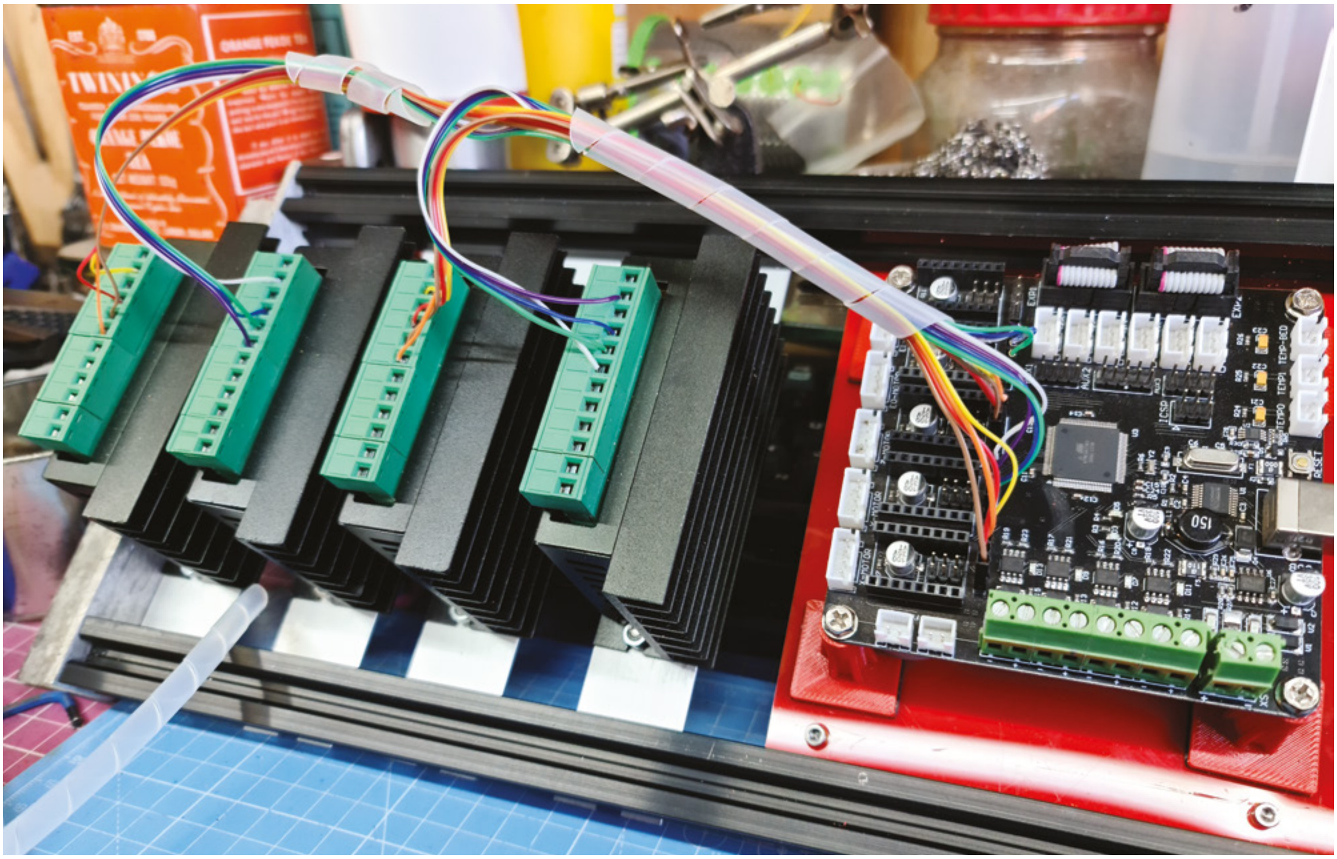


Above Marlin supports several LCD and touchscreen displays. Enabling a setting in the configuration will generally also update the menu and control settings available from the display

Right Relays and servo-controlled air valves like this one can be controlled by Marlin, giving you more control over certain cutting or engraving jobs

means adjustable bed heights and focussing are quite literally off the table. Using the original K40 board in a new machine would limit the functionality quite severely, especially when you consider the availability of control software like LightBurn.

The speed of the control board will determine the processing speed of the G-code that it receives, so when you're choosing a control board, you should look for a board with a faster processor. While an 8-bit Arduino with a CNC shield will work, you'll probably find that your board would struggle to deal with high-speed tasks like engraving. It's much better to use something with a faster 32-bit processor, and for now, it's best to look at something like the Cohesion3D LaserBoard (a direct drop-in replacement for the K40's M2 Nano), or the BigTreeTech SKR V1.3. The SKR V1.3 is mentioned here specifically because it's cheap, and has easy access to the STEP, DIR, ENABLE pins near the motor controller sockets, and that makes it a little bit easier to wire things up when you're using external motor controllers. At the price, you can always upgrade later if the board isn't powerful enough for your needs.



As for the choice of firmware that you are going to install on the controller, Marlin is probably the easiest to get started with. There's plenty of documentation on the Marlin website (**marlinfw.org**), but we'll have a quick look at some of the more important settings here.

Settings for the firmware are mostly located in the file **configuration.h**, which is used to define the basic type of machine and controls. You'll also need to edit the **Configuration_adv.h** file to enable specific laser-related features like PWM laser control. Enabling LASER_SETTING will enable the G-code commands necessary to control the laser, while the AIR_ASSIST setting will let you control air assist via a valve connected to a pin on the controller. Enabling AIR_EVACUATION and setting an AIR_EVACUATION_PIN will enable control of the exhaust fan by G-code and via the menu system (if you're using a menu screen).

PWM control of the laser needs to be enabled by setting SPINDLE_LASER_USE_PWM and defining the PWM pin used on the controller with SPINDLE_LASER_PWM_PIN. If you neglect this step, the laser will only have simple on/off control with no variable intensity. To maintain compatibility with LightBurn and LaserGRBL, you'll need to make sure that GCODE_MOTION_MODES is enabled.

Full instructions and firmware examples are available on the Marlin website. Marlin is not the only option for controlling a laser, but it is arguably the easiest to get started with and has mature documentation. At the time of writing, a Raspberry Pi Pico HAL (hardware abstraction layer) is under development for Marlin, and new control boards have been released that use the RP2040. It's worth checking on the progress of those features if you're thinking of buying a new board.

PLASTIC **POWER**

Although it's tempting to buy a high-quality flow sensor with a brass fitting, it's actually better to buy the cheaper plastic version for use with a coolant system. Distilled water and other coolants can react with metals like brass or stainless steel over time, increasing the conductivity of the coolant and causing the laser to arc out to the chassis.

Above
Having a control board with breakouts for external motor controllers can make life much neater inside the control box, and easier to upgrade or change in future

QUICK TIP

Most laser power supplies have a 5V output that can be used to power a simple microcontroller board, meaning that wiring up the sensors is even easier.

Power up your 3D prints

Our pick of the most useful advanced slicer features

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.

Cura slicer is one of the most powerful tools in the 3D printer user's arsenal. It lets you control a lot of different facets of how your designs will 3D-print. We're going to take a look at some of the lesser-known features of Cura that you can use to save filament, speed up your prints, or just make them look a bit funky.

Not all of these will be visible to the user by default. You can see all the slicing options by going to Preferences > Configure Cura > Settings, and clicking the Check All button. Now, if you go to the print setting menu (click on the layer height in the main screen), you can search for any of the available settings.

IRONING

Ironing (sometimes called 'Neosanding') smooths horizontal flat top surfaces of 3D prints by running a hot nozzle with very low flow over the finished print. The hot nozzle evens out any high bits, while the low flow will fill in any inconsistencies. The final result should be a much smoother print, though it will increase print time.

Ironing can't help with any surfaces that aren't flat or horizontal, though.

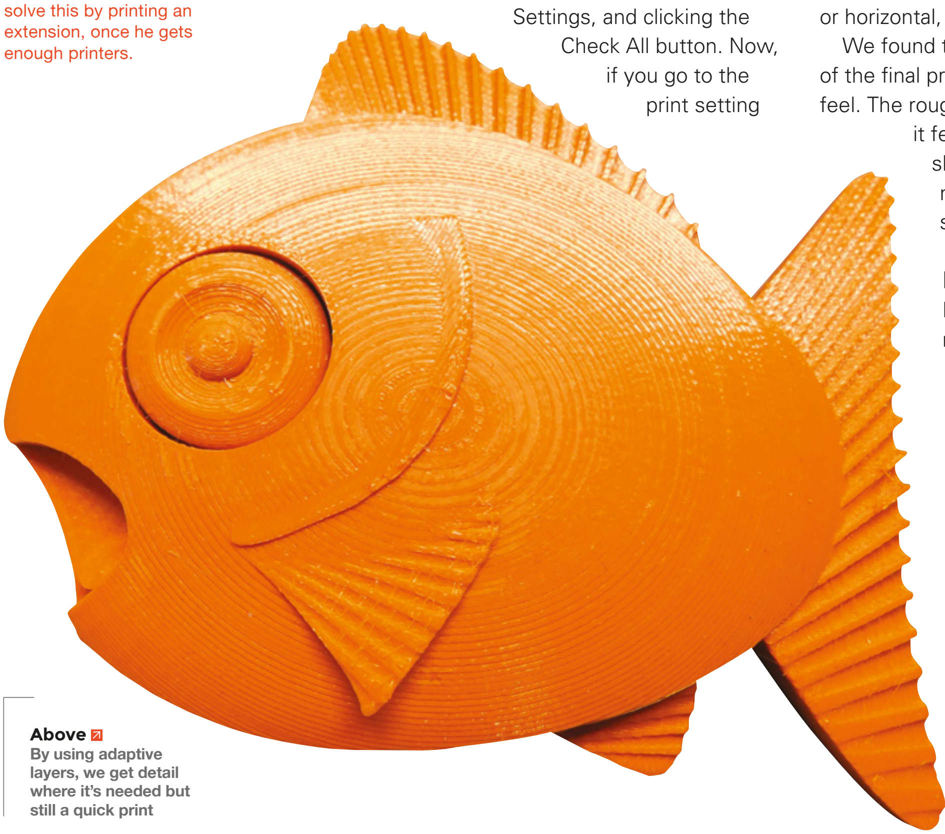
We found that, while it does improve the look of the final print, the biggest difference is to the feel. The rough surface is almost entirely gone, and it feels much more pleasant to touch. It should also help provide a better fit if you need to glue the top surface of a part to something else.

FUZZY SKIN

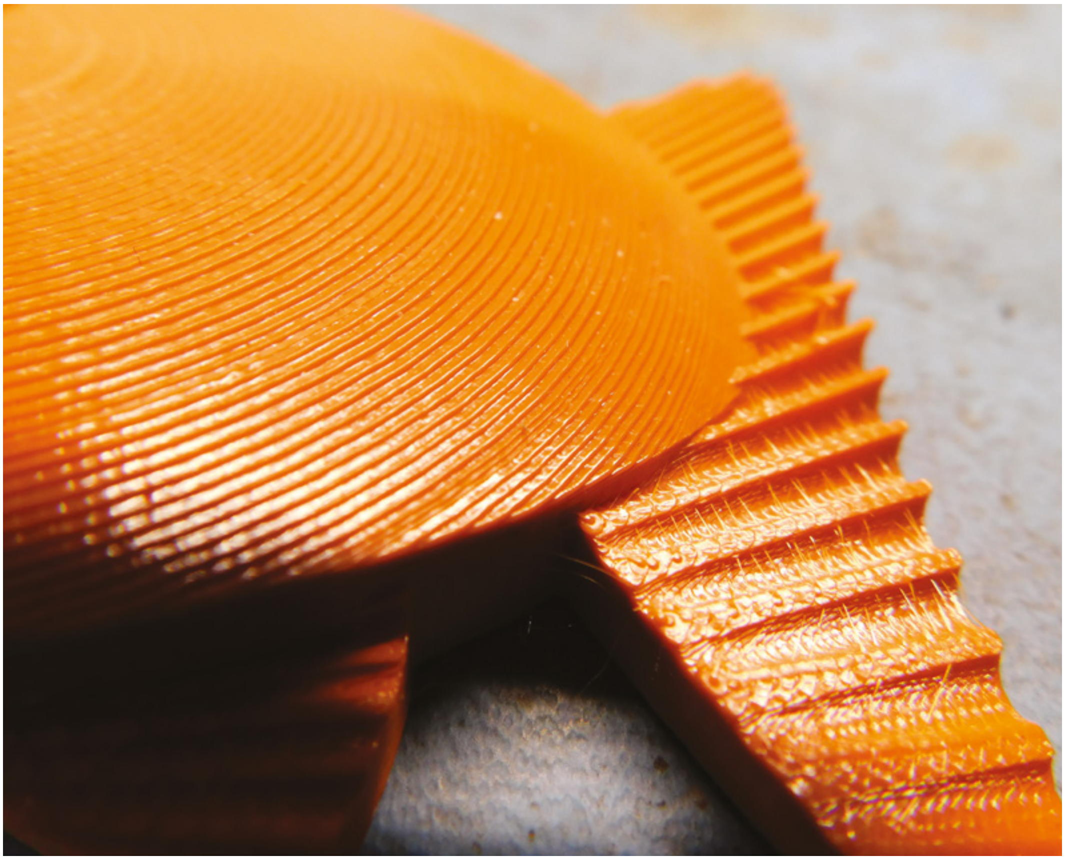
Fuzzy does pretty much what it says – it makes the final print a little bit fuzzy. OK, maybe fuzzy isn't the perfect description, but it 'wobbles' the nozzle to give it a textured look. This can be good for parts modelling animals or things with fur. It can also be good for grips, but you'll need to tune the parameters so they don't feel too rough for you.

BREAK SUPPORT INTO CHUNKS

If you have a large area that needs support, it can be a bit clunky to break off, as all the support material is connected together. Break Support Into Chunks will, as the



Above By using adaptive layers, we get detail where it's needed but still a quick print



name suggests, create unlinked blocks of support material so that it's a bit easier to break away. This doesn't affect the amount of area being supported.

This option will only be visible if Supports are enabled and the ZigZag pattern is selected.

DRAFT SHIELD

Some filaments are very susceptible to cool breezes. We found this is a particular problem with ASA. The draft shield option prints a single perimeter wall around the edge of the print (but not attached to the object being printed) to stop your main print cooling down too much. This may sound like it's not going to do much, but our experience is that if you have an unenclosed 3D printer, this can be the difference between being able to print some temperature-sensitive filaments and not being able to print them.

> **Some filaments are very susceptible to cool breezes.** We found this is a particular problem with ASA

LIGHTNING INFILL

What is infill on your 3D print for? That's not actually a rhetorical question, although it may sound like it. There are two basic things that infill provides:

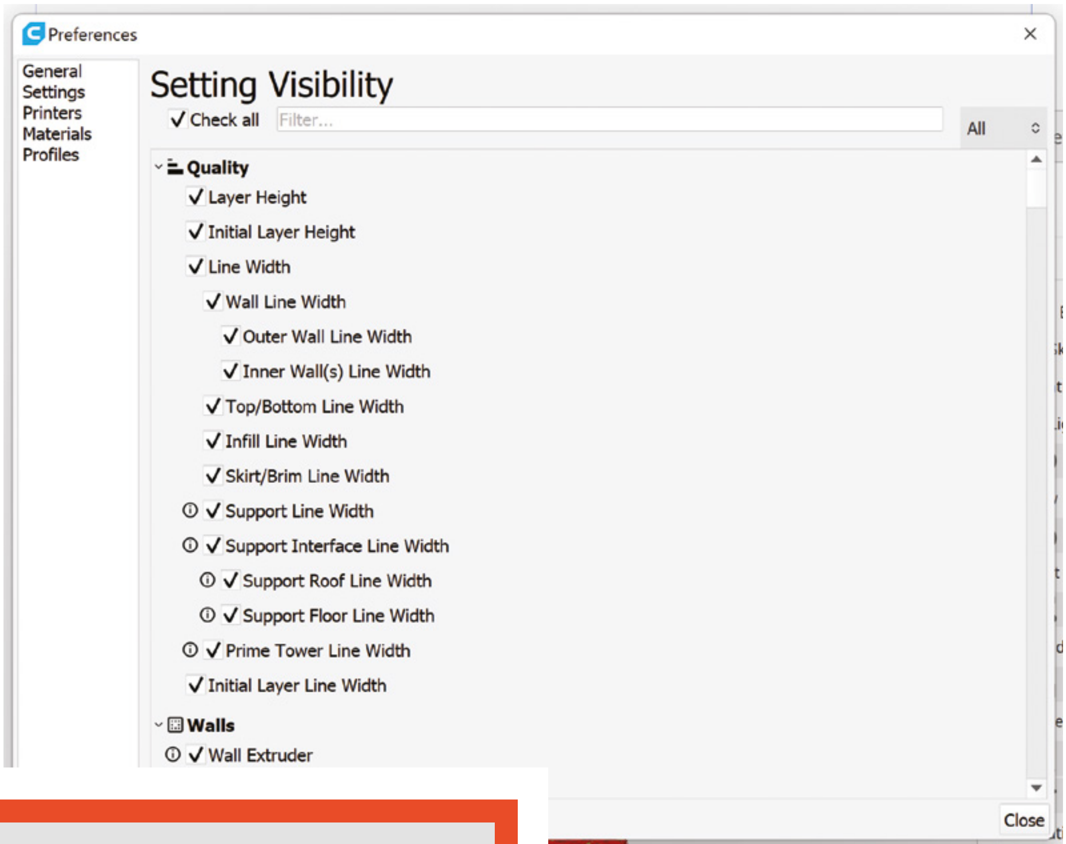
- **Printability** – if your part has a top, this may be unprintable without something to print this top on. Infill provides this something.
- **Strength** – a hollow part may not be strong enough for your needs. Adding infill can provide this extra strength.

If your part needs the first of these, but not the second – for example if it's an art piece – then you don't need infill, you need support for the top (and any internal overhangs). This is what lightning infill provides. Just a minimal amount of internal support material to make your part printable.

Take, for example, this rather fetching owl (overleaf). Printed with 10% gyroid infill, it will take 6 hours 57 minutes and use 60g of filament, while with lightning infill it will take 4 hours 42 minutes and use 35g of filament. The time and cost savings are dependent on the internal volume and its shape.

If you want some infill throughout the entire model, but need more to ensure the upper layers →

Above You can see where it shifts from thicker layers to thinner layers when detail is needed



Right By default, you won't see all the settings in Cura, so make sure to turn on the ones you want

PRUSASLICER

While PrusaSlicer doesn't have the range of options that Cura has, it does give you quite a bit of control over settings for different parts of your model.

Seam painter
While Cura lets you place the Z seam on a corner or randomly, PrusaSlicer lets you direct exactly where you want it on the model. In the 3D editor, 'Seam painter' is the bottom option on the left-hand menu. Select this and you can draw on your model where you want the Z seam to go.

Paint-on supports
Modern slicers are quite good at knowing where supports should go. However, they're not perfect. Particularly, if you're printing a lot of something, it can be very helpful to have control over exactly where supports go. Paint-on supports (the second-from-bottom option, left-hand menu) lets you draw on the model exactly where you want supports to go.

Modifier blocks
You can change certain print settings for different parts of your print. Not all settings can be changed, but you can change the infill percentage or number of perimeters, for example, if you know one part has to be strong while another part doesn't need to be. You need to change to Advanced mode, then right-click on your object in the list on the right-hand side. In the menu, you should be able to add a modifier block and use this to change the parameters for the area of the block.

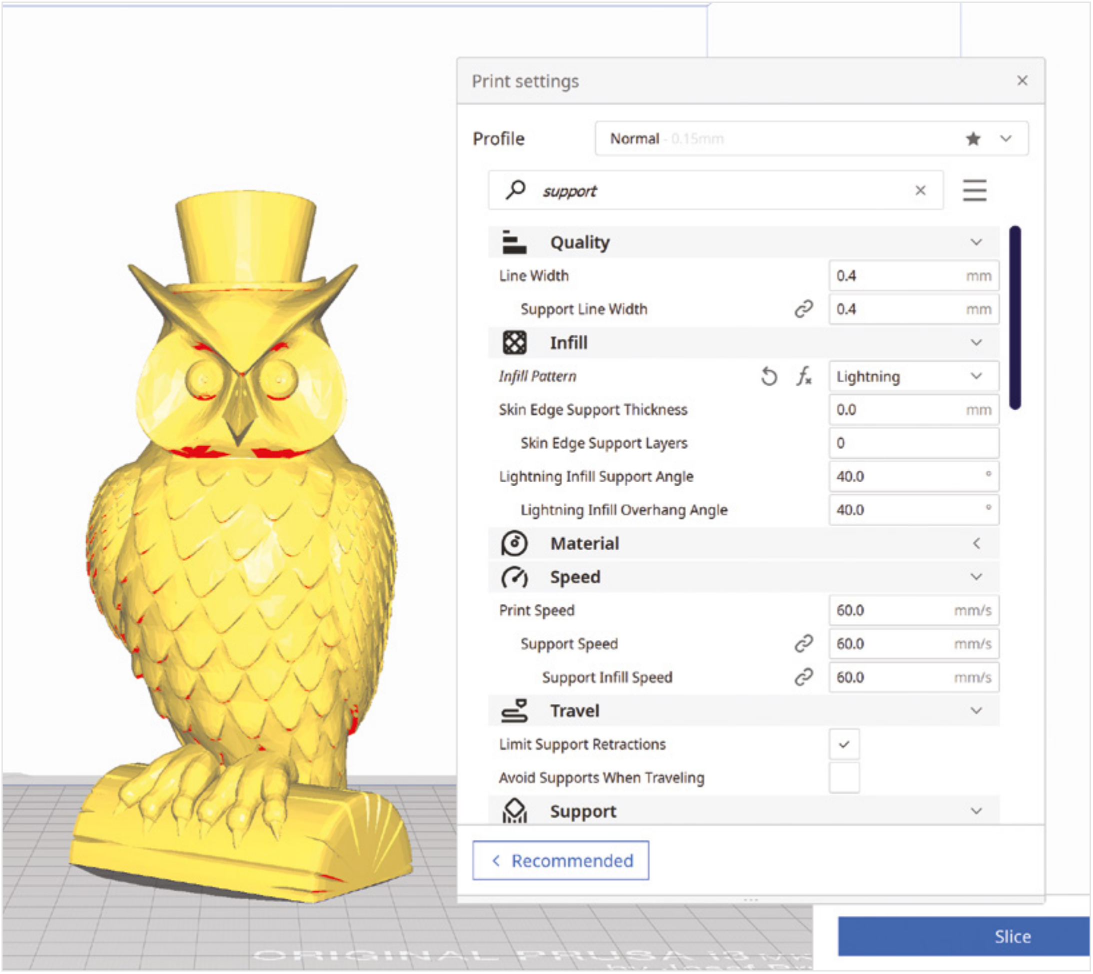
Adaptive layer height
This works much as it does in Cura, but you can also manually select where you'd like larger and smaller layer heights.

> "Picking the right layer height for your 3D print affects the look in many ways"

are properly supported, you can choose gradual infill, which will vary the infill density, increasing it near the top to improve printability.

ADAPTIVE LAYERS

Picking the right layer height for your 3D print affects the look in many ways, but how much it affects the look depends a lot on how much detail there is in the area. On vertical sides, chunky layers might not cause many problems, but on more detailed areas it might. For example, we sliced the Fish object (from **Thingiverse** user Sanraith – **hsmag.cc/FishThingiverse**) at 0.3 mm layer height with adaptive infill turned on with a range of 0.25 mm. With this, the base was printed with thick layers, while the



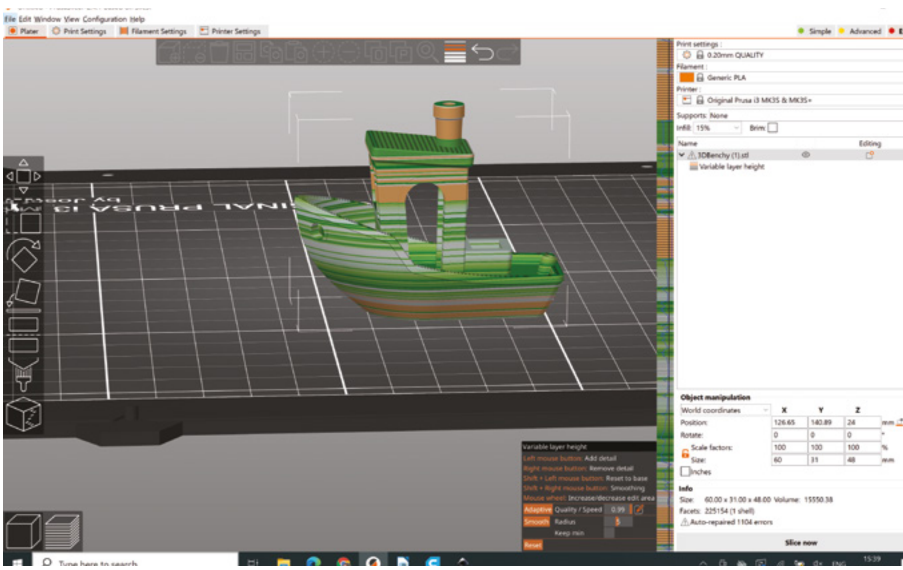
Left
Lightning infill means this owl can be almost completely hollow

top (where the detail is) was printed with super-fine 0.05mm layers. The result was a 4 hour 2 minute print. Without adaptive layers, printing at high detail (0.6mm layers) resulted in a 10 hour 25 minute print time.

Any print that has a significant vertical section with few details can benefit from this option.

Z SEAM

Each layer, the printer adds a new perimeter. Since this is (almost always) a loop, the nozzle has to start and finish this loop at some point, resulting in a slight imperfection on the surface of the print. By default, Cura will try and place this imperfection on the sharpest corner of the model. On models that have at least one sharp corner, this does quite a good job of hiding these imperfections, but if there isn't a sharp corner, all these imperfections line up one on top of the other to create a 'seam' that can be quite noticeable. You can't get rid of the imperfections entirely, but you can hide them better. Using the Z seam settings, you can select 'random' which means it will start each perimeter at a random place. This means there's no vertical seam that runs up the side of the model. □



Above
Adaptive layer height is also available in PrusaSlicer, and this gives you more manual control over where to add detail

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

ONLY THE BEST

Springtime electronics roundup

Gardening-related DIY electronics

By Marc de Vinck @devinck

Spring has arrived! And it's time to get your outdoor projects started. I've covered some outdoor-related projects in a previous springtime Best of Breed but, as the category has so many cool products, I thought I'd take another look again this year. There are, of course, the common IoT-related items like smart switches, plugs, and other connected devices, but I wanted to narrow down the category and dive a little deeper into a specific topic – gardening!

In this Best of Breed, I'll be looking at indoor and outdoor gardening-related products. Things that will make you and your plants happy! Automation, sensors, controllers – it's a mixed bag of products that have the common bond of keeping your plants alive. There are a couple of brand-new products too! Spring is here, so get your soldering iron tuned up, and let's get gardening.

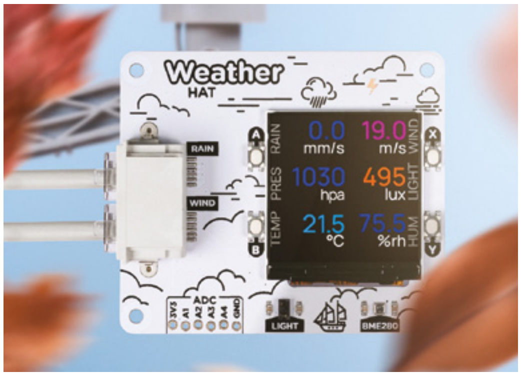
Weather HAT + Weather Sensors Kit vs Enviro + Air Quality for Raspberry Pi

PIMORONI ◈ $90.06 | pimoroni.com

PIMORONI ◈ $52.40 | pimoroni.com

When researching products for this roundup, one of the first things that I found was the all-new Weather HAT + Weather Sensors Kit from Pimoroni. And I have to say, it was a great find! This Weather HAT combo kit not only includes the HAT for your Raspberry Pi, but it also includes all the sensors needed to create a robust weather station.

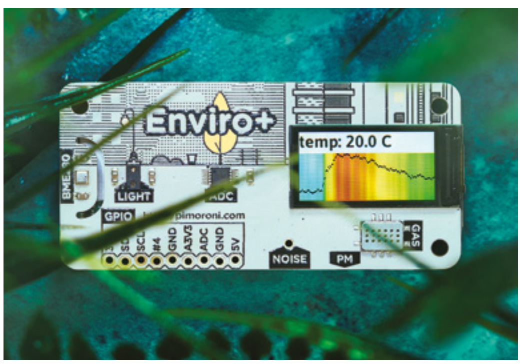
The Weather HAT is Pimoroni's all-in-one solution for hooking up multiple climate and environmental sensors to a Raspberry Pi. It features an LCD screen and sensors that can measure temperature, humidity, pressure, and light. It also has an RJ11 connector, the same type of connector that a typical landline phone uses, that will let you easily attach an external wind and rain sensor. This kit includes those sensors too, but you can also save some money and just start out with the Weather HAT. Just don't forget to pick up a Raspberry Pi to complete the system!



Left ◈ & Below While functionality is important, the design of these HATs from Pimoroni is absolutely stunning

Not to be outshone by the new Weather HAT, the Enviro+ , also from Pimoroni, has a lot of similar features, but focuses on indoor environmental sensors. The HAT includes a BME280 temperature, pressure, and humidity sensor, an LTR-559 light and proximity sensor, a MiCS-6814 analogue gas sensor, and a microphone. And if that's not enough, you can also add an optional PMS5003 Particulate Matter Sensor via a dedicated connector. And all that data gathered from the sensors can be viewed on its on-board colour screen.

The Enviro+ HAT was developed in collaboration with the University of Sheffield, and they have made it fairly simple to allow you to contribute real-time air quality data from your location to open-data projects like Luftdaten, also known as Sensor.Community. Head on over to the product page to learn more about this collaboration, and the importance of indoor environmental monitoring. →



VERDICT

Weather HAT + Weather Sensors Kit

A great kit to get you up and running.

10/10

Enviro + Air Quality for Raspberry Pi

Another great combination board from Pimoroni.

9/10

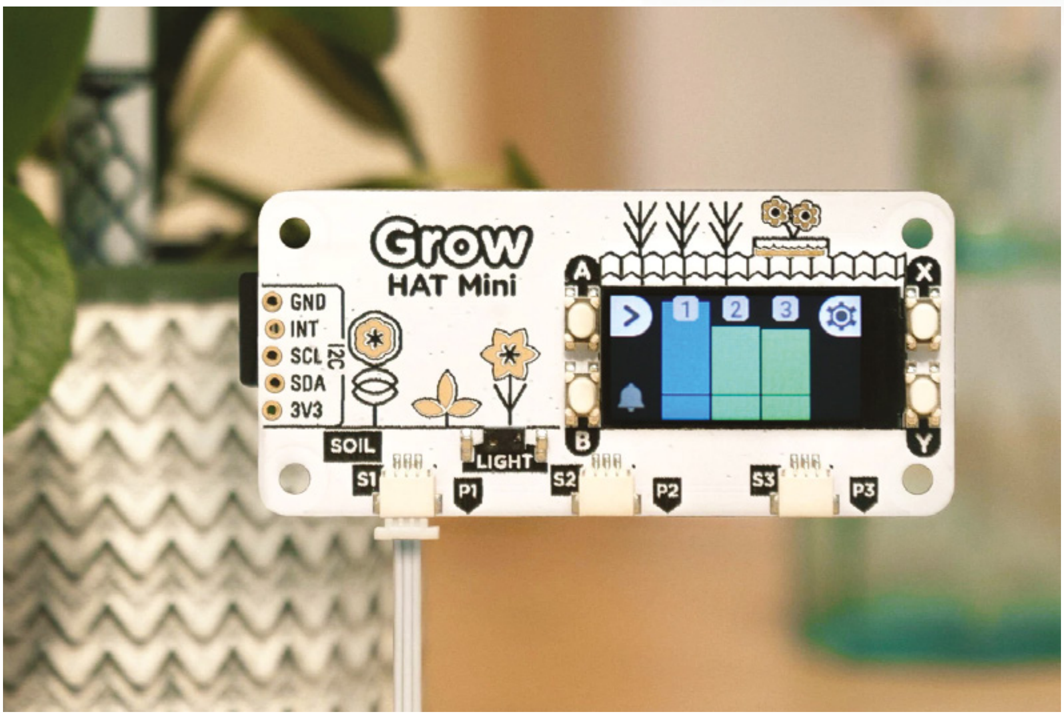
BEST OF BREED

Grow by Pimoroni

PIMORONI ◆ $32.75 | pimoroni.com

Pimoroni has another great addition to their environmental sensing line-up called Grow. This little HAT features a beautiful on-board IPS colour screen. You can display data from any of the sensors you connect via the JST connectors, or the integrated light sensor.

The board gets even more interesting when you plug in the included capacitive moisture sensors. Once you take a soil reading, you can also use the Picoblade 2P-compatible connectors for switching low-current 5 V devices on and off. It's a great way to get started with some simple automation and control logic. Just add a Raspberry Pi and you are off to automate your indoor garden. And if you don't have any plants, Pimoroni also sells a kit that includes everything you need to get started growing herbs or chilli peppers.



Left ◆ Keep your plants happy

VERDICT

Grow by Pimoroni

Another winner from Pimoroni.

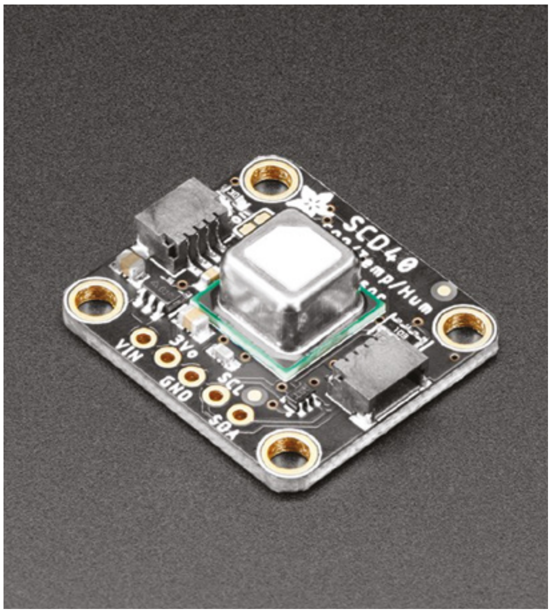
9/10

Adafruit SCD-40 - True CO_2, Temperature and Humidity

ADAFRUIT ◈ $49.50 | adafruit.com

If you have a greenhouse, exotic plants, or even a vivarium, then measuring CO_2 accurately may be critical to your ecosystem's success. And that's where the Adafruit SCD-40 breakout board comes into play. Unlike the typical 'gas' sensors out there that approximate the amount of VOCs in the air, the SCD-40 is a photoacoustic, 'true' CO_2 sensor that can tell you the specific CO_2 PPM composition of the surrounding ambient air. It's a true measurement of CO_2 concentration.

The board also features Adafruit's STEMMA QT form factor which allows you to wire it up quickly, without soldering, via their QT connectors and cables. And, if you need even more range in measuring CO_2, typically only in industrial situations, they also sell a version that features an SCD-41 sensor. It costs a little bit more, but you also get an extended range, and increased accuracy. Head on over to the product page to learn more about accurately reading CO_2 levels. →



Left ◈ Learn about how the planet is dying in real time

VERDICT

Adafruit SCD-40 - True CO_2, Temperature and Humidity

Accurate readings are what this is all about.

9/10

BRASS LIQUID SOLENOID VALVE - 12 V - 1/2 NPS

ADAFRUIT ◈ $24.95 | adafruit.com

If you are considering the addition of a flow sensor to your watering system, then you should also consider adding this brass solenoid valve from Adafruit. Just like the flow sensor, most of this fitting is made of brass for added durability. Now you can easily turn your water source on and off via your Raspberry Pi or microcontroller.

Just keep in mind, unlike many other sensors and components used in DIY electronics that run on 3.3 V or 5 V, this one operates on 12 volts. There are also some other important considerations before choosing a solenoid valve, so head over to the website for more information and to see if this is a good choice for your project.

BEST OF BREED

Liquid Flow Meter - Brass 1/2” Nominal Threaded

ADAFRUIT $24.95 | adafruit.com

There are a few different water flow meters out there, but this one has one obvious feature that sets it apart – it’s brass! Many of the flow meters you can find online are plastic, or have substantial plastic parts, but this one from Adafruit is mostly brass. This allows it to function in a much harsher environment, and it should last a lot longer than its plastic equivalents.

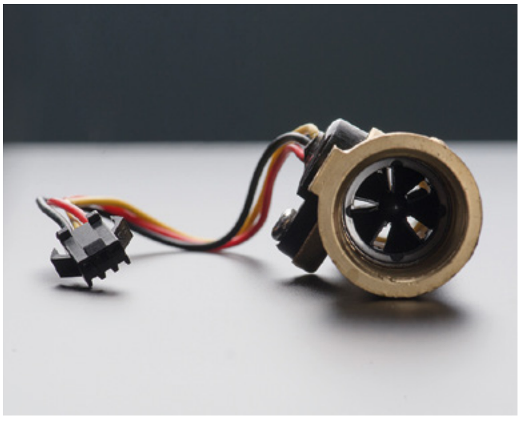
The flow sensor consists of a little pinwheel that spins inside the pipe fitting. You can count the rotations, via a small magnet and a Hall effect sensor, and get a good estimation of the amount of water that has passed through. The key here is estimation. Each pulse of the pinwheel represents 2 millilitres of fluid. That estimation is dependent on the flow-rate, orientation, and pressure. You can calibrate the sensor for your application and, for most gardening purposes, it is accurate enough with minimal calibration.

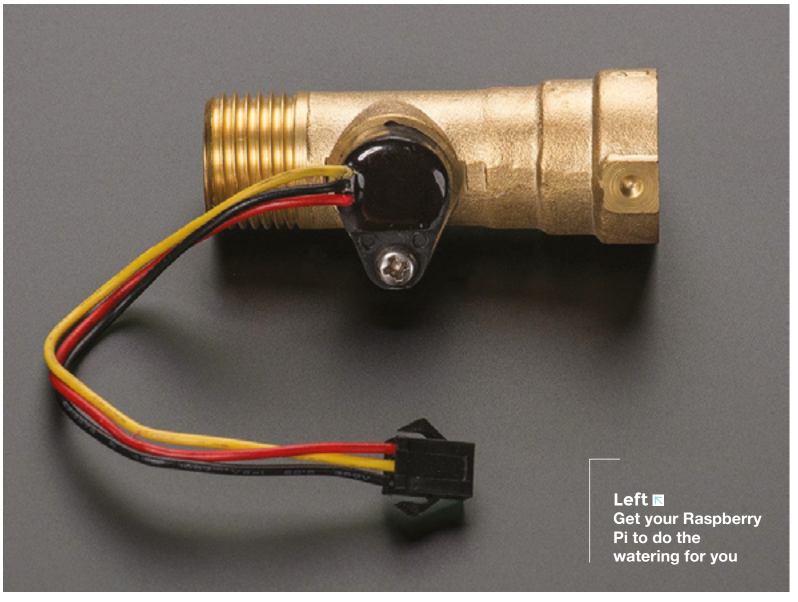
VERDICT

Liquid Flow Meter - Brass 1/2” Nominal Threaded

Robust enough for outdoors.

9/10

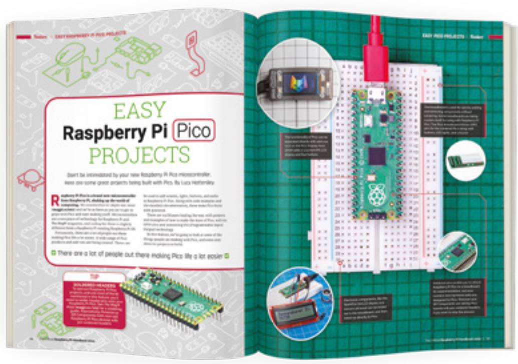




Left Get your Raspberry Pi to do the watering for you

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
Updated! QUICKSTART GUIDE
Start incredible projects with Raspberry Pi
FROM THE MAKERS OF MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

50 HACKS & HINTS
RASPBERRY PI OS TIPS



EASY Raspberry Pi Pico PROJECTS

Pimoroni Badger 2040

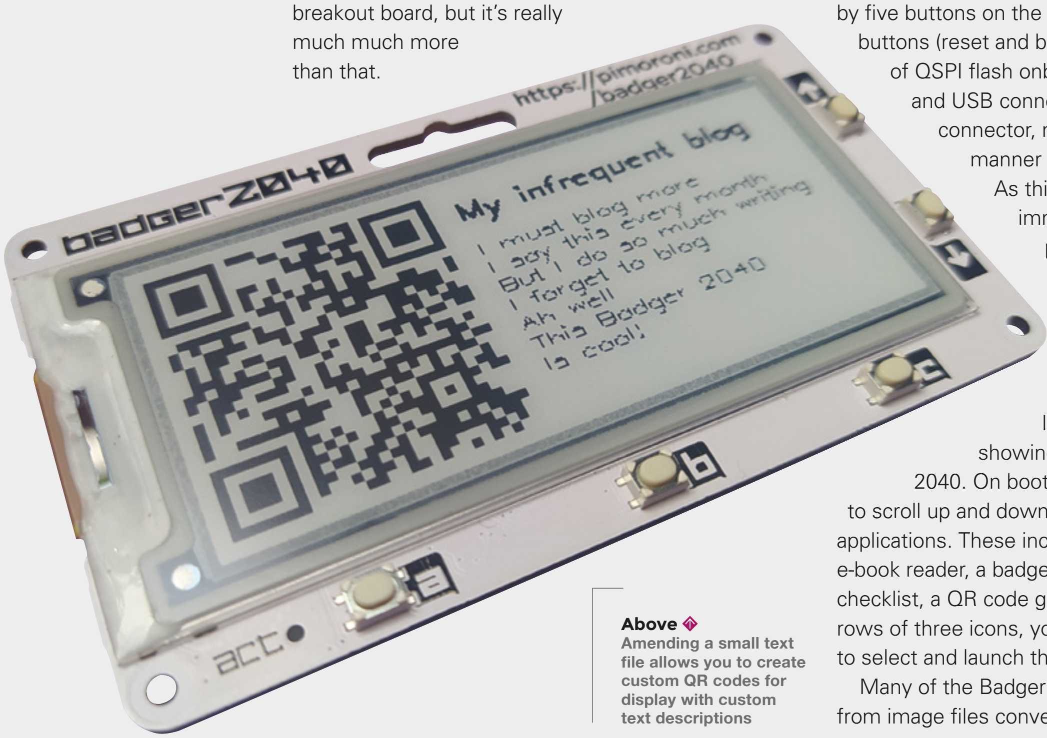
A versatile and programmable badge powered by Raspberry Pi RP2040

PIMORONI £12 (£19.50 with accessory set) | pimoroni.com

By **Jo Hinchliffe** @concreted0g

E-ink is an amazing technology. Its contrast and visibility, even in high sunshine, are great. Fabulous viewing angles and the fact that it can keep an image crisply held when powered down make it a versatile display technology. However, whilst this makes it sound incredibly serious, as we write this review we have the long time internet meme tune 'Badger, badger, badger, badger, badger, badger, badger / Mushroom, mushroom' washing around our heads.

Supplied fully assembled, the Badger 2040 from Pimoroni looks, at first glance, like an e-ink breakout board, but it's really much much more than that.

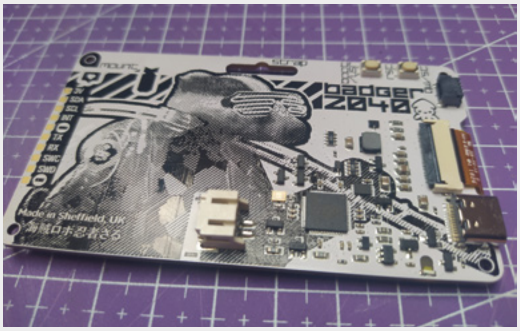


Above Amending a small text file allows you to create custom QR codes for display with custom text descriptions

It's a lovely 2.9" black and white e-ink display that's fully programmable with its onboard RP2040 microcontroller. It can be powered and programmed over the USB-C socket or, for mobile operation, you can add a battery pack via the JST PH connector on the rear. It will run happily on a voltage between 2.7 V and 6 V, and if you buy the accessory set, it includes a small 2 × AAA battery box terminated in a JST PH connector. You could also power this directly from a single LiPo cell, but it's worth considering how you will make that safe for portable use, especially as this is likely to be used as a wearable piece of technology.

Delving into the specs, the e-ink display is flanked by five buttons on the front of the PCB and two buttons (reset and boot) on the rear. There is 2MB of QSPI flash onboard and, as well as the battery and USB connectors, there is a Qw/St connector, meaning you can connect all manner of sensors and peripherals to it. As this is RP2040-based, you can immediately head 'off-piste' and program the Badger 2040 however you like, but Pimoroni has supplied a great set of examples bundled up into BadgerOS. BadgerOS features a launcher full of little applications showing the versatility of the Badger 2040. On boot you use the two side buttons to scroll up and down the launcher to select between applications. These include a clock, a font selector, an e-book reader, a badge, an image browser, a to-do checklist, a QR code generator, and more. Shown in rows of three icons, you then use the lower buttons to select and launch the applications.

Many of the BadgerOS examples are fed content from image files converted into binaries, or from text



Above Excellent artwork on the Badger 2040 PCB

files. Stored in the root directory of the Badger 2040, you can connect the Badger 2040 up to a computer and use Thonny to add your own content to the BadgerOS applications. Pimoroni has incorporated Badger 2040 into its broader Pirate brand of MicroPython for Pico repository, and downloading or cloning this repo gives you heaps of examples to play with. It also includes some handy Python scripts that easily enable you to convert images into the correct binaries for display within BadgerOS. There are C++ examples as well as MicroPython, and also, from the product page, Pimoroni has supplied links to get going with CircuitPython for Badger 2040.

If you buy the accessory pack, you get a recycled lanyard which you can clip to the Badger 2040, turning it into a wearable. As an ID badge for events, it's great, as you can tweak the Badge application to contain your own content. Converting an image for the Badge application entails creating a 104 × 128 pixel PNG image which you then convert to a binary using a small Python script from the Pimoroni MicroPython repository. All the instructions for this can be found on **hsmag.cc/PimoroniBadgerLearn**.

Text for the Badge application is in a simple text file that you can edit directly in Thonny and resave to the Badger 2040. This is similar for other applications, such as the QR code generator. The QR code generator reads a nine-line text file held in the root directory of the Badger 2040. The first line should be a URL which will be converted into a displayed QR code, and the remaining eight lines of text are displayed at the side of the QR code. It's really easy! Similarly, you can edit the e-reader application to include any text file you like – as many classics are available on Project Gutenberg, there's no reason not to carry a classic with you every day!

We are excited to see what the community does with the Badger 2040 – we have already seen a few interesting applications for it. One that particularly caught our eye is using the Badger 2040 as a waveform generator using the Qw/ST as an output for generated waveforms and the e-ink display and buttons as a menu system to change the frequency and duty cycle and more. The GitHub repository for that project is available here **hsmag.cc/WaveGen**.

All in all, we think the Badger 2040 is a great, affordable, and fun device that can simply be a great conversation piece, a fabulous conference badge, or a solid platform for learning MicroPython or other RP2040 programming approaches.

Above BadgerOS features a launcher menu with a host of small, fun applications to play with

Below The Badger 2040 makes for a good conference or event badge

VERDICT

A heap of fun and learning from just £12.

9/10

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

RETRO GAMING WITH RASPBERRY PI

2ND EDITION

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

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

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

Shaper Origin

The handheld router that tells you where to cut

SHAPER ◈ €2890 + VAT | shapertools.com

By **Ben Everard** @ben_everard

We're not quite sure what to call the Shaper Origin. It's not quite a CNC – though it is partly computer-controlled – and it's not quite a hand tool, though you do move it with your hands. To look at, it appears to be a chunky router. There's two handles and a spindle, much as you'd find on any other router, but there's also a bit on the front and a screen. Another clue that this isn't an ordinary router is the black and white 'domino' tape that you need to stick on the workpiece.

Below ◈
At first glance you could be forgiven for thinking this is a regular router

Inside Origin, there's a computer that you can load your designs onto (as SVG files). You can then scan the workpiece (there's an on-board camera). Origin can sense its position using the domino tape as an optical reference. The screen shows the position of the spindle and the cut lines in the design. Origin can automatically move the spindle a small amount (a couple of centimetres), so as long as you're close to the correct position, Origin can correct your movements to follow the cut lines exactly.

In theory, then, you get the benefits of a CNC (accurately cutting the design) and the benefits of a hand-held machine (small enough to store and easily transport). We wanted to test if you really do get the best of both worlds.

IN THE WORKSHOP

Our first impression of the machine came from the software. After booting up, you're presented with a menu on the touchscreen. The software was intuitive and we got started without any hiccups – there are video guides to follow, but it shouldn't take long to get to grips with it.

This is a really important point because the people who stand to gain the most from Origin are wood workers rather than electronics tinkerers. Making both design and use as straightforward as possible is really important for a tool for non-geeks.

Beyond the software, there really isn't much to using Origin beyond what there is to any other router. A button plunges the spindle to the assigned depth and then you move it according to the path shown on the screen. If you stray too far from the path (beyond the distance that Origin can compensate for), it will

Left The ShaperTape lets the Origin know where it is in 3D space

retract the spindle. For shallow cuts (around 3 mm or so), this retraction is usually fast enough that it doesn't noticeably cut the workpiece. For deeper cuts, you will get a gouge in the side if you stray too far from the assigned path. That said, you really shouldn't be going off path – it's not hard to stay on course and we only did to test it out.

We were impressed by the accuracy the machine managed. The parts looked comparable to fully CNCd parts and were about as dimensionally accurate as it's really possible to get with wood. We tested it out for cutting shapes out of sheet material and engraving designs on the surface of wood, and in both cases the results were excellent. The one caveat is that it's a 2.5D machine. This means that each path you cut with Origin goes to a specific depth. If you wanted to carve out a fully 3D shape from a block of wood, you'd need a different machine.

For a lot of purposes, Origin can replace a full CNC, and do the job really well. For space-constrained makers, it's a fantastic option for accurately cutting wood and other sheet material. There are some things that it can do much better than a full CNC – the most noticeable, working on things that can't fit into a CNC, like the top of an assembled table.

There are, however, a couple of things to be aware of. The domino tape comes in at about £14 per roll. That represents a significant running cost to the machine. Used properly, this covers one and a half 4×8 foot sheets. You can get away with spreading it a bit more thinly than this, but you may hit occasional issues if you do.

> "For space-constrained makers, it's a fantastic option for **accurately cutting wood and other sheet material**"

Another slight sticking point is your time. While Origin is reasonably quick at cutting, you do have to manually move the machine around the workpiece rather than just hitting a button and letting the machine do all the work as you would on a traditional CNC. We found that the height of your workbench had a big impact on how comfortable this was. When we had to stoop a little to do this, we very quickly ended up with aches in our lower back.

Overall, though, these issues we had with Origin are slight compared to how much additional capability it gave us. In our testing, Origin delivered exactly what was promised – the ability to accurately cut shapes in wood with a handheld tool. It worked for large shapes (such as cutting designs out of sheet material) and small shapes (such as carving letters and other designs into wood). For these purposes, we don't know of any other handheld tool that can perform the task. It's orders of magnitude faster and easier than trying to do the job with traditional hand tools. This all comes at a price, both for the tool and for the tape, but depending on what you're making, this could save you a lot of time.

VERDICT

The power of CNC with the small size of handheld tools.

9/10

Pimoroni Trilobot

A Raspberry Pi robot kit with some cool under-lighting

PIMORONI ◆ **£48** | pimoroni.com

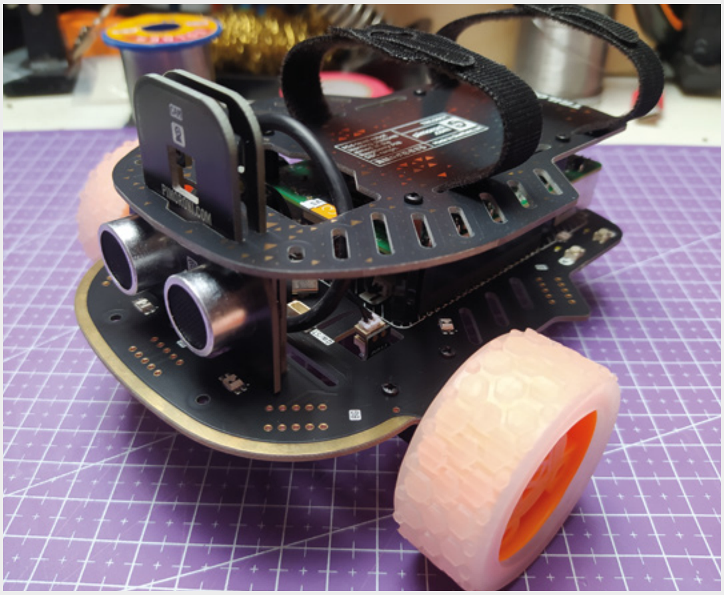
By **Jo Hinchliffe** @concreted0g

It's a great time for people to get into building robots – there's no end to the recipes and tutorials online. You could choose to totally scratch-build, or go for something more off-the-shelf to get you up and running fast. Firmly at the latter end of the spectrum sits the Pimoroni Trilobot kit.

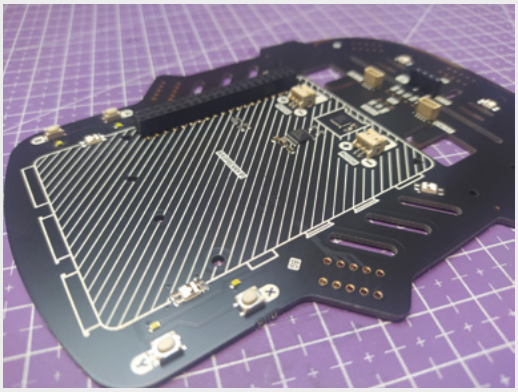
Definitely looking like a trilobite, the Trilobot kit allows the robo-curious to build a gorgeous little bot sporting two driven wheels at the front and a castor at the rear to allow quick, responsive turning. It comes in two different kit options; a base kit where you need to add a Raspberry Pi 4 and a USB-C power bank to get it up and running, or a full kit which includes a Raspberry Pi 4, power bank, SD card, and Raspberry Pi Camera. We've built the base kit for the purposes of this review.

All the motor driving electronics are built into the chassis, and the chassis structure is formed out of beautifully designed FR4 PCBs. FR4 is super-tough so, once assembled, the Trilobot feels really rugged and will survive a few crashes and tumbles. The assembly itself is pretty straightforward with good instructions: **hsmag.cc/AssemblingTrilobot**.

You begin attaching the micro gear motors to the underside of the chassis board. The boards are beautifully designed and the artwork and silkscreen are of the usual excellent Pimoroni quality. After adding the motors and wheels, you add the castor assembly. You then mount the Raspberry Pi 4 – you can mount the camera, but we momentarily couldn't find one, so skipped this! Included in the Base kit is an ultrasonic sensor which you can mount alongside a camera. Everything slots together well, and it all physically comes together in perhaps 20 minutes. You'll notice, when you plug in the motors, that the Trilobot has a pair of Qw/ST connectors. These are available so you can add extra sensors/hardware easily. In addition to the Qw/ST connectors, there are five 'Breakout Garden' sets of pins, ready to receive sockets (sold separately) which open up the Trilobot to a huge range of Pimoroni peripherals. Finally on pins, there is a set of three pin pads broken out specifically for a servo connection, and we imagine a classic first modification might be to try and pan the ultrasonic sensor or a camera to aid robot navigation.



Right The fully assembled Trilobot Base kit with Velcro straps, ready to receive a USB-C power bank



Above All the chassis components are FR4 PCBs, finished to Pimoroni's usual excellent standard

Adding the top plate, which is a reversible choice of two delicious artwork designs, you can attach a power bank with the included Velcro straps, and they also provide a short flexible USB-C right-angle cable. There's a nice gap in the top plate that allows easy access to the Raspberry Pi SD card slot, and also routes the USB-C power cable. We didn't use the slick-looking Pimoroni power bank that's included with the full kit – ours was quite large, but was still perfectly accommodated and held securely on the top plate.

Having dragged a copy of NOOBS onto the SD card, you need to hook up the Trilobot to a keyboard and monitor for first boot, and to tinker with the code examples. At the end of the assembly instructions there are details of how to enable the camera and I2C interfaces, and also how to clone and interact with the Pimoroni Trilobot GitHub repository which contains some great example Python scripts to get you started. The first suggestion is to run the **flash_underlights.py** example which shows off the incredibly bright LED under-lighting built into the chassis! Beyond flashing the lights, there's a great range of examples to get you started with all kinds of functionality.

Speaking of functionality, the PCB has several buttons built-in, and there are numerous examples of using the buttons to interact with the LEDs, and also a really simple example **single_button.py** which just shows how to detect if a button has been pressed. There are lots of examples around movement, from simple scripted short movements through to an example, **remote_control.py**, which allows you to choose from a range of wireless controllers and then you can soon be driving around at your leisure. Speaking of driving, we wondered how well the castor arrangement would work on a variety of surfaces. It's fair to say this isn't an outdoor all-terrain robot, but we found that it was perfectly happy on hardwood or lino flooring, but also did fine on carpets and rugs. If you have a long shag pile carpet, we aren't so sure!

Finally, on the subject of examples, there are numerous examples around adding and controlling servos to the Trilobot, as well as some very nice examples of using the ultrasonic sensor for distance measuring or object avoidance. These include **avoid_walls.py**, which does what it says on the tin, as well as interesting examples like **distance_lights.py**, where the LED under-lights indicate how far away they are from an object in view of the ultrasonic sensor.

To sum up, we feel the Trilobot is definitely a mid-level kit that's capable of some excellent robotics project work, as well as great fun to watch, drive, or navigate. There are some tiny nuts and bolts which may mean really young ones might need a hand or some supervision, but the excellent software examples give a huge opportunity for both robotics projects and learning Python. ◻

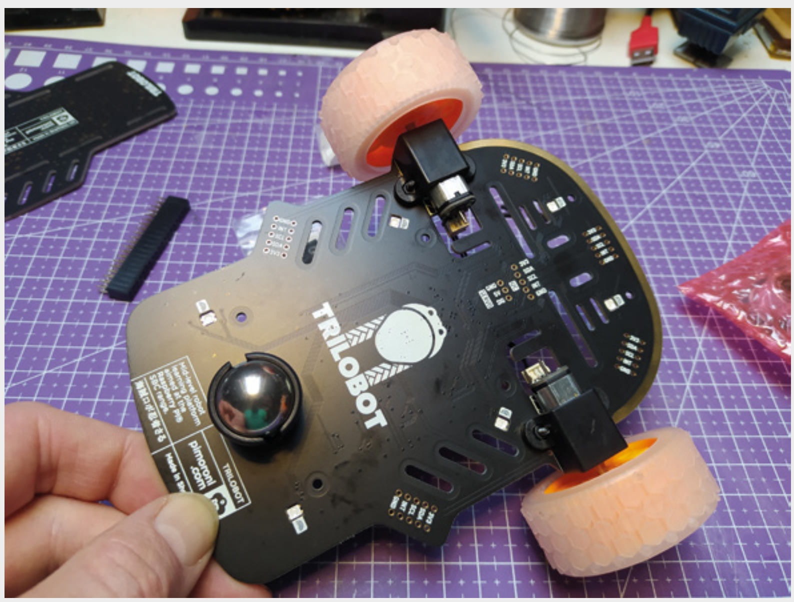
VERDICT

A great mid-level kit suitable for home fun and learning, or perfect in an education environment.

9/10

Left One of the first suggested examples is to play with the LED under-lighting – it's incredibly bright!

Below The wheels and castor system works well, and is easy to install



next month

issue #55

ON SALE
19 MAY

BUILD A BUDGET-FRIENDLY PLOTTER

ALSO

→ 3D PRINTING

→ ARDUINO

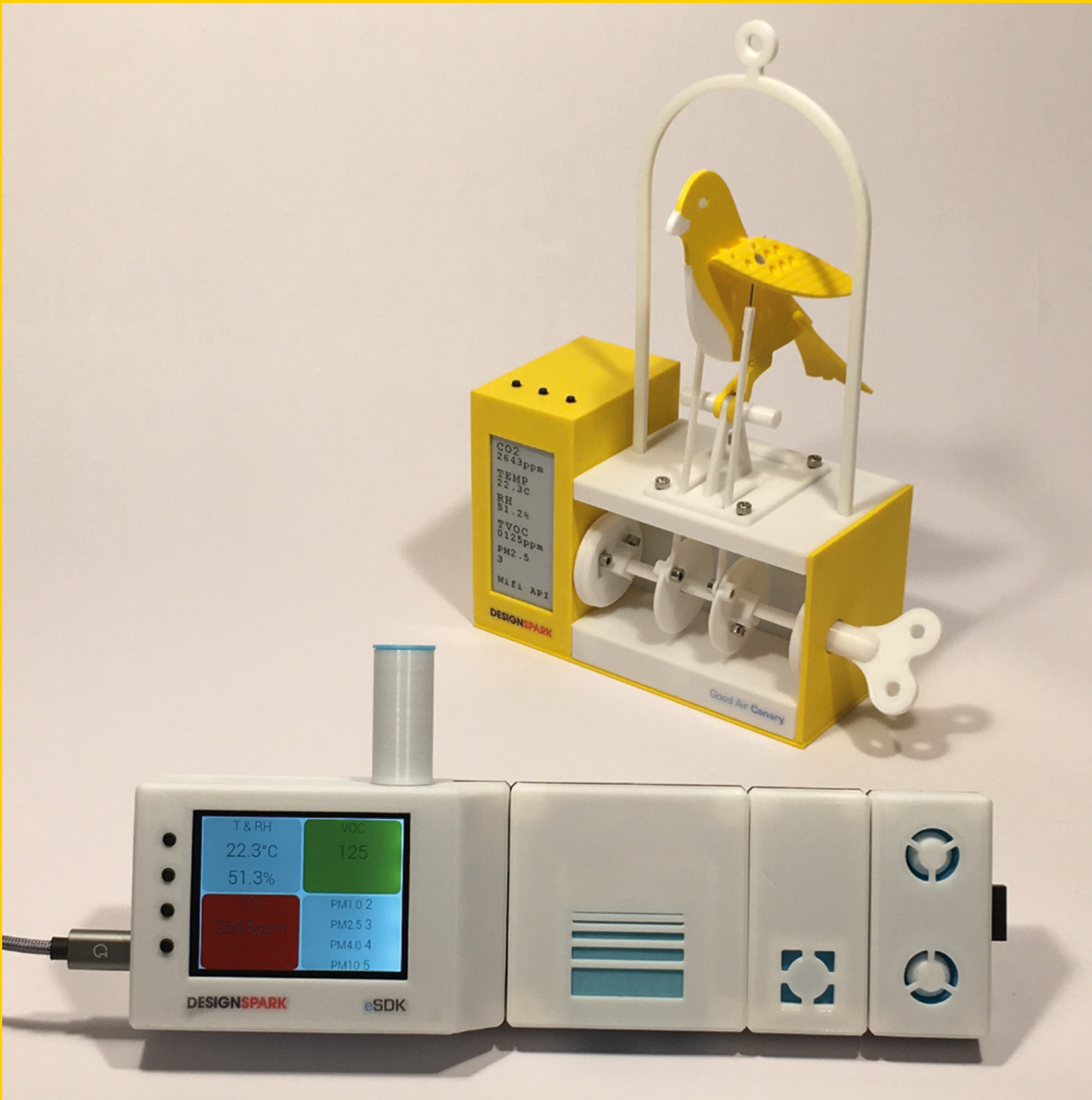
→ LASER CUTTING

→ RASPBERRY PI

→ AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe



It’s not the soldering that keeps me awake at night, or the 3D printing; it’s the ethics. Am I making the world a better place with my design, or will the thing I’m making end up in a landfill?

Jude Pullen